

Full-orbit particle simulations with sawtooth, RMP and ripple for AUG

Graduation thesis

S.Y.F. Cats, BSc¹

s.y.f.cats@student.tue.nl

Fusion Committee:

dr. H.J. de Blank (chair)^{1,2}
dr. E. Westerhof²
prof. dr. ir. G.T.A. Huijsmans^{1,3}
dr. ir. F. Felici¹
dr. ir. L.P.J. Kamp¹

Applied Physics Committee:

dr. ir. J. van Dijk (chair)¹
dr. H.J. de Blank^{1,2}
dr. E. Westerhof²
dr. R.W. van der Heijden¹
dr. ir. L.P.J. Kamp¹

Eindhoven, Friday 10th March, 2017

¹Eindhoven University of Technology, The Netherlands

²DIFFER, Eindhoven, The Netherlands

³ITER, Cadarache, France

Abstract

Energetic particles provide heat to the bulk plasma of magnetically confined fusion devices by Coulomb collisions [26]. Confinement of these particles is required to improve efficiency. Axisymmetry breaking fields have been found to affect magnetic stability, i.e. transient heat and particle fluxes caused by Edge Localized Modes (ELMs) [20]. These fields have also been found to deteriorate fast particle confinement [28], resulting in a so-called ‘density-pumpout’ [62].

‘Saddle coils’ are purposely build to break axisymmetry [84]. Optimal operation of these saddle coils requires understanding of both beneficial and detrimental effects of non-axisymmetric fields [87]. In this work the focus lies on the latter: deterioration of fast particle confinement. A full-orbit simulation code called `EBdyna.go` has been developed which ‘tracks’ particles in the magnetic field of the ASDEX Upgrade tokamak. In addition to the non-axisymmetric field of the saddle coils, the field ripple is also taken into account.

Sampling of the phase space reveals that the field ripple and saddle coils affect a different part of the phase space. Radial excursions of transit particle is found to be correlated with magnetic islands induced by the non-axisymmetric fields. Trapped particles are subject to the ‘ripple resonance’, a resonance related to bounce points and the field ripple [32, 58].

Here we show a similar resonance holds for the saddle coil field (as assumed by [90]). Trapped particles that meet the resonance condition have radial excursions of approximately 2.5 % of the plasma vessel within 2 ms. The contribution of the ripple resonance of the RMP field is an often overlooked contribution to the density-pumpout.

Energetic particles are primarily found within the core. Understanding of the redistribution by this ‘sawtooth’ instability can help develop a coherent control strategy [41]. An electromagnetically consistent model of the sawtooth crash by de Blank [15] has been implemented. Plasma profiles show the expected redistribution within the core, but the sawtooth does not affect particles at larger radii. Additional losses due to the sawtooth instability have not been found.

Collisions and induced currents in the plasma (the plasma response) have not been taken into account in this work.

Revisions since first version to the committee.

version	day	changes
1.90	27 th Feb	First version to committee
1.91	28 th Feb	- Spelling correction in title section 2.5.4 - Added ‘date accessed’ and ‘ISBN’ to some references
1.92	7 th Mar	Added unit of E_{kin} in figure 4.1(a)
1.93	8 th Mar	Corrected ψ to φ in equation (2.18)
1.94	9 th Mar	- Added L.P.J. Kamp as fusion committee member - Changed correction of Error Field Correction Coils to error fields, instead of ripple in section 2.4 (error noticed by G.T.A. Huijsmans)

Preface

Full-orbit particle simulations with sawtooth, RMP and ripple for AUG. What a sentence... We will dive deep in plasma physics and only talk about a very specific application: tokamaks. Here we will discuss simulations performed on particles to investigate synergy between the a magnetic instability, the ‘sawtooth’, and ‘Resonant Magnetic Perturbations’ (RMPs) or ‘the ripple’. Do these effects play an individual role? Do they affect each other? Is there some sort of *synergy*? All these questions will be discussed, but I can imagine we need to explain more of what we’re dealing with.

Maybe it is worth mentioning first why it is useful to investigate this topic. If we look at the energy balance and environment of the future we realize there are grim times ahead [26]. Energy resources currently developed either fall short or have major disadvantages (e.g. radioactive waste, depletion, environmental contamination). It is my belief that we, all humans, should leave no stone unturned and keep open minds in order to find a solution. This might well be the largest challenge we will face the coming century (or centuries).

It is therefore I’ve decided to make my contribution to a sustainable energy mix by doing research in nuclear fusion. Since fusion is quite complex, I could only focus on a particular aspect. However I hope this contributes to knowledge of the ‘Fusion society’. Together we might develop a sustainable solution for the energy problem.

This gives enough motivation to start the research. Besides that I must admit, fusion is simply fun! If any questions remain, one might try to contact me by mail: s.y.f.cats@student.tue.nl or s.y.f.cats@differ.nl. However, these addresses will not be available for long, so try selwyn@lagaria.nl if you’ve got no luck with the other two.

The first two committee members, H. de Blank and E. Westerhof, have been supervising this project. Here I would like to express my gratitude for the useful discussions that have made this project possible.

Contents

1	Introduction	2
1.1	Fusion power	2
1.2	Tokamaks	3
1.3	3D fields and sawteeth redistributing fast ions	4
1.4	Research question	5
1.5	AUG as modelling subject	5
2	Theory	7
2.1	Tokamak physics	7
2.2	Confinement of particles by magnetic fields	11
2.3	Magnetic structure	14
2.4	3D fields	21
2.5	Particle orbits	26
2.6	Sawteeth	31
3	Physics of the codes	35
3.1	FINESSE (and PF2q)	37
3.2	Biot-Savart solver BS3D (3D)	38
3.3	EBdyna.go	40
3.4	Sawtooth model	41
4	Results	47
4.1	Particle distributions	47
4.2	Losses in time and space	49
4.3	Losses in phase space	51
4.4	Resonant particle motion	51
4.5	Sawtooth redistribution	55
4.6	Synergy	57
5	Discussion	59
6	Conclusion	61
Bibliography		62
Appendices		69
A	Acknowledgements	70
B	Derivation of conserved quantities	71
B.1	Conservation of magnetic moment	71
B.2	Conservation of canonical toroidal angular momentum	74

C Inner workings of codes	77
C.1 Equilibrium reconstruction with PF2q and FINESSE	77
C.2 BS3D	78
C.3 EBdyna.go	80
C.4 Sawtooth	91
C.5 Verification and convergence study	94
C.6 Manual	100
D List of functions and parameters	107
D.1 Function list	107
D.2 Parameter list	109

Chapter 1

Introduction

1.1 Fusion power

Our topic starts with maybe the most famous formula ever: $E = mc^2$. Einstein has shown that mass and energy are essentially the same. Wouldn't it be great if we could utilize this conversion of mass to energy for our daily use? Unfortunately, nuclear reactions required for the conversion of mass to energy take some effort.

Fission reactions are by far the easiest. Some elements fissure spontaneously, whilst others need a little help. However, fissure is known to produce radioactive waste and accidents are hazardous (for example the accidents at Tsjernobyl in 1986 and Fukushima in 2011). These accidents remind us of the disadvantages of fission plants¹.

Let's make the bold statement that fusion would be an ideal alternative energy source. Fusion fuel is abundant and radioactive waste is marginal, the waste only needs to be stored for ~ 100 years [26]. Ever since researchers discovered fusion powers the sun, the quest began to harvest fusion power directly here on Earth. Lately other 'green energy sources' have sparked worldwide interest and a sustainable future without fusion power might be feasible. Nevertheless, the climate and energy problem is too large to omit research into feasible energy solutions like fusion.

Historically, fusion has proven to be extremely difficult and thus coal is still being burned for electricity. Now we face problems due to pollution and the danger of running dry. In the meantime, fusion research has gained knowledge on how to possibly construct a fusion reactor. One of the most important understanding is how to overcome Coulomb forces, which make positively charged nuclei repel each other (quantum mechanics helps a little with that). To overcome the repulsive force, we need the particles to approach each other 'fast enough'. Having more particles (a higher density) also increases the number of fusion reactions. The product of these requirements states that we need to confine the pressure for a 'long enough' period. Indeed each star does this, they use immense gravitational pressure. That's obviously not possible here on Earth. We have to resort to magnets to achieve a reasonable pressure for long enough, or choose the path of 'inertial confinement', which relies on huge large pressures for a short period.

For the fuel it is common to take a mixture of deuterium (D) and tritium (T) (in near equal amounts). This combination has the highest fusion power for realistic pressures. Proposals to use other nuclei as fuel will therefore not be discussed. Each DT-reaction produces an alpha particle with an energy of 3.52 MeV and a neutron with 14.1 MeV [26, 59]. Compared to the temperature of the tokamak (discussed later), these particles have a high energy and thus a high velocity.

¹Although research on new, safer fission reactors is ongoing and sparks my interest. Still, politicians are not likely to favour fission energy plants.

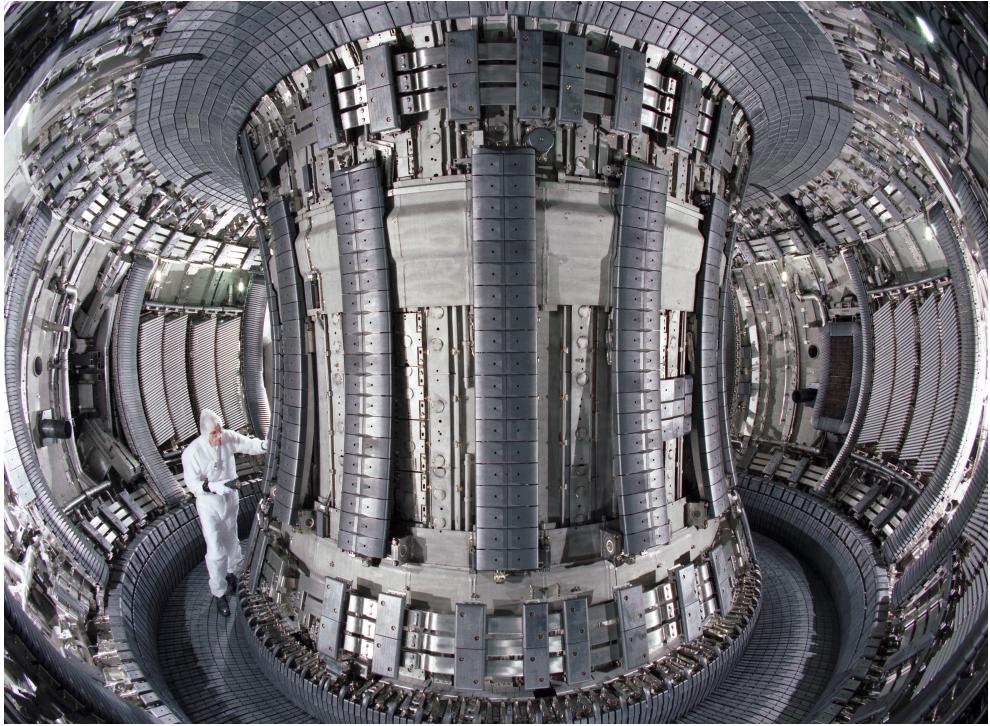


Figure 1.1: A photograph of the vacuum vessel at the Joint European Torus (JET) in Great-Britain from 1994 shows the doughnut-like shape and the axisymmetry (approximatiately). Figure reproduced from [18].

1.2 Tokamaks

Magnetic confinement has made a great leap forward with the Russian invention of the tokamak². It is considered to be the most fruitful approach towards fusion. Characteristic of the tokamak is its doughnut-like shape.

A doughnut is, just like a tokamak, axisymmetric. This means that turning your doughnut does not change any of the physical properties. The same holds for a tokamak, the inside of one is shown in figure 1.1. To extend the comparison, we can look at, for example, the amount of jam. To investigate this, it is sufficient to break the doughnut and only look at the cross section. Similar 2D simplifications are used to study tokamak physics, assuming axisymmetry. Symmetry breaking effects however do require a full 3D study [82].

Note that the most efficient temperature for a tokamak is extremely high: ~ 100 million °C [26, 54]. At these temperatures the ‘fuel’, deuterium and tritium (like all matter), will be fully ionized as electrons move freely. This state of matter is called a plasma.

The magnetic field only exerts a force on charged particles, i.e. on the plasma ions and electrons. Neutrons produced in the fusion reaction can escape freely, whilst α -particles stay in the tokamak and give off their energy to the bulk plasma by Coulomb collisions [26]. They ‘heat’ the plasma, along with various heating systems. High energy particles, like α -particles also interact with the magnetic field. Understanding ‘fast’ ion confinement is crucial for operating a tokamak reactor and optimally designing the magnetic field.

Fast particles can be created by fusion, but also ‘manually’. This is done either by injection, called Neutral Beam Injection (NBI) or by heating already present ions with radio waves (ICRH). NBI is usually performed with deuterium, which is ionized, accelerated to a high energy and neutralized again before injecting it into the tokamak. Neutralization is necessary for particles to

²Tokamak is an Russian acronym for ‘toroidal chamber’ and ‘magnetic coil’.

penetrate the magnetic fields and reach the center (the ‘core’). Thus confinement also shields the plasma from penetration of charged particles.

Radio waves are often used to heat an already present population of ions, e.g. helium, other impurities or any populations of choice. These are heated to far higher energies than the bulk plasma. Confinement of deuterium particles should generally be as high as possible, since they also function as fuel. Conversely, confinement of helium or α -particles only needs to be sufficient to transfer their energy to the plasma and become thermal. Thermalized alpha particles, referred from here on as helium, are detrimental for a fusion reactor. These dilute the plasma and radiate away energy. In summary, fast ion confinement has to deal with conflicting demands.

1.3 3D fields and sawteeth redistributing fast ions

This research will investigate the fast ion confinement, when axisymmetry is broken. Symmetry breaking contributions to the 2D fields are dubbed ‘3D fields’.

One of these 3D fields is due to the coils creating the main magnetic field. This field, directed toroidally, is produced by a discrete number of Toroidal Field coils. By construction they produce the ‘field ripple’, a variation of the field strength, also called the Toroidal Field Ripple (TFR) [22, 34]. The other 3D field in our study is generated by ‘saddle’ coils. These coils are installed to control stability of the magnetic field, i.e. ELMs³ [84, 85, 92].

3D fields have been found to deteriorate particle confinement [48, 62, 67, 94]. Especially fast particles are sensitive to 3D fields, since their collisionality is low and they have a relative long confined path length [82, 94]. Usually, the explanation for the deterioration of (fast) particle confinement is related to the stochasticization of the magnetic field at the edge of the plasma, e.g. [62, 67]. However, this does not completely explain the loss in confinement (the ‘density-pumpout’) [62, 67].

Fast particles are not only important for heating, but also for plasma stability and current drive in tokamaks. Therefore numerous studies have been performed to understand the physics of the fast particle confinement [3, 22, 28, 34, 79, 94]. In addition to loss of heating, plasma stability and current drive, there is concern that fast ions could damage plasma facing components [3, 82]. NBI produced fast particles could produce an additional heat load of $\sim 1 \text{ MW/m}^2$ [90]. Estimations of the lost power with saddle coils vary around 15 % of the injected NBI power [28, 90].

Simulations might help to obtain insight in how to operate the saddle coils with minimal fast ion losses. Most simulation studies have used the so-called guiding-centre approximation. This approximation neglects the variations (of e.g. magnetic field \mathbf{B}) within a gyration orbit. Due to the large gyration orbit of fast ions, these studies might fall short. Additionally they fail to determine the gyration phase consistently. A full-orbit code would include these effects [57].

Gyration is known to influence the ‘tips’ of orbits for some particles [57]. The ‘ripple resonance’ appears to be strongly connected with the toroidal location of these ‘tips’ [32, 58, 90]. All references we have found but one discuss this resonance for the TFR. Tani et al. [90] claims this ripple resonance also applies for the saddle coils, yet only refers to literature specific for the field ripple. They also fail to provide some quantitative analysis (and use a guiding-centre code).

The majority of the research in fast ion confinement has been performed with a static magnetic field. There is however a magnetic instability in the core of the tokamak, which tends to expel the core region outwards. Previous studies have examined the redistribution of this ‘sawtooth’ instability, but then again have been performed in axisymmetric fields [39, 41]. In this research we will study the combined effect of both these 3D fields and this core instability.

³Edge Localized Modes are periodic instabilities at the edge and one of the major ‘problems’ today. These are discussed in more detail in section 2.4.

1.4 Research question

The question we seek to answer is whether *additional loss of fast ions due to the combination of the sawtooth instability and 3D fields is present*. Both bulk plasma and fast ions are effectively expelled from the core of the plasma, so the particles are being ‘mixed’. We are interested if the sawtooth can refill the loss region in phase space, that is the region where particles are lost if one includes 3D fields.

To help our quest we have made some ‘sub-questions’.

- How can saddle coils and the TFR be properly modelled?
- How can the sawtooth instability be properly modelled?
- In which part of the phase space are ions lost due to 3D fields?
- Which physical processes cause additional fast ion loss when 3D fields are applied?
- Is there a combined effect of 3D fields and sawteeth on the loss of fast ions (a ‘synergy’)?

Classifying fast ions only based on their nucleus (deuterium, tritium and helium) does not provide the complete picture. Losses also depend on the position in the space (\mathbf{x}, \mathbf{v}). Since this 6D-space is hard to analyse, we will categorize particles based on the nature of their orbits. Thereto we use the magnetic moment μ and an angular momentum p_φ . The canonical toroidal angular momentum p_φ is conserved in axisymmetry, but not in 3D fields [57, 81]. Going from the full six dimensional phase space to these two variables drastically simplifies the phase space. However, we do not span the complete phase space (one important dimension is the initial φ position). Thus we will have to ‘project’ on the μ, p_φ space and resort to statistics.

For this research a full-orbit code called `EBdyna.go` is used. This code has first been developed by Jaulmes [40] during his work at DIFFER, which already included a model for the sawtooth. During this project, the particle tracking of `EBdyna.go` has been improved, a new sawtooth model is implemented and 3D fields are added. A package called `BS3D` has been developed to determine the vacuum 3D fields using the Biot-Savart law. Vacuum in this sense means the exclusion of induced currents in the plasma by the 3D fields (known as the plasma response). These codes will be thoroughly verified and discussed.

In order to examine the physical processes of fast particle loss, the vacuum field is sufficient. We choose to simulate only deuterium particles produced by the NBI, which is also assumed to be sufficient to answer this specific sub-question. Collisions will not be taken into account, with the advantage that the position in phase space (μ, p_φ) is not rapidly changing, which is helpful in classification of ions⁴. Neglecting collisions will however limit the simulation time to a short time scale and withhold us to determine a steady-state power loss.

First we will discuss the concept of the tokamak in detail. Special attention is given to particle drifts, their confinement and the structure of the magnetic field. Also the 3D fields and sawteeth are discussed here. In chapter 3 we will discuss the codes that have been used in this work. Afterwards the results are presented in chapter 4. These results are discussed in chapter 5. The work is concluded in chapter 6.

The reader is expected to have some basic knowledge of (plasma) physics. It is inevitable to ‘dive deep’ in the world of fusion physics for some concepts and definitions. Some fusion concepts are therefore only briefly treated. Any book explaining fusion physics more extensively could come in handy for those new to fusion (e.g. Friedberg, Wesson [26, 96]).

1.5 AUG as modelling subject

A tokamak named ASDEX Upgrade (AUG) has been built in Garching, Germany and has been equipped with saddle coils [74, 86]⁵. The fast ion diagnostics of AUG make it suitable for research

⁴Classification of particles based on their orbits is discussed in section 2.5, where we will show that some particles are ‘trapped’. Trapped particles will be subject to the ripple resonance, so the classification is of importance.

⁵AUG is short for ASDEX Upgrade, which is again short for Axially Symmetric Divertor Experiment Upgrade

1.5. AUG AS MODELLING SUBJECT

on fast ion confinement. Therefore we will model this tokamak in simulations and with the prospect to compare our results with experiments in future work.

AUG has a major radius of 1.65 m and a minor radius a of 0.5 m. With a total (plasma) volume of $\sim 13 \text{ m}^3$, it is seen as a medium sized tokamak. Typical electron densities reached are $\sim 1 \times 10^{20} \text{ m}^{-3}$ with a core temperature of 3 keV. To contain this plasma, the toroidal magnetic field is in the order of 3.1 T, delivered by 16 Toroidal Field Coils. A poloidal component of the field is produced by inducting a current of $\sim 1.6 \text{ MA}$ ⁶.

ASDEX Upgrade is characterized by its full tungsten wall. In addition AUG has 16 additional coils called ‘control’ or ‘saddle’ coils. These are dubbed ‘B’ coils and are divided in upper (Bu) and lower (Bl) coils. Each of these coils consists of 5 turns and can cope with a current of $\sim 0.96 \text{ kA}$, resulting in a total of 4.8 kAt. Nowadays ASDEX Upgrade has also been equipped with an extra 16 saddle coils around the midplane, called ‘A’ coils [74]. These are not considered since the configurations that have been studied did not involve these ‘A’ coils.

RMP studies are done at a large number of machines, among which DIII-D, KSTAR, MAST, AUG and JET [75, 85]. Full suppression of ELMs, as shown in e.g. DIII-D [20], has been hard to accomplish in AUG as reported by [19, 44, 85, 86]. Mitigation of ELMs has been observed, albeit under different conditions [27, 85]. Recently Suttrop et al. (2017) [87] reported full ELM suppression in AUG.

⁶Necessity of the current is explained in chapter 2. Details of these quantities will be further specified in chapter 2. The major radius is, lets say, half the diameter of the torus. The minor radius is half the diameter of a cross section.

Chapter 2

Theory

We start with a general introduction to the physics of a tokamak. Next follows an explanation of the magnetic field structure. This is required to understand the trajectory of particles, which is dealt with next. This chapter ends with theory of the sawtooth instability.

2.1 Tokamak physics

First we start with some background information of the plasma in a tokamak, that is its temperature, quasi neutrality and the Coulomb collisions. These help us estimate the time scale of collisions and the ‘slowing down’ of fast ions. Based on this we obtain an estimate when collisionless simulations lose physical meaning. Thus it provides us an upper limit for the duration of simulations.

2.1.1 Plasma confinement in a tokamak

The tokamak is characterized by a major radius R_0 and the minor radius a , which is the maximum of the radial coordinate r (see figure 2.1). By definition this minor radius is expressed by $(R_{max} - R_{min})/2$, where the R_{max} and R_{min} refer to the distance from the center of the tokamak to the most outer value and the most inner value of the ‘tube’ respectively. We use the symbol φ for the toroidal angle and θ for the poloidal angle.

The pressure, confined using the force of the magnetic coils, is determined by the product of temperature T and density n . In fusion and plasma physics, the temperature T is often expressed in electron volts¹. The optimal temperature for fusion power output given the (magnetic) pressure of a reactor is roughly 15 keV. The density is typically in the order of $1 \times 10^{20} \text{ m}^{-3}$.

Please note that the temperature is extremely high, roughly 200 million degrees Celsius. A temperature more than ten times higher than the (estimated) temperature in the center of the sun. All matter, i.e. deuterium and tritium, will convert into ‘the fourth state of matter’: plasma at this temperature. In a plasma the (positively charged) nuclei and (negatively charged) electrons move freely. Since the temperature is extremely high, particles are almost completely ionized and can therefore be confined by a magnetic field [26].

Charged particles, which make up the plasma, exert a Coulomb force on each other. Since opposite charges attract each other, a single particle charge will be surrounded by a ‘cloud’ of opposite charged particles. Their charge is effectively ‘screened’ on a short distance, known as the

¹Electron volt is actually a unit of energy; the temperature is expressed in the typical thermal energy of a particle. The temperature in Kelvin is converted to electron volts by multiplication with k_B/e , in which k_B is the Boltzmann constant and e the elementary charge. Equations in this thesis with the temperature T should be read as if $k_B T$ is written.

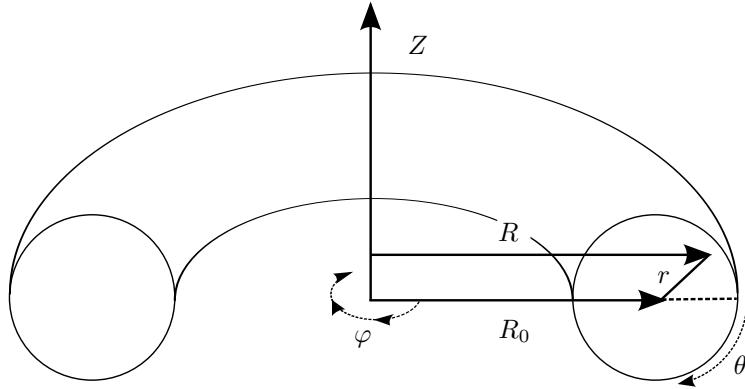


Figure 2.1: Two coordinate systems for a (cylindrical) tokamak. The coordinates (θ, r, φ) describe the same space as the toroidal coordinates (R, Z, φ) . The horizontal distance from the center of the torus to the center of the cross section is called the ‘major axis’ and dubbed R_0 . The radial distance r is defined from the center of cross section. The poloidal angle θ is rotating clockwise from this view, with $\mathbf{e}_\theta \equiv \mathbf{e}_r \times \mathbf{e}_\varphi$.

‘Debye length’. This distance is given by:

$$\lambda_D = \sqrt{\frac{\varepsilon_0 T_e}{n_e e^2}}, \quad (2.1)$$

with ε_0 the vacuum permittivity, T_e the electron temperature, n_e the electron density and e the elementary charge [26].

On length scales somewhat larger than the Debye length, the plasma is ‘quasi-neutral’. The charge density due to positively charged particles (ions) is approximately equal to the charge density of negatively charged particles (electrons). This relation is expressed using the sum of ion densities n_i as:

$$\sum_i Z_i n_i = n_e, \quad (2.2)$$

with Z_i the charge state of each ion species.

2.1.2 Coulomb collisions

Chances that two particles come close and ‘touch’ each other are very slim. The long-range Coulomb force often prevents this. Thus the (strong) nuclear force does not really play a role in the energy or momentum transfer (it does play a role in fusing nuclei of course). Hence we can estimate the time scale of the ‘slowing down’ of fast ions by only considering Coulomb collisions. Here we follow the derivation in Friedberg [26].

The momentum of a ‘test particle’ (k) will be lost after some typical time, dubbed the ‘collision time’ τ_k , i.e. this is when the direction of the velocity has become randomized. This typical time expresses the movement through phase space and is relevant for the maximum time one can simulate without collisions, before the simulation loses its relevance.

Intuitively, the collision frequency ν_k is defined as $1/\tau_k$. An estimation of the collision frequency can be made based on the interactions between different particle species, each defined with an individual collision frequency: $\nu_k = \sum_b \nu_{kb}$. The collision frequency of a particle k with the bulk b (either electrons e or ions i) is given by:

$$\nu_{kb}(v_k) = \left(\frac{\sqrt{2}}{8\pi\sqrt{\pi}} \frac{Z_b^2 Z_k^2 e^4 n_b}{\varepsilon_0^2 T_b \sqrt{T_b}} \frac{m_b \sqrt{m_b}}{m_k m_r} \ln(\Lambda) \right) \left(\frac{\sqrt{\pi}}{2u_k^3} \operatorname{erf}(u_k) - \frac{\exp(-u_k^2)}{u_k^2} \right), \quad (2.3)$$

with Z_b and Z_k the charge of a bulk particle and the test particle respectively, $\Lambda \equiv 12\pi n_e \lambda_D^3$ (and $\ln(\Lambda)$ is known as the Coulomb logarithm [26]), m_k is the mass of the test particle, m_r the reduced mass², ‘erf’ the error function and $u_k \equiv v_k/v_{T_b}$ a normalized velocity [26]. Here v_{T_b} is the thermal velocity of the bulk particles defined by³ $\sqrt{2T_b/m_b}$.

The last factor between brackets in equation (2.3) can be approximated by a more convenient expression (for what’s to come), which yields:

$$\nu_{kb}(v_k) = \left(\frac{1}{4\pi} \frac{Z_b^2 Z_k^2 e^4}{\varepsilon_0^2 m_k m_r} \ln \Lambda \right) \frac{n_b}{v_k^3 + 3\sqrt{\pi} v_{T_b}^3 / 4}, \quad (2.4)$$

from which we make use of the limiting cases: $v_k \gg v_{T_b}$ and $v_k \ll v_{T_b}$. The first applies for a fast ion and the bulk of ions, whilst the latter applies for a fast ion and the bulk of electrons, due to difference in mass. Even in a future reactor, the α -particles with an energy of 3.52 MeV, have a lower velocity than the thermal velocity of electrons with a bulk energy of 15 keV [26].

Let’s now try to solve the differential equation for fast ions f , slowing down by a bulk of electrons e and (deuterium) ions i :

$$m_f \frac{dv_f}{dt} = -m_f (\nu_{fe} + \nu_{fi}) v_f, \quad (2.5)$$

which will be normalized by the ‘critical velocity’ v_c : the velocity where $\nu_{fi} = \nu_{fe}$. Since $\nu_{fi} \propto v_b^{-3}$, we can reduce this equation to:

$$\frac{dw_f}{dt} = -\nu_{fe} \left(1 + w_f^{-3} \right) w_f, \quad (2.6)$$

with $w_f \equiv v_f/v_c$, the normalized velocity. Note that ν_{fe} is constant during the slowing down, since $v_{Te} \ll v_f$ and thus independent of the speed of the fast ion. Hence the differential equation can be solved analytically and one finds:

$$\nu_{fe} t = \frac{1}{3} \ln \left(\frac{1 + w_0^3}{1 + w_f^3} \right), \quad (2.7)$$

with w_0 the normalized start velocity of the particle. For a reactor sized tokamak the slowing down of a particle is shown in figure 2.2.

The critical velocity v_c can be found relatively straightforward by setting $\nu_{fi} = \nu_{fe}$, and making use of the two limiting cases. The ratio of kinetic energy at the moment the test particle has critical velocity with the thermal energy of electrons T_e (assuming quasi-neutrality: $n_e = n_i$) and is given by:

$$\frac{m_f v_c^2}{2T_e} = \left(\frac{3\sqrt{\pi}}{4} \right)^{2/3} \left(1 + \frac{m_f}{m_i} \right)^{2/3} \left(\frac{m_f}{m_e} \right)^{1/3}, \quad (2.8)$$

which can be determined a priori. This is useful to determine the normalized velocity at start of the slowing down: $w_0 = \sqrt{E_0/0.5m_f v_c^2} = \sqrt{v_0^2/v_c^2}$.

To summarize, the slowing down time t_f of fast particles is given by the formula:

$$t_f = \frac{1}{3\nu_{fe}} \ln (1 + w_0^3), \quad (2.9)$$

and can differ quite significantly from the typical (electron) collision time $\tau_{fe} \equiv 1/\nu_{fe}$ [26, 100]. For ASDEX Upgrade we find the typical times given in table 2.1.

Note that particles move slower through phase space (collision time $(\nu_{fe} + \nu_{fi})^{-1}$) than the slowing down time t_f . Simulations without collisions are thus limited by (a fraction of) the slowing down time t_f (~ 2 ms).

² $m_r \equiv m_k m_b / (m_k + m_b)$

³In calculating the collision frequency we assume the ‘bulk’ is Maxwellian distributed.

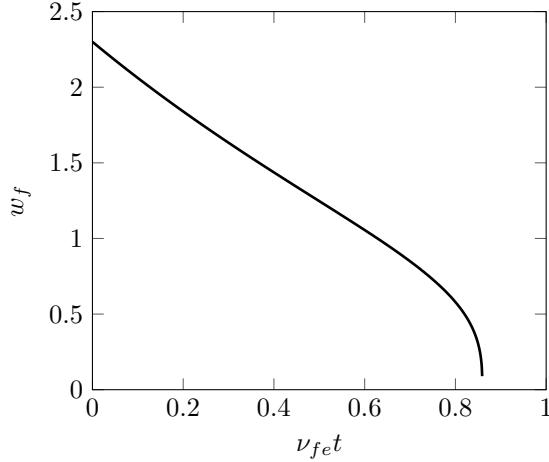


Figure 2.2: Normalized velocity of a fast particle w_f as function of the normalized time ν_{fet} . The start velocity is based on an alpha particle with an energy of 3.5 MeV slowing down in an ion and electron population with a temperature of 15 keV. Ion collision frequency increases during the slowing down and is equal to the electron collision frequency at $w_f = 1$. After this point the velocity w_f drops rapidly.

Table 2.1: Typical collision and slowing down time with a bulk temperature of $T = 3$ keV and a density of $n_e = n_i = 1 \times 10^{20} \text{ m}^{-3}$ (AUG). Note that the ion collision time drops rapidly:
 $\tau_{fi} = \nu_{fi}^{-1} \propto v_f^3$.

	D - 40 keV (NBI) [ms]	D - 3 MeV (ICRH) [ms]
τ_{fe}	107	126
$\tau_{fi}(t = 0)$	32	21×10^3
t_f	10	190

2.2 Confinement of particles by magnetic fields

First the basic idea of how the magnetic field confines particles is discussed. The structure of the magnetic field lines and some coordinates is dealt with in the next section.

2.2.1 Magnetization

The tokamak is characterized by a dominant magnetic field \mathbf{B} in the toroidal direction (going around). Let's for now assume this is the only component of the field. The resulting Lorentz force is given by:

$$F_l = q\mathbf{v} \times \mathbf{B}, \quad (2.10)$$

with q the charge of the particle, \mathbf{v} the velocity vector and \mathbf{B} the magnetic field (vector). Since the velocity parallel to the magnetic field will not result in any force, we have chosen to differentiate between parallel and perpendicular velocity. For this we introduce the unit vector of the magnetic field: $\mathbf{b} \equiv \mathbf{B}/|\mathbf{B}|$. This makes it easy to define the parallel velocity as $\mathbf{v}_{\parallel} = (\mathbf{v} \cdot \mathbf{b})\mathbf{b}$ and the perpendicular velocity as $\mathbf{v}_{\perp} = \mathbf{b} \times (\mathbf{v} \times \mathbf{b})$.

The Lorentz force will cause charged particles to perform a gyro-motion (circular motion) perpendicular to the toroidal field. The radius of this motion, called the Larmor radius, is known to be:

$$r_L = \frac{mv_{\perp}}{|q|B}, \quad (2.11)$$

with m the mass of the particle, $v_{\perp} = |\mathbf{v}_{\perp}|$ the magnitude of the perpendicular velocity and $B = |\mathbf{B}|$ the magnitude of the magnetic field.

From this equation we can see that increasing the magnitude of the magnetic field B reduces the Larmor radius and thereby prevents particles (whose center of gyration is further from the wall than their Larmor radius) to hit the walls directly.

Even so, there are still particle losses by diffusion, caused by Coulomb collisions. However, the magnetic field reduces the *radial step size* (or mean free path length) of the particles. This drastically tempers the diffusion and we can speak of a confined plasma⁴.

Fusion plasma have a relatively high magnetic field. The resulting ion Larmor radius is much smaller than the typical length scale of the vessel (in this case the minor radius a). Therefore we can speak of both ions and electrons as *magnetized*.

Another interesting property of the magnetization is the Larmor frequency, or cyclotron frequency, which is $2\pi/T$, with T the period of one gyration. This happens to be independent of the velocity and given by:

$$\omega_c = \frac{|q|B}{m}. \quad (2.12)$$

This frequency is of great importance for this work. Let's demonstrate this by estimating the Larmor frequency of deuterium. First we express the charge q as the product of charge state Z_i and elementary charge⁵ e . Ionized deuterium has a charge state ($Z_i = 1$). We take a (relatively small) magnetic field of 2.5 T and find $\omega_{ci} = Z_i e B / m_i \sim 1.2 \times 10^8$ rad/s. Time steps of kinetic codes should remain below the period of this motion (~ 52 ns) by roughly a factor of 50. Thus a reasonable time step Δt is ~ 1 ns. Examining physics on time scale of milliseconds therefore costs ($\sim 10^6$) time steps, which becomes computationally intensive.

For this reason some simulation codes decouple the gyration from the motion of the gyro center. However, we have chosen not to have such a *gyrokinetic* or *guiding center* code. A full-orbit code is necessary for fast particles in 3D fields, since the variation of B within a gyration is not small compared to for example the ripple. Fast particles have a Larmor radius r_L of ~ 10 cm.

⁴In section 2.5 we will discuss the particle orbits in more detail and find (especially for fast particles) a much larger step size than the Larmor radius ρ_L . Still the concept of a confined plasma remains the same: particles do not touch the wall.

⁵From now on we will use $Z_i e$ to express charge instead of q . The symbol q will be used for something completely different later on.

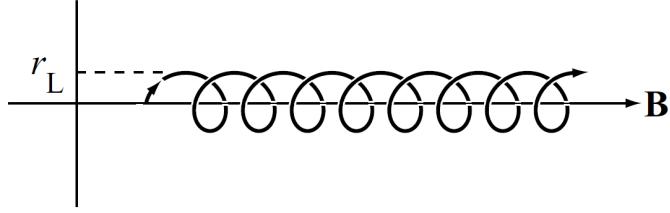


Figure 2.3: Helical orbit of a charged particle along a field line with a Larmor radius r_L . The perpendicular motion is a gyration, whilst the parallel motion is unimpeded. Figure reproduced from [26].

Variation of the magnetic field B within the gyro orbit is approximately $2\frac{\partial B}{\partial R}r_L/B \approx \sim 1\%$ ⁶. For comparison, the (radial) perturbation of saddle coils is 0.1% [86] and the ripple varies with $\sim 1\%$ (see figures 2.9(b) and C.17). A guiding center code could result in an incorrect particle orbit as discussed in [58, 81]. Resonances between orbit and 3D fields are sensitive to the gyration, which will be discussed in section 2.5.5.

2.2.2 Drifts

Up to now we have seen that particles move helically along a magnetic field line (see figure 2.3). The parallel motion is unimpeded, whilst the perpendicular motion is gyrating. We can state that charged particles tend to ‘follow’ the field lines [59]. However, some small corrections have to be made to this first order description. The resulting drifts are the reason why a current needs to be induced in the tokamak. Secondly we will refer to them when categorizing particles in their ‘orbit category’ later on.

∇B drift

First of all we will discuss ∇B drift. Previously we mentioned the relation of the gyro radius and the magnetic field; a higher field results in a smaller gyro radius. Now consider in what manner the magnetic field is constructed. Around the vessel, coils are placed to produce a toroidal magnetic field. These so-called toroidal field coils (TF coils) lead to a magnetic field stronger on the inner side of the torus (small R) than on the outer edges (large R), dropping off with $\propto R^{-1}$ [26].⁷

Because of the inevitable difference in magnetic field, we will call the inner half of the vessel the *high field side* (HFS) and the outer half the *low field side* (LFS) from now on. Let’s look at the effects by assuming a quite large, but still smooth transition within the gyro orbit.

If one tries to draw a curve with a small radius on the left (HFS) and a large radius on the right (LFS) we find the orbits shown in figure 2.4. A vertical drift is seen, which can be either upward or downward, depending on direction of the gyration. Turning clockwise or counterclockwise depends on the direction of the magnetic field and the sign of the charge. The direction of this drift is given by $q\mathbf{B} \times \nabla B$, which is indeed charge dependent.

Curvature B drift

If one considers the top view of a torus, it is clear the magnetic field lines are bending (considering axisymmetry). This makes the particle corner, since they ‘follow’ field lines. However, the curvature of the magnetic field also results in a vertical drift.

⁶Larmor radius r_L is based on particles with an energy in the range of 20 – 60 keV. The factors $\frac{\partial B}{\partial R}$ and B are based on an equilibrium reconstructed by FINESSE, which is used in the results as well.

⁷For a better understanding of the $\propto R^{-1}$ behaviour of the vacuum field by the TF coils, one can draw a top view of the tokamak with two circles inside the plasma (see figure 2.9(b)). These circles enclose the same TF coils and thus the same current, independent of R . This results in $B \propto R^{-1}$ according to general electromagnetic theory [33].

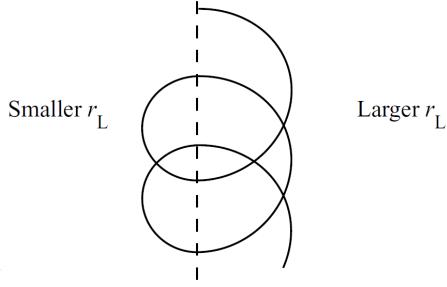


Figure 2.4: Orbit of charge particle with a ∇B drift. Note that ions and electrons will gyrate in opposite direction and therefore drift in the opposite direction. The magnetic field direction is either ‘paper in’ or ‘paper out’. Figure reproduced from [26].

We will try to describe this ‘curvature B drift’, whilst avoiding formulas. Please consider an ion moving exactly parallel to the magnetic field at first instance. Although no (magnetic) force is acting on the particle, this situation can not hold. The magnetic field alters direction and the particle misses (for a brief moment) the corner. Now the perpendicular velocity component v_{\perp} is nonzero and directed to the LFS. The resulting force is directed vertical, resulting in a vertical curvature B drift.

A more sophisticated conception relies on the frame of reference of the magnetic field line. In this frame, the magnetic field is straight instead of curved. However, some force \mathbf{F} seems to act on the particle, directed to the LFS. This (centrifugal) force is the result of an apparent electric field [26], i.e. with $q\mathbf{E} = \mathbf{F}$. The combination of an electric field and a magnetic field results in a drift in the direction of $\mathbf{E} \times \mathbf{B}$, which is indeed in the vertical direction. The next section treats the $\mathbf{E} \times \mathbf{B}$ drift more generally.

The direction of the curvature B drift is given by $q\mathbf{R}_c \times \mathbf{B}$, with \mathbf{R}_c the vector from the center of the torus to the particle (so directed to the LFS). Also here we see that the drift is charge dependent. Since ∇B is directed to the HFS, it strengthens the current resulting from the curvature B drift⁸ [26].

E × B drift

The ∇B and curvature B drifts create a *vertical* current. This separates the charges and results in an ambipolar electric field \mathbf{E} . In general plasma physics we would expect this field to result in a harmless plasma oscillation.

But these particles are magnetized, with the electric field perpendicular to the magnetic field. Particles therefore accelerate on one side of the gyro orbit, whilst they’re decelerating during the other half. The combined effect results in the $\mathbf{E} \times \mathbf{B}$ drift. A sketch of the orbits helps to create a conceptual idea of this drift, see figure 2.5.

The direction of this drift is, as the name suggest, in the direction of $\mathbf{E} \times \mathbf{B}$, regardless of the sign of the charge. Although electrons and ions gyrate in the opposite direction, they react opposite to an electric field. If nothing is done to prevent the $\mathbf{E} \times \mathbf{B}$ drift, a confined plasma is unattainable.

2.2.3 Current

The solution to this is to ‘stir’ or ‘mix’ the plasma. The idea is that the field lines can bring particles from the top, back to the bottom and the other way around. After all, a particle follows the field line (in good approximation).

⁸This is proven in the case plasma currents are small and the magnetic field is therefore in good approximation the vacuum magnetic field. For tokamaks this approximation holds and we can safely say that both drifts are in the same direction.

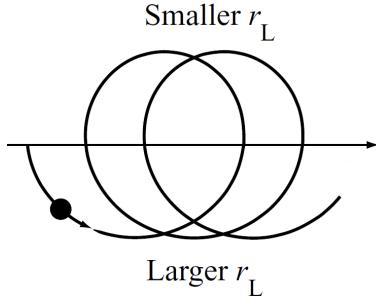


Figure 2.5: $\mathbf{E} \times \mathbf{B}$ drift of a gyrating particle. The electric field is directed downward. For an ion we note that the bottom half of the gyro orbit has a larger Larmor radius r_L than the upper half. The asymmetry of the gyro motion causes a drift to the right. Ions and electrons gyrate in the opposite direction, but the effect of the electric field is also charge dependent. Therefore the $\mathbf{E} \times \mathbf{B}$ drift is in the same direction for ions and electrons. Figure reproduced from [26].

Up to now we have only discussed field lines which remained in the same poloidal position and only moving toroidally. Our solution requires them to be wound helically. There are multiple methods to produce this *rotational transform*. If the winding is built inherently in the construction of magnetic field coils, we call the device a *helical fusion device*, e.g. stellarators.

Tokamaks on the other hand, make use of a current running toroidally in the plasma. The near axisymmetry of the tokamak makes it more suitable to our modelling of 3D fields as perturbations. Another reason why we focus our research on tokamaks has to do with their relevance in the fusion community. They have been studied the most and have achieved the best overall performance [26].

For tokamaks the current is primarily driven by induction. A large coil is placed through the torus (the hole in the doughnut), called the ‘solenoid’. Current is driven through the solenoid, with continuously increasing magnitude. An electric field is induced within the plasma, resulting in the plasma current.

This method for current drive has the advantage of simplicity, but there are many disadvantages. Coils saturate after ~ 1 minute for present day devices and 15 minutes in ITER (expected in 2025). In addition, the plasma current endangers the device when the plasma is lost. The running current is then likely to dissipate in the vacuum vessel, resulting in a huge blow to the device.

Alternative methods for current drive exist, which avoid the saturation of the central plasma coil (the solenoid). For example Electron Cyclotron Current Drive (ECCD), which makes use of microwaves injected by antennas, tuned to the Larmor frequency of electrons. Also a phenomenon called the *bootstrap current* is considered, a form of current induced by collisions amongst particles with specific orbits in combination with a high pressure gradient. However, we will focus on experiments with mainly inductive current drive, which is the ‘de facto’ scenario.

Fast particles considered in this research have 3000 round trips during the 2 ms simulation time in the most extreme cases. With a loop voltage of ~ 1 V [26] the energy would only change 3 keV. For fast particles with an energy of several tens of keV, this can be neglected. The ratio of loop voltage energy with the initial kinetic energy is typically 5 %.

2.3 Magnetic structure

Now we will focus on the structure of the magnetic field (lines). This helps with understanding the kinetic behaviour of ions. First we discuss the magnetic field in axisymmetry. Later in this section the effect of perturbations is discussed.

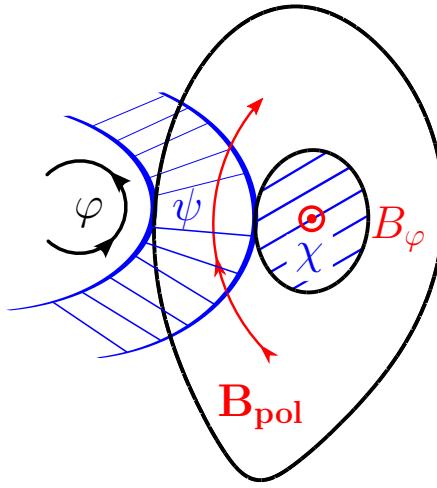


Figure 2.6: Indication of the toroidal field B_φ and the toroidal flux coordinate χ in combination with the poloidal field \mathbf{B}_{pol} and the poloidal flux coordinate ψ . Note the surface of the poloidal flux ψ spans across the toroidal direction (hence it is related to the magnetic field by $2\pi R$). The LCFS is indicated as well as the flux surface with $\bar{\psi} = 0.5$ for AUG. The poloidal flux $2\pi\psi$ is defined as the area integral from the HFS (left in image).

2.3.1 Flux surfaces

If we follow a helically wound field line for one toroidal round trip, a large chance exist it will not exactly close on itself. Even after an infinite number of round trips, the chance of it ending up in *exactly* the same position is practically non-existent. In this case the field line spans a surface *ergodically*. We will call this a *flux surface*, the reason for this name will become clearer soon.

Since particles follow the field lines quite well, any perturbation on such a field line is spread out rapidly across the flux surface. Plasma parameters, such as temperature and density, are therefore by good approximation constant on a flux surface. This means we can analyse plasma parameters in 1D only (e.g. temperature and density as function of the surface).

2.3.2 Radial coordinate

Note that a real tokamak is not actually cylindrical (for efficiency), but the shape is a bit stretched and triangulated. We need new ‘radial coordinate’ to distinguish flux surfaces, as the distance from the center of the vessel r is not a label for the flux surfaces. Some other physical quantities are used which are based on the magnetic field.

Since no field lines penetrate the flux surface, the amount of *magnetic flux* enclosed by any contour on a flux surface is constant. This makes sense, since flux is the surface integral of the magnetic field *normal* to the surface.

Both the *poloidal* flux $2\pi\psi$ as the *toroidal* flux $2\pi\chi$ can be used as radial coordinate. The poloidal flux is used more often and is related to the poloidal field \mathbf{B}_{pol} . The poloidal field is nothing else but the total field without the toroidal component B_φ . This component is responsible for the helical winding of field lines. The relation between the poloidal flux coordinate ψ and the poloidal field \mathbf{B}_{pol} , yields:

$$\psi = \frac{1}{2\pi} \iint \mathbf{B}_{\text{pol}} \cdot d\mathbf{S}_\psi, \quad (2.13)$$

with $d\mathbf{S}_\psi$ the normal vector of an infinitesimal surface element. The total area spans from the HFS to the flux surface and over the toroidal coordinate φ . The area for such an integral is indicated in figure 2.6 with blue.

2.3. MAGNETIC STRUCTURE

Conversely, the poloidal field can be determined using the poloidal flux $2\pi\psi$ in axisymmetry:

$$\mathbf{B}_{\text{pol}} = \frac{\mathbf{e}_\varphi \times \nabla\psi}{R} = \begin{pmatrix} B_R \\ B_z \\ 0 \end{pmatrix} = \frac{1}{R} \begin{pmatrix} -\frac{\partial\psi}{\partial z} \\ \frac{\partial\psi}{\partial R} \\ 0 \end{pmatrix}. \quad (2.14)$$

with \mathbf{e}_φ the unit vector in the φ direction.

The surface with $\psi = 0$ is called the *Last Closed Flux Surface* (LCFS). This LCFS is surrounded by field lines which do not span a surface inside the vacuum vessel. Field lines directly outside the LCFS hit the bottom of the device, due to a coil just below the vacuum vessel. These hit a surface called the *divertor*, which is therefore subject to a high particle and heat flux. The layer of field lines which do this is ~ 1 cm thick and called the *Scrape Off Layer* (SOL). Especially fast ions form a threat to the divertor, making it relevant for this research.

By definition ψ is zero at the edge (LCFS) and maximum at the center of the plasma on the *magnetic axis* with the value ψ_0 . In literature often a flux coordinate is used which is zero at the center of the plasma and increasing towards the LCFS⁹. So when convenient we will use $\tilde{\psi} = \psi_0 - \psi$, e.g. when discussing the sawtooth in section 2.6. Normalization of the radial coordinate is often performed, here defined as:

$$\bar{\psi} = \frac{\psi_0 - \psi}{\psi_0} \equiv \frac{\tilde{\psi}}{\psi_0}, \quad (2.15)$$

which runs from zero at the magnetic axis to one at the LCFS.

In addition to the poloidal flux, we could also use the toroidal flux: $2\pi\chi$. This involves the flux of the toroidal field through the flux surface, looking at the poloidal cross section (see figure 2.6). Similarly to the poloidal flux, we can normalize this. Since it is already zero at the magnetic axis the normalization is given by:

$$\bar{\chi} = \frac{\chi}{\max(\chi)}. \quad (2.16)$$

On top of all these radial coordinate, we discuss two additional ones. It is common to use the square root of the flux coordinates as radial coordinates as well. These are given by: $\rho = \sqrt{\bar{\psi}}$ and $\rho_{\text{tor}} = \sqrt{\bar{\chi}}$. We will encounter them in some of the codes (as described in chapter 3 and appendix C). One advantage of ρ is that we zoom in on the core, which might be of larger interest when discussing energy, pressure or other profiles. The radial coordinate ρ_{tor} is an almost linear coordinate, thus it is the closest to the radial distance r .

2.3.3 Safety factor

We mentioned the flux lines often span a surface ergodically. However, at some radial positions, field lines do exactly close on themselves. This happens after m toroidal and n poloidal turns. The ratio of m/n is called the safety factor q . An irrational value of the winding m/n indicates that field lines span a surface ergodically.

As the q factor describes the helicity of the magnetic field structure, it is properly defined by the ratio of change in toroidal flux χ and poloidal flux ψ :

$$q \equiv \frac{d\chi}{d\psi}. \quad (2.17)$$

Take note that the helicity is constant on a flux surface, and q is therefore called a *flux surface quantity* or *flux function*. This means that also q can be used (and thus is used) as radial coordinate¹⁰. As the name suggests, the safety factor is important for the operation of the tokamak as many magnetic instabilities are related to rational values of q .

⁹Some tokamaks have the poloidal field in the opposite direction. In these cases another definition of ψ, θ or φ might be more convenient.

¹⁰Some non-inductive scenarios might not have a monotonically increasing safety factor. These cases cannot use q as radial coordinate.

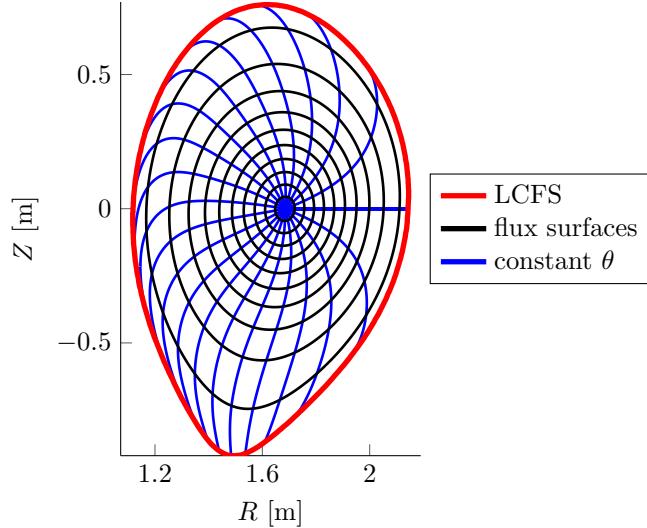


Figure 2.7: Poloidal cross section of ASDEX Upgrade with some flux surfaces indicated and with contours for some poloidal angles θ drawn. The poloidal angle θ in straight field line-coordinates is sparse at the LFS (right) and has a high gradient $\partial\theta/\partial R$ at the HFS (left).

The safety factor is approximately 1 in the center, i.e. $q_0 \approx 1$. Inductive scenarios have a monotonically increasing safety factor from magnetic axis to edge. At the edge the safety factor is ~ 5 , thus the field lines take five toroidal round trips to complete one poloidal turn. The edge safety factor is denoted as q_a ¹¹.

2.3.4 Poloidal angle

Previously we mentioned the poloidal angle θ , which was part of the cylindrical coordinate system (r, θ, φ) . We already ‘declined’ the radial coordinate r , due to the non-circular shape of the plasma. We replaced this with a radial coordinates based on the magnetic structure, i.e. the poloidal flux $2\pi\psi$. Now we redefine the poloidal angle θ based on the magnetic structure. If we define the angle by

$$\frac{\mathbf{B} \cdot \nabla\psi}{\mathbf{B} \cdot \nabla\theta} = q, \quad (2.18)$$

we acquire a coordinate system in which a field line appears as a straight line plotted when in θ, φ -space, having a slope of $1/q$. It is also essential for the sawtooth model described in section 3.4. Coordinates ψ, θ, φ are called ‘straight field line-coordinates’, or ‘flux coordinates’ [31]. In figure 2.7 we can see lines of constant ψ and θ illustrated in a poloidal cross section (a position with a certain φ).

Keep in mind that these ψ and θ relate to the magnetic structure and are thus dependent on the experiment and vary in time. The magnetic structure can have an offset from the center of the machine¹². Therefore the newly defined poloidal angle θ in the plasma center is not necessarily the same as the previously used geometrical angle.

We use a convention in which the poloidal angle θ rotates in the direction of the poloidal field \mathbf{B}_{pol} . Thus in the cross sections shown in this document, it rotates clockwise (see figure 2.6). The toroidal angle φ is directed along the magnetic field direction, which is (from a top view) in the clockwise direction as well. Note that the current is directed opposite to the (toroidal) magnetic field in AUG.

¹¹In practice one used q_{95} , the q factor at $\bar{\psi} = 0.95$, since q diverges to ∞ beyond the LCFS.

¹²The offset originates from a vertical instability and the ‘Shafranov shift’. The vertical instability can be controlled by coils. The Shafranov shift originates from plasma pressure and has no relevant consequences for this work.

2.3. MAGNETIC STRUCTURE

Table 2.2: Overview of coordinates systems. The vertical shift of the plasma with respect to the geometrical center is z_{axis} . The abbreviation ‘sfl’ is used for straight field line coordinates.

symbol	coordinate system	zero at:	possible signs / domain
x, y, z	Cartesian	torus center, mid height	+,-
r	cylindrical	R_0	$[0, a]$
R	toroidal	torus center, along z -axis	+
Z	toroidal	nearest grid point to center of plasma	+,-
θ	cylindrical / sfl	mid plane LFS at $y = 0, x > 0$	$[0, 2\pi)$
φ	cylindrical / toroidal / sfl	(position of TF coil nr. 1)	$[0, 2\pi)$
ψ	sfl	edge of plasma	+
$\tilde{\psi}$	sfl	center of plasma	+
$\bar{\psi}$	-	center of plasma	$[0, 1]$
χ	-	center of plasma	+
$\bar{\chi}$	-	center of plasma	$[0, 1]$

Coordinates used in this work have been summarized in table 2.2. We will denote the offset of the plasma center from the machine center as (R_{axis}, z_{axis}) . EBdyna.go makes use of the Z -coordinate, where the capital indicates its origin ($Z = 0$) to be slightly different. Here $Z = 0$ is defined at the vertical position in a 2D grid that is closest to the magnetic axis. Sometimes we refer to the $Z = 0$ plane, which is also referred to as the *mid plane*¹³. The R, Z, φ coordinate system is dubbed the toroidal coordinate system.

2.3.5 Neoclassical Tearing Modes (NTMs)

Rational values of q tend to be susceptible to instabilities. In the ‘normal’ case a surface is spanned ergodically by a single field line. However, at rational values of q multiple field lines span the surface, each penetrating a poloidal cross section a finite number of times.

Thus at a single radial position with rational q , say at q_1 , multiple (on themselves closing) field lines are present. A (resonant) perturbation could drive one of these field lines to q_2 somewhere on the orbit around the tokamak, whilst the other field lines remain at q_1 . Plasma parameters like temperature, density etc. are now almost the same on q_1 and q_2 . This can be measured in temperature and density profiles as a ‘shortcut’, i.e. the profiles are flat.

The region of field lines with the same q value, now forms a surface when projected to the poloidal cross section. This surface, called an ‘island’, repeats itself every n toroidal turns and m poloidal turns, i.e. with the periodicity of the field lines. The center of such an island is called the ‘O’-point. Positions where no radial short circuit is present are called ‘X’-points. Another name for an island, which ‘tears’ the magnetic structure, is Neoclassical Tearing Mode, where the ‘neoclassical’ refers to transport theory in tokamaks.

Surfaces that are not ‘torn’ are called *intact* flux surfaces, or Kolmogorov-Arnold-Moser (KAM) surfaces [92, 97]. Note that particles would not cross the ‘intact’ surface next to an Neoclassical Tearing Mode if it wasn’t for collisions or specific particle resonances.

2.3.6 Poincaré plot

There is a way to look at the magnetic structure. Put a dot every time a magnetic field line intersects with a poloidal cross section (e.g. when it crosses $\varphi = 0$) and after enough toroidal turns the magnetic structure appears. Such a *Poincaré plot* can also be used to visualize any

¹³The mid plane actually lies on $\theta = 0$ and $Z = Z_{axis}$. However, the difference is minute in comparison to the distance between the magnetic axis and z_{axis} (vertical coordinate from the machine center).

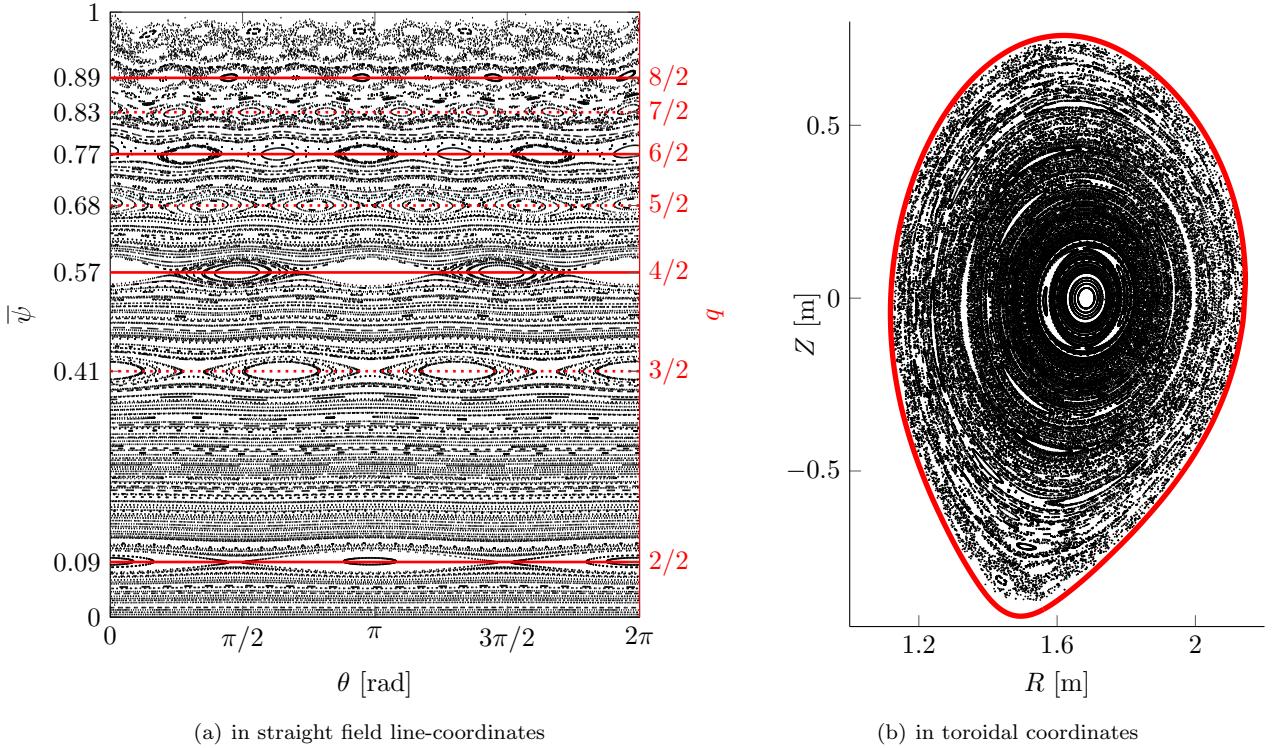


Figure 2.8: Poincaré plot for a $n = 2$ perturbation by RMPs (perturbation repeats itself twice in one toroidal turn). Neoclassical Tearing Modes occur at q surfaces resonant to the perturbation.

islands. Plotted in ψ, θ , the Poincaré plot of ergodically spanned surfaces without perturbations appear as horizontal lines; each field line remains on the same ψ value. A field line exactly on a rational surface would appear as dots (or a single dot on the $q = 1$ surface).

For example, take a field line on the $q = 3/2$ surface, which does $2/3$ of the poloidal turn in one round trip around the torus ($\Delta\varphi = 2\pi$). It crosses the $\varphi = 0$ cross section on three different θ positions and hence three dots appear in the Poincaré plot. Since it has the same three θ positions at $\varphi = \pi$, the Poincaré plot would look the same here. The magnetic structure repeats itself 3 times in a poloidal turn and 2 times every toroidal turn.

When a radial perturbation is applied, the field lines can be distorted and NTMs appear. The Poincaré plot for this case is shown in figure 2.8. Islands show up at the q surfaces which are resonant with the perturbation. This perturbation is repeated twice along the torus, resulting in 2 islands at the $q = 1$ surface, 3 at the $q = 3/2$ surface etc. [11, 64].

2.3.7 Stochastic region

With perturbations, the magnetic field structure at the edge ($\psi \approx 0$) can become *stochastic*. In this case, islands become large enough to destroy intermediate KAM surfaces, and are ‘overlapping’. This allows particles to travel radially without being confined by KAM surfaces. The criteria for island overlapping is the ‘Chirikov’s resonance overlap criterion’, which is given by:

$$\sigma_{Chir} = \frac{\Delta\psi_{m,n} + \Delta\psi_{m+1,n}}{2|\psi_{m+1,n} - \psi_{m,n}|} \geq \sigma_c, \quad (2.19)$$

or in words: the distance between neighboring islands $|\psi_{m+1,n} - \psi_{m,n}|$ becomes smaller than their average width $(\Delta\psi_{m,n} + \Delta\psi_{m+1,n})/2$ with some correction factor σ_c [1, 12]. Due to secondary

formed islands and the geometry in real space, the factor σ_c is smaller than unity, approximately 0.7 [12]. The ‘Chirikov parameter’ σ_{Chir} has been given in a discrete manner.

A more continuous expression for the Chirikov parameter can be found, which also provides insight in the physics which increase the island widths. This expression yields:

$$\sigma_{\text{Chir}} = 4 \sqrt{\left| \frac{hq^2 R}{a} \frac{b_{m=nq}}{B} \right|}, \quad (2.20)$$

with $h = d \log(q)/d \log(r)$ is the normalized shear and $b_{m=nq}$ the pitch aligned component of the magnetic perturbation [75]¹⁴. Notice that maximizing the pitch aligned components of a perturbation increases the stochasticity.

2.3.8 Magneto-Hydro-Dynamics (MHD) and the Grad-Shafranov-equation

A powerful ‘picture’ for the dynamics of magnetic fields in a tokamak is obtained by describing the plasma as a ‘fluid’, interacting with the Maxwell equations. Such analysis is named Magneto-Hydro-Dynamics (MHD) and can describe the (2D) equilibrium. We cut some corners and immediately postulate that, in steady state, the pressure gradient is balanced by magnetic forces:

$$\mathbf{j} \times \mathbf{B} = \nabla p, \quad (2.21)$$

with \mathbf{j} the current density [31]. From this it follows that $\mathbf{B} \cdot \nabla p = 0$, and thus that ∇p is directed perpendicular to the flux surfaces. The current we speak of here is running over the flux surface, demonstrated by the inner product with the pressure gradient: $\mathbf{j} \cdot \nabla p = 0$.

Ampère’s law provides an expression for the current density: $\mu_0 \mathbf{j} = \nabla \times \mathbf{B}$ [26, 31, 33]. Together with equation (2.21) we find a closed set of equations for the axisymmetric magnetic equilibrium. A partial differential equation (PDE) can be found, known as the Grad-Shafranov-equation. In order to derive this we require associated expressions for j_φ and the ‘stream function’ F , following Goedbloed et al. [31]. These expressions are given by:

$$\mu_0 R j_\varphi = R \left(\frac{\partial B_z}{\partial R} - \frac{\partial B_R}{\partial Z} \right) = R \frac{\partial}{\partial R} \left(\frac{1}{R} \frac{\partial \psi}{\partial R} \right) + \frac{\partial^2 \psi}{\partial Z^2} \equiv \Delta^* \psi, \quad (2.22)$$

$$\mu_0 \mathbf{j} = \frac{\nabla F \times \mathbf{e}_\varphi}{R} = \frac{1}{R} \begin{pmatrix} \frac{\partial R B_\varphi}{\partial Z} \\ -\frac{\partial R B_\varphi}{\partial R} \end{pmatrix} \rightarrow F \equiv R B_\varphi, \quad (2.23)$$

$$\mathbf{B} = \frac{\mathbf{e}_\varphi \times \psi}{R} = -\frac{1}{R} \begin{pmatrix} \frac{\partial R A_\varphi}{\partial Z} \\ -\frac{\partial R A_\varphi}{\partial R} \end{pmatrix} \rightarrow \psi \equiv -R A_\varphi, \quad (2.24)$$

with a Laplace-like operator Δ^* defined as shown in equation (2.22) [31]. Please notice that the relation between current density and stream function resembles the relation between the poloidal field and the poloidal flux ψ . The stream function F is, consequently, also constant on a flux surface (a flux function, $F = F(\psi)$). Hence we can express components of the pressure gradient as:

$$\frac{\partial p}{\partial R} = j_Z B_\varphi - j_\varphi B_Z = \left(-\frac{F F'}{R^2} - \frac{j_\varphi}{R} \right) \frac{\partial \psi}{\partial R}, \quad (2.25)$$

$$\frac{\partial p}{\partial Z} = j_\varphi B_R - j_R B_\varphi = \left(-\frac{F F'}{R^2} - \frac{j_\varphi}{R} \right) \frac{\partial \psi}{\partial R}, \quad (2.26)$$

¹⁴More about the Chirikov parameter can be found in Abdullaev, Bécoulet et al., Chirikov, Ghendrih et al., Rechester and Rosenbluth [1, 7, 12, 30, 73].

$$p' = -\frac{FF'}{R^2} - \frac{\mu_0 j_\varphi}{R}, \quad (2.27)$$

with the prime indicating a differentiation to ψ [31]. Combining these we find the Grad-Shafranov-equation:

$$R \frac{\partial}{\partial R} \left(\frac{1}{R} \frac{\partial \psi}{\partial R} \right) + \frac{\partial^2 \psi}{\partial Z^2} = -FF' - \mu_0 R^2 p' = \mu_0 R j_\varphi, \quad (2.28)$$

which needs to be solved iteratively, since j_φ is *not* a flux function (constant on a flux surface) [96]. A self-consistent solution provides the poloidal magnetic fields (ψ) and the toroidal field ($F = RB_\varphi$) of an axisymmetric equilibrium [1, 31, 96]. Since the poloidal field can be reconstructed from the poloidal flux using equation (2.14), the total (2D) equilibrium, including the safety factor q , is ‘reconstructed’ by solving equation (2.28).

2.4 3D fields

The tokamak concept is axisymmetric, but in practice there are perturbations which break the toroidal symmetry. First of all there are ‘error fields’. These are unavoidable and due to the necessity for ‘ports’ to allow e.g. measuring and due to inaccuracies or compromises when building the tokamak. These error fields are, however, negligible compared to the Toroidal Field Ripple and Resonant Magnetic Perturbations [1]. Some tokamaks are equipped with Error Field Correction Coils (EFCCs) to decrease these error fields, but these can also be used for the application of Resonant Magnetic Perturbations [51]. Only the Toroidal Field Ripple and Resonant Magnetic Perturbation coils are considered in this work. The positions of the TF and RMP coils for a section of ASDEX Upgrade are shown in figure 2.9(a) to provide an impression of the dimensions and structure we’re talking about.

2.4.1 Toroidal Field Ripple (TFR)

A well known 3D field is the ‘ripple’, sometimes called Toroidal Field Ripple (TFR). It exists due to the finite number of Toroidal Field coils, producing a toroidal variation of the magnitude of the toroidal field. Exactly at the toroidal position of an TF coil, the (toroidal) field is larger than in between the coils. It extends radially (meaning there is more space between coils at larger R). The TFR is often seen as ‘given’, since there is no way to tune or control it. That is, apart from building a new tokamak of course.

Research is ongoing to use ‘ferritic inserts’ to reduce the ripple, but that’s outside the scope of this research (and AUG does not use any of these techniques up to now) [3, 22, 46, 47, 76, 79, 84].

Figure 2.9(b) shows a representation of the ripple (parameter) η as seen from the top. This parameter η is defined by:

$$\eta(\theta, \psi, \varphi) = \frac{B(\theta, \psi, \varphi) - \overline{B(\theta, \psi)}}{\overline{B(\theta, \psi)}}, \text{ with } \overline{B(\theta, \psi)} \equiv \frac{B_{max}(\theta, \psi) + B_{min}(\theta, \psi)}{2}, \quad (2.29)$$

where B_{max} and B_{min} are the respectively the maximum and minimum of B at a specific point in the poloidal cross section.

2.4.2 Resonant Magnetic Perturbations (RMPs)

Unlike the TFR, the saddle coils creating the RMPs are controllable. Research is ongoing to find the optimal way to operate them, e.g. [55, 62, 98] [19]. There’s a lot to say about RMPs and how to operate them, so we will deal with several topics separately.

One of the questions dealt with in this section is when to call a perturbation *resonant*. And why is the resonance of importance to the particle confinement? One might also wonder if plasma currents are able to ‘screen’ the RMP fields. Before we answer these questions, let’s first explain the concept of RMP fields.

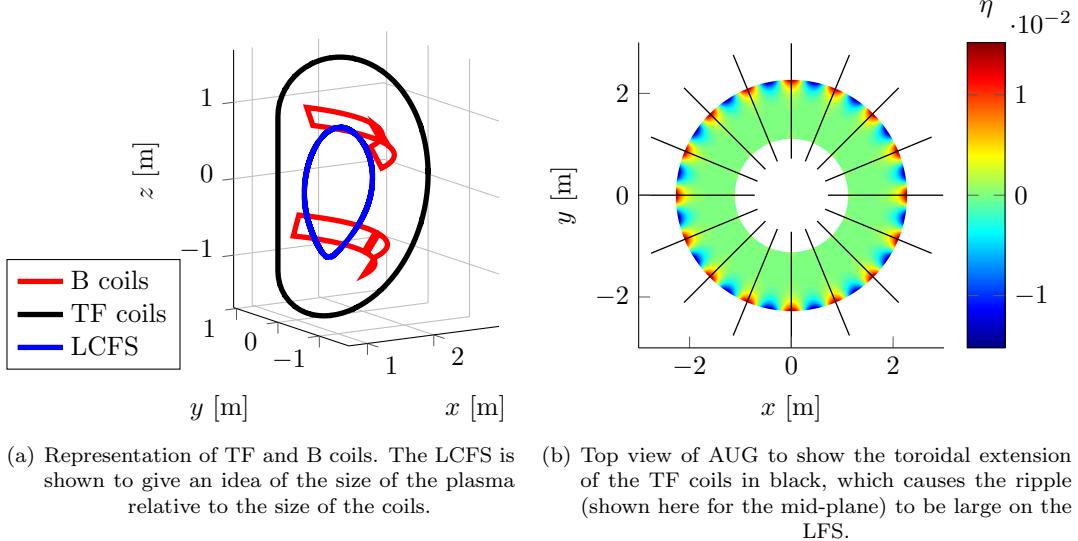


Figure 2.9: The two 3D fields in question are produced by the coil geometry, shown here. In ASDEX Upgrade the ripple is an artifact of the 16 TF coils. The RMP fields are produced by A (not shown) and B-coils. We only study the B-coils, which consist of 8 upper (Bu) and 8 lower (Bl) coils. Note that the standard Cartesian coordinates x, y, z are used.

Firstly, we will discuss ‘modes’ and resonances in the plasma. Next we will introduce Edge Localized Modes (ELMs) and their negative effects. This is followed by the ‘proposed idea’ of how RMPs could control ELMs. After this we define ‘parity’ and the phase difference $\Delta\Omega$ in order to distinguish RMP configurations. Lastly we will present some effects of the *plasma response*. Plasma response is the induction of currents in the plasma, which alter the perturbation of the RMP.

Modes

We have already seen the mode numbers m and n when we mentioned the safety factor $q = m/n$. Here these mode numbers expressed the winding and thereby the periodicity of the field lines on the flux surface. Recall that a field line closes in on itself after m toroidal and n poloidal turns. Thus, the field has a repetition m times poloidally and n times toroidally.

A Fourier transform of the magnetic field over the θ or φ coordinate would find the dominant modes m and n respectively. However, the poloidal Fourier transform would also find more ‘pure harmonic’ m ; the field consists of multiple harmonics. We need to separate the (resonant) *mode* with the *harmonics* it consists of [75].

We speak of an $n = 2$ RMP configuration when the currents through the saddle coil are configured such that they are repeated twice in one toroidal turn. Other harmonic components n for this RMP configuration exist. The largest of these is the ‘complementary mode’, caused by the finite size and finite number of saddle coils [90]. So when for example 8 saddle coils are installed at different toroidal positions (as in AUG), and the RMP configuration is an $n = 2$ mode, also a large $n = 6$ complementary mode (harmonic) co-exists.

When a Fourier Transform is made of the *radial component*, i.e. $B_r \equiv \mathbf{B}_{\text{RMP}} \cdot \nabla\psi / |\nabla\psi|$, we find the components of the perturbations $b_{m,n}$. Mode spectra in m, n -space are point symmetric (around $m = n = 0$) [84]. It is thus sufficient to look at the $n > 0$ spectrum in order to analyse the penetration of each harmonic m . Such analysis is made for two $n = 2$ configurations in figure 2.10. Here only the $n = 2$ harmonic is shown¹⁵. Note that m is a discrete quantity, yet the nomenclature

¹⁵The difference for these two configurations will be explained in section 2.4.2.

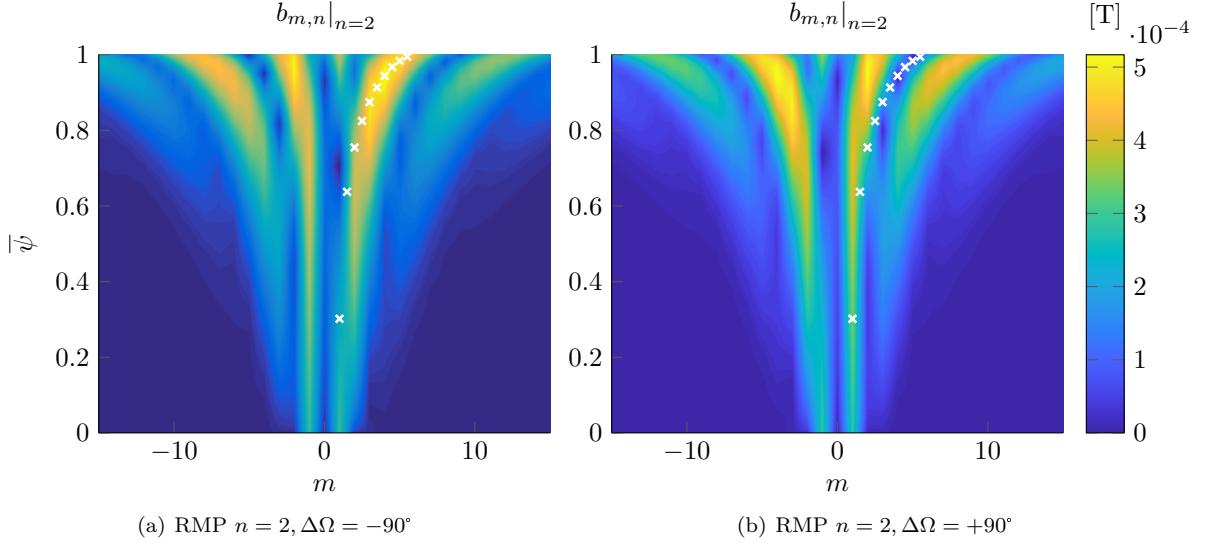


Figure 2.10: Magnitude of $n = 2$ component of the radial perturbation by two RMP configurations presented in a ‘nomenclature’ or spectrogram. White crosses indicate the field aligned components at the rational q surfaces and show that the $\Delta\Omega = -90^\circ$ configuration is resonant. Configurations are based on [87].

is smoothed (as usually done) to make structures easily discernible [75].

With such a nomenclature one can determine the resonant components $b_{m=nq}$, based on the equilibrium field (shown with white crosses). These resonant components enlarge the Chirikov parameter and thus create stochasticity. At the edge higher m harmonics are present in the spectrum, whilst only the low m modes penetrate deep in the plasma.

ELMs

With the first tokamaks, the required confined pressure could not be achieved by far. One of the obstacles was the unexpectedly large turbulence.

Unexpectedly, confinement increased near the edge of the plasma when a large power flux crossed the LCFS. This appears to ‘rip apart’ turbulent flow loops (known as eddies) [50]. Resulting edge gradients effectively raise temperature, density and pressure profile as if put on a *pedestal*. A new operation mode was found where a high confinement could be reached, hence its name: H-mode [26].

However, sometimes the pedestal collapses, due to some sort of instability called Edge Localized Mode. It is believed to be a ‘peeling-ballooning’ mode, where small n couple to higher n , which drive the edge unstable. During an ELM, a large amount of particles and heat is blasted onto the plasma facing components. This transient character of ELMs is a major issue for the divertor [52].

The averaged heat flux in ITER is expected to be 10 MW/m^2 , which can already be called extreme [9]. Even materials with high melting temperatures, like tungsten, have difficulty handling this. Keep in mind that plasma facing components in a reactor have to withstand heat fluxes for long periods for economical feasibility. Transient heat fluxes caused by ELMs make it even harder for plasma facing components to stay below melting temperature [62]. ELMs therefore need to be mitigated or avoided altogether [52, 75].

Apart from melting the divertor, ELMs could also ‘sputter’ wall material into the plasma. Ions from the divertor are knocked out of their lattice and dilute the plasma. Heavy particles, e.g. tungsten with atomic number $Z_i = 74$, ionize up to a high degree. Radiation losses are thus increased when tungsten penetrates the plasma. Additionally, these impurities will decrease the

fuel density, drastically reducing the fusion rate.¹⁶

Research into liquid metal divertors to cope with larger heat fluxes is ongoing. ELMs might be helpful to flush impurities out of the plasma. So maybe, in the future, we can cope with (small) ELMs. Nevertheless, there is interest in means of mitigating or ‘controlling’ ELMs, even if it only were for optimizing the reactor [75].

Why RMPs mitigate ELMs

Experiments with ergodic edges by means of additional coils have first been performed in TEXTOR and Tore Supra. One of the goals of this research was to avoid high power fluxes by enlarging the strike area [1, 2, 24].

Later saddle coils have shown to increase the ELM frequency, reducing the transient energy and heat flux [20, 80, 85]. In some occasions saddle coils have suppressed ELMs altogether [48, 95], sometimes without (significant) loss of the plasma stored energy [20]. A main focus of saddle coil application has therefore become to ‘control’ ELMs. Saddle coils are hence called ‘ELM control coils’ sometimes. However, first employment of saddle coils was to control and influence MHD instabilities in general [36]. Other applications are controlling the ‘resistive wall mode’, controlling the plasma rotation (plasma moving toroidally) and enabling MHD spectroscopy [92].

The first understanding of the ELM control relied on the ‘resonance’ of the radial perturbation with the plasma, i.e. increasing components $b_{m=nq}$. Resulting islands are likely to overlap at the edge, creating a stochastic region near the pedestal of ELMs [30, 62, 63, 75]. The stochastic region at the edge of the plasma will slightly deconfine the plasma edge, reducing the pressure pedestal [62]. This might prevent the peeling-balloonning trigger and thereby fully suppress the ELMs.

Secondly RMPs are believed to lower the peeling-balloonning stability limit. The ELMs are therefore more frequent and have less energy / particle build up, hence smaller [62]. The exact mechanism of ELM mitigation is still unclear, but it is likely correlated to the coupling of the kink response and plasma response [8, 61, 62]. Non-linear coupling of these have diminished the original ‘resonant perturbation’ idea. ELM mitigation can therefore also be achieved with non-resonant configurations [27, 83, 85].

The application of RMPs in AUG (and other tokamaks) has shown increased fast ion losses, a reduction of density and a reduction of current to the divertor, associated with mitigated ELMs [28]. One AUG discharge (or ‘shot’) is shown in figure 2.11. Here we see a drop in density, known as ‘density pump-out’ [62, 67]. The exact origin of density-pumpout can only partially be explained (with modelling). Density pumpout observed in experiments is larger than modeled [49, 61, 62].

Phase difference

To align the (radial) perturbation with the equilibrium field, the upper (Bu) and lower (Bl) coils can be operated individually (maximizing $b_{m=nq}$). It is common to use a sinusoidal signal for the current through the coils. Let’s define this for the upper row as $I_c = I_0 \sin(n\varphi_c + \Omega_u)$, with I_c the current through the coil and φ_c its average toroidal position. The lower row can be given a phase difference $\Delta\Omega$, e.g. $I_c = I_0 \sin(n\varphi_c + \Omega_u + \Delta\Omega)$. The radial perturbation of two $n = 2$ configurations with different phase difference $\Delta\Omega$ is shown in figure 2.12.

Some experiments also choose to vary Ω_u or $\Delta\Omega$ in time. The first is used to smoothen heat load asymmetries [19, 91]. Varying the phase alignment, $\Delta\Omega$, is experimentally used to study the effect on ELM mitigation and confinement [19, 44].

¹⁶Tungsten might not be fully ionized, so dilution might not be that drastic. However, radiation scales with Z_i^2 and if not fully ionized, could even radiate energy from the core. Therefore radiation is seen as the dominant problem [26, 69].

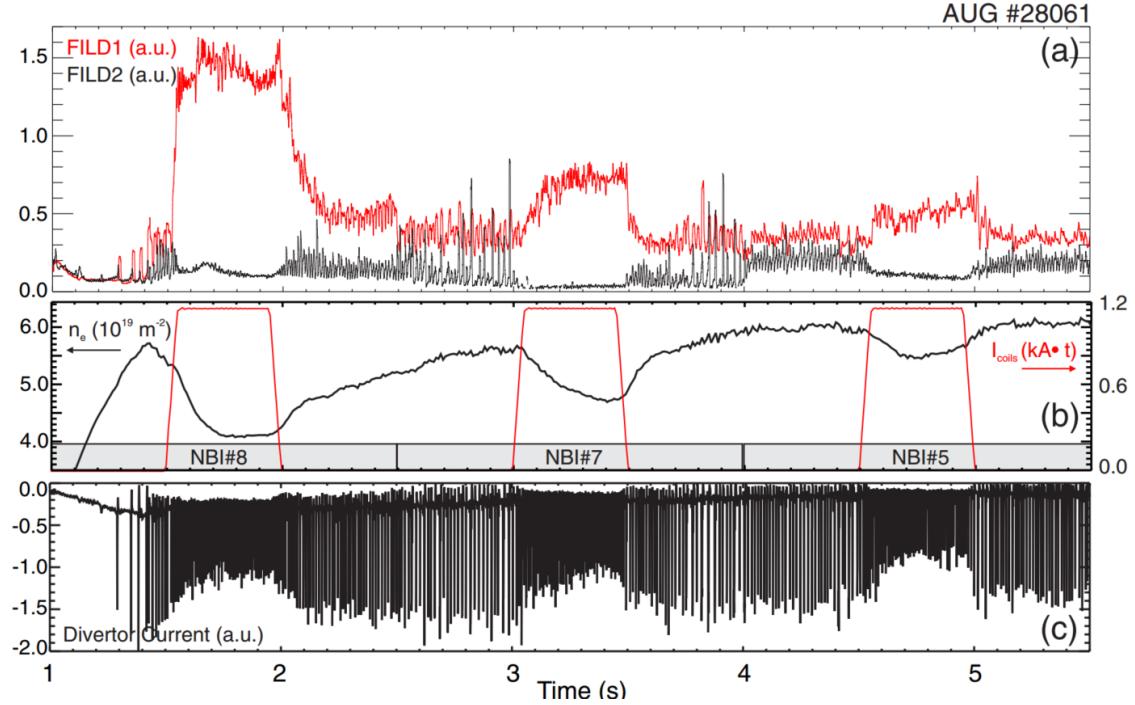


Figure 2.11: AUG discharge where the signals of a) the Fast Ion Loss Detertor (FILD) 1 and 2 b) the line integrated density and current through RMP coils and c) the divertor current show the influence of RMPs. Firstly the density is reduced, called ‘density-pumpout’, secondly the FILD detect loss of high energy ions and thirdly the ELMs are mitigated. Figure reproduced from [28].

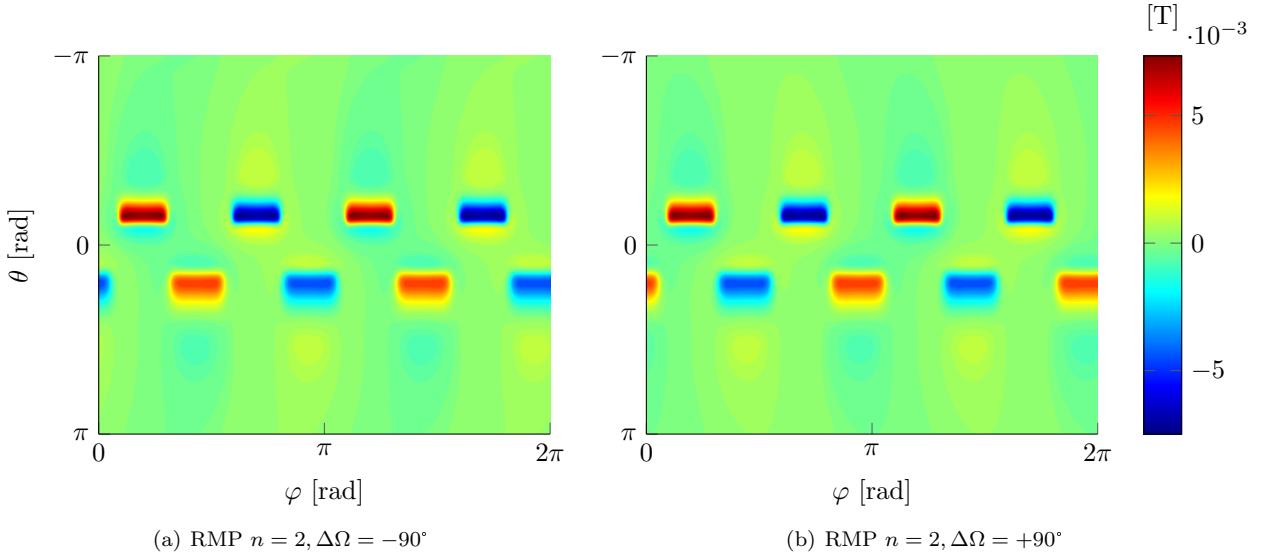


Figure 2.12: Radial component of perturbation at $q = 4$ surface by a two RMP configurations where the phase difference between upper and lower B coils is given by $\Delta\Omega$ (positive is outward directed).

Plasma response

Lastly we will discuss the ‘plasma response’ and its relevance for this work. An ideal plasma response would ‘screen’ any perturbation, removing any stochasticity [68].

Early work expected any plasma response to ‘shield’ the perturbation, thus the vacuum field would provide a ‘worst case scenario’, e.g. [6, 77, 79, 90]. However, more recent work contrasts this statement [27, 68, 92, 98].

In the core, the temperature is much higher than at the edge. Therefore, the resistivity $\eta \propto T^{-3/2}$ is larger at the very edge $\bar{\psi} \gtrsim 0.97$ [62]. Screening currents are thereby prevented in this region. However, at the pedestal region the perpendicular electron flow is large, screening currents at especially the rational q surfaces [62, 75]. Moving inwards we encounter less screening currents at $q \lesssim 3$ due to a small perpendicular electron velocity [62].

Coupling with the $m+2$ ‘kink mode’ and the core ‘kink’ response can amplify the perturbation, drastically changing the nomenclature [75]. Plasma response cannot only amplify the perturbation, but also change the resonant coil phase. The resonant coil phase can easily change with $\sim 60^\circ$ [75].

However, DIII-D experiments show a correlation to the overlap of magnetic island (Chirikov parameter) when the *vacuum* field is considered [23].

To summarize, in reality one ends up with ‘totally’ different characteristics of the field when the plasma response is taken into account. A model for the plasma response is preferred, but the strong nonlinearity of the plasma response complicates the models [92]. Plasma response is therefore beyond the scope of this work.

Although we lack the plasma response and are thus ‘stuck’ with the vacuum field, we do not pretend to have a ‘worst-case’ scenario. Instead we will investigate the *characteristics of ions* that react on 3D fields more generally, similarly to Tani et al. [90].

2.5 Particle orbits

Knowledge of particle orbits will help us understand the results of simulations. Most physics concerning the ion motion dealt with here are specific to the toroidal configuration of the tokamak (the bending of the doughnut). All physical effects in this section apply for electrons as well as for ions. Let’s focus on ions for simplicity.

2.5.1 Equation of motion

The kinetic code needs to solve the equation of motion for each particle. When collisions are neglected, the only relevant forces are from electric and magnetic fields. The equation of motion then yields:

$$m\dot{\mathbf{v}} = q [\mathbf{E} + \mathbf{v} \times \mathbf{B}]. \quad (2.30)$$

For simplification, we will leave the electric field out for the moment. Thus we will consider static magnetic field. We need only little simplification to analyse the particle motion ourselves and find some characteristic particle orbits.

With the equation of motion (2.30), we can derive two constants of motion (appendix B.1 and appendix B.2). Firstly, we will discuss the ‘first adiabatic invariant’, the magnetic moment μ . Secondly we find a conserved momentum, the ‘canonical toroidal angular momentum’ p_φ . Later we will implement the equation of motion in the code by using an ‘integration scheme’ called *BORIS*. The latter is discussed in chapter 3.

2.5.2 Magnetic moment

The magnetic moment is, as the name suggests, a property that can be assigned to a particle in a magnetic field. Don’t underestimate its importance, since it is useful to classify particles and we will use it as a coordinate in phase space. Here we will provide the definition of the magnetic

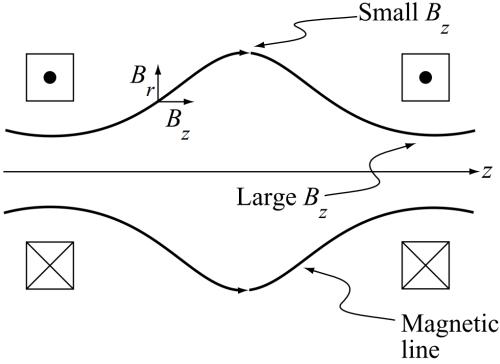


Figure 2.13: A representation of the inhomogeneous magnetic field with a parallel gradient. This forms the basis for the *mirror machine*. Image replicated from [26].

moment, ‘the first adiabatic invariant’. It is often seen as a constant of motion, but it can change. This can occur when the (perpendicular) magnetic field gradient is large within the gyro-orbit.

In appendix B.1 the derivation of the magnetic moment is given. One can read this to get an impression of the invariance, i.e. how well it is conserved. Note that the magnetic moment is derived without any collisions. Collisions do change the magnetic moment (among other things).

It turns out that the ratio of perpendicular energy $E_{\perp} \equiv mv_{\perp}^2/2$ and the magnetic field B is approximately conserved. This ratio is the magnetic moment μ and given by:

$$\mu \equiv \frac{mv_{\perp}^2}{2B} = \frac{E_{\perp}}{B}. \quad (2.31)$$

Often the magnetic moment is normalized with the kinetic energy E_{kin} and magnetic field on axis B_0 . The magnetic moment then relates to the maximum pitch χ_p , which is the angle between the velocity v and the magnetic field \mathbf{B} . However, the pitch isn’t conserved along the orbit, whilst the magnetic moment is.

One could separate the gyration from the gyro center motion, since the gyration is small and quick. Any fluctuations (of for example the magnetic field) are therefore ‘averaged out’. The gyration adds a parallel force to the gyro center of $F_{\parallel} = -\mu\nabla_{\parallel}B$, which is used in *gyro center codes*. However, we will use a *full orbit code*, as discussed in section 2.4.

2.5.3 Passing and trapped

Magnetic mirror

If we now consider the magnetic field between two coils in a 2D geometry (see figure 2.13) we see a parallel gradient of the magnetic field. Here the parallel force is given by equation (B.17).

Now consider a particle initially in a region with a low magnetic field B_l , moving towards the higher magnetic field B_h . There could be a magnetic field large enough that all velocity is in the perpendicular motion. Since we have conservation of energy we find:

$$\mu B_l + \frac{1}{2}mv_{\parallel,l}^2 = \mu B_h. \quad (2.32)$$

In this case, the particle can not move towards a higher magnetic field, since its kinetic energy is conserved and μ is constant. A detailed analysis shows that the particle will revert its parallel direction and *bounce* back to the lower magnetic field. So this particle does not have enough parallel directed velocity/momentum to penetrate all the way to the highest magnetic field strength of the field line, where the guiding center is lying on. This effect is called the *magnetic mirror* [59].

The geometry of figure 2.13 shows one of the earliest devices to confine ions and is called the mirror machine. Unfortunately, good confinement could never be achieved this way as the

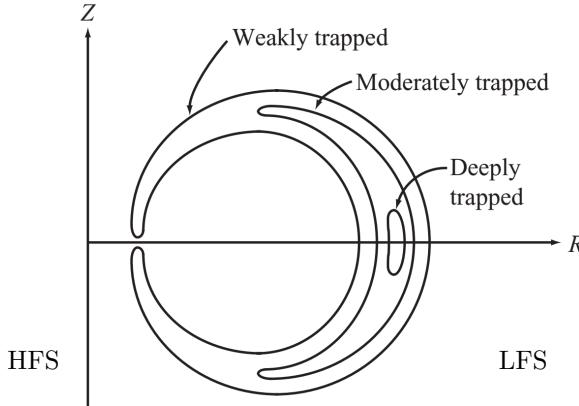


Figure 2.14: Poloidal cross section of three banana particles with varying degree of trapping. One particle comes close to crossing the HFS mid-plane. For these particles the boundary between passing and trapping becomes vague, since the gyro phase comes into play. The radial drifts cancel out over the orbits, so the orbits are closed. Figure reproduced from [26] with the addition of HFS and LFS.

end losses would drive instabilities [59]. Connect the ends and one obtains a tokamak, yet still with particles bouncing when encountering a high magnetic field, i.e. the HFS (inner side) of the tokamak. These bouncing particles are called *trapped*.

Banana particles

In a tokamak the ions move toroidally along helical magnetic field lines from the LFS to the HFS (or back to the LFS and later to the HFS again). A large fraction of these ions will bounce back when the field is high (recall $B \sim R^{-1}$). The radial drifts (both the ∇B as the curvature B drift) are directed vertically. The resulting vertical drift is inward in one half of the midplane, but outward on the other, cancelling out over a complete orbit (one bounce in the upper and one in the lower half of the poloidal cross section). A banana-like shape arises from the poloidal projection of the trapped particles (see figure 2.14). These trapped particles are therefore often called *banana particles*. The *bounce frequency* or ‘banana orbit frequency’ ω_b is defined as the frequency of this orbit, i.e. number of complete orbits per second.

Since the step size of particles has increased to (twice) the size of the banana width, transport by diffusion will increase.

If the collision frequency is (much) higher than the bounce frequency, a banana particle would rapidly leave its position in phase space. This possibly ‘de-traps’ the particle. To examine this the collision frequency is often normalized to the banana frequency [37, 38] and called the collisionality ν^* .

As the tokamaks of the future will be larger, the plasma will become hotter. The faster the particles are, the less they’re influenced by other charged particles. Therefore the Coulomb collisions are less pronounced, i.e. the collisionality ν^* decreases [26]. When the collisionality is low, a banana particle is hardly influenced and can complete multiple banana orbits. A high collisionality on the other hand, frees the banana particles from their trapped state. So the behaviour of banana particles will become more relevant in larger machines.

Precession of trapped particles

Let’s for now assume the collisionality is indeed low and the banana particle can complete several banana orbits. Due to the bouncing, a particle will always move in opposite φ direction in the inner and outer ‘leg’ of the banana. However, take note that the φ translation of these legs will not cancel out in general. First observe that the change in poloidal angle between the tips is the

same for the outer and inner ‘leg’ of the banana. By definition, the average radial position on the outer leg is larger. The helicity is thus lower (a higher q factor). The (absolute) φ translation on the outer leg of the banana is therefore larger than on the inner leg .

This unbalance results in a *precession*, i.e. the banana particle is rotating (in the φ direction) in accordance to the φ translation of the outer leg. Precession can be measured with the precession frequency ω_d , the number of toroidal round trips per unit of time. The slope of a linear fit of the φ, t -plot with ‘many’ banana orbits is equal to ω_d .

For ASDEX Upgrade we can determine in which direction these particles precess, i.e. the sign of ω_d can be determined. Please recall the poloidal cross section shown in figure 2.6, where the HFS is on the left and the LFS on the right. The magnetic field in ASDEX Upgrade is coming towards ‘us’ as indicated in figure 2.6. The poloidal field \mathbf{B}_{pol} is in the clockwise direction, as is the positive direction of θ . Thus field lines have a helicity such that they’re moving clockwise in the poloidal cross section (positive θ direction) and ‘towards us’ (positive φ direction).

Since the ion is gyrating clockwise from our viewpoint, we can see that the ∇B drift is directed downwards (as is the curvature B drift). The outer leg is therefore the leg moving in the negative poloidal direction and thus the negative φ direction. Now we know that for ASDEX Upgrade the precession will be in the negative φ direction, which is counter-clockwise as seen from the top.

Co- and countercurrent passing particles

Particles which aren’t trapped are dubbed *passing*, and their orbits are roughly a circle in the poloidal cross section. Again there is a radial excursion over the orbit, deforming the circle, but the orbit still closes. Some very weakly trapped particles can sometimes cross the HFS mid-plane or bounce depending on the gyro-phase. This slightly blurs the boundary between passing and trapped particles.

We have two main categories in the passing particles. Passing particles can move either parallel (co-field) or anti-parallel (counter-field) to the toroidal field. Again the winding of the field lines tells us that co-field particles (positive φ direction) will rotate in the positive poloidal direction (clockwise in our view from figure 2.6).

Consider two particles, with the same (gyro-center) position on the LFS. A co-field particle will move away from the center and have a smaller R -value at the HFS mid-plane. On the other hand the counter-field particle, moving in counter-clockwise direction, will drift inwards. If the particles are barely passing and the magnetic moment happens to change, e.g. induced by 3D fields, the particle could bounce. A counter-field particle would bounce onto the inner leg, drifting further inwards. On the other hand a co-field particle will bounce on the outer leg, which increases the radial excursion and it might be lost to the wall due to a transition to the trapped phase space.

Now consider two passing particles with the same average flux surface $\langle\psi\rangle$. The co-field particle will be shifted to the HFS. Thus a difference is present between the trapped-passing boundary for co- and counter-field particles, even if they have the same average radial position. As soon as the co-field particle bounces, the ‘average flux surface’ is suddenly more outward. Therefore a gap can appear in the phase space of co-field particles and trapped particles. We will find a quantity for the ‘average flux surface’ later on.

In ASDEX Upgrade it is common to separate passing particles based on the direction of the current, instead of the toroidal field. We will therefore categorize passing particles as either ‘co-current’ or ‘counter-current’. Note that co-current is counter-field and counter-current is co-field.

Other particle categories

We have not mentioned three other particle categories. Not all passing particles have a complete circular orbit, some passing particles have a radial excursion such that they do never cross the HFS and stagnate on the LFS. These are so-called ‘stagnation LFS’ particles. On the other hand some particles never cross the LFS, the ‘stagnation HFS’ particles.

In addition, not every trapped particle has an orbit which looks like a banana. It is possible for a trapped particle to cross at the HFS of the tokamak. To discriminate these we call them

'potatoes'. However, since these are somewhat exotic populations, we will focus on banana and 'normally' passing particles, so just forget about these¹⁷.

2.5.4 Canonical toroidal angular momentum

In addition to the magnetic moment μ we can find another conserved quantity in axisymmetric systems. Whilst μ is related to the pitch, this one is related to the radial position. Therefore it can be used as a coordinate for the 'averaged flux surface'.

Lagrangian physics can be used to find the angular momentum of a particle rushing through the torus. This is done in the frame of reference of the particle, where the centrifugal force is imposed as a real force. Therefore, we speak of the *canonical* toroidal angular momentum p_φ .

A derivation of the conservation of p_φ for axisymmetric systems can be found in appendix B.2. The expression for p_φ in both 2D as 3D yields:

$$p_\varphi \equiv mRv_\varphi + (Z_ie)(RA_\varphi - \psi), \quad (2.33)$$

where A_φ is the φ component of the vector potential \mathbf{A} by 3D fields, i.e. not including the axisymmetry¹⁸.

The evolution of p_φ when 3D fields are applied is also derived in appendix B.2. From this we find analytically the discrete increase of the canonical toroidal angular momentum we expect in a discrete simulation code, which is:

$$\Delta p_\varphi^n = Z_ie\Delta t \sum_{k=0}^n \left(\sum_{j=R,Z,\varphi} v_j \left(\frac{\partial A_j}{\partial \varphi} \right) \right) \Big|_{t=t_k}, \quad (2.34)$$

with k the step number, A_j components of the vector potential \mathbf{A} and v_j components of the velocity vector \mathbf{v} .

One can verify `EBdyna.go` in axisymmetry by checking p_φ remains unchanged. With 3D fields we need to check if p_φ evolves according to equation (2.34). The Biot-Savart code `BS3D` that is used to determine the 3D fields can also be used to determine the derivatives $\partial A_j / \partial \varphi$.

2.5.5 Ripples in the magnetic field

When axisymmetry is broken the conservation of p_φ is lost. However, in a toroidally varying field the orbit averaged value $\langle p_\varphi \rangle$ is approximately conserved. We will discuss two loss processes which result in radial excursions (i.e. causing $\langle p_\varphi \rangle$ not to be conserved that well). In both cases the radial drifts don't cancel out and orbits do not close (seen in the poloidal projection). Additionally, particles will be lost due to stochasticity of the field, especially at the edge.

Ripple trapping

Consider firstly the magnetic field strength along a field line, as shown in figure 2.15. Here the particle is trapped not in the main variation, produced by the HFS and LFS of the torus, but trapped in a local *well*, i.e. in the ripple.

As a result, the particle might spend most time of its orbit on one half of the poloidal plane ($Z > 0$ or $Z < 0$). Consequently, the radial drifts (∇B and curvature B drifts) are not cancelled out over a complete orbit. Therefore, they result in either an upward or downward drift. A ripple trapped particle can be lost quite rapidly if collisionality is low [26, 35]. Simulations from this work have shown that ripple trapped particles are lost in ~ 0.05 ms. Since the RMPs are low in magnitude with a radial perturbation of $\sim 0.1\%$ of B [86], they do not produce significant magnetic wells.

¹⁷We're not so much interested in the projection of the orbit on the poloidal cross section. It is far more interesting if particles react to ripples in the field due to 3D fields as mentioned in section 2.5.5.

¹⁸Note that magnetic field lines do not lie on contours of constant ψ when axisymmetry is broken, i.e. $\mathbf{B} \cdot \nabla \psi \neq 0$

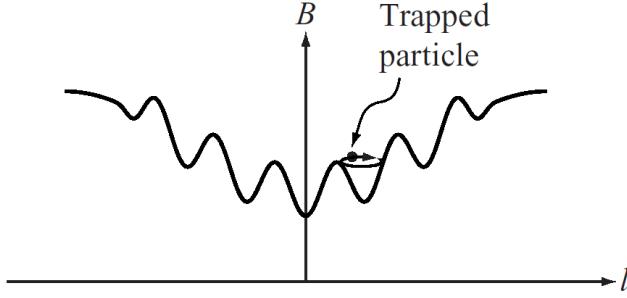


Figure 2.15: Magnetic field strength B along magnetic field line parametrized with l . The field strength B main variation is caused by the $\propto R^{-1}$ drop of the toroidal field. On top of this, a ripple can trap particles locally, of which an example is shown here. Banana particles are trapped in this $\propto R^{-1}$ variation, which is on a larger scale than ‘ripple trapped particles’. The latter might become trapped within the upper or lower half of the poloidal plane ($Z > 0$ or $Z < 0$). Radial drift does not cancel out over the complete orbit anymore. Figure reproduced from [26].

Ripple resonance

Another possible process resulting in radial excursion is resonance of bounce positions with the 3D perturbation. During most of the orbit, any radial excursion varies with the toroidal variation, ‘self-cancelling’ the excursions [32]. However, near the banana tip (or ‘turning point’) the excursions increase in size. Therefore, p_φ can abruptly change at every banana tip [57, 58, 90].

This ‘ripple resonance’ is related to the bounce and precession frequency of banana particles [57, 58, 90]. The ratio of bounce frequency ω_b and the precession frequency ω_d dictates the number of banana tips in one toroidal round trip (see figure 2.16). In order to correctly model the bounce physics, it is essential to use a *full-orbit code*. Otherwise the toroidal position of the bounce point might deviate significantly [57]. Some codes, e.g. ASCOT, can switch between guiding-center en full-orbit algorithms near turning points. This form is called a hybrid code [81]. We have chosen not to implement this optimization to avoid any artifacts. On top of that we expect that only a full-orbit code can be used when a sawtooth is simulated.

Goldston et al. [32] found a theoretical limit on the ripple, known as the GWB criterion. When this limit is broken, particles with a large banana (low μ) will be lost more rapidly [89]. Ripple resonance is widely known and has been studied extensively *for the ripple*. Conversely, we have only found a single publication discussing the ‘ripple resonance’ for the RMP fields, namely Tani et al. (2012) [90]. However, Tani et al. [90] appear to ‘assume’ the ripple resonance to be present for RMP fields.

2.6 Sawteeth

Diagnostics like Soft X-Ray (SXR) and Electron Cyclotron Emission (ECE) have shown a periodic build up and sudden loss of core pressure in the tokamak (see figure 2.17(a)). Pressure (density and temperature) consequently increases in the outer region. The exact trigger of this ‘sawtooth crash’ is yet to be found [40, 56]. It is known that the sawtooth crash occurs when q drops below unity at the center of the tokamak. This instability is ‘pressure driven’ and has a single helicity, i.e. the mode numbers are $m = n = 1$. This unstable mode is sometimes called the ‘internal kink’ mode [96].

Field lines of the equilibrium field lie, by construction, on surfaces of constant $\tilde{\psi}$, i.e. $\mathbf{B} \cdot \nabla \tilde{\psi} = 0$ ¹⁹. Perturbations of the flux can destroy this equality. Let’s assume the sawtooth has a *pure* $n = m = 1$ mode. One can then describe a flux coordinate as a function of $m\theta - n\varphi$, with a helical variation.

¹⁹Here we make use of $\tilde{\psi} = \psi_0 - \psi$ as done in most literature (e.g. Kadomtsev, Porcelli et al., Wesson [42, 70, 96]).

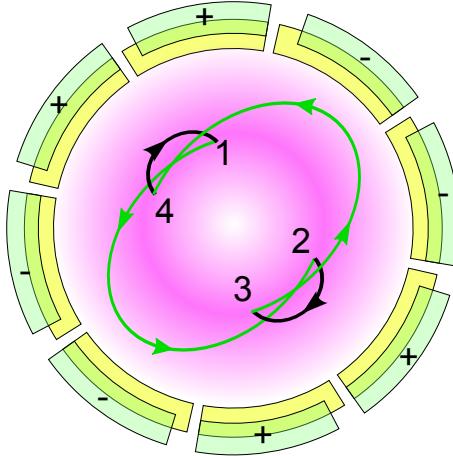


Figure 2.16: Top view of ASDEX Upgrade and the trajectory of a (fast) particle. The ‘Bu’ and ‘Bl’ coils are shown, with signs depicting their individual radial perturbation. A periodicity of half a toroidal rotation is observed, hence an $n = 2$ mode applies. The particle is trapped and bounces accordingly, with the outer leg (green) travelling in the negative φ direction. Evidently, the bouncing, with $|\omega_b/\omega_d| = 2$, is resonant with the $n = 2$ perturbation.

2.6.1 Helical flux

Since a single helicity is considered, there exists a flux coordinate known as the helical flux ψ_* with the property $\mathbf{B} \cdot \nabla \psi_* = 0$. Note that the existence of the helical flux implies that no stochastic fields are present, a consequence of considering only the $n = m = 1$ harmonic.

In order to find ψ_* , we make use of the perturbed flux ψ_H and subtract this from $\tilde{\psi}$. This perturbed flux is equal to the flux through a helically wound surface spanning from the magnetic axis as shown in figure 2.17(b) (hence the H subscript).

Since $\psi_H = \frac{n}{m}\chi$, we can find ψ_H for the $m = n = 1$ perturbation by:

$$\psi_* \equiv \tilde{\psi} - \frac{n}{m}\chi = \tilde{\psi} - \chi, \quad (2.35)$$

where in the last step we substituted $n = m = 1$ [42, 70, 96].

2.6.2 Kadomtsev model

The characteristic time for core ejection by sawteeth ($\sim 100\mu\text{s}$) is much shorter than the characteristic resistivity time $\tau_R = \sim 10\text{ ms}$ [96]. A model to explain this speed has been proposed by Kadomtsev (1975) [42]. This model explains that fast ‘reconnection’ of helical flux is responsible for the creation of a magnetic island, which takes the place of the original core [96]. Within the island lie nested flux surfaces, i.e. no stochasticity, since we only consider a single helicity ($n = m = 1$). These flux surfaces can thus be described by a helical flux ψ_* , which will help us find the magnetic field during a sawtooth.

Let’s define a new radial coordinate r , not to be confused with the earlier mentioned radial coordinate in meters. This radial coordinate is defined as: $r \equiv \sqrt{2\chi}$, with χ the toroidal flux coordinate.

The initial helical flux ψ_- is given by $\psi_-|_{t=0}$. This profile starts out as zero at $r = 0$, increasing till the maximum at $q = 1$ and dropping to negative values at larger r as shown in figure 2.18. After the sawtooth crash, the helical flux will have a very different profile ψ_+ , starting out positive (with $\psi_+(0) = \psi_-(r_0)$) and monotonically decreasing.

The Kadomtsev model makes use of conservation of both helical and toroidal flux to construct the ψ_+ profile. According to this model the maximum of the helical flux will reconnect and form an

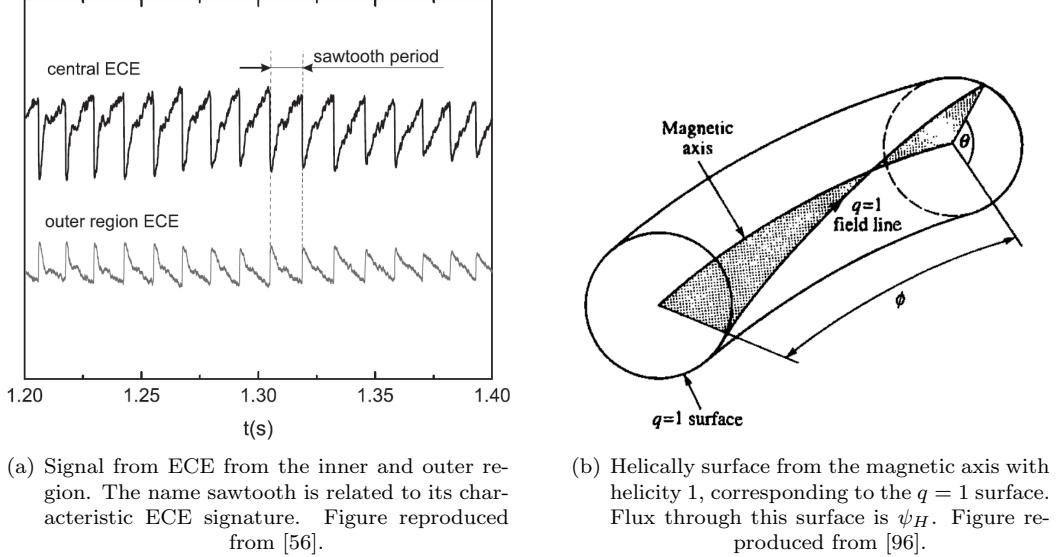


Figure 2.17: Figures providing an impression of both the experimental as the theoretical background of the sawtooth instability.

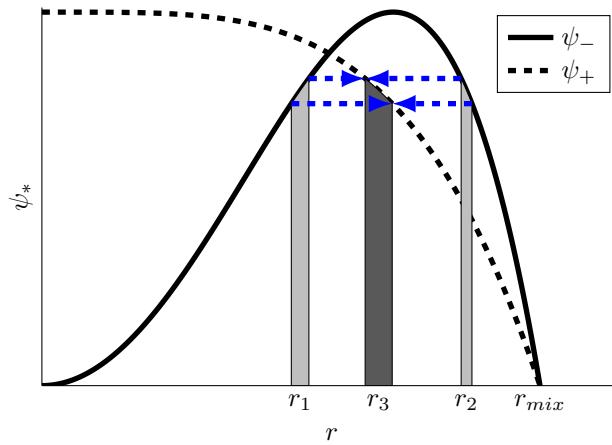


Figure 2.18: Profiles of the helical flux ψ_* before and after the sawtooth crash (ψ_- and ψ_+ respectively). Helical flux progressively reconnects, whilst conserving both poloidal and toroidal flux. This helical flux conservation implies that ψ_* is equal at both sides of the separatrix (r_1 and r_2); toroidal flux conservation implies that the area (gray) is conserved, i.e. $r_1 d(r_1) + r_2 d(r_2) = r_3 d(r_3)$. The sawtooth only affects the center up to the mixing radius r_{mix} .

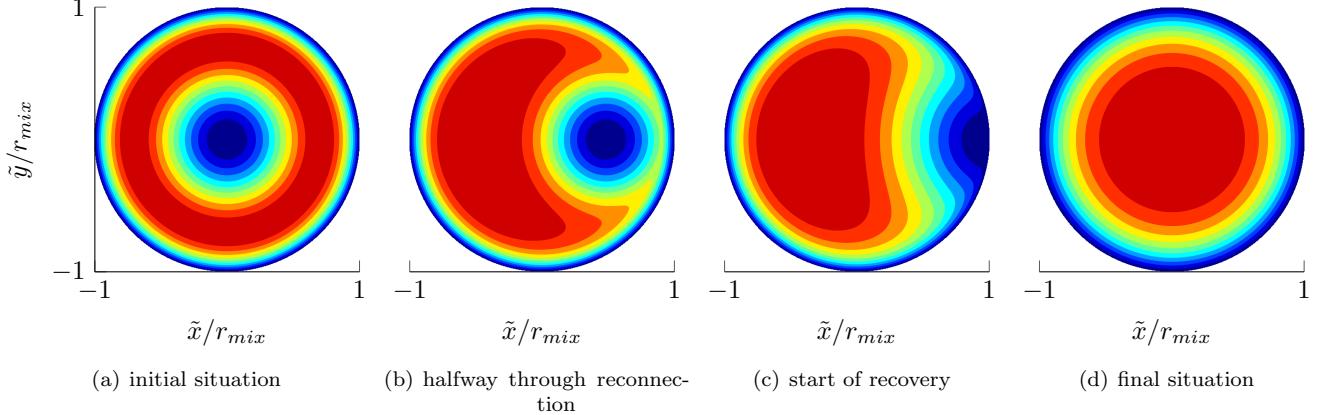


Figure 2.19: Evolution of ψ_* profile in \tilde{x}, \tilde{y} -space during collapse with two phases (colors range from 0 to $\max(\psi_*)$). Note that reconnection starts at the ‘X’-points at $r = r_0$ on the $q = 1$ surface (red in figure 2.19(a)). The ‘X’-point moves outwards till $r = r_{mix}$ (outer blue shell).

island during the first phase ($\sim 100 \mu\text{s}$). Numerous terms can contribute to an ‘incomplete reconnection’, where this phase ends prematurely [70]. Here we assume ‘complete reconnection’, where the original core has disappeared completely, which coincides well with experiments [70]. During a second phase ($\sim 200 \mu\text{s}$), known as the ‘recovery phase’ the poloidal symmetry is restored [70].

At the start of the first phase an ‘X’-point appears at the maximum of ψ_* , which is then on the $q = 1$ surface ($r = r_0$). For simplicity, let’s assume this is at $\theta = 0$ at the $\varphi = 0$ position. Since the perturbation has a 1/1 helicity, reconnection occurs all around the torus at $\theta = \varphi$, forming a magnetic island. ‘Layer by layer’, the magnetic island replaces the original core. The plasma is then ‘mixed’ up till the mixing radius r_{mix} , which is the (non trivial) position where $\psi_- = 0$. Plasma outside of the mixing radius remains *unaffected*.

The expelling of the core and formation of the island is shown in figure 2.19. Here we made use of straight field line-coordinates to transform to a space in which the sawtooth has an symmetric $m = 1$ structure. This transformation is defined by:

$$\tilde{x} = r \cos(\theta - \varphi) \quad \tilde{y} = r \sin(\theta - \varphi), \quad (2.36)$$

where θ *must* be chosen in the straight field line-coordinate system. Note that the definition ensures that the coordinate system ‘moves along’ with the helicity of $q = 1$.

As reconnection continues, the ‘X’-point moves radially outwards, reconnecting helical flux from the original core (originally at $r_1 < r_0$) with equal helical flux at $r_2 > r_0$. Conservation of helical flux dictates that these helical fluxes are equal, so $\psi_-(r_1) = \psi_-(r_2) = \psi_+(r_3)$, with r_3 the radial coordinate of this flux after the sawtooth crash (including the recovery phase). These relations are shown in figure 2.18. Note that during the reconnection process, the helical flux at $r > r_2$ is still unaffected.

Toroidal flux first covered by the disk of $r_2 d(r_2)$ is divided over the island and original core. The conservation of this toroidal flux yields: $r_2 d(r_2) = r_1 d(r_1) + r_3 d(r_3)$, from which we can determine $r_3 = \sqrt{r_2^2 - r_1^2}$ [70, 96]. This is consistent when considering the surface of the island in \tilde{x}, \tilde{y} -space, which is $\pi r_2^2 - \pi r_1^2$.

Together with the condition that the helical flux is equal at r_1, r_2 and r_3 , we can construct the profile ψ_+ . Although the Kadomtsev model provides us with insight of flux reconnection and provides the eventual shape of the flux, it does not provide the exact time dependency of the sawtooth evolution, nor the shape of the helical flux ψ_* during the crash. Some choices for these have to be made in order to determine **E** and **B** fields. These choices are described in chapter 3.

Chapter 3

Physics of the codes

Several code packages work together. Here we will describe these packages, their functionality and the relevant physics. Detailed information about codes, like verification and the manual are found in appendix C. Lists of parameters and functions are found in appendix D. In this work `EBdyna.go` has been improved and a functionality called `BS3D` has been added to the `EBdyna.go` framework. These two ‘packages’ are found in the SVN repository of `EBdyna.go`:

```
svn co --force https://www.differ.nl/svn/shared/EBdyna.go/trunk
```

However, before these can be used, one needs to solve the Grad-Shafranov-equation to reconstruct the 2D equilibrium. This has been done with a `FORTRAN` code called `FINESSE`. A `Python` tool, named `PF2q`, has been used in order to find the solution of the Grad-Shafranov-equation best matching an experiment in ASDEX Upgrade. The general workflow is shown in figure 3.1.

Whilst modelling and calculating the magnetic field, it is seen more important to keep the magnetic field *divergence free* then to have an accurate representation of an experiment. A magnetic field which isn’t divergence free would result in noisy output (e.g. p_φ variations in axisymmetry).

Some time scales need to be considered for these simulations. Most have already been mentioned, but table 3.1 holds a complete overview. Based on this and the convergence study in appendix C.5 we have chosen the fundamental time step $\Delta t = 1\text{ ns}$ and the simulation length $t_{sim} = 2\text{ ms}$. Note that the 2D equilibrium has been reconstructed by `FINESSE` only once and does not evolve, apart from when a sawtooth is applied in the simulation.

Table 3.1: Time scales of deuterium ions with an energy in the range of $20 - 60\text{ keV}$ in AUG ($T = 3\text{ keV}$ and $n_e = 1 \times 10^{20}\text{ m}^{-3}$). Collisional time scales (and slowing down) are based on section 2.1.2, whilst others are based on simulation output.

name	description	typical time
gyration time	gyration of ion, depends on B	$33 - 64\text{ ns}$
precession (passing)	time for one toroidal round trip (two directions directions)	$\sim 1\text{ }\mu\text{s}$
precession (trapped)	time for one toroidal round trip (negative direction)	$10\text{ }\mu\text{s or smaller}$
orbit (passing)	duration for completing (a circular) orbit in poloidal projection	$5 - 80\text{ }\mu\text{s}$
orbit (trapped)	duration for completing (a banana) orbit in poloidal projection	$20 - 100\text{ }\mu\text{s}$
slowing down time	time for a fast particle to slow down by collisions to bulk energy	$3.8 - 17\text{ ms}$
ion collision time	fast particle momentum change by bulk of ions	$11.5 - 59.6\text{ ms}$
electron collision time	fast particle momentum change by bulk of electrons	107 ms

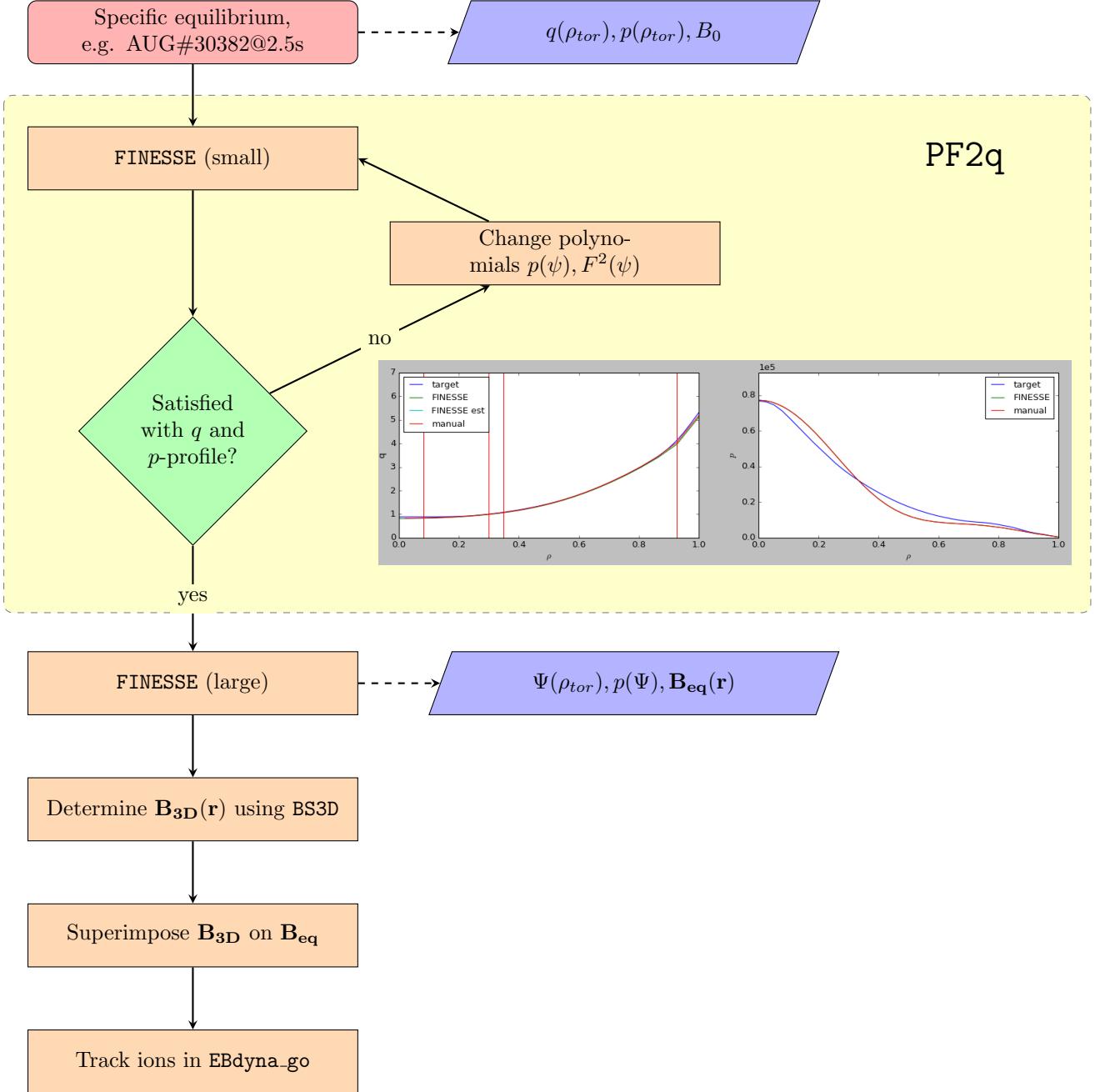


Figure 3.1: Main workflow to determine fast ion motion in 3D field at ASDEX Upgrade. The roles of the different packages are shown (PF2q, FINESSE, EBdyna.go, BS3D). FINESSE can be performed with a (radial and poloidal) resolution to acquire the q and p profile for comparison in PF2q (usually 33×33). Once the user is satisfied with the p and q profiles, FINESSE can be performed at a higher resolution (e.g. 513×513). Next the 3D fields are determined by BS3D and EBdyna.go can start the simulation using a superposition of the 2D and 3D fields. Blue trapezia indicate which (key) data are acquired or required from the specific process.

3.1 FINESSE (and PF2q)

FINESSE is the successor of HELENA, a well known code to solve the Grad-Shafranov-equation. It is written in FORTRAN and requires polynomials of the \tilde{p} and \tilde{F}^2 , i.e. the normalized p and F^2 profile (see appendix C). These profiles are polynomials of $\bar{\psi}$, i.e. $\tilde{p} = \tilde{p}(\bar{\psi})$, $\tilde{F}^2 = \tilde{F}^2(\bar{\psi})$.

In order to acquire these polynomials one is advised to make use of PF2q [93]. PF2q is a PYTHON code, which opens a Graphical User Interface (GUI). Here the experimental q and p -profiles are plotted along with the profiles reconstructed by the last FINESSE iteration. PF2q enables the user to alter $p(\bar{\psi})$ and $F^2(\bar{\psi})$ polynomials and run FINESSE from the GUI. Profiles of q , p and F are shown as functions of ρ_{tor} . The user can iteratively converge these polynomials to best match the q and p -profiles of the experiment. This is primarily based on visual inspection.

On top of the $\tilde{p}(\bar{\psi})$ and $\tilde{F}^2(\bar{\psi})$ profiles, FINESSE requires additional data from the experiment. Most importantly it requires the shape of the Last Closed Flux Surface and the parameters α and ϵ used to normalize the $p(\bar{\psi})$ and $F^2(\bar{\psi})$ profiles. PF2q determines these based on the minor radius a , the current on axis I_0 and the magnetic field on axis B_0 (which is purely toroidal).

PF2q is written to cope with data provided by ASDEX Upgrade. Both the p and q -profiles are provided as functions of ρ_{tor} , i.e. $p = p(\rho_{tor})$ and $q = q(\rho_{tor})$. This means that PF2q will perform the following tasks:

- Read AUG data from a .mat-file.
- plot the $p(\rho_{tor})$ and $q(\rho_{tor})$ profiles in the GUI.
- determine α, ϵ from the AUG data.
- Extract an initial guess for the $p(\bar{\psi})$ and $F^2(\bar{\psi})$ polynomials from the current input file of FINESSE.
- Let the user change these polynomials ($p(\bar{\psi}), F^2(\bar{\psi})$) until the button ‘RUN FINESSE’ is pressed (which then changes to ‘RERUN FINESSE’).
- Determine the normalized polynomials $\tilde{p}(\bar{\psi})$ and $\tilde{F}^2(\bar{\psi})$.
- Update the FINESSE input file with the new $\tilde{p}(\bar{\psi})$ and $\tilde{F}^2(\bar{\psi})$ -profiles.¹
- Convert the resulting $p(\bar{\psi})$ and $q(\bar{\psi})$ profiles to $p(\rho_{tor})$ and $q(\rho_{tor})$ and plot these along with the experimental profile. PF2q utilizes the q -profile for this conversion.

The user can iterate this process until the q and p profile are satisfactory. Although error margins for the q -profile are larger than for the pressure profile, more value is added to match the q -profile due to the sensitivity for the sawtooth modelling (shown by Jaulmes, Jaulmes et al. [40, 41]).

Especially near the center of the plasma emphasis lies on an accurate q -profile, caused by the sensitivity of the helical flux ψ_* and the implications for the sawtooth nodelling. Differences in the reconstructed and experimental profiles are shown for four different radial positions on which the user can base the polynomials.

Once the experimental and reconstructed profile from FINESSE agree, one can take the input file that PF2q has produced. The number of radial and poloidal points should be increased to reconstruct the equilibrium with the desired resolution. Typically one would use 33×33 grid points while iterating with PF2q. For this work the resolution has been increased to the maximum 513×513 grid points after iterating with PF2q.

FINESSE provides all quantities on a grid of radial and poloidal points. That includes geometric quantities like $R - R_0$ and Z , with R_0 the major radius. Some scripts in the EBdyna.go repository are used to convert quantities in a 2D grid of R, Z . Most quantities are smoothly fitted, e.g. $\psi(R, Z)$. This could be done for each component of \mathbf{B} , but it was found that reconstructing \mathbf{B}_{pol} from ψ (with equation (2.14)) improved conservation of p_φ .

¹The LCFS is stored in a separate file that FINESSE uses, but PF2q does not.

Since the reconstruction of the equilibrium is time-consuming (i.e. the work, not the calculation itself) and the parameter space is already large (due to different configurations of RMP coils), we've only considered a single equilibrium. This equilibrium is AUG shot #30382, at the time 2.5 s, which is just before a sawtooth collapse.

3.2 Biot-Savart solver BS3D (3D)

For this work a package for `EBdyna.go` has been developed, which is given the name `BS3D` (Biot-Savart 3D). It provides 3D vacuum fields from specified coils using the Biot-Savart law [33]. With the Biot-Savart law a high precision is acquired, known to outmatch Finite Element Methods [3].

3.2.1 Implementation of Biot-Savart law

Eddy currents are produced in the plasma as response to 3D fields. These alter the perturbation rather complexly. The response depends on several plasma parameters, e.g. plasma rotation² and density. Therefore we have restricted ourselves by only considering the ‘vacuum field’ of 3D fields, i.e. the field as if no plasma is present.

`BS3D` makes use of the geometry of coils made available by ASDEX Upgrade. It produces a grid based on either straight field line or toroidal coordinates. The magnetic field in each grid point from any coil (with ‘contour’ C) can be determined using Biot-Savart law, given by:

$$\mathbf{B}(\mathbf{s}) = \frac{\mu_0}{4\pi} \int_C \frac{\mathbf{I} \times \mathbf{r}}{r^3} dl, \quad (3.1)$$

with \mathbf{r} the vector from the coil to grid position \mathbf{s} ($r = |\mathbf{r}|$), \mathbf{I} the current vector and μ_0 the vacuum permeability [33].

Furthermore the vector potential is found with a similar integral [33]:

$$\mathbf{A}(\mathbf{s}) = \frac{\mu_0}{4\pi} \int_C \frac{\mathbf{I}}{r} dl, \quad (3.2)$$

which comes directly from Ampère’s law. One can verify $\mathbf{B} = \nabla \times \mathbf{A}$ is indeed consistent with these formulas. One could also reconstruct \mathbf{B} from \mathbf{A} alone, instead of using equation (3.1). In appendix C.5.3 it is shown that the Biot-Savart law produces a magnetic field with a higher accuracy. The vector potential does have other uses, e.g. verification of `EBdyna.go` by determining the evolution of p_φ .

Coils are described as fine wire filaments, i.e. without thickness. `BS3D` discretizes a wire filament as a combination of (ideally infinitesimal) elements. Typically, these wire elements have a length of ~ 1 mm. When the coil has been given with a larger spacing, `BS3D` will use linear interpolation to decrease the element size. An RMP (Bu or Bl) coil typically consists of ~ 3400 wire elements and a TF coil of ~ 9300 wire elements. The discrete Biot-Savart law yields:

$$\mathbf{B}(\mathbf{s}) = \frac{\mu_0 I}{4\pi} \sum_k \frac{d\mathbf{l}_k \times \mathbf{r}_k}{r_k^3}, \quad (3.3)$$

with I the current through a coil, k the counter for the filament elements and $d\mathbf{l}_k$ the vector describing the filament size and direction (see figure 3.2). The vector \mathbf{r}_k is determined by $\mathbf{r}_k = \mathbf{s} - \mathbf{l}_k$. The discrete version of equation (3.2) yields:

$$\mathbf{A}(\mathbf{s}) = \frac{\mu_0 I}{4\pi} \sum_k \frac{d\mathbf{l}_k}{r_k}. \quad (3.4)$$

To verify the integration scheme of `EBdyna.go` we make use of the evolution of the toroidal angular momentum p_φ . This requires the φ -derivatives of the vector potential $\partial A_j / \partial \varphi|_{j=R,Z,\varphi}$.

²The plasma in a tokamak tends to rotate toroidally as a whole. This can influence the perturbations caused by 3D fields.

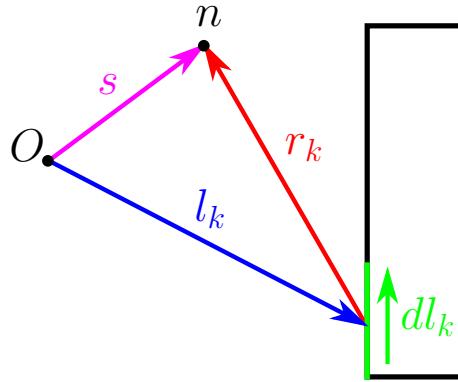


Figure 3.2: Schematic image of vectors used in the BS3D code. The vector \mathbf{s} indicates the evaluation point n from the origin O (which is the design center of the tokamak at $x = y = z = 0$). The vector \mathbf{l}_k indicates filament k and \mathbf{r}_k points from the filament to the evaluation point n . Take note that the length of the filament dl_k (green) should be small for an accurate magnetic field.

In order to accurately determine these, an auxiliary grid is made by BS3D. This auxiliary grid consists of four grid points for every grid point in the main grid, having a slight offset in the toroidal coordinate φ . Based on the toroidal difference in the main grid $\Delta\varphi$ BS3D will put auxiliary grid points at $\varphi \pm \Delta\varphi/20$ and at $\varphi \pm \Delta\varphi/10$. A finite difference scheme is used to determine the φ -derivatives (see appendix C.5.3).

3.2.2 Toroidal Field Ripple calculation in BS3D

When BS3D is set to determine the field of the Toroidal Field coils, one finds a total 3D field, consisting of a 2D field when averaged toroidally and the ‘ripple’ itself. The FINESSE equilibrium field differs slightly from the 2D vacuum field found by BS3D. Pressure and current of the plasma is taken into account consistently by FINESSE, yet not by BS3D; FINESSE provides a valid solution for the Grad-Shafranov-equation. However, on the LCFS, the FINESSE code assumes there is no plasma and should thus find the same field as BS3D. We can accurately estimate the current through the TF coils by comparing these two. The ripple is then ‘extracted’ from the BS3D field. EBdyna.go can superimpose this ripple on the FINESSE equilibrium later on. To ‘extract’ the ripple BS3D will perform the following steps:

1. Determine the 3D vacuum field from the TF-coils on the LCFS. BS3D will use a certain current I_{BS3D} through the TF coils.
2. Average the 3D vacuum field over the toroidal coordinate for the 2D vacuum field on the LCFS³.
3. Compare the vacuum field with the FINESSE output on the LCFS to get a *scaling factor*.
4. Determine the 3D vacuum field on the toroidal grid (R, Z, φ) with current I_{BS3D} .
5. Scale this 3D field with the previously determined scaling factor.
6. Determine the 2D field from BS3D by averaging over the toroidal coordinate.
7. Subtract this 2D vacuum field from the 3D vacuum field to get the Toroidal Field Ripple.

³EBdyna.go and BS3D have a 3D grid where the first and last toroidal coordinate coincide ($\varphi = 0$ and $\varphi = 2\pi$). Thus the toroidal average should exclude one of these grid points.

For the shot analysed in this thesis, the difference in B on the LCFS between FINESSE and BS3D had a standard deviation of 0.33 %, which is considered acceptable.

Geometry of a Toroidal Field coil has kindly been provided by B. Geiger (employed at IPP, the institute housing AUG). The TF coils are D-shaped and have a finite width. BS3D assumes the 16 TF coils are evenly distributed over the toroidal angle φ , thus the geometry of this single coil is sufficient to determine the TF field.

3.2.3 Resonant Magnetic Perturbation calculation in BS3D

BS3D is able to use data of the A, Bu and Bl coils stored in the `EBdyna.go` directory⁴. Vacuum fields from the coils have been determined with a current of 4.8 kAt, to model a 5 turn coil with a current of 0.96 kA. The magnitude of the current is an estimate based on [84, 85, 87]⁵.

Three RMP configurations have been compared in this work, namely:

- $n = 2, \Delta\Omega = -90^\circ$, where the lower coils (Bl) have a phase difference of -90° compared to the upper coils (Bu). Thus the perturbation from the lower coils is 45° ahead of the upper coils in the toroidal direction φ (see figure 2.12(a)).
- $n = 2, \Delta\Omega = 90^\circ$, where the lower coils (Bl) have a phase difference of -90° compared to the upper coils (Bu). Thus the perturbation from the lower coils is -45° behind the upper coils in the toroidal direction φ (see figure 2.12(b)).
- $n = 4$, ‘even’ parity, i.e. $\Delta\Omega = 0^\circ$, where we’ve chosen not to use a sinusoidal signal. The current through Bu and Bl coils is alternating in direction (indicated with + or -), yet constant in magnitude. The signs of these currents are $+ - + - + -$ and equal in both rows.

The $n = 2$ configurations are also used in recent work (experimental and theoretical) [61, 87]⁶. Performance restraints allowed for a 2D grid with 1100×1800 nodes (unique R, Z) and a difference in toroidal angle of $\pi/256$ radians between grid points. A complete description on how 3D fields are determined is provided in appendix C.2.

3.3 EBdyna.go

Particle simulations have been performed by (an improved version of) `EBdyna.go`⁷. `EBdyna.go` utilizes the `BORIS`-scheme to integrate the equation of motion (2.30) and thereby simulate the particle motion [66, 71]. The details and implementation of `EBdyna.go` and `BORIS` are described in appendix C.

Since RMPs and the TFR produce stochastic fields, the orbit of a particle within these regions becomes sensitive to initial conditions and parameters. One cannot expect the same radial excursion if, for example, the time step Δt is modified. When 3D fields are used in `EBdyna.go`, we have to look at collections of particles, using *statistics*.

⁴ Results in this thesis were produced with the Bu and Bl coil data loaded incorrectly. The φ -coordinate in the data files was defined rotating in the opposite direction. This wouldn’t matter if it wasn’t for the direction of the magnetic field and current. However, we do not expect any major influence. The loading of coil data has been corrected in `EBdyna.go` SVN repository #1004.

⁵In hindsight a current of 6.5 kAt would’ve matched better with the configurations in [87]. ITER RMP coils are foreseen to have currents an order of magnitude higher [21, 53].

⁶ Nomenclatures shown by Suttrop et al. [87] correspond with figure 2.10. However, due to a different equilibrium, our resonant configuration is with a phase offset $\Delta\Omega = -90^\circ$, instead of $\Delta\Omega = +90^\circ$ shown by Suttrop et al. [87]. We suspect that the φ coordinate is defined to rotate opposite from ‘our’ φ coordinate. As a consequence, the signs of m and $\Delta\Omega$ are opposite to our definition as well, thus producing the same nomenclatures. Strangely, we defined θ rotating in the opposite direction, since a previous article by Suttrop et al. [86] clearly shows the opposite definition. Since the m, n space is point symmetric, the nomenclatures would not correspond, i.e. the nomenclature which defined here as the $\Delta\Omega = -90^\circ$ should have corresponded with the $\Delta\Omega = +90^\circ$ in Suttrop et al. [87]

⁷The name `EBdyna.go` is based on the capability to simulate full (*gyration*) *orbits* for *dynamic E and B* fields.

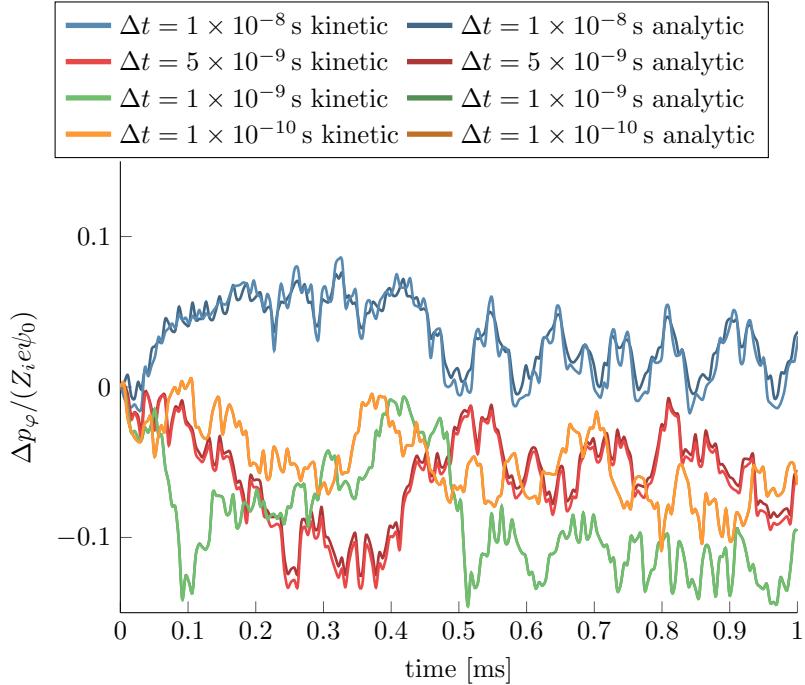


Figure 3.3: Although the evolution of p_φ from both equations (2.33) and (2.34) overlap, the path has not been converged yet. Here the particle has been given an energy of ($E_{kin} = 90 \text{ keV}$) and the ripple was applied.

In figure 3.3 the evolution of p_φ is shown for a 90 keV particle with different values of Δt . Although the two expressions for p_φ match and the grid has converged, we would like to emphasize the *path* of the particle is clearly not converged. Due to the stochasticity, convergence of the orbit of a single particle cannot be attained, yet the statistically averaged results *do* converge. More about grid convergence is discussed in appendix C.5.

3.4 Sawtooth model

Models which describe ψ_* and Φ during the sawtooth are based on the reconnection model of Kadomtsev [42]. Previously, the model of Kolesnichenko and Yakovenko [45] has been used in `EBdyna.go` for the research of Jaulmes et al. [41]. However, this model does not provide an analytic relation between the coordinate and the enclosed surface, i.e. the flux. Determination of the helical flux requires a numerical integral. Thus the electric and magnetic fields cannot be expressed explicitly and are not necessarily continuous during the complete sawtooth (reconnection and recovery phase).

`EBdyna.go` has been implemented with a new model for the sawtooth reconnection. This new model is based on private communication with de Blank [15] and is ‘electromagnetically consistent’, i.e. evolving **E** and **B** fields are consistent, yet the MHD force balance is not.

Here we describe the newly implemented of de Blank [15], with the focus on the implementation, rather than presenting a complete derivation. Some assumptions have to be made to avoid singularities in **E** and **B** fields and also because the time evolution of the sawtooth is not precisely known.

- Potential and flux are continuous in space, thereby avoiding infinitely large electric or magnetic fields.
- Magnetic fields outside the mixing radius r_{mix} remain unchanged in accordance with the

3.4. SAWTOOTH MODEL

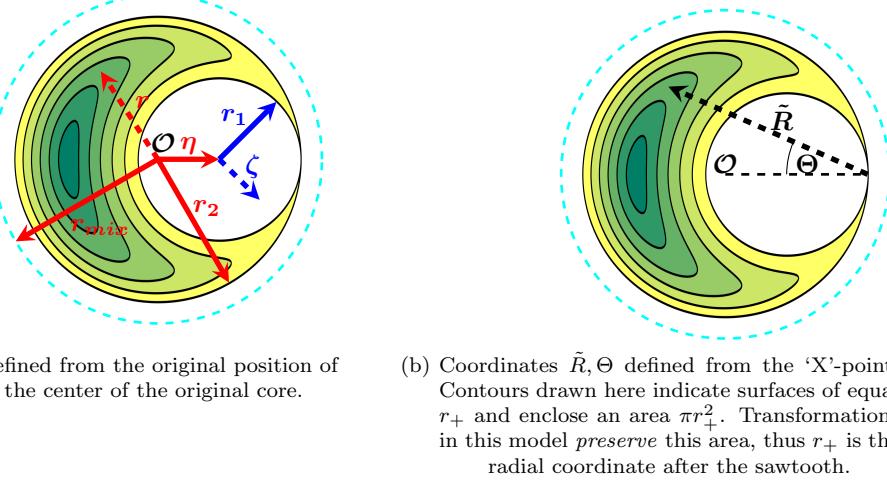


Figure 3.4: Coordinates used in the sawtooth model in \tilde{x}, \tilde{y} -space. Colors indicate the origin of each vector. Dashed lines are vectors with arbitrary length. The direction of each vector is arbitrary, except for η . The cyan circle indicates the eventual size of the island and has radius r_{mix} .

Kadomtsev model.

- There is no interaction of (fast) particles with the sawtooth. Driven or damping forces on magnetic configurations are *not* present (i.e. we exclude Landau damping). However, magnetic fields do affect (fast) particles based on the Lorentz force.
- The sawtooth consists of a pure $n = 1, m = 1$ harmonic.
- Coupling between modes is not included, i.e. 3D fields do not interact with the sawtooth. This is reasonable since the sawtooth consists of a pure $n = 1$ harmonic and the applied 3D fields have higher toroidal mode numbers n .
- The $m = 2$ harmonic of a sawtooth is not included. This component can be responsible for island creation at $q = m/n = 2/1$ during experiments, but will not appear in our modelling. In this model, sawtooth effects are only present in the area $r < r_2(t) \leq r_{mix}$.
- During reconnection, the original core moves with constant speed in \hat{x}, \hat{y} -space, i.e. r_1 is constant.
- After reconnection, a ‘recovery’ phase sets in where a ‘deformation parameter’ κ (introduced later) linearly decreases to 0, thereby restoring axisymmetry.
- After recovery, the (now poloidally symmetric) helical flux ψ_* stops evolving.
- After recovery, electric fields (suddenly) disappear.
- The core is ideally conducting during the sawtooth reconnection (no resistivity), since the temperature is high. Therefore the toroidal component of the electric field E_φ can be determined with $\mathbf{B} \cdot \mathbf{E} = 0$.

Now it is time to describe the 2D geometry of the sawtooth region in \hat{x}, \hat{y} -space. Consider figure 3.4, where the sawtooth has been shown during the reconnection phase. Three regions exist during the reconnection phase:

1. the original core ($\zeta < r_1$),

2. the ‘unaffected’ area ($r > r_2$).

3. the island $r \leq r_2$ and $\zeta > r_1$,

For the unaffected area (2), the ψ_* and thus the ψ profile are functions of the radial coordinate r (red). The original core (1) has moved outwards with its center at $\eta \equiv r_2 - r_1$. Conservation of helical flux requires the original ψ_* profile to hold within area (1). It is useful to define the radial coordinate $\zeta \equiv \sqrt{(\tilde{x} - \eta)^2 + \tilde{y}^2}$ within this core.

More creativity is required to obtain the helical flux within the island (3). If found, we can determine ψ_* within the island at any position \hat{x}, \hat{y} and given time. First we introduce polar coordinates \tilde{R} (distance to the ‘X’-point) and the ‘helical angle’ Θ as shown in figure 3.4(b). Note that these ‘live’ within \tilde{x}, \tilde{y} -space and are shifted by the (time dependent) ‘X’-point. The separatrix (line between island and original core) consists of tangent circles C_1 and C_2 , with radii r_1 and r_2 respectively. These circles are given by $\tilde{R} = 2r_1 \cos(\Theta)$ and $\tilde{R} = 2r_2 \cos(\Theta)$ respectively.

Between C_1 and C_2 , i.e. within the island, we define contours of constant helical flux ψ_* , dubbed $C(r_+)$. The label r_+ is given by:

$$r_+^2 = \kappa_r^{-1} \left(\tilde{R} - 2r_1 \cos(\Theta) \right) \left(\tilde{R} - 2r_2 \cos(\Theta) \right) + r_3^2, \quad (3.5)$$

with the parameter κ_r given by:

$$\kappa_r = (r_2 - r_1)/(r_1 + r_2). \quad (3.6)$$

Here we note that $r_+ = r_3$ on the separatrix (on C_1 and C_2). This is in accordance with the helical flux at the separatrix, which is $\psi_-(r_1) = \psi_-(r_2) = \psi_+(r_3)$ (see figure 2.18). The mid of the island, the ‘O’-point, corresponds with $r_+ = 0$. One can verify that the area enclosed by $C(r_+)$ is exactly πr_+^2 .

The construction of ψ_+ yields discontinuities of $\nabla \psi_*$ at the separatrix. In practice, these would resemble singular current sheets. However, Kolesnichenko and Yakovenko [45] argue that these should not occur during the reconnection phase. This can be ‘repaired’ using an area-preserving deformation of the contours labeled with r_+ . Thus the shape of the contours C_+ change in shape during reconnection, but preserve the area πr_+^2 . The preservation of area ensures the labels r_+ correspond to the radial coordinate after reconnection r_3 . Then the helical flux can be determined by $\psi_+(r_+)$.

Here we introduce the ‘deformation’ parameter κ , a function of Θ . Now we choose the transformation $\tilde{R}^2 \rightarrow \tilde{R}^2 + \kappa(r_+^2 - r_3^2)$. We obtain from equation (3.5) the following equation for r_+ :

$$r_+^2 = \kappa_r^{-1} \left(\sqrt{\tilde{R}^2 + \kappa(r_+^2 - r_3^2)} - (r_1 + r_2) \cos(\Theta) \right)^2 + r_3^2 \sin^2(\Theta), \quad (3.7)$$

which we can rewrite to a quadratic equation by defining $\rho = \sqrt{r_+^2/r_3^2 - \sin^2(\Theta)}$.

Some might be glad, whilst others may be disappointed, that we do not discuss the mathematics in great detail. Cutting some corners we see that to solve the quadratic equation, one needs to determine the following sets of quantities:

$$c = \kappa_r^{-1} + \kappa - \frac{\tilde{R}^2}{r_3^2 \cos^2(\Theta)}, \quad (3.8)$$

$$\rho = \frac{c \cos(\Theta)}{1 + \sqrt{1 + (\kappa - \kappa_r)c}}, \quad (3.9)$$

$$r_+ = r_3 \sqrt{\rho^2 + \sin^2(\Theta)}. \quad (3.10)$$

Now we introduce the radial gradient of ψ_i , denoted as $\nabla_i \psi_*$. Here $i = 1$ or 2 , corresponding to the gradient in r_1 or r_2 (in area (1) or area (3)). Next we write these gradients as $\nabla_i \psi_* =$

3.4. SAWTOOTH MODEL

$\psi'_i \mathbf{n}_c = \frac{d}{dr}(\psi_-(r_i)) \mathbf{n}_c$, with \mathbf{n}_c the normal vector pointing ‘outwards’ on the contour enclosing the island (3). The gradients normal on contour C_i are given by

$$\nabla_i \psi_* = \left(\frac{1}{\psi'_i} + \left(\frac{r_2}{\psi'_2} - \frac{r_1}{\psi'_1} \right) \frac{\kappa - \kappa_c}{2r_i} \right)^{-1} \mathbf{n}_c, \quad (3.11)$$

where κ_c is defined by

$$\kappa_c \equiv \frac{2}{r_1 + r_2} \frac{\psi'_1 + \psi'_2}{\frac{\psi'_2}{r_2} - \frac{\psi'_1}{r_1}} = \frac{2}{r_1 + r_2} \frac{\psi'_1 + \psi'_2}{q_1 - q_2}, \quad (3.12)$$

and called the ‘reconnection parameter’ [15]. Continuity of $\nabla \psi_*$ is ensured during reconnection when $\kappa = \kappa_c$, i.e. when deformation and reconnection are coupled in an area-preserving manner. We will come back to this condition when looking at the continuity of the potential, which is required since current sheets need to be absent [45].

Also the toroidal flux (coordinate) χ is conserved [42], which yields

$$\psi'_1 = q_2(1 - q_1)r_1, \quad (3.13)$$

$$\psi'_2 = q_1(1 - q_2)r_2, \quad (3.14)$$

where q_i is determined with interpolation of the original $q(r)$ -profile at r_i . With equation (3.13) one can determine the parameter κ_c and thereby κ at any time during the reconnection phase.

Now we present the potential Φ and flux ψ for the three regions. One can find ψ based on the position in \tilde{x}, \tilde{y} -space, the time dependent parameters r_1, r_2 and r_+ from equations (3.8) to (3.10).

$$\psi_* = \begin{cases} \psi_- \left(\sqrt{(\tilde{x}^2 + \tilde{y}^2)} \right), & r > r_2 \rightarrow \text{unaffected area} \\ \psi_- \left(\sqrt{(\tilde{x} - \eta)^2 + \tilde{y}^2} \right), & \zeta < r_1 \rightarrow \text{original core} \\ \psi_+(r_+), & \text{otherwise} \rightarrow \text{island} \end{cases} \quad (3.15)$$

$$\Phi = \begin{cases} 0, & r > r_2 \\ (r_2 - r_1) \tilde{R} \sin(\Theta), & \zeta < r_1 \\ \Theta \left[\frac{1}{2} (r_+^2 - r_3^2) (\dot{\kappa} - \dot{\kappa}_r) + (\kappa - \kappa_c) r_3 \dot{r}_3 \right] \\ + \sin(\Theta) \left[\dot{r}_2 \tilde{R} - (r_1 \dot{r}_2 + r_2 \dot{r}_1) \cos(\Theta) + \dot{r}_3 r_3 \rho \right], & \text{otherwise} \end{cases} \quad (3.16)$$

The potential Φ presented here is zero by definition at the $\Theta = 0$ line and thus $\Phi = 0$ in the unaffected area $r > r_2$. With the integral over the contours $C(r_+)$ (constant ψ_*), we find the potential to be

$$\Phi = \dot{r}_2 \tilde{R} \sin(\Theta) - \frac{\partial}{\partial t} \int_{C(r_+)} \frac{1}{2} \tilde{R}^2 d\Theta, \quad (3.17)$$

where the first term describes the velocity of the ‘X’-point, i.e. the velocity of the (\tilde{R}, Θ) coordinate frame. This is the ‘stream function’ for a flow with velocity \dot{r}_2 in the $\Theta = \pi$ direction.

The potential within the original core is given by $(\dot{r}_2 - \dot{r}_1) \tilde{R} \sin(\Theta)$. Similarly to the expression for the potential at the ‘X’-point, this expresses the rigidly moving original core with velocity $\dot{\eta} = \dot{r}_2 - \dot{r}_1$. At the separatrix the potential is given by

$$\Phi = \begin{cases} \Theta(\kappa - \kappa_c) r_3 \dot{r}_3 + (\dot{r}_2 - \dot{r}_1) r_1 \sin(2\Theta), & C = C_1 \\ \Theta(\kappa - \kappa_c) r_3 \dot{r}_3, & C = C_2, \end{cases} \quad (3.18)$$

which should be continuous both at the separatrix and at $r = r_2$ (and thus 0 here). Thus during the reconnection, when a separatrix is present, we require $\kappa = \kappa_c$ for electromagnetically consistent fields (i.e. a continuous potential Φ). During the recovery phase $\dot{r}_3 = 0$, so the condition $\kappa = \kappa_c$ does not need to be satisfied any longer.

Note that the condition $\kappa = \kappa_c$ is an artifact of the area preserving transformation $\tilde{R}^2 \rightarrow \tilde{R}^2 + \kappa(r_+^2 - r_3^2)$. Benefits of the chosen transformation are the explicit expression for r_+ and the finite velocity at $R = 0$ when the reconnection phase ends. Once the reconnection is completed the helical flux is not poloidally symmetric ($\kappa = \kappa_c$). A second phase follows, where poloidal symmetry is restored. In this ‘recovery phase’ $r_1 = r_2 = r_3 = 0$, thus κ can decrease although κ_c remains constant.

Up to now the model has a defined evolution in terms of shape and amount of reconnection. But the speed of the reconnection and recovery is still arbitrary. In this work the choice is made to have a linear decrease of r_1 during the reconnection phase (100 µs), followed by a linear decrease of κ during the recovery of poloidal symmetry (200 µs) (based on previous studies [41] and literature [96]).

Electric and magnetic fields can be extracted by $\mathbf{E}_{\text{pol}} = -\nabla\Phi$ and $\mathbf{B}_{\text{pol}} = \frac{1}{R}\nabla\tilde{\psi} \times \mathbf{e}_\varphi$. `EBdyna.go` provides these gradients by determining Φ and $\tilde{\psi} = \psi_* + \chi$ at the particle positions and at positions $R + \epsilon, Z$ and $R, Z + \varepsilon$, with $\epsilon, \varepsilon \sim 1 \times 10^{-9}$ m $\ll 1$. Implementation of this sawtooth model, parameters and the issues that arose are discussed in appendix C.4.

With this sawtooth model we can determine ψ_* (and thus ψ) and Φ in \hat{x}, \hat{y} space. Here we emphasize that straight field line-coordinates are necessary for the transformation to this space, i.e. R, Z, φ coordinates of the particle are ‘mapped’ to \hat{x}, \hat{y} space, where the helical flux ψ_* and Φ are determined. In figure 3.5 these two quantities are shown in both R, Z and \hat{x}, \hat{y} space, showing that the sawtooth mode is not circular in R, Z space.

The equilibrium determines the profiles of χ and θ . Hence the mapping from R, Z to \hat{x}, \hat{y} space (and vice versa) is also equilibrium dependent.

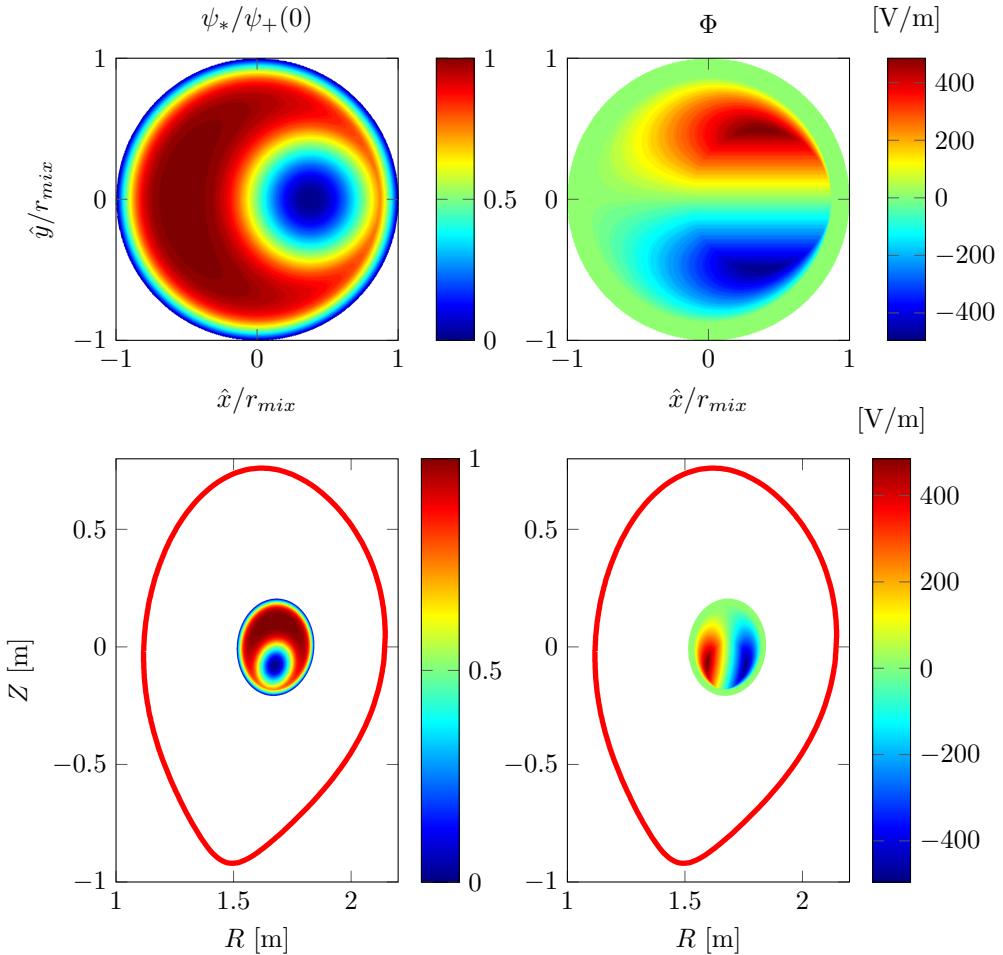


Figure 3.5: Comparison of the helical flux ψ_* and potential Φ in both the R, Z (at $\varphi = \pi/2$) and \hat{x}, \hat{y} space when reconnection is at 30 % (in time). The distorted shape of both in R, Z -space makes the importance of straight field line-coordinates apparent. Note that the helical flux ψ_* is normalized to the maximum of the helical flux $\psi_+(r = 0)$.

Chapter 4

Results

4.1 Particle distributions

Results for `EBdyna-go` shown in this chapter are based on two distributions:

- NBI-distribution, ionized deuterium particles originating from an NBI beam. The distribution is ‘steady state’, so some particles in this distribution are recently ionized, whilst others have already been slowed down and travelled radially (outward). Losses within a certain time interval are cancelled by radially travelling particles and newly ionized NBI particles such that the distribution remains unchanged. This distribution is the result of the `TRANSP` code. Unfortunately, documentation of the equilibrium and NBI-beams is unavailable (there are multiple NBI’s on AUG). This NBI distribution consists of 2.0×10^6 particles.
- ‘Test distribution’, which is used with the intention to cover phase space evenly. Or at least, sample the complete phase space. It is hard to come up with a distribution function that evenly spreads in μ and p_φ . Therefore a homogeneous distribution in real space \mathbf{x} and ‘flat’ energy distribution between 20 – 60 keV has been chosen. Both position and energy are randomly chosen for all 6.0×10^6 particles. The pitch $\chi_p = v_{\parallel} / |\mathbf{v}|$ is also chosen at random between -1 and 1.

In order to compare simulations with or without sawtooth (ST) or with different 3D fields, initialization of these distributions is based on the 2D equilibrium. Initial position and velocity are constructed by the following steps:

1. Determine the magnetic field in the *2D equilibrium* at the gyro center $\mathbf{B}(\mathbf{x}_{gc}) \equiv \mathbf{B}_{gc}$.
2. Choose a random gyro phase¹, i.e. a ‘normal’ vector \mathbf{N} (perpendicular to $\mathbf{b}_{gc} \equiv \mathbf{B}_{gc} / |\mathbf{B}_{gc}|$).
3. Determine $|\mathbf{v}|$ from E_{kin} .
4. Determine $v_{\perp} = \sqrt{|\mathbf{v}|^2 - v_{\parallel}^2}$ (always positive), with v_{\parallel} from the distribution.
5. Determine $\mathbf{v} = v_{\parallel} \mathbf{b}_{gc} + v_{\perp} \mathbf{N}$ (in R, Z, φ components).
6. Determined the initial position $\mathbf{x} = \mathbf{x}_{gc} - \frac{m \mathbf{v} \times \mathbf{b}_{gc}}{Z_i e |\mathbf{B}_{gc}|}$.
7. The velocity is ‘assumed’ to be at half the time step back.

Although this scheme does not result in the precise positioning of the gyro center, one can safely assume the small error made does not disturb the distribution too much.

¹ The choice is made to generate a random gyro phase with a constant seed. The gyro phase repeats itself every 2000 particles. In this manner a simulation is reproducible.

4.1. PARTICLE DISTRIBUTIONS

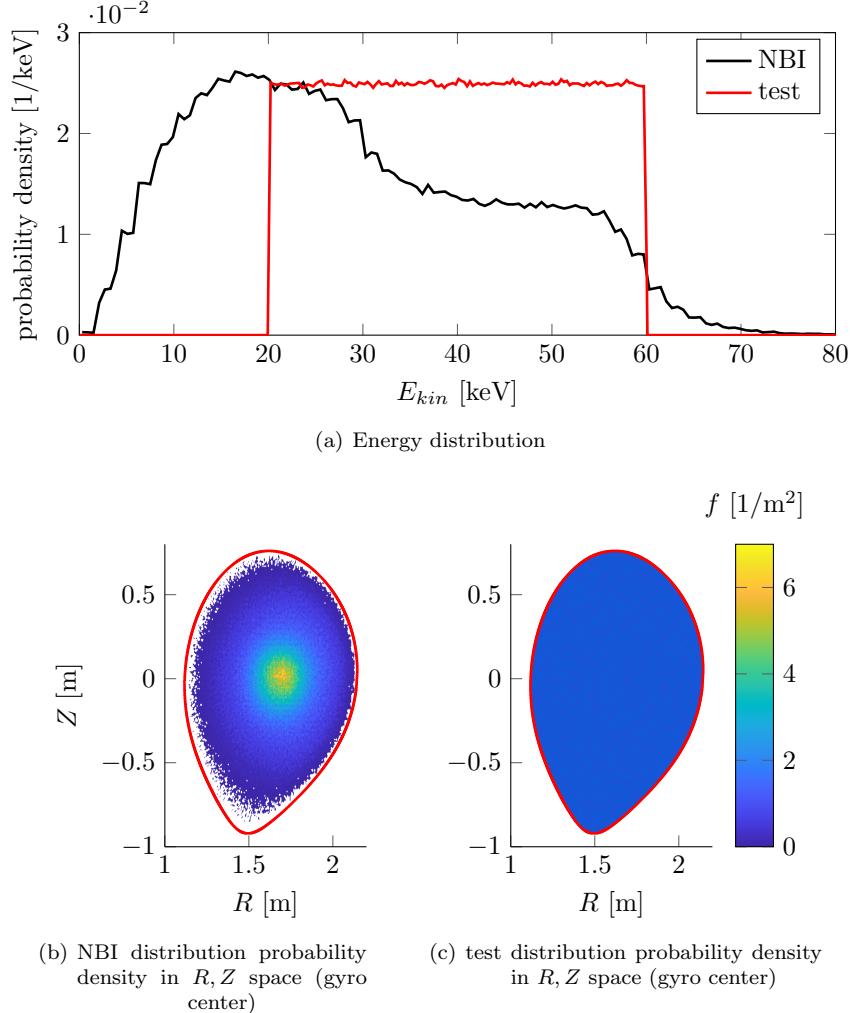


Figure 4.1: Probability density f for the two distributions used. This shows the probability density of both distributions in the poloidal projection R, Z as in energy. Both distributions are evenly distributed in the toroidal angle φ .

The probability densities f for energy and R, Z space for both distributions are shown in figure 4.1. The value f represents the probability to encounter a particle between E_{kin} and $E_{kin} + dE_{kin}$ in figure 4.1(a). Integration of f over the whole energy range adds up to 1. For example, the energy probability of the test distribution has a width of 40 keV, hence the probability density has an average value of 0.025 keV^{-1} . Similarly the surface integral in figures 4.1(b) and 4.1(c) adds up to 1.

These distributions can be projected on the phase space spanned by μ and p_φ as shown in figure 4.2. Here μ is normalized to B_0/E_{kin} , with B_0 the magnetic field on axis and p_φ is normalized to $Z_i e \psi_0$. The latter normalization factor is equal to the radial excursion from magnetic axis to LCFS, when not considering the velocity component v_φ .

Green, red and black lines enclose the population of trapped, co-current passing and counter-current passing particles respectively. For the co-current passing particles, p_φ can be less than -1, due to the velocity component in p_φ ($v_\varphi < 0$). Similarly for all counter-current passing particles $p_\varphi > -1$ holds ($v_\varphi > 1$). Close to the edge (right) we note that counter-current particles can have positive p_φ ($v_\varphi > 1$) and co-current particles always have $p_\varphi < 0$. Trapped particles can overlap with passing particles in a relatively small region.

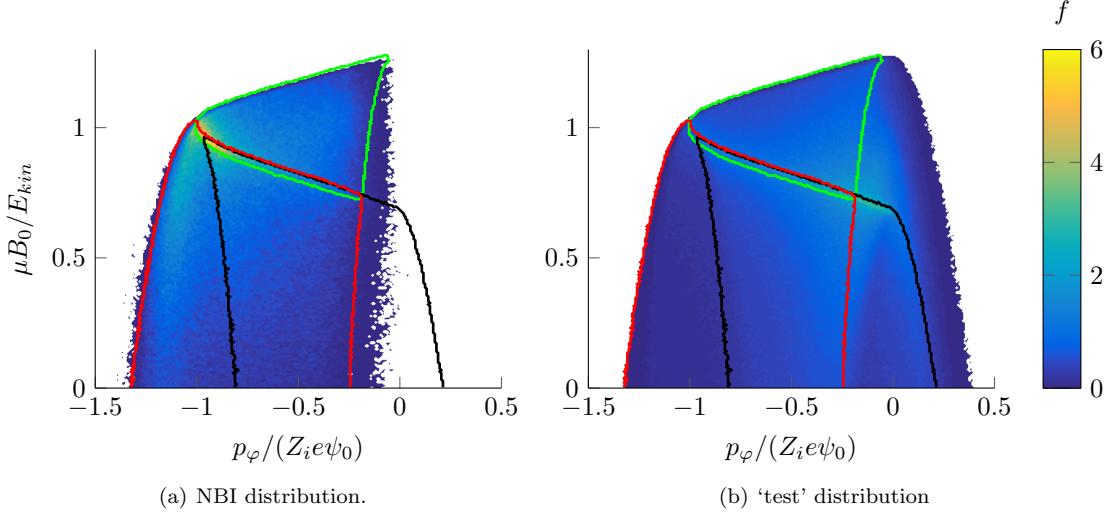


Figure 4.2: Probability density f shows a more even distribution for the ‘test’ distribution than for the NBI distribution. Lines indicate the regions with different particle categories based on the test distribution. At the top (high μ) trapped particles have a green boundary. The red line encloses co-current passing and the black line counter-current passing particles. Particles on the right, outside these boundaries, are not confined in the 2D equilibria (prompt-losses).

4.2 Losses in time and space

Some particles might be lost before completing an orbit. Those losses are dubbed ‘prompt losses’ and occur quite rapidly. In figure 4.3(a) the loss as function of time is shown for several 3D fields with the NBI distribution. Prompt losses occur in the first ~ 0.2 ms, after which the 2D simulation confines all remaining particles. Here a particle is considered lost when the field cannot be determined due to a grid point outside the LCFS. The field outside the LCFS is not obtained by FINESSE when solving the Grad-Shafranov-equation.

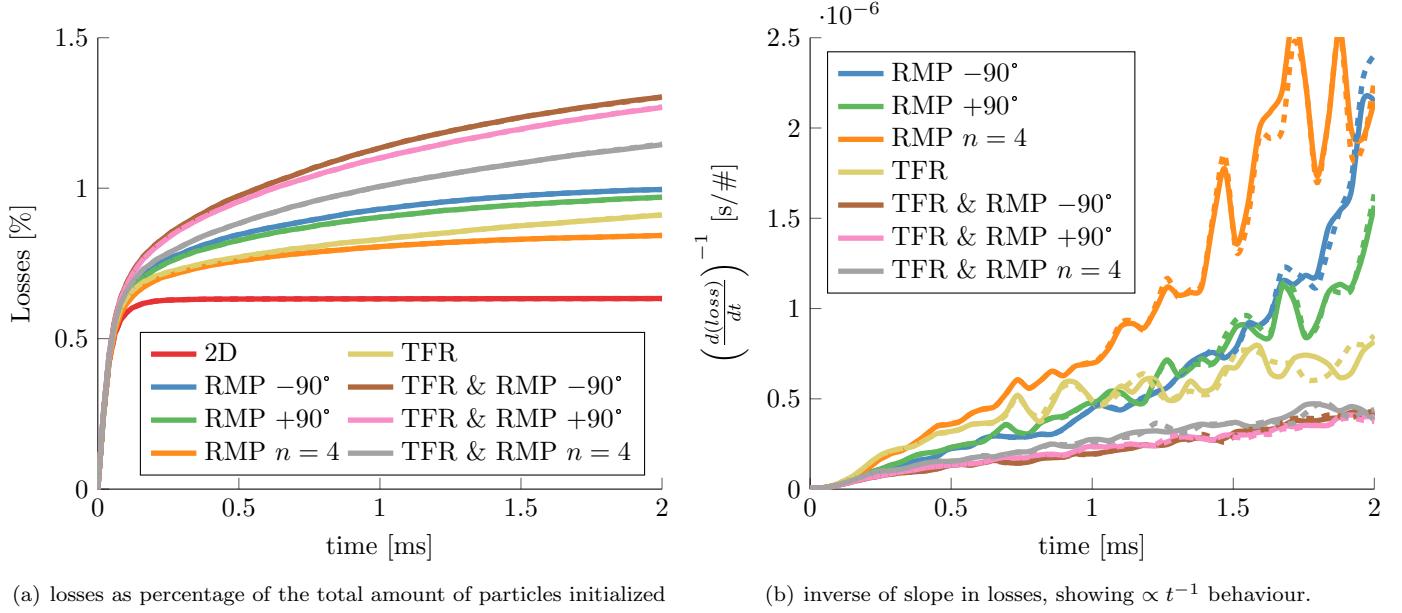
3D fields give rise to losses on longer time scales. Invisible in figure 4.3(a) are dashed lines, with color corresponding to the same 3D fields, but with a sawtooth crash in the first 300 μ s. The absence of any difference indicates there are little to none particles influenced by the sawtooth which end up being ‘ejected’ out of the plasma. Apparently the intact KAM-surfaces in between the mixing radius r_{mix} and the LCFS withhold particles from moving further outwards.

Shown in figure 4.3(b) is the inverse of the time derivative of the losses, i.e. the inverse of the slope in figure 4.3(a). Stochasticity gives rise to an initial $\propto t^{-1}$ behaviour, which holds during the complete simulation for the configurations with both ripple and RMP field.

Since a $\propto t^{-1}$ relation does not converge for $t \rightarrow \infty$ and the number of particles is finite, the relation ceases to exist if the ‘loss region’ in phase space is drained, e.g. the RMP $n = 4$ configuration (without TFR). In our understanding this is due to the spectrum in island size, approximately scaling with the inverse of the island size.

Positions of particles are frequently recorded to study the location before the particle is considered lost. For the TFR ($n = 16$) and RMP ($n = 2$) the distribution over the φ coordinate is shown for the NBI distribution in figure 4.4.

The 3D fields have been implemented with perfect periodicity (i.e. the TFR is periodic with $2\pi/16$ radians in the toroidal direction). Variations that are not periodic with this mode cannot be due to the 3D field, but require another explanation. Firstly, the recorded position could be the position before the ‘fatal’ time step, but could also have been maximal $100 \Delta t$ before crossing the LCFS. Secondly the initialization has a flat, but random φ distribution. The $n = 4$ RMP losses are not shown, due to the high level of noise (low number of ejected particles). Losses in



(a) losses as percentage of the total amount of particles initialized

 (b) inverse of slope in losses, showing $\propto t^{-1}$ behaviour.

Figure 4.3: Losses for the NBI-distribution in several 3D fields. Losses in equilibrium (2D) are particles initialized on ‘loss-orbits’. Both RMP and TFR fields result in losses over longer periods, where the combinations result in most losses. Losses show an $\propto t^{-1}$ behaviour at start. As the area of loss in phase space becomes depleted, the slope decreases and its inverse starts to diverge.

In this region the simulations with sawtooth (dashed) start to deviate.

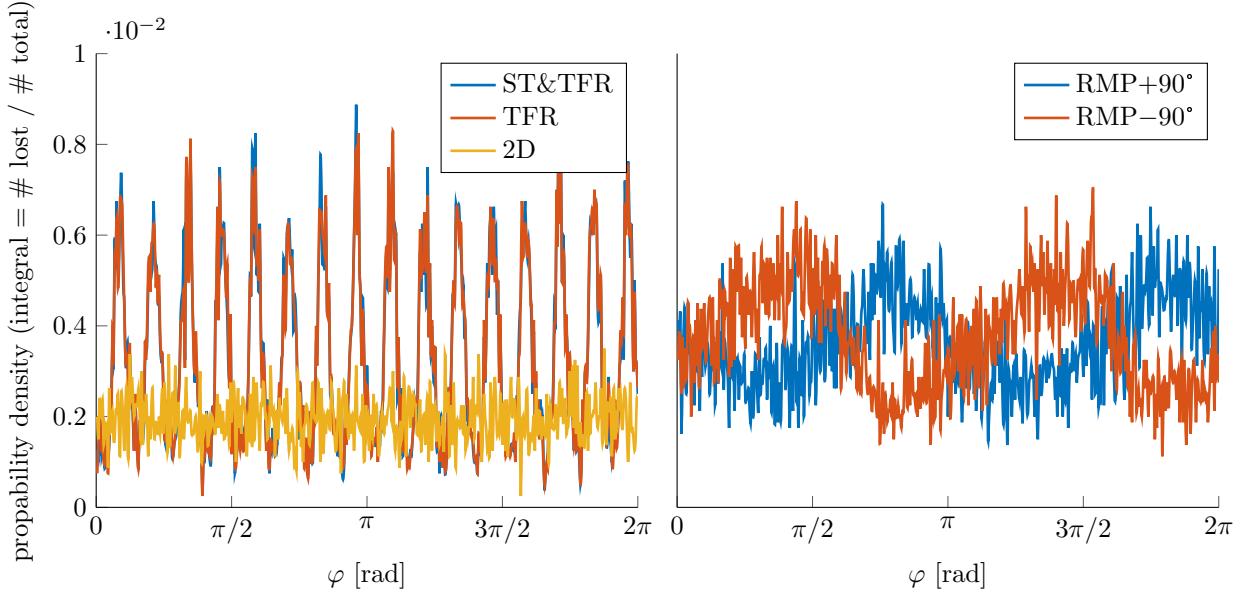


Figure 4.4: Losses as function of φ for the 3D fields (NBI distribution). These plots for simulations with a sawtooth are no different, so are only shown for the TFR. Clearly the $n = 16$ character of the ripple is shown (left) as well as the $n = 2$ character of the RMP fields (right). The parity causes particles to be lost at a toroidal angle difference of $\sim \pi$. Note that the toroidal position of losses particle is highly localized.

the 2D equilibrium are especially noisy for this reason, but shown nevertheless as a ‘baseline’.

Clearly visible are the dominant modes for the TFR ($n = 16$) and RMP ($n = 2$) in the toroidal position of the losses. Static 3D field could locally exceed the tolerance for plasma facing components, due to this high localized loss. To avoid localized losses due to RMPs, some experiments are performed in which the RMP fields are rotating [14, 84, 90, 91, 94]².

When a sawtooth is applied at the start of the simulation, there is little to no effect on the location of the losses. Nevertheless a small effect is shown in some figures up to now. Some particles might enter the sawtooth region, which lies far from the LCFS (see figure C.8). Thus a ‘synergy’, i.e. combined effect, of RMP and sawtooth appears to be present. Eventually, this appeared to be a numerical error and will be discussed in section 4.6.

A broad variation in toroidal angle φ of the position where particles are lost is present for the $n = 2$ RMPs. The two $n = 2$ configurations lose particles at a toroidal difference of $\sim \pi$. Any small variation might lead to such a phenomenon, since particles are moving rapidly in the toroidal direction. We do not expect to find an explanation to relate the difference in $\Delta\Omega$ to the φ -coordinate of the losses.

4.3 Losses in phase space

A projection of losses is made on the μ, p_φ space in figure 4.5. Especially the variation in the initial φ is not taken in account and we therefore need to ‘bin’ particles and show the percentage of particles lost. This can be interpreted as a ‘chance’ of a particle to be ejected within the 2 ms simulation as function of μ, p_φ . Boxes with less than 10 particles do not have representative statistics and are therefore coloured white.

When the ripple is taken into account, losses are present in the high- μ region (figures 4.5(a), 4.5(e) and 4.5(f)). These are particles which are ‘deeply’ trapped and thus possibly ‘ripple trapped’. Counter-current passing particles appear to be well confined when only the TFR applies (little loss near the black line in figure 4.5(a)).

RMP fields also affect trapped particles, but then the less deeply trapped (low μ). RMPs seem to be more effective in ejecting passing particles, now also the counter-current passing. A band structure is visible in the co-current region ($p_\varphi \approx -0.4Z_i e\psi_0$). Comparing the $n = 2$ and $n = 4$ ejection of co-current particles, one can observe 2 or 4 bands respectively.

The TFR and RMP seem to affect separate regions with little overlap. High μ losses are present with the TFR and lower μ and passing losses with the RMP. When both the TFR and RMP apply, all these regions are affected. The resulting high loss is consistent with figure 4.3. Variations between RMP configurations not very pronounced. Especially the two RMP $n = 2$ configurations are not that different.

To check if particles which are lost when the TFR is applied were ‘ripple trapped’, an ‘effective midplane’ is defined as discussed in appendix C.3.4. The idea was that particles which bounce more than three times, but only pass this midplane once or less could be identified as ripple trapped. The condition on the number of bounces is to exclude the population of prompt losses. Only a small fraction of the test distribution met all conditions (0.05 %) when the TFR applies. Some orbits of this population do drift downwards in a manner one expects from the ripple trapped loss. Unfortunately, not all of the orbits were indeed ripple trapped; we failed to reliably identify the ripple trapped population.

4.4 Resonant particle motion

4.4.1 Trapped particles

To quantify the radial transport, we again make use of the canonical toroidal angular momentum p_φ . The ratio of the bounce and precession frequency ω_b/ω_d is hard to distribute evenly as shown

²One could ‘easily’ implement a rotation of the RMP field by subtracting a time dependent angle when interpolating in the 3D grid. Note that this model rotates the coils instead of the current.

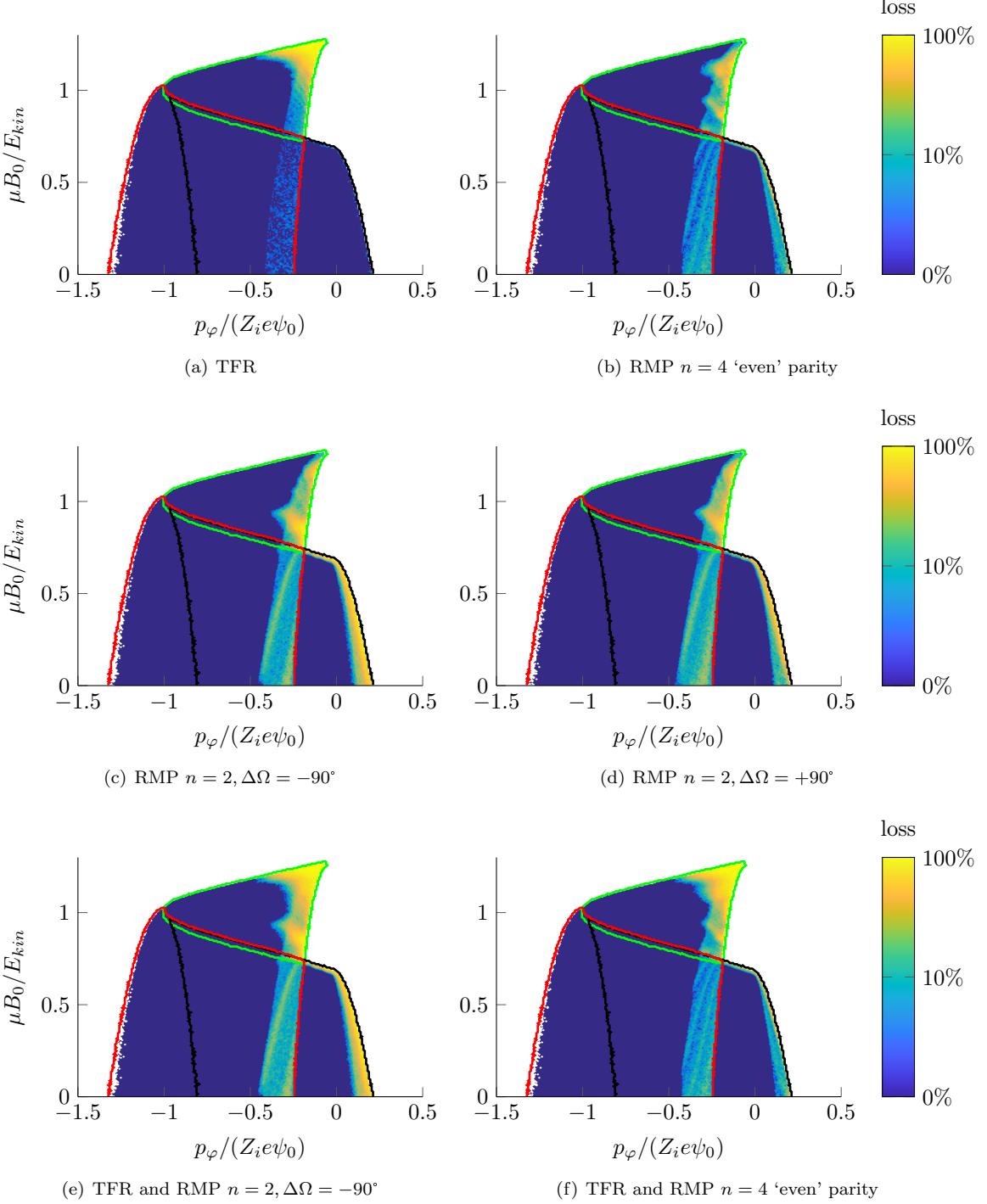


Figure 4.5: Loss in μ, p_φ phase space as percentage of the ‘bin’. The different parity of the two $n = 2$ modes shows hardly any difference in this phase space. The ripple (TFR) affects particles in another position of phase space, i.e. high μ . Lines in green, red and black enclose the region of trapped, co-current passing and counter-current passing particles respectively. Ejection of co-current passing particles by the RMPs shows a radial structure corresponding to the applied mode.

White coloured ‘bins’ have less than 10 particles.

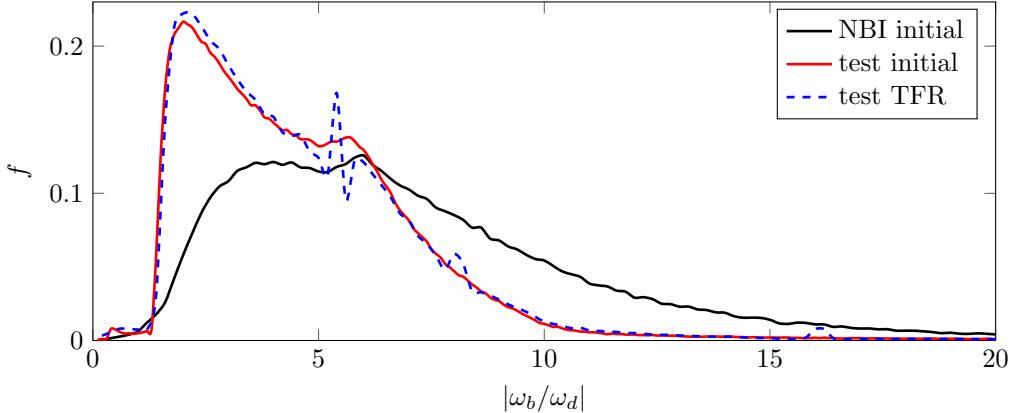


Figure 4.6: Probability density in $|\omega_b/\omega_d|$ of the two distributions. Although the NBI distribution shown a more evenly spread in $|\omega_b/\omega_d|$, the test distribution has been used since it has three times as many particles. Since ω_b is defined positive, the sign can change for ω_d . As AUG has negative sign trapped particles, the sign of ω_b/ω_d is negative as discussed in section 2.5 (given ω_d has been correctly determined). When a ripple is included, some particles change their bouncing behavior and a ‘bump’ shows up at $|\omega_b/\omega_d| = 8$ and another one at $|\omega_b/\omega_d| = 16$. This adopted distribution has only been examined with the test distribution.

in figure 4.6.

Later we found that the ω_b/ω_d distribution changed when a 3D field was applied. Only for the test distribution we shown this slightly altered distribution due to the TFR. Some particles tend to bounce more frequently in this case and get a value of $|\omega_b/\omega_d| = 8$ or 16.

Although the ‘test distribution’ is homogeneously distributed in R, Z, φ , energy and pitch, the distribution is less even in $|\omega_b/\omega_d|$ than the NBI distribution. Nevertheless we proceed with the test distribution since it has three times as many particles and might therefore produce better statistics. Proper normalization is necessary due to this uneven distribution.

The radial excursion p_φ is presented as function of ω_b/ω_d in figure 4.7 for the RMP $n = 2$ modes without ripple. Here we only consider particles which have not been ejected. These figures show the distribution in Δp_φ for each ‘bin’ containing similar values of ω_b/ω_d . Both positive and negative values of Δp_φ are present, representing outward and inward radial excursions respectively.

Evidently, the radial excursions show peaking at rational values for ω_b/ω_d . Especially at $|\omega_b/\omega_d| = 2$ a peak is shown, corresponding with $n = 2$. A slightly smaller peak is present at $|\omega_b/\omega_d| = 6$, consistent with the complementary mode [90].

When a ripple is applied we also have to consider $|\omega_b/\omega_d| > 10$, due to the small distribution of particles with $|\omega_b/\omega_d| = 16$. In figure 4.8(a) only the ripple is applied. Also here some specific values of $|\omega_b/\omega_d|$ are correlated with high radial excursions. With the TFR the correlation is less pronounced than with the RMPs.

When both a ripple and an $n = 2$ RMP apply, the peak at $|\omega_b/\omega_d| = 2$ is slightly decreased, but is still the dominant one. The graph gets more blurry for larger $|\omega_b/\omega_d|$.

Consider also the RMP configuration with an $n = 4$ mode, shown in figure 4.9. Also in this figure a correlation is visible between the radial transport and $\omega_b/\omega_d = n$. Secondary peaks show up at $\omega_b/\omega_d = 2$ and 6. Similarly to the $n = 2$ RMPs a blurring of the signal occurs when a ripple is applied. This strengthens the belief that resonant bouncing is an important effect of RMPs, rather than only stochasticity.

Note that some particle have radial excursions of $\Delta p_\varphi \approx \sim 0.05 Z_i e \psi_0$. This corresponds to roughly 5% excursions in the ψ coordinate. Note that ψ is not a linear coordinate, ρ comes closer to being a linear coordinate³. Since $\rho = \sqrt{\psi}$, we find a radial excursion of roughly 2.5% within

³Although ρ_{tor} is a closer to a linear coordinate than ρ .

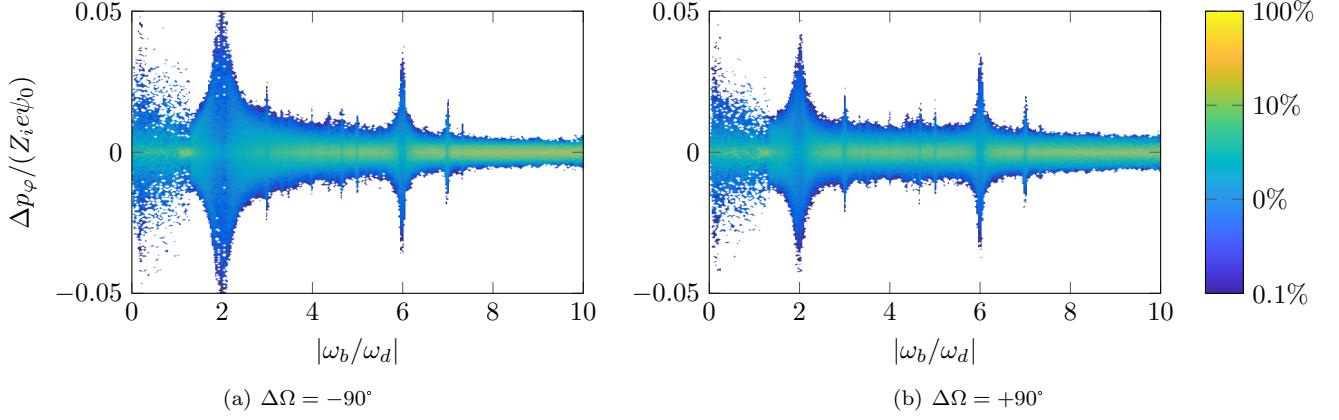


Figure 4.7: Radial excursion with $n = 2$ RMPs quantified by p_φ as function of ω_b/ω_d as percentage of ‘binned’ particles with similar ω_b/ω_d (test distribution). Percentages with less than 0.1 % have been filtered and appear white. Especially the rational values at 2 and 6 (main and complementary mode) have radial transport both inwards and outwards.

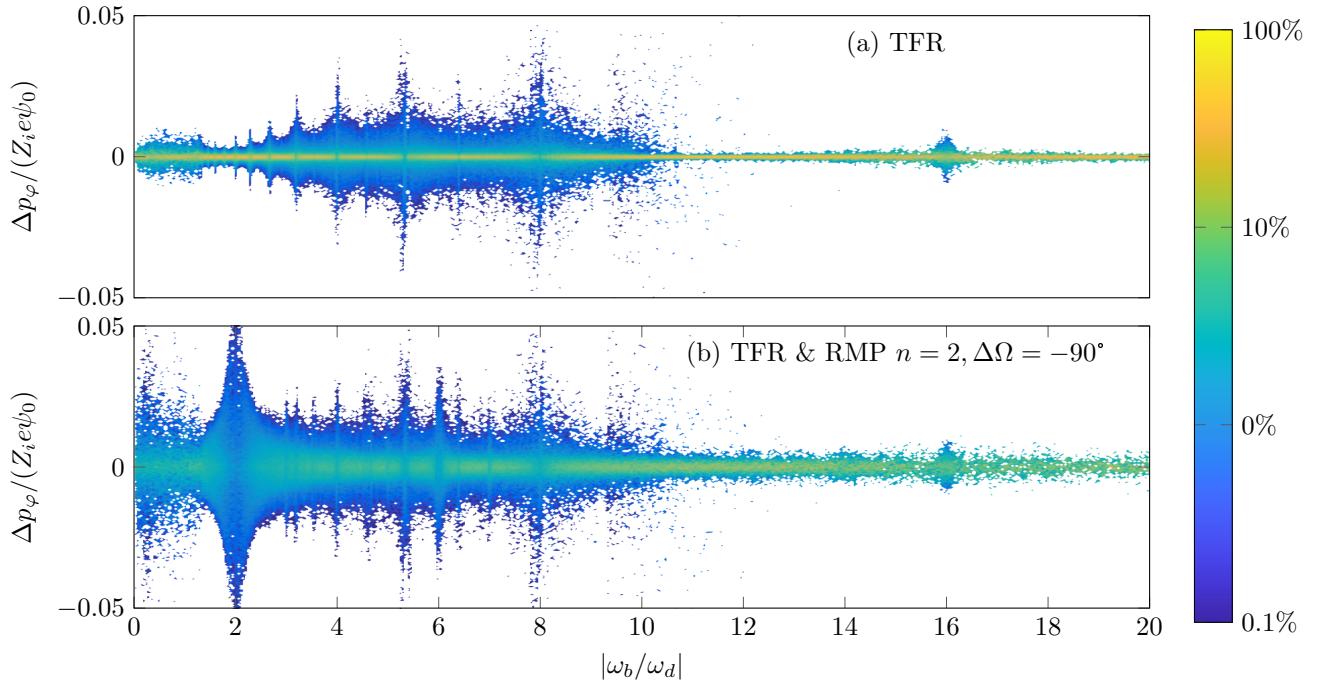


Figure 4.8: Radial excursion when a ripple is applied with and without RMP. Here p_φ is presented as function of ω_b/ω_d with a given percentage of particles with similar ω_b/ω_d . Percentages with less than 0.1 % have been filtered and appear white. Rational values of ω_b/ω_d appear to have increased radial excursions.

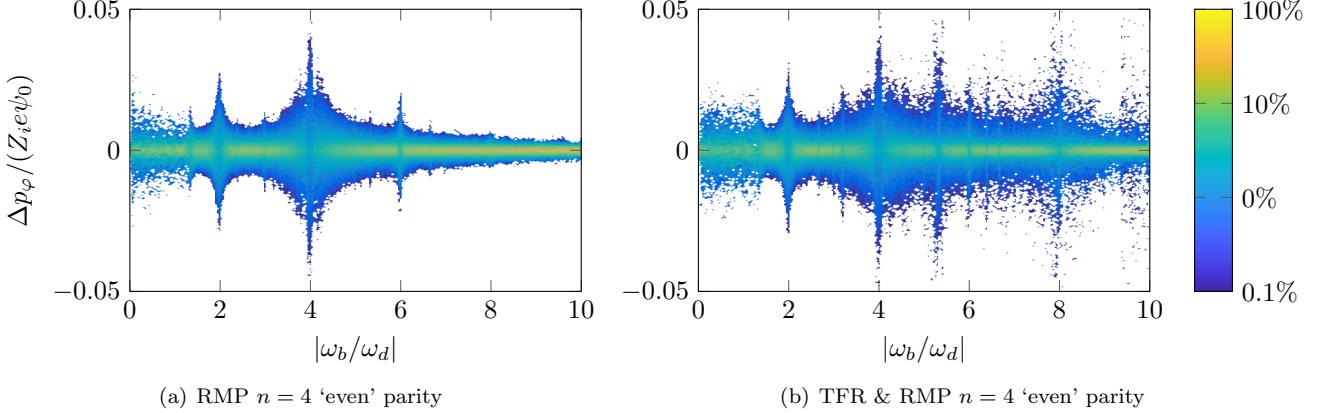


Figure 4.9: Loss as function of ω_b/ω_d as percentage of ‘binned’ particles with similar ω_b/ω_d made with the test distribution. The normalization corrects for the uneven distribution in ω_b/ω_d . Percentages with less than 0.1% have been filtered and appear white. (Bin sizes are kept constant.)

2 ms due to the ripple resonance.

The magnitude of the radial excursions have been examined as function of time. For this we select all trapped particles, which have not been ejected during the 2 ms and have a frequency ratio of $\omega_b/\omega_d \in (-10, 0)$. We take the root-mean-square value of p_φ over these particles and plot them as a function of time in figure 4.10.

An explanation of the decrease in $(\Delta p_\varphi)_{rms}$ has not been found. The double logarithmic plot shows an almost constant for the TFR simulation. This case has been fitted, revealing the time dependency: $(\Delta p_\varphi)_{rms} \propto t^{0.2}$. However, when the selection of ω_b/ω_d is made narrower around the $n = 2$ peak for the $n = 2$ RMP configurations, this power more than doubles. A similar results was obtained with the test distribution, i.e. $(\Delta p_\varphi)_{rms} \propto t^{0.2}$.⁴

4.4.2 Passing particles

Passing particles do not have the same ‘ripple resonance’. They are more likely influenced by stochasticity and the islands created by RMPs or the TFR. The average radial excursion, quantified by Δp_φ is shown in figure 4.11 as function of the average q -value, which is determined using an 1 ms orbit in the 2D equilibrium.

Locations where the $n = 2$ perturbation creates islands do show radial excursions for passing particles. Those particles on the inner side of the island travel outwards and thus have a positive change in p_φ and conversely, particles on the outer sides of the island move inwards. Note that the radial transport in combination with collisions does have a net deconfining effect.

Due to radial drifts (∇B and curvature B) the orbits are shifted for energetic particles. For co-current particles this shift is outwards and for counter-current inwards. This is apparent in the horizontal shift of the resonance in figure 4.11.

Correlation between trapped particles and the resonant q -surfaces is not observed. As discussed by [90], the resonance condition for trapped and passing particle is different.

4.5 Sawtooth redistribution

Radial particle and energy profiles are dominantly affected by the sawtooth, see figure 4.12. The probability distribution drops slightly near the edge when 3D fields apply. Any other variation

⁴The slopes of the test distribution is similar, but not the offset, i.e. a constant factor is found between the two distribution due to the total amount of particles.

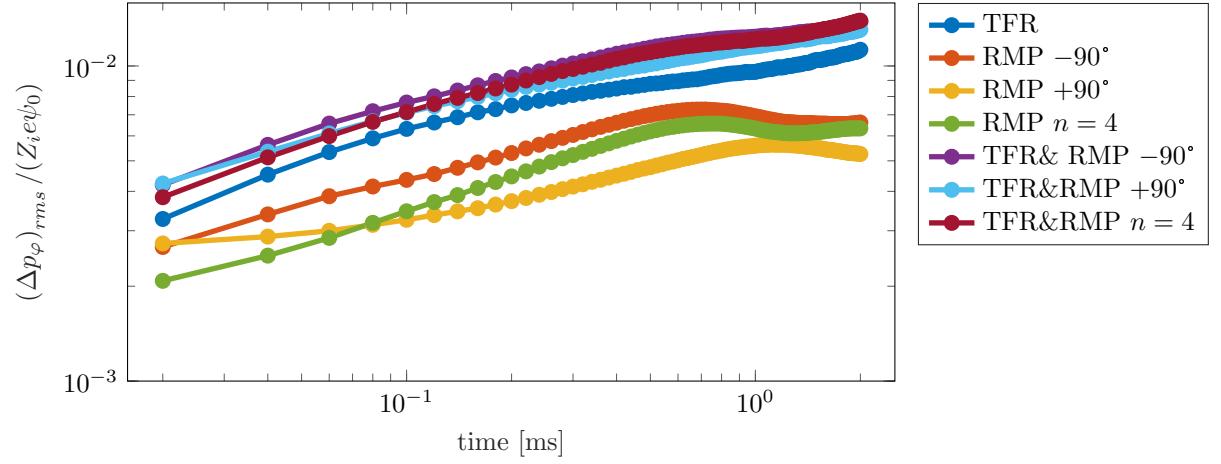


Figure 4.10: Evolution of the rms-value of p_φ for the selection of confined particles with $\omega_b/\omega_d \in (-10, 0)$ of the NBI-distribution. RMP fields do not always result in a monotonic increasing rms-value, which is unexpected. An explanation for the decrease is not found. The TFR seems to scale with $\propto t^{0.2}$, which is a lower power than diffusion. When particles in the narrow range around a $|\omega_b/\omega_d|$ peak are selected, the power (slope when plotted on a double logarithmic scale) increases.

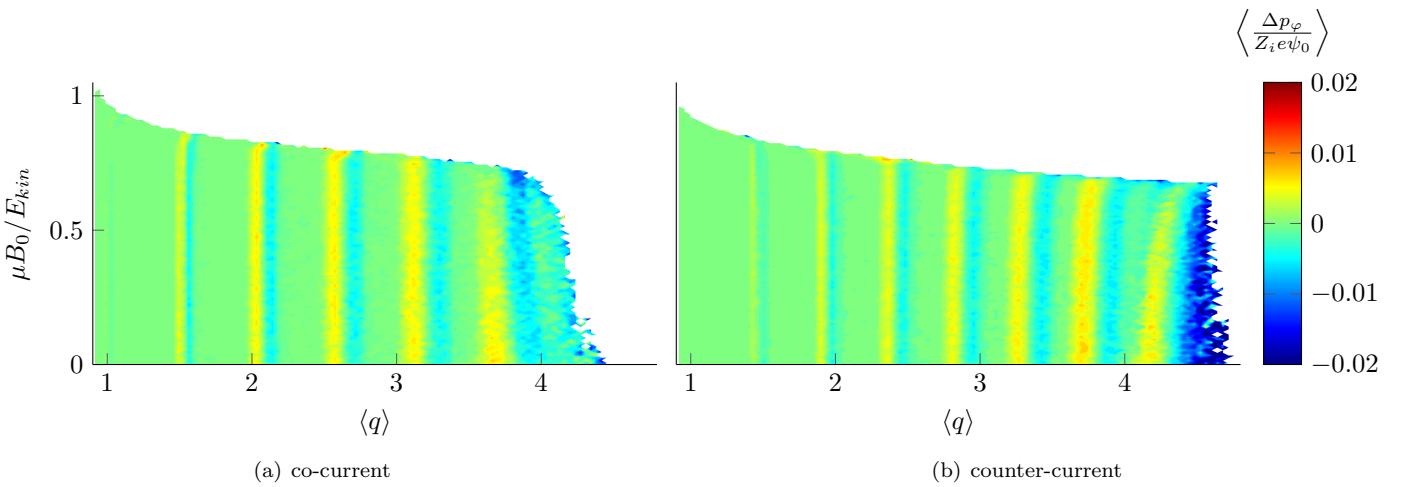


Figure 4.11: Radial transport, quantified by the averaged value of Δp_φ , is correlated to the island locations for passing particles. This has been shown here for the RMP $n = 2, \Delta\Omega = -90^\circ$ configuration. Here $\langle q \rangle$ is the average q value over the orbit in equilibrium. Only data from passing particles which are confined are shown.

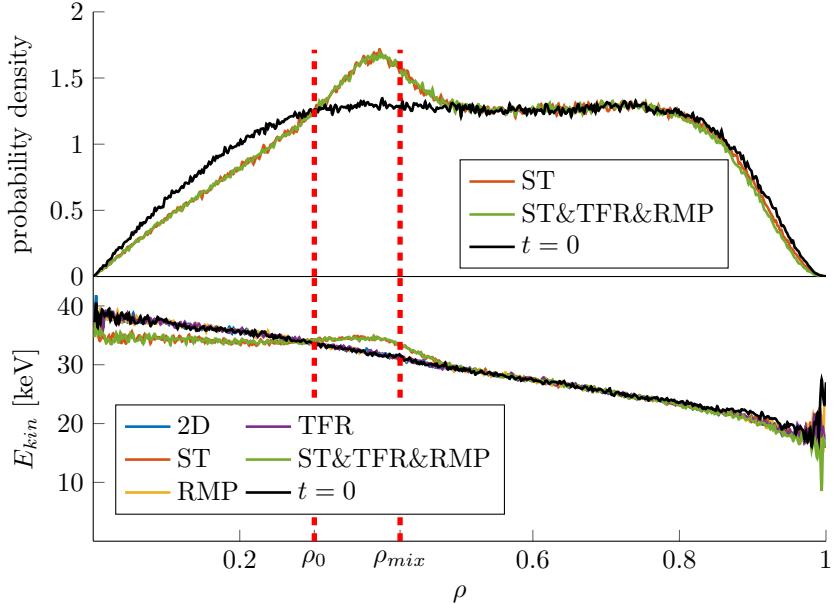


Figure 4.12: Redistribution of particles (top) and energy (bottom) of fast NBI ions after 2 ms. Significant redistribution due to the sawtooth (ST) is visible in both profiles, but 3D fields (TFR and/or an RMP with $n = 2, \Delta\Omega = -90^\circ$) hardly make a difference. Near the LCFS ($\rho = 1$) variations occur, in the energy profile, due to bad statistics (even in the initial distribution). This time the probability density has been normalized such that the integral represent the fraction of particles that are left after the simulation. This enables a quantitative comparison of the profiles. Dashed red lines indicate the positions $\rho_0 \equiv \rho(r = r_0)$ and $\rho_{mix} \equiv \rho(r = r_{mix})$.

due to 3D fields is barely visible within the noise. Statistics drop especially near the LCFS and near the magnetic axis.

A flattening of the energy profile is observed. The inversion radius, the position where average energy and density remain unchanged, is roughly r_0 .

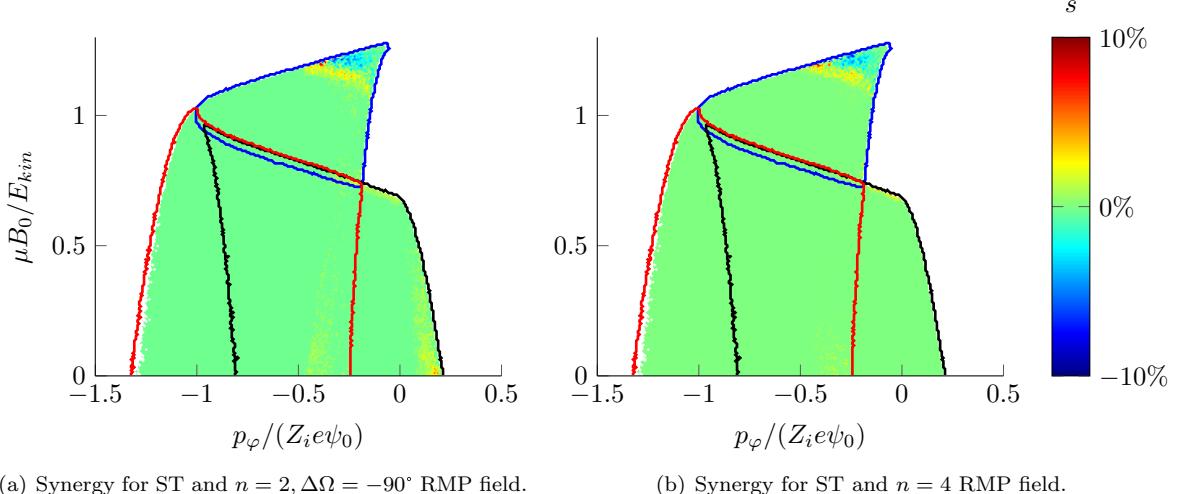
4.6 Synergy

One of the research questions considered synergy of 3D fields and the sawtooth. The focus lies on whether or not we can consider the two effects separately. In this work we have only considered the synergy of losses. We examine if there exist a part in phase space which is affected specifically when 3D fields and the sawtooth are considered.

In order to assess this we introduce a synergy parameter s . This parameter is set to 1 if a particle is not lost when situation A *or* B applies, but is lost when A and B apply simultaneously. Similarly it is set to -1 if the combination of A and B simultaneously does not result in a loss, whilst both A and B separately do lose the particle. Thus the *synergy* is deconfining (positive s) or confining (negative s). As seen in figure 4.13, both positive and negative areas are present in phase space.

Particles with a positive or negative synergy parameter s for the $n = 2, \Delta\Omega = -90^\circ$ case has been investigated in more detail. This revealed that the alleged ‘synergy’ was a round-off error.⁵

⁵When a sawtooth is applied in the simulation all particles outside the sawtooth region will obtain a \mathbf{B} field based on two interpolations, one in the 2D field \mathbf{B}_{2D} and the other in a 3D grid with the perturbation \mathbf{B}_{3D} . At the start of the sawtooth model a linear time based interpolation of fields is performed, due to instability in some parameters, e.g. κ_c (see appendix C.4). Particles outside the sawtooth region experienced a slightly different


 (a) Synergy for ST and $n = 2, \Delta\Omega = -90^\circ$ RMP field.

 (b) Synergy for ST and $n = 4$ RMP field.

Figure 4.13: Regions of positive (synergy is deconfining) and negative (synergy is confining) areas are visible in phase space. Here the synergy parameter s characterizes the deconfining effect of the 3D fields and the sawtooth. All simulations had the TFR on. Particles are binned according to the initial position in phase space with only the TFR field. Blue dictates the region of trapped particles, red the region of co-current particles and black regions of counter-current particles.

The round-off error in combination with the orbit being close to the edge and stochasticity of the magnetic field is responsible for the artefact. A second simulation, where the round-off error had been avoided, showed that no synergy is present in the losses. Avoiding the round-off error required a consistent scheme for particles outside the sawtooth region which is exactly the same whether or not a sawtooth is simulated.

Despite this unfortunate round-off error, this does provide an underestimate of the unavoidable numerical errors `EBdyna.go` will make. Non-zero values in figures 4.3(b) and 4.4 do show parts of phase space sensitive to every possible model parameter.

numerical scheme due to this time based interpolation. Although it has been performed with two identical fields, say \mathbf{B}_1 , a round-off error in calculating $\mathbf{B} = \gamma\mathbf{B}_1 + (1-\gamma)\mathbf{B}_1$ resulted in $\mathbf{B} \neq \mathbf{B}_1$ (γ is a time dependent parameter).

Chapter 5

Discussion

Simulations and models have yet to provide a satisfactory explanation for ‘density-pumpout’ found when using RMPs [62]. Previous particle simulations have often used guiding-center codes or hybrid codes, e.g. [6, 47, 81, 90]. We have presented results from a full-orbit code, which is required for accurate modelling in rippled or perturbed fields, i.e. to correctly find the toroidal coordinate where the bounce occurs [57] and cope with the high gradients during the sawtooth collapse. Other codes exist with collisional models and the plasma response, but rarely with both and rarely in combination with a full-orbit code. `EBdyna.go` distinguishes itself from other codes with a model for the sawtooth collapse.

Radial transport has been quantified using a ‘new parameter’, the ratio of bounce and precession frequency $|\omega_b/\omega_d|$. This parameter allows the characterization of the ripple resonance discussed in [58, 90]. Strong correlations have been found between this parameter and both the main and complementary mode of the RMP field (n and $8-n$) as argued by [90]. To our knowledge, other studies have not shown this ripple resonance for the RMP field. The resonance condition seems to be blurred when both RMPs and the ripple are applied, as if a mixing of the modes broadens the bandwidth of resonance. Radial excursion of trapped particles (which are not in the stochastic edge region) is governed by the ripple resonance, rather than movement within an island. Passing particles however are not influenced by this resonance and follow the island.

Although `EBdyna.go` can be used to study fast particle confinement, it fails to produce a reliable heat load on plasma facing components. Estimations of power loss and heat load would require some source, e.g. α -particles, or NBI particles and a collisional model. On top of that `EBdyna.go` lacks a reliable magnetic field outside the LCFS. `FINESSE` only determines the equilibrium (by solving the Grad-Shafranov-equation) within the LCFS¹. Other codes for equilibrium reconstructions might be used in the future to include the region outside the LCFS. Studies have shown ‘lobes’ in this region, i.e. structures which increase the local heat load [43, 53, 60, 98].

Physics of fast ion confinement loss with 3D fields have been studied using a distribution which covers a large section of phase space. Time dependency of these losses has been examined with an NBI distribution. Both distributions consist of deuterium particles. Confinement and heating of heavier ions has not been studied (e.g. minorities heated by ICRH, helium ash or other impurities). These minorities might be considered in a follow-up study. Especially the ‘flushing’ of the core by sawteeth [41], could be examined further with the new sawtooth model in `EBdyna.go`.

Future work might also incorporate validation of `EBdyna.go` simulations with experiments². Cooperation with other institutes could help with including plasma response, turbulence and/or collisions, which might help with this validation.

Although an attempt has been made, this work failed to reliably identify ripple trapped

¹With `BS3D` one could obtain the vacuum field outside the LCFS (one should also consider field due to the plasma current).

²Comparing a code to experiments might be done similar to Asunta et al. [6]. In this paper a diagnostic signal is modelled (the FILD) and a ‘good correspondence’ is claimed. It did use a hybrid code and a vacuum field perturbation.

particles. Therefore no conclusion is drawn whether the ripple trapping of particles is responsible for the high μ losses with the TFR. Ripple trapped particles might be identified in the future using a more sophisticated approach and by using distributions which occupy this part of phase space better. RMP coils are deemed too weak to trap particles locally (in a ripple), since their radial perturbation is relatively low $\sim 10^{-3}B$ [86].

The $\propto t^{-1}$ behavior of the losses is consistent with loss due to stochasticity. Or, to be more specific, consistent with the distribution of ‘connection lengths’ of field lines which reach into the loss region [13]. The radial excursions of the trapped particles due to the ripple resonance seem to have a lower scaling with time. A future study might examine the time dependency of the ripple resonance in more detail. This, in combination with collisions, might help estimate the contribution of the ripple resonance to the density-pumpout.

Good confinement of fast particles with a sawtooth has also been confirmed in experiments and other simulations, reported in Geiger et al. [29]. No losses were induced by the sawtooth when 3D field were applied. Influence on the core redistribution by 3D fields was also absent. Synergy as defined in this work has not been found. Other definitions or more detailed work might be carried out in the future. Plasma response to the RMP field can be amplified by the core kink response [75]. A possibly higher penetration of the RMP field might result in synergy.

Experiments show a periodic crashing of the sawtooth with a frequency of ~ 100 Hz. However, after the crash, fields do not evolve anymore in the presented model. During the remaining of the simulation the q -profile has a discontinuity, indicating a current sheet. Future work might consider the evolution of the (helical) flux after the single sawtooth collapse used in this work.

Chapter 6

Conclusion

In this work the `EBdyna.go` has been improved on multiple levels. Both the sawtooth model and the ‘particle pusher’ `BORIS` have been revised. Vacuum 3D fields are determined by the Biot-Savart solver `BS3D` with a high precision.

RMP and ripple fields result in non-prompt losses which evolve by $\propto t^{-1}$ until the loss region in phase space nears depletion. Since the regions affected by RMPs (passing and low μ) and the TFR (high μ) hardly overlap, the loss region is largest when both are modelled in the simulation.

Radial transport of trapped particles shows a strong correlation to the ratio of the bounce and precession frequencies. Those who meet the resonant condition can have radial excursions of $\sim 2.5\%$ of the minor radius within 2 ms. Ripple resonance might be relevant for explaining the ‘density-pumpout’ phenomenon observed in RMP experiments. Ripple trapping particles are sparse and not reliably identified.

Passing particles have, on average, a lower radial displacement than trapped particles. Their radial displacement is due to the movement within magnetic islands that appear at the rational q surfaces (as previously argued in [90]). Trapped particles do not show such a correlation, indicating the resonance at the banana tips is dominant.

Although a sawtooth effectively redistributes the core, it does not lead to an additional loss. Loss regions are too far outward for a sawtooth crash to refill them. Initially some synergy in the loss was observed between 3D fields and the sawtooth crash, but this turned out to be a numerical artefact.

Bibliography

- [1] S. S. Abdullaev, *Magnetic Stochasticity in Magnetically Confined Fusion Plasmas*, ser. Springer Series on Atomic, Optical, and Plasma Physics. Cham: Springer International Publishing, 2014, vol. 78. [Online]: <http://link.springer.com/10.1007/978-3-319-01890-4> [ISBN: 978-3-319-01889-8]
- [2] S. S. Abdullaev, K. H. Finken, M. W. Jakubowski *et al.*, “Overview of magnetic structure induced by the TEXTOR-DED and the related transport,” *Nuclear Fusion*, vol. 43, pp. 299–313, 2003.
- [3] S. Äkäslompolo, O. Asunta, T. Bergmans *et al.*, “Calculating the 3D magnetic field of ITER for European TBM studies,” *Fusion Engineering and Design*, vol. 98-99, no. 99, pp. 1039–1043, 2015. [Online]: <http://dx.doi.org/10.1016/j.fusengdes.2015.05.038>
- [4] B. Amberg, “Image Interpolation (ba_interp2),” 2010. [Online]: <https://nl.mathworks.com/matlabcentral/fileexchange/20342-image-interpolation--ba-interp2-> [Accessed: 2016-01-01]
- [5] B. Amberg, “3D Volume Interpolation with ba_interp3: Fast interp3 replacement,” 2008. [Online]: <https://nl.mathworks.com/matlabcentral/fileexchange/21702-3d-volume-interpolation-with-ba-interp3--fast-interp3-replacement> [Accessed: 2016-10-29]
- [6] O. Asunta, S. Äkäslompolo, T. Kurki-Suonio *et al.*, “Simulations of fast ion wall loads in ASDEX Upgrade in the presence of magnetic perturbations due to ELM-mitigation coils,” *Nuclear Fusion*, vol. 52, no. 9, p. 094014, 2012. [Online]: <http://stacks.iop.org/0029-5515/52/i=9/a=094014>
- [7] M. Bécoulet, E. Nardon, G. Huysmans *et al.*, “Numerical study of the resonant magnetic perturbations for Type I edge localized modes control in ITER,” *Nuclear Fusion*, vol. 48, no. 2, p. 024003, feb 2008. [Online]: <http://stacks.iop.org/0029-5515/48/i=2/a=024003>
- [8] M. Bécoulet, F. Orain, G. T. A. Huijsmans *et al.*, “Mechanism of edge localized mode mitigation by resonant magnetic perturbations,” *Physical Review Letters*, vol. 113, no. 11, p. 115001, sep 2014. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.113.115001>
- [9] H. Bolt, V. Barabash, G. Federici *et al.*, “Plasma facing and high heat flux materials needs for ITER and beyond,” *Journal of Nuclear Materials*, vol. 307-311, no. 1 SUPPL., pp. 43–52, dec 2002. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0022311502011753>
- [10] J. P. Boris, “Relativistic Plasma SimulationOptimization of a Hybrid Code,” in *Numerical Simulation of Plasma*, no. NOVEMBER 1970, 1970, p. 65. [Online]: <http://www.dtic.mil/dtic/tr/fulltext/u2/a023511.pdf>

- [11] P. Cahyna, M. Bécoulet, R. Pánek *et al.*, “Resonant magnetic perturbations and edge ergodization on the COMPASS tokamak,” *Plasma Physics Reports*, vol. 34, no. 9, pp. 746–749, sep 2008. [Online]: <http://link.springer.com/10.1134/S1063780X08090080>
- [12] B. V. Chirikov, “A universal instability of many-dimensional oscillator systems,” *Physics Reports*, vol. 52, no. 5, pp. 263–379, may 1979. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/0370157379900231>
- [13] T. Constantino, S. V. F. Levy, A. M. C. Souza, and R. Maynard, “Statistical-Mechanical Foundation of the Ubiquity of Lévy Distributions in Nature,” *Physical Review Letters*, vol. 75, no. 20, pp. 3589–3593, 1995. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.75.3589>
- [14] G. D. Conway, S. Fietz, H. W. Müller *et al.*, “Impact of magnetic perturbation coils on the edge radial electric field and turbulence in ASDEX Upgrade,” *Plasma Physics and Controlled Fusion*, vol. 57, no. 1, p. 014035, 2014. [Online]: <http://stacks.iop.org/0741-3335/57/i=1/a=014035>
- [15] H. J. de Blank, “Electromagnetically consistent model of complete reconnection (Personal communication),” Eindhoven, 2016.
- [16] G. Delzanno and E. Camporeale, “On particle movers in cylindrical geometry for Particle-In-Cell simulations,” *Journal of Computational Physics*, vol. 253, pp. 259–277, nov 2013. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0021999113004798>
- [17] J. Dillon, “Sharedmatrix,” 2011. [Online]: <https://nl.mathworks.com/matlabcentral/fileexchange/28572-sharedmatrix?requestedDomain=www.mathworks.com> [Accessed: 2016-10-29]
- [18] EURO Fusion, “JET vacuum vessel,” 2011. [Online]: <https://www.euro-fusion.org/2011/08/internal-view-of-stripped-out-jet-vacuum-vessel/?view=gallery-428> [Accessed: 2016-03-18]
- [19] T. E. Evans, “Erratum: Resonant magnetic perturbations of edge-plasmas in toroidal confinement devices (2015 Plasma Phys. Control. Fusion 57 123001),” *Plasma Physics and Controlled Fusion*, vol. 58, no. 4, p. 049601, 2016. [Online]: <http://stacks.iop.org/0741-3335/58/i=4/a=049601>
- [20] T. E. Evans, R. A. Moyer, P. R. Thomas *et al.*, “Suppression of Large Edge-Localized Modes in High-Confinement DIII-D Plasmas with a Stochastic Magnetic Boundary,” *Phys. Rev. Lett.*, vol. 92, no. 23, p. 235003, 2004. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.92.235003>
- [21] T. Evans, D. Orlov, A. Wingen *et al.*, “3D vacuum magnetic field modelling of the ITER ELM control coil during standard operating scenarios,” *Nuclear Fusion*, vol. 53, no. 9, p. 093029, sep 2013. [Online]: <http://stacks.iop.org/0029-5515/53/i=9/a=093029>
- [22] A. Fasoli, C. Gormenzano, H. L. Berk *et al.*, “Chapter 5: Physics of energetic ions,” *Nuclear Fusion*, vol. 47, no. 6, pp. S264–S284, jun 2007. [Online]: <http://stacks.iop.org/0029-5515/47/i=6/a=S05>
- [23] M. E. Fenstermacher, T. E. Evans, T. H. Osborne *et al.*, “Effect of island overlap on edge localized mode suppression by resonant magnetic perturbations in DIII-D,” *Physics of Plasmas*, vol. 15, no. 5, p. 056122, may 2008. [Online]: <http://aip.scitation.org/doi/10.1063/1.2901064>
- [24] K. H. Finken, S. Abdullaev, W. Biel *et al.*, “Overview of Experiments with the Dynamic Ergodic Divertor on TEXTOR,” *Contributions to Plasma Physics*, vol. 46, no. 7-9, pp. 515–526, sep 2006. [Online]: <http://doi.wiley.com/10.1002/ctpp.200610038>

BIBLIOGRAPHY

- [25] B. Fornberg, "Generation of finite difference formulas on arbitrarily spaced grids," *Mathematics of Computation*, vol. 51, no. 184, pp. 699–699, 1988. [Online]: <http://www.ams.org.sire.ub.edu/journals/mcom/1988-51-184/S0025-5718-1988-0935077-0/home.html>
- [26] J. P. Friedberg, *Plasma Physics and Fusion Energy*. Cambridge: Cambridge university press, 2007 [ISBN: 0521851076].
- [27] M. Garcia-Munoz, S. Äkäslompolo, O. Asunta *et al.*, "Fast-ion redistribution and loss due to edge perturbations in the ASDEX Upgrade, DIII-D and KSTAR tokamaks," *Nuclear Fusion*, vol. 53, no. 12, p. 123008(9), 2013. [Online]: <http://stacks.iop.org/0029-5515/53/i=12/a=123008>
- [28] M. García-Muñoz, S. Äkäslompolo, P. de Marne *et al.*, "Fast-ion losses induced by ELMs and externally applied magnetic perturbations in the ASDEX Upgrade tokamak," *Plasma Physics and Controlled Fusion*, vol. 55, no. 12, p. 124014, 2013. [Online]: <http://stacks.iop.org/0741-3335/55/i=12/a=124014>
- [29] B. Geiger, M. Weiland, A. Mlynek *et al.*, "Quantification of the impact of large and small-scale instabilities on the fast-ion confinement in ASDEX Upgrade," *Plasma Physics and Controlled Fusion*, vol. 57, no. 1, p. 014018, 2015. [Online]: <http://iopscience.iop.org/0741-3335/57/1/014018/article/>
- [30] P. Ghendrih, A. Grosman, and H. Capes, "Theoretical and experimental investigations of stochastic boundaries in tokamaks," *Plasma Physics and Controlled Fusion*, vol. 38, no. 10, pp. 1653–1724, oct 1996. [Online]: <http://stacks.iop.org/0741-3335/38/i=10/a=002>
- [31] J. P. Goedbloed, R. Keppens, and S. Poedts, *Advanced Magnetohydrodynamics*. Cambridge: Cambridge University Press, 2010. [Online]: <http://ebooks.cambridge.org/ref/id/CB09781139195560> [ISBN: 9781139195560]
- [32] R. J. Goldston, R. B. White, and A. H. Boozer, "Confinement of High-Energy Trapped Particles in Tokamaks," *Physical Review Letters*, vol. 47, no. 9, pp. 647–649, aug 1981. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.47.647>
- [33] D. J. Griffiths and R. College, *Introduction to Electrodynamics*, 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall Inc., 1999, vol. 3 [ISBN: 0-13-805326-X].
- [34] W. Heidbrink and G. Sadler, "The behaviour of fast ions in tokamak experiments," *Nuclear Fusion*, vol. 35, no. 2, pp. 243–243, feb 1995. [Online]: <http://stacks.iop.org/0029-5515/35/i=2/a=514>
- [35] P. Helander, C. D. Beidler, T. M. Bird *et al.*, "Stellarator and tokamak plasmas: a comparison," *Plasma Physics and Controlled Fusion*, vol. 54, no. 12, p. 124009, 2012. [Online]: <http://stacks.iop.org/0741-3335/54/i=12/a=124009>
- [36] T. Hender, R. Fitzpatrick, A. Morris *et al.*, "Effect of resonant magnetic perturbations on COMPASS-C tokamak discharges," *Nuclear Fusion*, vol. 32, no. 12, pp. 2091–2117, dec 1992. [Online]: <http://stacks.iop.org/0029-5515/32/i=12/a=I02>
- [37] ITER Physics Basis Editors, ITER Physics Expert Group Chairs an Co-Chairs, ITER Joint Central Team and Physics Unit, and ITER EDA Naka Joint Work Site, "Chapter 1: Overview and summary," *Nuclear Fusion*, vol. 39, no. 12, pp. 2137–2174, dec 1999. [Online]: <http://stacks.iop.org/0029-5515/39/i=12/a=301>
- [38] ITER Physics Expert Group on Confinement and Transport, ITER Physics Basis Editors, and ITER EDA Naka Joint Work Site, "Chapter 2: Plasma confinement and transport," *Nuclear Fusion*, vol. 39, no. 12, pp. 2175–2249, 1999. [Online]: <http://stacks.iop.org/0029-5515/39/i=12/a=302>

- [39] F. Jaulmes, B. Geiger, T. Odstrčil *et al.*, “Numerical and experimental study of the redistribution of energetic and impurity ions by sawteeth in ASDEX Upgrade,” *Nuclear Fusion*, vol. 56, no. 11, p. 112012, nov 2016. [Online]: <http://stacks.iop.org/0029-5515/56/i=11/a=112012>
- [40] F. Jaulmes, “Kinetic behaviour of ions in tokamak inductive scenarios,” Ph.D. dissertation, Eindhoven University of Technology, 2016. [Online]: <http://repository.tue.nl/841352>
- [41] F. Jaulmes, E. Westerhof, and H. de Blank, “Redistribution of fast ions during sawtooth reconnection,” *Nuclear Fusion*, vol. 54, no. 10, p. 104013, oct 2014. [Online]: <http://stacks.iop.org/0029-5515/54/i=10/a=104013>
- [42] B. B. Kadomtsev, “Disruptive instability in tokamaks,” *Soviet Journal of Plasma Physics*, vol. 1, no. 5, pp. 710–715, 1975. [Online]: <http://adsabs.harvard.edu/abs/1975FizPl...1..710K>
- [43] A. Kirk, I. Chapman, Y. Liu *et al.*, “Understanding edge-localized mode mitigation by resonant magnetic perturbations on MAST,” *Nuclear Fusion*, vol. 53, no. 4, p. 043007, apr 2013. [Online]: <http://stacks.iop.org/0029-5515/53/i=4/a=043007>
- [44] A. Kirk, W. Suttrop, I. T. Chapman *et al.*, “Effect of resonant magnetic perturbations on low collisionality discharges in MAST and a comparison with ASDEX Upgrade,” *Nuclear Fusion*, vol. 55, no. 4, p. 043011, 2015. [Online]: <http://stacks.iop.org/0029-5515/55/i=4/a=043011>
- [45] Y. Kolesnichenko and Y. Yakovenko, “Theory of fast ion transport during sawtooth crashes in tokamaks,” *Nuclear Fusion*, vol. 36, no. 2, pp. 159–172, feb 1996. [Online]: <http://stacks.iop.org/0029-5515/36/i=2/a=I04>
- [46] G. Kramer, R. White, R. Nazikian, and H. Berk, “Fusion-born Alpha Particle Ripple Loss Studies in ITER,” in *22th IAEA Fusion Energy Conference, 13-18 October, 2008, Geneva, Switzerland*, no. c, 2008, pp. IT/P5-3 (1-8). [Online]: http://www-naeweb.iaea.org/napc/physics/fec/fec2008/papers/it_p6-3.pdf
- [47] T. Kurki-Suonio, O. Asunta, T. Hellsten *et al.*, “ASCOT simulations of fast ion power loads to the plasma-facing components in ITER,” *Nuclear Fusion*, vol. 49, no. 9, p. 095001, 2009. [Online]: <http://stacks.iop.org/0029-5515/49/i=9/a=095001>
- [48] M. Lanctot, R. Buttery, J. de Grassie *et al.*, “Sustained suppression of type-I edge-localized modes with dominantly $n = 2$ magnetic fields in DIII-D,” *Nuclear Fusion*, vol. 53, no. 8, p. 083019, aug 2013. [Online]: <http://stacks.iop.org/0029-5515/53/i=8/a=083019>
- [49] P. Lang, A. Loarte, G. Saibene *et al.*, “ELM control strategies and tools: status and potential for ITER,” *Nuclear Fusion*, vol. 53, no. 4, p. 043004, 2013. [Online]: <http://stacks.iop.org/0029-5515/53/i=4/a=043004>
- [50] A. W. Leonard, “Edge-localized-modes in tokamaks,” *Physics of Plasmas*, vol. 21, no. 9, 2014. [Online]: <http://dx.doi.org/10.1063/1.4894742>
- [51] Y. Liang, H. R. Koslowski, P. R. Thomas *et al.*, “Active control of type-I edge-localized modes with $n = 1$ perturbation fields in the JET tokamak,” *Physical Review Letters*, vol. 98, no. 26, p. 265004, jun 2007. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.98.265004>
- [52] A. Loarte, G. Saibene, R. Sartori *et al.*, “Characteristics of type I ELM energy and particle losses in existing devices and their extrapolation to ITER,” *Plasma Physics and Controlled Fusion*, vol. 45, no. 9, pp. 1549–1569, sep 2003. [Online]: <http://stacks.iop.org/0741-3335/45/i=9/a=302>

BIBLIOGRAPHY

- [53] A. Loarte, G. Huijsmans, S. Futatani *et al.*, “Progress on the application of ELM control schemes to ITER scenarios from the non-active phase to DT operation,” *Nuclear Fusion*, vol. 54, no. 3, p. 033007, mar 2014. [Online]: <http://stacks.iop.org/0029-5515/54/i=3/a=033007>
- [54] D. J. MacKay, *Sustainable Energy without the hot air*. Cambridge: UIT Cambridge Ltd., 2009, no. 3.5.2. [Online]: www.withouthotair.com [ISBN: 9780954452933]
- [55] A. F. Martitsch, S. V. Kasilov, W. Kernbichler *et al.*, “Effect of 3D magnetic perturbations on the plasma rotation in ASDEX Upgrade,” *Plasma Physics and Controlled Fusion*, vol. 58, no. 7, p. 074007, 2016. [Online]: <http://stacks.iop.org/0741-3335/58/i=7/a=074007>
- [56] O. Merkulov, “Control of plasma profiles and stability through localised Electron Cyclotron Current Drive,” Ph.D. dissertation, University of Utrecht, 2006.
- [57] H. Mimata, K. Tani, K. Tobita *et al.*, “Finite Larmor radius effects on ripple diffusion in tokamaks,” *Progress in Nuclear Energy*, vol. 50, no. 2-6, pp. 638–642, mar 2008. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0149197007002077>
- [58] H. MIMATA, K. TANI, H. TSUTSUI *et al.*, “Numerical Study of the Ripple Resonance Diffusion of Alpha Particles in Tokamaks,” *Plasma and Fusion Research*, vol. 4, pp. 008–008, 2009. [Online]: <http://joi.jlc.jst.go.jp/JST.JSTAGE/pfr/4.008>
- [59] J. Ongena, R. Koch, R. Wolf, and H. Zohm, “Magnetic-confinement fusion,” *Nature Physics*, vol. 12, no. 5, pp. 398–410, may 2016. [Online]: <http://www.nature.com/doifinder/10.1038/nphys3745>
- [60] F. Orain, M. Bécoulet, G. Dif-Pradalier *et al.*, “Non-linear magnetohydrodynamic modeling of plasma response to resonant magnetic perturbations,” *Physics of Plasmas*, vol. 20, no. 10, p. 102510, oct 2013. [Online]: <http://aip.scitation.org/doi/10.1063/1.4824820>
- [61] F. Orain, A. Lessig, M. Hözl *et al.*, “Non-linear modeling of the Edge Localized Modes and their control by Resonant Magnetic Perturbations in ASDEX Upgrade,” in *43rd EPS Conference on Plasma Physics*. European Physical Society, 2016. [Online]: <http://matthias-hoelzl.org/2016-08-Orain-EPS-P1.017.pdf>
- [62] F. Orain, M. Hözl, E. Viezzer *et al.*, “Non-linear modeling of the plasma response to RMPs in ASDEX Upgrade,” *Nuclear Fusion*, vol. 57, no. 2, p. 022013, feb 2017. [Online]: <http://stacks.iop.org/0029-5515/57/i=2/a=022013>
- [63] F. Orain, “Edge Localized Mode control by Resonant Magnetic Perturbations in tokamak plasmas,” Ph.D. dissertation, Université d’Aux-Marseille, 2014. [Online]: <https://tel.archives-ouvertes.fr/tel-01124954/>
- [64] G. Papp, M. Drevlak, T. Fülop *et al.*, “Runaway electron losses caused by resonant magnetic perturbations in ITER,” *Plasma Physics and Controlled Fusion*, vol. 53, no. 9, p. 095004, sep 2011. [Online]: <http://stacks.iop.org/0741-3335/53/i=9/a=095004>
- [65] S. Parker and C. Birdsall, “Numerical error in electron orbits with large $\omega_{ce}\Delta t$,” *Journal of Computational Physics*, vol. 97, no. 1, pp. 91–102, nov 1991. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/002199919190040R>
- [66] L. Patacchini and I. H. Hutchinson, “Explicit time-reversible orbit integration in Particle In Cell codes with static homogeneous magnetic field,” *Journal of Computational Physics*, vol. 228, no. 7, pp. 2604–2615, 2009. [Online]: <http://dx.doi.org/10.1016/j.jcp.2008.12.021>

- [67] C. Paz-Soldan, R. Nazikian, S. R. Haskey *et al.*, “Observation of a Multimode Plasma Response and its Relationship to Density Pumpout and Edge-Localized Mode Suppression,” *Physical Review Letters*, vol. 114, no. 10, p. 105001, mar 2015. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.114.105001>
- [68] D. Pfefferl , C. Misev, W. a. Cooper, and J. P. Graves, “Impact of RMP magnetic field simulation models on fast ion losses,” *Nuclear Fusion*, vol. 55, no. 1, p. 012001, 2015. [Online]: <http://stacks.iop.org/0029-5515/55/i=1/a=012001>
- [69] V. Philipps, R. Neu, J. Rapp *et al.*, “Comparison of tokamak behaviour with tungsten and low- Z plasma facing materials,” *Plasma Physics and Controlled Fusion*, vol. 42, no. 12B, pp. B293–B310, dec 2000. [Online]: <http://stacks.iop.org/0741-3335/42/i=12B/a=322>
- [70] F. Porcelli, D. Boucher, and M. N. Rosenbluth, “Model for the sawtooth period and amplitude,” *Plasma Physics and Controlled Fusion*, vol. 38, pp. 2163–2186, 1996. [Online]: <http://stacks.iop.org/0741-3335/38/i=12/a=010>
- [71] H. Qin, S. Zhang, J. Xiao *et al.*, “Why is Boris algorithm so good?” *Physics of Plasmas*, vol. 20, no. 8, p. 084503, aug 2013. [Online]: <http://aip.scitation.org/doi/10.1063/1.4818428>
- [72] L. Qu val, “Biot Savart magnetic Toolbox,” 2015. [Online]: <https://nl.mathworks.com/matlabcentral/fileexchange/50434-biot-savart-magnetic-toolbox> [Accessed: 2016-10-29]
- [73] A. B. Rechester and M. N. Rosenbluth, “Electron Heat Transport in a Tokamak with Destroyed Magnetic Surfaces,” *Physical Review Letters*, vol. 40, no. 1, pp. 38–41, jan 1978. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.40.38>
- [74] M. Rott, U. Seidel, B. Streibl *et al.*, “Electro-magnetic modeling of the planned active in-vessel coils at ASDEX Upgrade,” *Fusion Engineering and Design*, vol. 84, no. 7-11, pp. 1653–1657, 2009. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0920379608004341>
- [75] D. A. Ryan, Y. Q. Liu, A. Kirk *et al.*, “Toroidal modelling of resonant magnetic perturbations response in ASDEX-Upgrade: coupling between field pitch aligned response and kink amplification,” *Plasma Physics and Controlled Fusion*, vol. 57, no. 9, p. 095008, 2015. [Online]: <http://stacks.iop.org/0741-3335/57/i=9/a=095008>
- [76] G. Saibene, R. Sartori, P. D. Vries, and A. Loarte, “Toroidal field ripple effects on H-modes in JET and implications for ITER,” *EfdaJetCp(07)03/62*, vol. 31, no. 07, pp. 8–11, 2007. [Online]: <http://www.iop.org/Jet/fulltext/EFDC070362.pdf>
- [77] M. Schaffer, J. Menard, M. Aldan *et al.*, “Study of in-vessel nonaxisymmetric ELM suppression coil concepts for ITER,” *Nuclear Fusion*, vol. 48, no. 2, p. 024004, feb 2008. [Online]: <http://stacks.iop.org/0029-5515/48/i=2/a=024004>
- [78] N. Schl mer, “matlab2tikz/matlab2tikz,” 2016. [Online]: <https://nl.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz-matlab2tikz> [Accessed: 2016-01-01]
- [79] K. Shinohara, T. Kurki-Suonio, D. Spong *et al.*, “Effects of complex symmetry-breakings on alpha particle power loads on first wall structures and equilibrium in ITER,” *Nuclear Fusion*, vol. 51, no. 6, p. 063028(12), 2011. [Online]: <http://stacks.iop.org/0029-5515/51/i=6/a=063028>
- [80] T. Shoji, H. Tamai, Y. Miura *et al.*, “Effects of ergodization on plasma confinement in JFT-2M,” *Journal of Nuclear Materials*, vol. 196-198, no. C, pp. 296–300, dec 1992. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0022311506800498>

BIBLIOGRAPHY

- [81] A. Snicker, T. Kurki-Suonio, and S. K. Sipila, “Realistic Simulations of Fast-Ion Wall Distribution Including Effects Due to Finite Larmor Radius,” *IEEE Transactions on Plasma Science*, vol. 38, no. 9, pp. 2177–2184, sep 2010. [Online]: <http://ieeexplore.ieee.org/document/5545410/>
- [82] D. A. Spong, “Three-dimensional effects on energetic particle confinement and stability,” *Physics of Plasmas*, vol. 18, no. 5, p. 056109, 2011. [Online]: <http://scitation.aip.org/content/aip/journal/pop/18/5/10.1063/1.3575626>
- [83] U. Stroth, J. Adamek, L. Aho-Mantila *et al.*, “Overview of ASDEX Upgrade results,” *Nuclear Fusion*, vol. 53, no. 10, p. S622, oct 2013. [Online]: <http://stacks.iop.org/0029-5515/53/i=10/a=104003>
- [84] W. Suttrop, O. Gruber, S. Günter *et al.*, “In-vessel saddle coils for MHD control in ASDEX Upgrade,” *Fusion Engineering and Design*, vol. 84, no. 2-6, pp. 290–294, jun 2009. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0920379608004912>
- [85] W. Suttrop, T. Eich, J. C. Fuchs *et al.*, “First observation of edge localized modes mitigation with resonant and nonresonant magnetic perturbations in ASDEX upgrade,” *Physical Review Letters*, vol. 106, no. 22, pp. 1–4, 2011. [Online]: <http://link.aps.org/doi/10.1103/PhysRevLett.106.225004>
- [86] W. Suttrop, J. C. Fuchs, R. Fischer *et al.*, “Mitigation of edge localised modes with magnetic perturbations in ASDEX Upgrade,” *Fusion Engineering and Design*, vol. 88, no. 6-8, pp. 446–453, 2013. [Online]: <http://www.sciencedirect.com/science/article/pii/S0920379613000239>
- [87] W. Suttrop, A. Kirk, R. Nazikian *et al.*, “Experimental studies of high-confinement mode plasma response to non-axisymmetric magnetic perturbations in ASDEX Upgrade,” *Plasma Physics and Controlled Fusion*, vol. 59, no. 1, p. 014049, jan 2017. [Online]: <http://stacks.iop.org/0741-3335/59/i=1/a=014049>
- [88] T. Tajima, “Plasma physics via computer simulation,” *Computer Physics Communications*, vol. 42, no. 1, pp. 151–152, sep 1986. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/0010465586902407>
- [89] K. Tani, T. Takizuka, M. Azumi, and H. Kishimoto, “Ripple loss of suprathermal alpha particles during slowing-down in a tokamak reactor,” *Nuclear Fusion*, vol. 23, no. 5, pp. 657–665, may 1983. [Online]: <http://stacks.iop.org/0029-5515/23/i=5/a=007>
- [90] K. Tani, K. Shinohara, T. Oikawa *et al.*, “Effects of ELM mitigation coils on energetic particle confinement in ITER steady-state operation,” *Nuclear Fusion*, vol. 52, no. 1, p. 013012, 2012. [Online]: <http://stacks.iop.org/0029-5515/52/i=1/a=013012>
- [91] A. Thornton, A. Kirk, P. Cahyna *et al.*, “ELM mitigation via rotating resonant magnetic perturbations on MAST,” *Journal of Nuclear Materials*, vol. 463, pp. 723–726, aug 2015. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0022311514006795>
- [92] A. D. Turnbull, “Plasma response models for non-axisymmetric perturbations,” *Nuclear Fusion*, vol. 52, no. 5, p. 054016, 2012. [Online]: <http://stacks.iop.org/0029-5515/52/i=5/a=054016>
- [93] K. van de Plassche, “Pre-sawtooth crash q-profile reconstruction,” Eindhoven, 2015. [Online]: <https://github.com/Karel-van-de-Plassche/PF2q>
- [94] M. A. Van Zeeland, N. M. Ferraro, W. W. Heidbrink *et al.*, “Modulation of prompt fast-ion loss by applied $n = 2$ fields in the DIII-D tokamak,” *Plasma Physics and Controlled Fusion*, vol. 56, no. 1, p. 015009(10), 2014. [Online]: <http://stacks.iop.org/0741-3335/56/i=1/a=015009>

- [95] M. R. Wade, R. Nazikian, J. S. DeGrassie *et al.*, “Advances in the physics understanding of ELM suppression using resonant magnetic perturbations in DIII-D,” *Nuclear Fusion*, vol. 55, no. 2, p. 23002, 2015. [Online]: <http://stacks.iop.org/0029-5515/55/i=2/a=023002>
- [96] J. Wesson, *Tokamaks*, 3rd ed. Oxford: Oxford University Press, 2004 [ISBN: 0198509227].
- [97] R. B. White, “Determination of broken KAM surfaces for particle orbits in toroidal confinement systems,” *Plasma Physics and Controlled Fusion*, vol. 57, no. 11, p. 115008, 2015. [Online]: <http://stacks.iop.org/0741-3335/57/i=11/a=115008>
- [98] A. Wingen, N. M. Ferraro, M. W. Shafer *et al.*, “Connection between plasma response and resonant magnetic perturbation (RMP) edge localized mode (ELM) suppression in DIII-D,” *Plasma Physics and Controlled Fusion*, vol. 57, no. 10, p. 104006, 2015. [Online]: <http://dx.doi.org/10.1088/0741-3335/57/10/104006>
- [99] M. Winkel, R. Speck, and D. Ruprecht, “A high-order Boris integrator,” *Journal of Computational Physics*, vol. 295, pp. 456–474, aug 2015. [Online]: <http://linkinghub.elsevier.com/retrieve/pii/S0021999115002685>
- [100] S. J. Zweben, R. V. Budny, D. S. Darrow *et al.*, “Alpha particle physics experiments in the Tokamak Fusion Test Reactor,” *Nucl. Fusion*, vol. 40, no. 1, pp. 91–149, 2000. [Online]: <http://stacks.iop.org/0029-5515/40/i=1/a=307>

Appendix A

Acknowledgements

- H. de Blank, F. Jaulmes and E. Westerhof, DIFFER and J. van Dijk, Eindhoven University of Technology
For the fruitful discussions, supervision and the help with this project.
- D.C. Daan van Vugt, Eindhoven University of Technology
Special thanks to ir. D.C. (Daan) van Vugt from the Eindhoven University of Technology for helpful discussions on computational subjects.
- K. v.d. Plassche, University of Eindhoven / DIFFER
Thanks to K. (Karel) v.d. Plassche from the Eindhoven University of Technology for the help with the PF2q-tool he wrote during an internship at DIFFER and for helpful discussions and his companionship.
- Recognition for contributions to File Exchange of MATLAB®:
 - Dillon [17] for the *sharedmatrix* function, which is implemented to share the RAM on a node between different processes. This enabled large 3D matrices on *SurfSara Lisa*.
 - Amberg [4]/[5] for the fast interpolation schemes *ba_interp2/ba_interp3* (2D surface/3D volume interpolation).
 - Schröder [78] for the script *matlab2tikz*, which has been used frequently to export figures from MATLAB® to this document.
 - Quéval [72] for the script *BSmag* which has inspired *BS3D*. However, *BS3D* is a complete rewrite. *BS3D* is faster, can determine the vector potential and also switches between multiple algorithms. In addition *BS3D* reads out the coil data etc. specially made for the 3D fields of ASDEX Upgrade.
- DIFFER, www.differ.nl



For providing me with the opportunity to perform this work (and for the free coffee).

Appendix B

Derivation of conserved quantities

Particles in an axisymmetric (static) magnetic field have three conserved quantities. Energy conservation is a trivial one [26, 33], but the other two aren't.

First, we will derive the conservation of the magnetic moment μ . This quantity is not exactly conserved, as the derivation will show. Secondly, the conservation of the canonical toroidal angular momentum p_φ is derived. Additionally, the last section is devoted to the evolution of canonical toroidal angular momentum when 3D fields apply.

B.1 Conservation of magnetic moment

To derive the invariance of the magnetic moment we follow the dedicated chapter in the book of Friedberg [26]. We consider a particle in a static magnetic field. Here we assume the magnetic field is ‘large’, so the gyration is ‘small’. To be particular, the parallel gradient length L is much larger than the perpendicular gradient length of order r_L .

We start by finding the equation of motion. Once we have a suitable expression we ‘solve’ it, by assuming the gyro motion is almost a perfect harmonic oscillation. From this we find μ as a constant (both from the perpendicular and parallel part of the equation of motion).

B.1.1 Formalizing the equation of motion

Start by imagining a particle in a static inhomogeneous magnetic field. Averaged over the gyro orbit the magnetic field is directed in the \mathbf{e}_z direction. In addition there is a small component in the x -direction, which we need to satisfy the divergence free condition of the magnetic field and at the same time have a parallel magnetic field gradient. Without loss of generality we assume there is no y -component of the magnetic field¹.

The equation of motion (2.30) becomes:

$$\frac{d}{dt} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \frac{q}{m} \begin{bmatrix} v_y B_z \\ v_z B_x - v_x B_z \\ -v_y B_x \end{bmatrix}. \quad (\text{B.1})$$

The magnetic field is primarily in the z -direction: $\mathbf{B} = B\mathbf{b} \approx B_z \mathbf{e}_z$. Note that while \mathbf{e}_z is constant, the unit vector \mathbf{b} changes direction over the gyro orbit. Let's define a complete basis with \mathbf{b} to average over the gyro orbit.

The other two unit vectors \mathbf{e}_1 and \mathbf{e}_2 of this orthogonal basis are perpendicular to the magnetic field. For convenience we define $\mathbf{e}_2 = \mathbf{e}_y$ and $\mathbf{e}_1 = \mathbf{e}_2 \times \mathbf{b}$. The velocity components in this basis

¹The coordinates x, y, z used in this section are Cartesian coordinates, but have little to do with the previously defined Cartesian coordinate system for a tokamak. We're now dealing with more general situation.

and in the Cartesian basis are given by:

$$\begin{aligned} v_{\parallel} &= b_z v_z + b_x v_x & v_z &= b_z v_{\parallel} - b_x v_1 \\ v_2 &= v_y & v_y &= v_2 \\ v_1 &= b_z v_x - b_x v_z & v_x &= b_x v_{\parallel} + b_z v_1. \end{aligned}$$

With this the equation of motion becomes:

$$\frac{d}{dt} \begin{bmatrix} v_1 \\ v_2 \\ v_{\parallel} \end{bmatrix} + \begin{bmatrix} -\omega_c v_2 \\ \omega_c v_1 \\ 0 \end{bmatrix} = \begin{bmatrix} Kv_{\parallel} \\ 0 \\ -Kv_1 \end{bmatrix}, \quad (\text{B.2})$$

where

$$K = b_x \frac{db_z}{dt} - b_z \frac{db_x}{dt} \quad \text{and} \quad \omega_c = \frac{qB}{m}. \quad (\text{B.3})$$

This is quite straightforward to derive, if we keep in mind that $b_x^2 + b_z^2 = 1$ and therefore $b_x \frac{db_x}{dt} + b_z \frac{db_z}{dt} = 0$. Now we divide both sides by ω_c , theretofore we need to introduce a new typical time τ of the gyration, defined implicitly by:

$$\tau = \int_0^1 \omega_c(t) dt. \quad (\text{B.4})$$

We end up with:

$$\frac{d}{d\tau} \begin{bmatrix} v_1 \\ v_2 \\ v_{\parallel} \end{bmatrix} + \begin{bmatrix} -v_2 \\ v_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{K}v_{\parallel} \\ 0 \\ -\hat{K}v_1 \end{bmatrix}, \quad (\text{B.5})$$

B.1.2 Solving the equation of motion

To solve this equation we make use of the ‘long-thin’ approximation, which assumes the parallel gradient length L is large compared to the transverse gradient length. Since the perpendicular motion has a small characteristic size of $x \sim r_L$ (scale of gyro-orbit), we can neglect the perpendicular gradients for now. Combined with $\nabla \cdot \mathbf{B} = 0$ we find:

$$B_z = B_z(z) \quad (\text{B.6})$$

$$B_x = -x \frac{dB_z}{dz}, \quad (\text{B.7})$$

So the field is mainly in the z -direction, whilst B_x has an oscillatory character due to the gyration.

Our job is to find a suitable expression for \hat{K} to solve the equation of motion. We can do this by rewriting it as follows:

$$\hat{K} = b_z^2 \left(\frac{b_x}{b_z^2} \frac{db_z}{d\tau} - \frac{1}{b_z} \frac{db_x}{d\tau} \right) = \frac{B_z^2}{B^2} \frac{d}{d\tau} \left(-\frac{B_x}{B_z} \right) = \frac{B_z^2}{B^2} \frac{d}{d\tau} \left(\frac{x}{B_z} \frac{dB_z}{dz} \right). \quad (\text{B.8})$$

In our situation B_z and its z -derivative vary little over the gyro orbit, whilst B_x varies a lot. Therefore the leading order term of \hat{K} consists of the term with $dx/d\tau$ only. For this we make use of $x \sim r_L$ and the ratio $B_x/B_z \sim r_L/L \ll 1$. The latter follows from $L \equiv B_z/\frac{dB_z}{dz}$ and equation (B.7).

This might be somewhat unclear, so maybe looking at it differently helps. We can neglect some terms of the τ -derivatives. The first is the parallel gradient, which is assumed not to vary over the gyro orbit: $\frac{d}{d\tau} \frac{dB_z}{dz} \approx 0$. The other term $\frac{B_z^2}{B^2} \frac{d}{d\tau} \left(\frac{B_x}{B_z} \right) \frac{dB_z}{dz}$ is also neglected since B_z itself hardly varies over one gyro-orbit ($a/L \ll 1$) and thus $\frac{dB_z}{dz}$ can be considered small. Its prefactor $\frac{B_z^2}{B^2} \frac{d}{d\tau} \left(\frac{B_x}{B_z} \right) \sim \frac{B_x}{B^2}$ is also small for this case ($B \approx B_z \gg B_x$).

Now we can find a simplified expression for \hat{K} to solve the equation of motion. We find:

$$\hat{K} \approx \frac{B_z^2}{B^2} \frac{dx}{d\tau} \left(\frac{1}{B_z} \frac{dB_z}{dz} \right) \approx \frac{v_1}{\omega_c B_z} \frac{dB_z}{dz} \approx \frac{v_1}{v_{||} B_z} \frac{dB_z}{d\tau} \quad (\text{B.9})$$

Where in the last step we made use of $dz = (v_z/\omega_c)d\tau \approx (v_{||}/\omega_c)d\tau$. All that's left to do is to close the set of equations by assuming an almost harmonic gyration:

$$v_1 = v_{\perp}(\tau) \cos [\tau + \epsilon(\tau)], \quad (\text{B.10})$$

$$v_2 = -v_{\perp}(\tau) \sin [\tau + \epsilon(\tau)], \quad (\text{B.11})$$

with $\epsilon/\tau \ll 1$ and $\mathbf{v}_{\perp} = \mathbf{b} \times (\mathbf{v} \times \mathbf{b})$.

From this we can 'solve' the equation of motion equation (B.5). For the perpendicular part we can eliminate ϵ and we find:

$$\frac{dv_{\perp}}{d\tau} = \frac{v_{\perp}}{2B_z} \frac{dB_z}{d\tau} [1 + \cos(\tau + \epsilon)]. \quad (\text{B.12})$$

We can rewrite this equation as:

$$\frac{1}{\mu} \frac{d\mu}{dt} = \left(\frac{1}{B_z} \frac{dB_z}{dt} \right) \cos(\tau + \epsilon), \quad (\text{B.13})$$

with

$$\mu = \frac{mv_{\perp}}{2B}. \quad (\text{B.14})$$

The magnetic moment is only constant when the cosine averages out to zero, which is not the case if ϵ varies a lot over the gyro orbit. This happens when the (perpendicular) magnetic field gradient is large within the gyro orbit [26].

For the parallel motion we find:

$$\frac{dv_{||}}{d\tau} = -\frac{v_1^2}{\omega_c B_z} \frac{dB_z}{dz} = -\frac{v_{\perp}^2}{2\omega_c B_z} \frac{dB_z}{dz} [1 + \cos(\tau + \epsilon)]. \quad (\text{B.15})$$

If we average this over one gyro orbit, we can neglect the cosine term and we find:

$$m \frac{dv_{||}}{dt} = -\frac{mv_{\perp}^2}{2B_z} \frac{dB_z}{dz}, \quad (\text{B.16})$$

or more generic the result that the parallel force is given by:

$$F_{||} = -\mu \nabla_{||} B. \quad (\text{B.17})$$

So, whilst instantaneously there is no parallel force by the magnetic field on the guiding center, there is a higher order effect due to the gyro orbit. This force is used in guiding center codes, whilst we use the exact equation of motion with the local magnetic field in our code.

In addition it is possible, if you're not convinced yet, to find the invariance of the magnetic moment differently. We only use the solution of the parallel part of the equation of motion and conservation of energy. If we multiply both sides of equation (B.17) with $v_{||}$, we find the following set of equations:

$$\frac{1}{2} m \frac{dv_{||}^2}{dt} = mv_{||} \frac{dv_{||}}{dt} = v_{||} F_{||} = -\mu v_{||} \nabla_{||} B = -\mu \frac{dB}{dt} \quad (\text{B.18})$$

$$\frac{1}{2} m \frac{dv_{||}^2}{dt} + \frac{1}{2} m \frac{dv_{\perp}^2}{dt} = 0 \quad (\text{B.19})$$

$$\frac{1}{2} m \frac{dv_{\perp}^2}{dt} = \frac{d}{dt} E_{\perp} = \frac{d}{dt} \mu B = \mu \frac{dB}{dt} + \frac{d\mu}{dt} B \quad (\text{B.20})$$

Combining these equations results in $\frac{d\mu}{dt} = 0$ for nonzero magnetic fields ($B \neq 0$) as expected.

B.2 Conservation of canonical toroidal angular momentum

First we start by finding the toroidal angular momentum p_φ in an axisymmetric tokamak. Here we can make use of ψ as a sort of potential. For the 3D case we have to go back to the linear momentum \mathbf{P} to find the toroidal angular momentum p_φ ².

B.2.1 Toroidal angular momentum in axisymmetry

We start from the equation of motion. In the toroidal coordinates R, Z, φ the acceleration of a particle is given by equation (B.21).

$$\frac{d^2\mathbf{x}}{dt^2} = (\ddot{R} - R\dot{\varphi}^2) \mathbf{e}_R + (R\ddot{\varphi} + 2\dot{R}\dot{\varphi}) \mathbf{e}_\varphi + \ddot{Z} \mathbf{e}_z. \quad (\text{B.21})$$

We are only interested in the conserved quantity: the toroidal angular momentum. Therefore we exclude the toroidal component and find a more suitable expression:

$$\left(\frac{d^2\mathbf{x}}{dt^2} \right)_\varphi = \frac{1}{R} \frac{d}{dt} (Rv_\varphi), \quad (\text{B.22})$$

which is substituted in the force balance (so it is equal to the Lorentz force in the toroidal direction).

$$\frac{m}{R} \frac{d}{dt} (Rv_\varphi) = Z_i e \left[E_\varphi + (\mathbf{v} \times \mathbf{B})_\varphi \right]. \quad (\text{B.23})$$

The φ -component of the $\mathbf{v} \times \mathbf{B}$ is given by: $v_R B_z - v_z B_R$. Since this term only depends on the poloidal field we can write it in terms of the poloidal flux ψ by using:

$$\mathbf{B}_{\text{pol}} = \frac{\mathbf{e}_\varphi \times \nabla \psi}{R} = \begin{pmatrix} B_R \\ B_z \\ 0 \end{pmatrix} = \frac{1}{R} \begin{pmatrix} -\frac{\partial \psi}{\partial z} \\ \frac{\partial \psi}{\partial R} \\ 0 \end{pmatrix}. \quad (\text{B.24})$$

We substitute this in equation (B.23) and find:

$$\frac{m}{R} \frac{d}{dt} (Rv_\varphi) = Z_i e \left[E_\varphi + \frac{1}{R} \mathbf{v}_{\text{pol}} \cdot \nabla \psi \right] \quad (\text{B.25})$$

Please note that the magnetic field in toroidal coordinates is given by [33]:

$$\begin{aligned} \mathbf{B} &= \nabla \times \mathbf{A} \\ &= \left(\frac{\partial A_\varphi}{\partial z} - \frac{1}{R} \frac{\partial A_Z}{\partial \varphi} \right) \mathbf{e}_R + \frac{1}{R} \left(\frac{\partial A_R}{\partial \varphi} - \frac{\partial (RA_\varphi)}{\partial R} \right) \mathbf{e}_z + \left(\frac{\partial A_z}{\partial R} - \frac{\partial A_R}{\partial z} \right) \mathbf{e}_\varphi \end{aligned} \quad (\text{B.26})$$

Here we note that the poloidal flux ψ is related to the vector potential \mathbf{A} by the relation: $\psi = -RA_\varphi$, which makes it consistent with equation (B.24).

We can substitute the expression for the electric field from the potentials: $\mathbf{E} = -(\partial \mathbf{A} / \partial t + \nabla V)$, with Φ the electric potential [33]. To summarize we now write $E_\varphi = (\partial \psi / \partial t) / R - \mathbf{e}_\varphi \cdot \nabla \Phi$ and substitute this in the force balance:

$$\frac{1}{R} \frac{d}{dt} (Rv_\varphi) = Z_i e \left[\frac{1}{R} \frac{\partial \psi}{\partial t} - \mathbf{e}_\varphi \cdot \nabla \Phi + \frac{1}{R} \mathbf{v}_{\text{pol}} \cdot \nabla \psi \right]. \quad (\text{B.27})$$

Now we need to realize the partial derivative to time of ψ represents only its time evolution, which is zero in an axisymmetric, static field. However, a particle may experience a variation in ψ since it is travelling through the field. Hence we can replace the partial derivative to time by:

$$\frac{\partial \psi}{\partial t} = \frac{d\psi}{dt} - (\mathbf{v}_{\text{pol}} + \mathbf{v}_\varphi) \cdot (\nabla \psi). \quad (\text{B.28})$$

²The symbol \mathbf{p} will be used for angular momentum. Our symbol for linear momentum is \mathbf{P} .

By using this last substitution in the force balance and bringing the term $d\psi/dt$ to the left hand side we find:

$$\frac{d}{dt} (mRv_\varphi - Z_i e\psi) = Z_i e \left[-\frac{\partial \Phi}{\partial \varphi} - \frac{v_\varphi}{R} \frac{\partial \psi}{\partial \varphi} \right]. \quad (\text{B.29})$$

In axisymmetric systems the term on the right hand side turns out to be zero. Hence we find a conserved quantity, called the canonical toroidal angular momentum, which is given by:

$$p_\varphi = mRv_\varphi - (Z_i e)\psi. \quad (\text{B.30})$$

The radial character is obvious for banana tips, were $v_\varphi = 0$. We can normalize p_φ with $e\psi_0$. Note that p_φ provides a verification of the code, by checking its variation.

B.2.2 Toroidal angular momentum in 3D fields

With the result in axisymmetric fields we are able to validate the code without 3D fields. However, we wish to use 3D fields and still check if the radial excursion is determined with good precision. To do this, we find an expression how p_φ evolves in a known 3D field.

For this, we define the canonical linear momentum as:

$$\mathbf{P} = m\mathbf{v} + Z_i e\mathbf{A}, \quad (\text{B.31})$$

where the vector potential component is related to the ‘canonical’ part, i.e. the magnetic field is taken into account. The canonical angular momentum as $\mathbf{p} = R\mathbf{P}$. Again we will only look at the φ -components.

We can find an expression for the evolution of the linear momentum. Again, we make use of a force balance with the Lorentz force.

$$\begin{aligned} \frac{1}{Z_i e} \frac{d\mathbf{P}}{dt} &= \mathbf{E} &+ \mathbf{v} \times \mathbf{B} &+ \frac{d\mathbf{A}}{dt}, \\ &= -\frac{\partial \mathbf{A}}{\partial t} - \nabla \Phi &+ \mathbf{v} \times (\nabla \times \mathbf{A}) &+ \frac{\partial \mathbf{A}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{A}, \\ &= -\nabla \Phi &+ \mathbf{v} \times (\nabla \times \mathbf{A}) &+ \mathbf{v} \cdot \nabla \mathbf{A}, \end{aligned} \quad (\text{B.32})$$

We use a general vector identity to rewrite the term $\mathbf{v} \times (\nabla \times \mathbf{A})$ [33]:

$$\mathbf{v} \times (\nabla \times \mathbf{A}) = \nabla(\mathbf{v} \cdot \mathbf{A}) - \mathbf{A} \times (\nabla \times \mathbf{v}) - (\mathbf{A} \cdot \nabla)\mathbf{v} - (\mathbf{v} \cdot \nabla)\mathbf{A}. \quad (\text{B.33})$$

The velocity vector \mathbf{v} is the velocity of a particle, which is not a vector field and trivially does not depend on space. Therefore the ∇ -operator does not act on \mathbf{v} and hence the second and third term are zero. Additionally we can rewrite the last term to $-\mathbf{v} \cdot (\nabla \mathbf{A})$ to cancel the last term in equation (B.32). Now we find for the evolution of \mathbf{P} :

$$\frac{1}{Z_i e} \frac{d\mathbf{P}}{dt} = \nabla(\mathbf{v} \cdot \mathbf{A} - \Phi). \quad (\text{B.34})$$

Now that is out of the way, we can find the evolution of $p_\varphi = RP_\varphi$. Please note that in this coordinate system $\partial \mathbf{e}_R / \partial \varphi = \mathbf{e}_\varphi$ and $\partial \mathbf{e}_\varphi / \partial \varphi = -\mathbf{e}_R$.

$$\begin{aligned} \frac{d}{dt}(RP_\varphi) &= R \frac{d}{dt}(\mathbf{e}_\varphi \cdot \mathbf{P}) && + \frac{dR}{dt} P_\varphi, \\ &= R \mathbf{e}_\varphi \cdot \frac{d}{dt}(\mathbf{P}) &+ R \mathbf{P} \cdot \frac{d}{dt}(\mathbf{e}_\varphi) &+ v_R P_\varphi, \\ &= Z_i e R \mathbf{e}_\varphi \cdot \nabla(\mathbf{v} \cdot \mathbf{A} - \Phi) &+ R \mathbf{P} \cdot (\mathbf{v} \cdot \nabla(\mathbf{e}_\varphi)) &+ v_R P_\varphi, \\ &= Z_i e \frac{\partial}{\partial \varphi}(\mathbf{v} \cdot \mathbf{A} - \Phi) &+ \mathbf{P} \cdot \left(v_\varphi \frac{\partial \mathbf{e}_\varphi}{\partial \varphi} \right) &+ v_R P_\varphi, \\ &= Z_i e \mathbf{v} \cdot \frac{\partial \mathbf{A}}{\partial \varphi} - Z_i e \frac{\partial \Phi}{\partial \varphi} &- v_\varphi P_R &+ v_R P_\varphi. \end{aligned} \quad (\text{B.35})$$

Again we have used that \mathbf{v} does not depend on the spatial coordinates ($\partial\mathbf{v}/\partial\varphi = 0$). What remains from the left term is $\mathbf{v} \cdot \frac{\partial \mathbf{A}}{\partial \varphi}$, which is given by³:

$$\begin{aligned}\mathbf{v} \cdot \frac{\partial \mathbf{A}}{\partial \varphi} &= \mathbf{v} \cdot \left(\frac{\partial}{\partial \varphi} (A_R \mathbf{e}_R + A_Z \mathbf{e}_Z + A_\varphi \mathbf{e}_\varphi) \right), \\ &= \mathbf{v} \cdot \left(\frac{\partial A_R}{\partial \varphi} \mathbf{e}_R + \frac{\partial A_Z}{\partial \varphi} \mathbf{e}_Z + \frac{\partial A_\varphi}{\partial \varphi} \mathbf{e}_\varphi + A_R \mathbf{e}_\varphi - A_\varphi \mathbf{e}_R \right), \\ &= \frac{\partial A_R}{\partial \varphi} v_R + \frac{\partial A_Z}{\partial \varphi} v_Z + \frac{\partial A_\varphi}{\partial \varphi} v_\varphi + A_R v_\varphi - A_\varphi v_R.\end{aligned}\quad (\text{B.36})$$

In addition we know from the definition of the linear momentum \mathbf{P} :

$$-v_\varphi P_R + v_R P_\varphi = -v_\varphi (mv_R + Z_i e A_R) + v_R (mv_\varphi + Z_i e A_\varphi) = Z_i e (-v_\varphi A_R + v_R A_\varphi). \quad (\text{B.37})$$

Substituting both equation (B.36) and equation (B.37) in equation (B.35) we find our final expression for the evolution of p_φ :

$$\frac{1}{Z_i e} \frac{dp_\varphi}{dt} = \frac{\partial A_R}{\partial \varphi} v_R + \frac{\partial A_Z}{\partial \varphi} v_Z + \frac{\partial A_\varphi}{\partial \varphi} v_\varphi - \frac{\partial \Phi}{\partial \varphi}. \quad (\text{B.38})$$

If we assume there is no toroidal rotation, we find $\Phi = 0$ and can compute an ‘analytic’ evolution of p_φ by:

$$\Delta p_\varphi^n = Z_i e \Delta t \sum_{k=0}^n \left(\sum_{j=R,Z,\varphi} v_j \left(\frac{\partial A_j}{\partial \varphi} \right) \right) \Big|_{t=t_k}, \quad (\text{B.39})$$

with k the step number.

This can be computed during the simulation if the spatial derivatives of the vector potential are known.

³In general $\frac{\partial \mathbf{A}}{\partial \varphi} \neq \mathbf{0}$, not even in axisymmetry. In axisymmetry the components of \mathbf{A} are independent of φ , yet the direction of \mathbf{A} does change.

Appendix C

Inner workings of codes

Chapter 3 has been devoted to the general concepts of the code, focused on the physics. Any other coding aspect, for instance the interpolation and parameters will be dealt with in this appendix. We will discuss the three codes used: `FINESSE`, `EBdyna.go` and the `BS3D` code. After general information about the codes, some verification results have been shown. The last section provides a (step-by-step) manual.

C.1 Equilibrium reconstruction with PF2q and FINESSE

The `FINESSE` code is a well established code, the successor of the (well-known) code `HELENA`. `FINESSE` is an abbreviation for ‘`FINite Element Solver for Stationary Equilibria`’. It makes use of ‘finite elements’ to solve the Grad-Shafranov-equation (equation (2.28)). This work has used a single equilibrium, representative for a shot in AUG. Previous work on sawteeth (Jaulmes [40]) have used the same equilibrium, but reconstructed with `FINESSE` with a lower resolution. Studies for other equilibria are outside the scope of this work.

The `FINESSE` code is currently housed by the ITER organization, but also available within DIFFER. Instructions for the installation can be found at:

<http://wiki.differ.nl/differ-wiki/index.php/Finesse>, which requires a DIFFER account for access.

`FINESSE` makes use of the scaled p -and F -profiles¹, indicated with \tilde{p} and \tilde{F} [40]. These profiles are scaled by the inverse aspect ratio $\epsilon \equiv a/R_0$ and the ‘inverse flux parameter’ α , given by:

$$\alpha = \frac{a^2 B_0}{-\psi_0} \quad (\text{C.1})$$

Typically one would encounter the definition of α with the flux at the edge in other literature. Since ψ_0 is defined as the flux in on the magnetic axis, here one should use $-\psi_0$ to end up with the same equilibrium.

Both the \tilde{p} -profiles and the \tilde{F}^2 -profiles are given to `FINESSE` as 9th order polynomials. These can be found using the `PF2q`-tool [93]. `PF2q` enables one to try several p and F^2 -profiles with `FINESSE` easily to see which one reproduces the q -profile from an experiment best.

This Python tool, along with documentation, is currently hosted at:
<https://github.com/Karel-van-de-Plassche/PF2q>.

Note that `PF2q` ‘runs’ the `FINESSE` code to find the q -profile. The `main_script.py`-file redirects to a directory, in which is is required that the following directories are present:

- **DATA** in which the file `boundary.dat` is stored. This holds a boundary description with ~ 33 Fourier components.

¹ $F = RB_\varphi$ is the stream function in the Grad-Shafranov-equation given by equation (2.28) [31].

- **DATABASES** an empty directory used for temporary storage of FINESSE
- **INPUT** a directory with some FINESSE input files, e.g. error messages. Important is the input file `finesse.inp`. This file hold the \tilde{p} and \tilde{F}^2 -profiles (put there by PF2q) as well as other settings, e.g. the number of radial and poloidal grid points. We have used an equilibrium with 513 radial and 513 poloidal grid points ($= 2^9 \times 2^9$ finite elements), which is the maximum resolution.
- **OUTPUT** in which a log-files and the final output `.dat`-file is stored.

The output file that ends with `.dat` from FINESSE has to be put in the top directory of the `EBdyna.go` repository. Some scripts in this top directory can be used to import this data and convert it to MATLAB®-files (`.mat`). The directory `rebuild_equilibrium` holds numerous files to read this data and store them in the `data_tokamak` folder to be used by `EBdyna.go` and BS3D. These scripts are part of previous work and documentation is sparse. During this conversion, most parameters are converted from straight field line-coordinates (θ, ψ) to a grid defined by (R, Z) -coordinates. We assume from now on that this conversion is done and the equilibrium (2D) field is readily available.

C.2 BS3D

BS3D is housed in the `biot_savart_solver` directory in the `EBdyna.go` repository. Some parts of BS3D are devoted to RMPs in AUG or the TFR in AUG, but they have a similar structure. In figure C.1 an overview of functions and processes is given. Several ‘methods’, i.e. implementations, of the Biot-Savart law can be chosen. Some only determine **B**, some only **A** and some both. Differences in speed and amount of RAM required are also present.

Let’s first present the several ‘methods’ one can choose.

1. Loop over all filament elements, use vector based calculation for all evaluation points (all at once).
2. Loop over all evaluation points, use vector based calculation for all filament elements (all at once).
3. Loop over all filament elements, use vector based calculation for all evaluation points (all at once). This time a large cross product matrix is avoided to limit the demand on the memory.
4. Identical to method 3, but with calculation of the vector potential **A**. (*default method*)
5. Identical to method 3, but with only the calculation of the vector potential **A** and not the magnetic field **B**.
6. Calculation of **B** and **A** without loops, so in theory the fastest method. However, large matrices are required in the memory, making this method not applicable for our case.
7. Identical to method 6, but with only the calculation of the vector potential **A** and not the magnetic field **B**.

Restraints on the available RAM might limit the number of toroidal points in the 3D grid. In this work the vacuum field has been determined on 513 toroidal points². BS3D utilizes the toroidal periodicity of the perturbation, i.e. only 257 grid points are stored for the $n = 2$ mode and a mere 33 points for the TFR with $n = 16$. The matrix size stored in the RAM during `EBdyna.go` simulations is therefore roughly $1100 \times 1800 \times 257 \times 4$. These are the number of R, Z and φ grid points for four grid quantities. Three quantities determine **B** on the grid and the remaining one is A_φ . The latter is necessary to determine p_φ after simulations with equation (2.33).

²The first and last grid points overlap and are used to avoid interpolation errors.

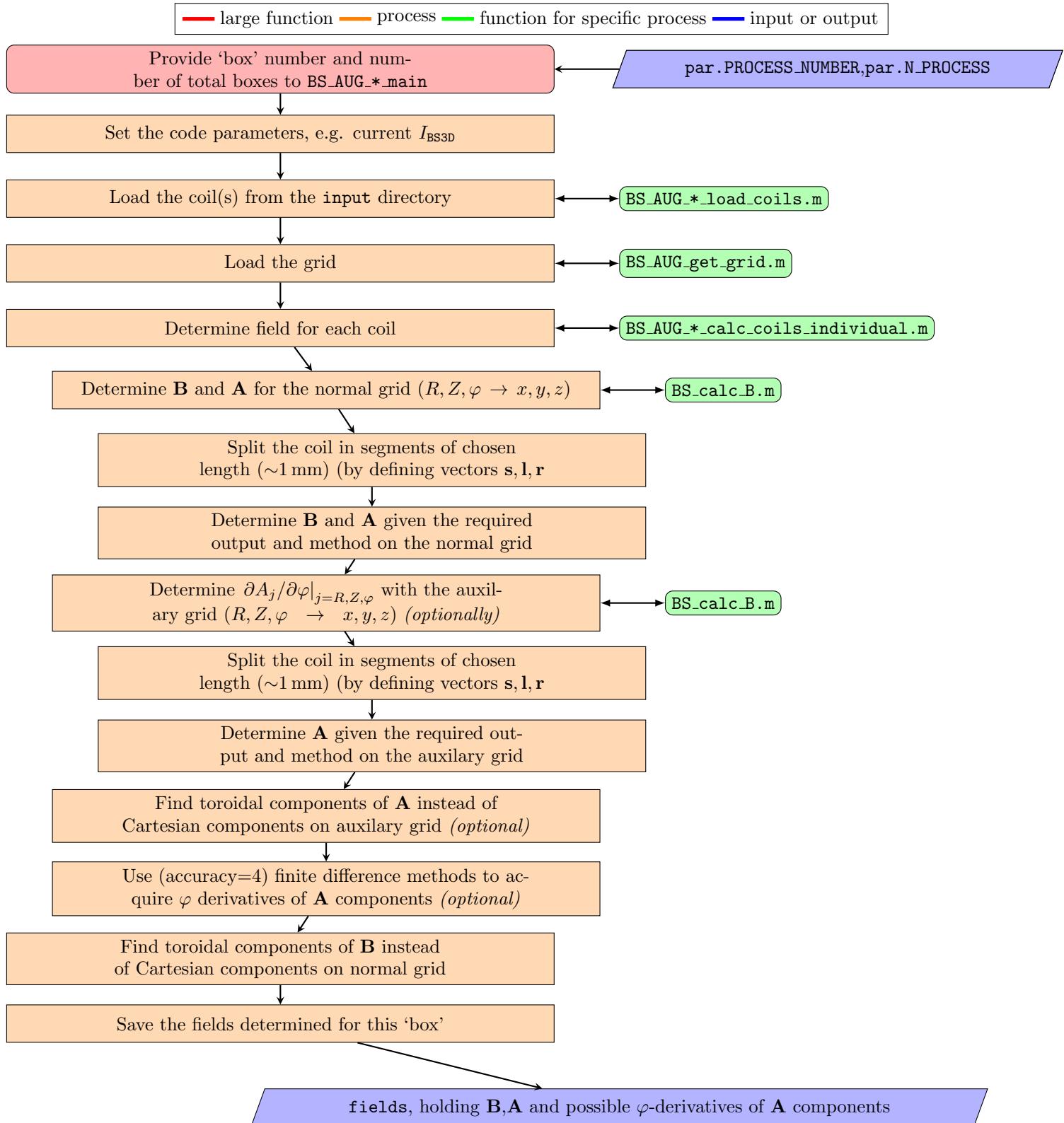


Figure C.1: Simplified overview of BS3D structure.

These quantities take up ~ 32 GB of RAM, but need to fit twice during the loading of the RAM (see appendix C). It is therefore the limit on the 64 GB nodes of **SurfSara Lisa**. Note that 3D validation using equation (2.34) require the three additional quantities $\partial A_R / \partial \varphi$, $\partial A_Z / \partial \varphi$ and $\partial A_\varphi / \partial \varphi$. Such 3D validations should thus be done on a node with more RAM or require a higher mode number n .

List of functions and parameters of BS3D can be found in appendix D.

C.3 EBdyna_go

Here we discuss the working of the `EBdyna.go`-code used for this thesis, which is housed in the `particles_equilibrium` directory. Other directories (e.g. `particles_collapse`) have been used in previous work ([40]), since then functionalities had been separated. The `EBdyna.go` code in this work makes use of ‘switches’ to simulate particles in either the original equilibrium, with 3D fields and/or with a sawtooth collapse.

First we will discuss some important functions and the structure of `EBdyna.go`. Next the interpolation schemes that are available are given. The `BORIS`-scheme, which finds the particle trajectory is discussed next. This sections ends with some verifications of the `EBdyna.go`-code. One can find a complete manual of `EBdyna.go` in appendix C.6.

C.3.1 Structure of EBdyna_go

Let’s have a look at *what* each function does. An overview is shown in figure C.2. The user starts by executing `G.eq`, the main function. In this script the user can alter parameters for the code (e.g. which 3D files to use or which interpolation scheme) (see appendix C.3.2). It will call several scripts that e.g. define the time step, the names of the files that are saved and initialize the fields and particles. Next the main simulation code `GT.eq`³ is called.

The main part of `GT.eq` is a loop. Beforehand, `EBdyna.go` will check if a previous simulation has save any intermediate data, called `RAW`-files. `EBdyna.go` will continue from these files if possible. It will also look for data about the trajectories, stored in `PREC`-files or in the distribution file itself. Each ‘loop’ or ‘time step’, consist of temporarily saving positions and velocity \mathbf{x}, \mathbf{v} . After ~ 100 time steps, these temporarily saved parameters are stored if the particle has been ejected to examine the positions of ejection. Afterwards output is evaluated by `evaluate_output.m`. For example, quantities as μ and p_φ are determined and saved in the `FULL`-file.

The full list of functions and what each one does is described in appendix D.1.3.

C.3.2 Interpolation

In order to find the magnetic field at the location of a particle we make use of a pre-defined field on a grid. This grid is either 2D or 3D, where the 3D grid can be based on straight field line-coordinates or toroidal coordinates. Interpolation has been used to find the best estimate given the position of the particle within a grid cell.

Interpolation grid

Physically, the magnetic field is ‘divergence free’, i.e. $\nabla \cdot \mathbf{B} = 0$ [33]. Unavoidably, the particle experiences discretization errors. A high resolution grid is therefore necessary to reduce the divergence of the magnetic field. The resolution of the grid is limited by the RAM of a computation node on the **SurfSara Lisa** which is 64 GB.

Initially a grid based on straight field line-coordinates has been used. One advantage is the low RAM requirement, since all the grid points are within the LCFS. However, problems arose

³The name `GT.eq` originates from *Gyro Tracking* in equilibrium. This is misleading, for which my sincere apologies. It is has to do with legacy, back when functionalities as e.g. sawtooth were housed in separate directories. The script *before* `GT.eq` has been given a shorter name: `G.eq`, no deep thoughts behind that....

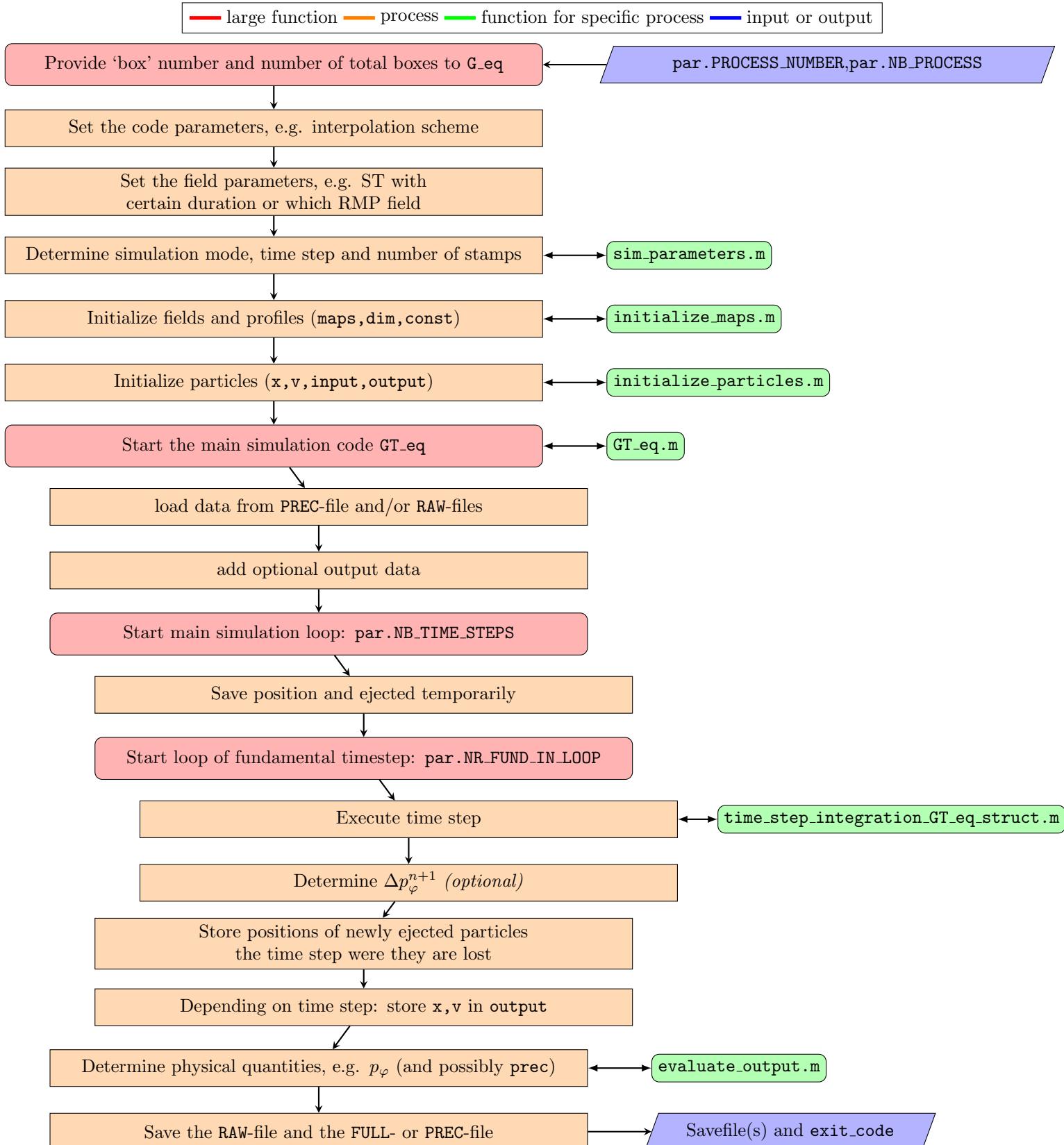


Figure C.2: Simplified overview of EBDYNA.GO.

due to the large spatial distance near $\theta \approx 0$ and $\psi \approx 0$ (around the LFS midplane and LCFS). In addition, θ has a very large gradient at the HFS mid-plane also near the LCFS. It is therefore hard to determine the poloidal angle θ by interpolation in (R, Z) -space.

Thirdly, it is computationally intensive, since the following steps had to be performed:

- Interpolate the equilibrium field (3 components) in an (R, Z) grid (3 interpolations)
- Interpolate the straight field line coordinates ψ and θ in an (R, Z) grid (2 interpolations)
- Interpolate the 3D magnetic field (3 components) on an (θ, ψ, φ) grid (3 interpolations)

Considering the numerical problems at $\psi \approx 0$ and $\theta \approx 0$ and these devious steps, the 3D grid has been revised. Presently, the 3D grid is based on toroidal coordinates (R, Z, φ) . Since the equilibrium field can now be superimposed on the 3D fields, only 3 interpolations in the (R, Z, φ) grid are necessary.

To accommodate for the increased demand on the RAM an additional function has been implemented. This function called `sharedmatrix` written by Dillon [17] enables the shared use of a section of the RAM for all processes on a node. This avoids having 16 times an identical magnetic field stored in the memory. During initialization, the total 3D field needs to be in the RAM twice, maximizing the size to 32 GB. Therefore, the choice has been made to use a grid with dimensions $1198 \times 1962 \times 513$ points in (R, Z, φ) ⁴. This, in MATLAB®-enviroment, still has a size of 64 GB. However, only half of the points is stored, since the field has a repetition of $n = 2$ in the worst case ($n = 16$ for the TFR).

This adds up to 32 points per oscillation of the ripple; enough to avoid effects due to divergence of the 3D field. Having more toroidal points during a TFR simulation would be possible, but problems might occur when trying to superimpose the RMP and TFR fields (see appendix C.2). Also the calculation of the Toroidal Field Ripple with BS3D would become costly, since it is required to determine the vacuum field on the all toroidal points for a single TF coil (see appendix C.2).

Interpolation schemes

Using the 4 (2D) or 8 (3D) nearest grid points, one can linearly interpolate the magnetic field to find an estimate for the magnetic field at the position of the particle. Cubic interpolation schemes make use of more surrounding points. In figure C.3 an overview of some interpolation schemes is given. For the interpolation method of choice one should consider that it is most important to have a divergence free magnetic field, i.e. $\nabla \cdot \mathbf{B} = 0$. This is most crucial for particle tracking in 3D [81].

One additional interpolation method has been implemented, which is only based on \mathbf{A}, ψ and $B_{eq,\varphi}$ and ensures a divergence free magnetic field. Here $B_{eq,\varphi}$ is the toroidal field determined by FINESSE. The gradients of \mathbf{A} and ψ are determined prior to the simulation and serve to calculate the magnetic field directly. Expectations arose that since one defines a vector field \mathbf{A} and a poloidal flux function ψ , $\nabla \cdot \mathbf{B} = 0$ is inherently ensured.

Unfortunately, this interpolation scheme turned out to suffer from discontinuities in the magnetic field at grid borders. To overcome this, very small time steps are necessary, making it useless for the large particle simulations. It can nevertheless be used as an extra check for the determination of \mathbf{B} by other interpolations.

A trade-off between grid size (which also reduces the divergence), step size Δt and the order of interpolation can occur. EBDYNA.GO is therefore equipped with several interpolation methods. These have been numbered and are shortly explained below:

0. Self-written ‘divergence-free’ interpolation scheme.

This interpolation scheme is analytically divergence free. Inside a grid cell we define coordinates $dr, dz, d\varphi$, which run from 0 to 1. We combine the 2D equilibrium by defining a combined ‘poloidal flux’: $\psi = \psi_{eq} - RA_\varphi$, with A_φ from the 3D field. The toroidal field

⁴The first two dimensions in (R, Z) have a historical background. During the initialization, this is reduced to 1100×1800 , which encloses the LCFS only just.

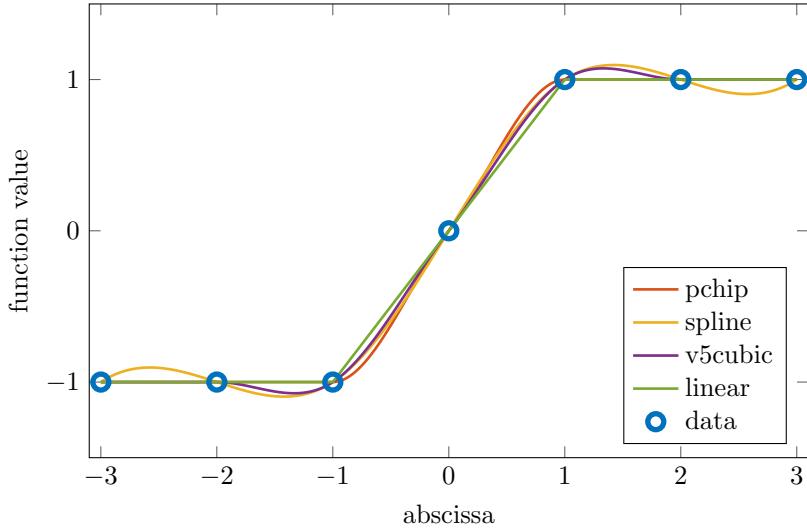


Figure C.3: Basic MATLAB® example of several interpolation methods. Note that Piecewise Cubic Hermite InterPolation ('pchip') is not an interpolation method available in `EBdyna.go`.

of the equilibrium $B_{eq,\varphi}$ is still interpolated directly. Linear interpolation of 2D quantities requires 4 neighboring grid points, indexed by 00,01,10 and 11, spanning a rectangle. Here the first number indicated the value of dr and the second of dz . Similarly, 3D quantities (e.g. A_R, A_Z and ψ) are interpolated using rectangles with 8 grid points, indexes with a third digit (the value of $d\varphi$). So we will define the following quantities in a cell (rectangle):

$$\begin{aligned} B_{eq,\varphi} &= [B_\varphi^{00}(1-dr) + B_\varphi^{10}dr] (1-dz) + [B_\varphi^{01}(1-dr) + B_\varphi^{11}dr] dz, \\ A_R &= \{[A_R^{000}(1-dr) + A_R^{100}dr] (1-dz) + [A_R^{010}(1-dr) + A_R^{110}dr] (dz)\} (1-d\varphi) \\ &\quad + \{[A_R^{001}(1-dr) + A_R^{101}dr] (1-dz) + [A_R^{011}(1-dr) + A_R^{111}dr] (dz)\} d\varphi, \\ A_Z &= \{[A_Z^{000}(1-dr) + A_Z^{100}dr] (1-dz) + [A_Z^{010}(1-dr) + A_Z^{110}dr] (dz)\} (1-d\varphi) \\ &\quad + \{[A_Z^{001}(1-dr) + A_Z^{101}dr] (1-dz) + [A_Z^{011}(1-dr) + A_Z^{111}dr] (dz)\} d\varphi, \\ \psi &= \{[\psi^{000}(1-dr) + \psi^{100}dr] (1-dz) + [\psi^{010}(1-dr) + \psi^{110}dr] (dz)\} (1-d\varphi) \\ &\quad + \{[\psi^{001}(1-dr) + \psi^{101}dr] (1-dz) + [\psi^{011}(1-dr) + \psi^{111}dr] (dz)\} d\varphi, \end{aligned}$$

From this we define a (divergence free) magnetic field, given by:

$$\mathbf{B} = \begin{pmatrix} \frac{1}{R} \left(-\frac{\partial A_Z}{\partial \varphi} - \frac{\partial \psi}{\partial Z} \right) \\ \frac{1}{R} \left(\frac{\partial \psi}{\partial R} + \frac{\partial A_R}{\partial \varphi} \right) \\ -\frac{\partial A_R}{\partial z} + \frac{\partial A_Z}{\partial R} + B_{eq,\varphi} \end{pmatrix} \quad (\text{C.2})$$

Note that $B_{eq,\varphi}$ is divergence free, since it is axisymmetric. All other fields are written in either \mathbf{A} or ψ , and thereby inherently divergence free.

To acquire the \mathbf{B} -field during a simulation, we require the rotation of \mathbf{A} , and therefore the spatial derivatives. Derivatives of \mathbf{A} to R, Z, φ are determined for each 'cell' a priori. To do

this as efficiently as possible we define the following quantities:

$$\beta_R = - (A_Z^{001} - A_Z^{000}) / \Delta\varphi - (\psi^{010} - \psi^{000}) / \Delta Z, \quad (C.3)$$

$$\beta_Z = + (\psi^{100} - \psi^{000}) / \Delta R + (A_R^{001} - A_R^{000}) / \Delta\varphi, \quad (C.4)$$

$$\beta_\varphi = - (A_R^{010} - A_R^{000}) / \Delta Z + (A_Z^{100} - A_Z^{000}) / \Delta R + B_\varphi^{00}, \quad (C.5)$$

$$\beta_{\varphi,R} = - (A_R^{110} - A_R^{100} - A_R^{010} + A_R^{000}) / \Delta R / \Delta Z + (B_\varphi^{10} - B_\varphi^{00}) / \Delta R, \quad (C.6)$$

$$\beta_{\varphi,Z} = + (A_Z^{110} - A_Z^{100} - A_Z^{010} + A_Z^{000}) / \Delta R / \Delta Z + (B_\varphi^{01} - B_\varphi^{00}) / \Delta Z, \quad (C.7)$$

$$\beta_{R,Z} = - (A_Z^{011} - A_Z^{010} - A_Z^{001} + A_Z^{000}) / \Delta R / \Delta\varphi, \quad (C.8)$$

$$\beta_{Z,R} = + (A_R^{101} - A_R^{100} - A_R^{001} + A_R^{000}) / \Delta R / \Delta\varphi, \quad (C.9)$$

$$\beta_{Z,\varphi} = - (\psi^{101} - \psi^{100} - \psi^{001} + \psi^{000}) / \Delta R / \Delta\varphi, \quad (C.10)$$

$$\beta_{R,\varphi} = + (\psi^{011} - \psi^{010} - \psi^{001} + \psi^{000}) / \Delta Z / \Delta\varphi, \quad (C.11)$$

$$\beta_{\varphi,RZ} = + (B_\varphi^{11} - B_\varphi^{10} - B_\varphi^{01} + B_\varphi^{00}) / \Delta R / \Delta Z, \quad (C.12)$$

$$\gamma_R = - (A_R^{111} - A_R^{110} - A_R^{011} + A_R^{010}) / \Delta R / \Delta\varphi + \beta_{Z,R}, \quad (C.13)$$

$$\gamma_Z = - (A_Z^{111} - A_Z^{110} - A_Z^{011} + A_Z^{010}) / \Delta Z / \Delta\varphi - \beta_{R,Z}, \quad (C.14)$$

$$\gamma_\varphi = - (\psi^{111} - \psi^{110} - \psi^{011} + \psi^{010}) / \Delta R / \Delta\varphi - \beta_{Z,\varphi}, \quad (C.15)$$

$$\alpha_R = - (A_Z^{101} - A_Z^{100}) / \Delta\varphi - (\psi^{110} - \psi^{100}) / \Delta Z - \beta_R, \quad (C.16)$$

$$\alpha_Z = + (A_R^{011} - A_R^{010}) / \Delta\varphi + (\psi^{110} - \psi^{010}) / \Delta R - \beta_Z, \quad (C.17)$$

These quantities enable a fast calculation of the **B**-field by:

$$\beta_{R,R} = \alpha_R + dz \Delta Z \gamma_Z + d\varphi \Delta\varphi \gamma_\varphi,$$

$$\beta_{Z,Z} = \alpha_Z - dr \Delta R \gamma_R - d\varphi \Delta\varphi \gamma_\varphi,$$

$$B_R = (\beta_R + dr \Delta R \beta_{R,R} + dz \Delta Z \beta_{R,Z} - d\varphi \Delta\varphi \beta_{R,\varphi}) / R,$$

$$B_Z = (\beta_Z + dr \Delta R \beta_{R,Z} + dz \Delta Z \beta_{Z,Z} - d\varphi \Delta\varphi \beta_{R,\varphi}) / R,$$

$$B_\varphi = \beta_\varphi + dr \Delta R \beta_{\varphi,R} + dz \Delta Z (\beta_{\varphi,Z} + dr \Delta R \beta_{\varphi,RZ} / RZ) - d\varphi \Delta\varphi (\beta_{Z,Z} + \beta_{R,R}),$$

which boils down to exactly equation (C.2). Thus based on the grid solely, the divergence is exactly 0. The 2D variant can be simplified further (which has been done in the `EBdyna_go` code).

1. Self-written linear interpolation

An obsolete collection of functions is housed in the `01 general functions` directory. This interpolation method determines (in MATLAB® code) the indexes of the four or eight neighboring grid points of each particle. By using the ‘slopes’ dr , dz and possibly $d\varphi$, the interpolation method makes a linear interpolation. This method is faster than `interp`, slightly slower than `griddedInterpolant` and slower than `ba_interp`.

2. Linear interpolation by MATLAB® function: `interp`

For comparison one can use linear interpolation by a standard MATLAB® function: `interp`. It is relatively slow and provides the same results as `ba_interp`. `EBdyna_go` switches to `interp2` or `interp3` according to the 2D or 3D (surface or volume) interpolation.

3. Linear interpolation by custom function: `ba_interp` [4, 5]

Similarly to the `interp`-function, the two functions `ba_interp2` and `ba_interp3` are used for 2D and 3D interpolations. This method is fastest and therefore recommended and RAM costs are low. These functions have to be compiled (once) on the system used, which can be

done with the functions: `make_ba_interp2_function` and `make_ba_interp3_function` (also stored in `01_general_functions` directory)⁵.

4. **Linear interpolation by MATLAB® function:** `griddedInterpolant`

The `griddedInterpolant` is a MATLAB® function optimized if for multiple interpolations of a given (constant) quantity, e.g. B_R . A function (e.g. F) is declared a priori. During the first interpolation, this function is created and polynomials for the interpolation are stored. The second time around, only these polynomials have to be evaluated. Note that the (polynomials of) the produced function *cannot* be in a shared memory.

5. **Cubic interpolation by MATLAB® function:** `interp` A cubic interpolation method, which can differ between ‘`pchip`’ or ‘`v5cubic`’ depending on the MATLAB®-version.⁶

6. **Cubic interpolation by custom function:** `ba_interp` [4, 5] A faster cubic interpolation method (‘`v5cubic`’).

7. **Cubic interpolation by MATLAB® function:** `griddedInterpolant` An also quite fast cubic interpolation method (‘`v5cubic`’). However, the RAM demand is higher than `ba_interp`.

8. **Spline interpolation by MATLAB® function:** `interp`

9. **Spline interpolation by MATLAB® function:** `griddedInterpolant`

Some interpolation schemes produce an error to avoid usage, since they result in painfully slow or RAM intensive simulations. One could remove these error messages from `EBdyna.go`.

One should take care that the definition of ‘cubic’ interpolation has changed in recent MATLAB® versions and is currently the ‘Piecewise Cubic Hermite Polynomial’ interpolation. When used, consider verifying which one is used exactly. Specifying `pchip` for the Piecewise Cubic Hermite Polynomial interpolation is recommended. Using cubic interpolation is risky, since the polynomials can result in an oscillating function near the LCFS (see figure C.3).

Cubic interpolation with `ba_interp` is doable, but requires much more time than linear interpolation. `ba_interp` only supports one ‘type’ of cubic interpolation, similar to the traditional ‘cubic’ interpolation (dubbed `v5cubic`). Cubic interpolation of `griddedInterpolant` is possible, yet also care needs to be taken with the ‘type’ of cubic interpolation and the higher demand on the RAM. When the argument ‘`cubic`’ is given to `griddedInterpolant`, the performed interpolation is similar to the `v5cubic` interpolation of `interp`. Since the enormous calculation time and the RAM requirement, the ‘`spline`’ interpolation of `interp` is not advised. Spline interpolation of `griddedInterpolant` might be possible for a low RAM grid (e.g. 2D interpolation). It is a higher order interpolation than cubic, but at the cost of (a lot of) RAM. Not every interpolation method is compatible with the sawtooth simulation, simply because it has not been implemented.

In order to test interpolation scheme #0, from hereon dubbed **A** based interpolation, we consider the conservation of canonical toroidal angular momentum p_φ . We compare the **A** based interpolation in the most fine grid for several fundamental time steps Δt to the ‘default’ `EBdyna.go` interpolation scheme: #3 a linear interpolation by `ba_interp` with $\Delta t = 0.1$ ns. In figure C.4 the evolution of p_φ normalized to $Z_i e \psi_0$ is shown for two particles.

A similar conservation of p_φ is reached once $\Delta t \leq 0.1$ ns in the **A** based interpolation scheme. This is considered too small for realistic lengths of simulations. Yet it does provide a kind of comparison with the other interpolation methods. Secondly we conclude that although the field is completely divergence free on the grid, a particle might experience a divergence. Discontinuity

⁵When compiling e.g. the `ba_interp2`-function on a WINDOWS machine, one should first remove `ba_interp2.cpp` and then rename the file `ba_interp2_windows.cpp` to `ba_interp2.cpp`. Similarly one needs to rename compilation files for `ba_interp3`. After compiling one can revert the changes to these .cpp-files.

⁶Note that method 5 is meant to be ‘normal cubic’, i.e. ‘`v5cubic`’. The `interp1`-function in some MATLAB®-versions uses an alternative cubic interpolation known as Piecewise Cubic Hermite Polynomial (‘`pchip`’) when the argument ‘`cubic`’ is passed. Nevertheless, `griddedInterpolant`, `interp2` and `interp3` do use ‘`v5cubic`’.

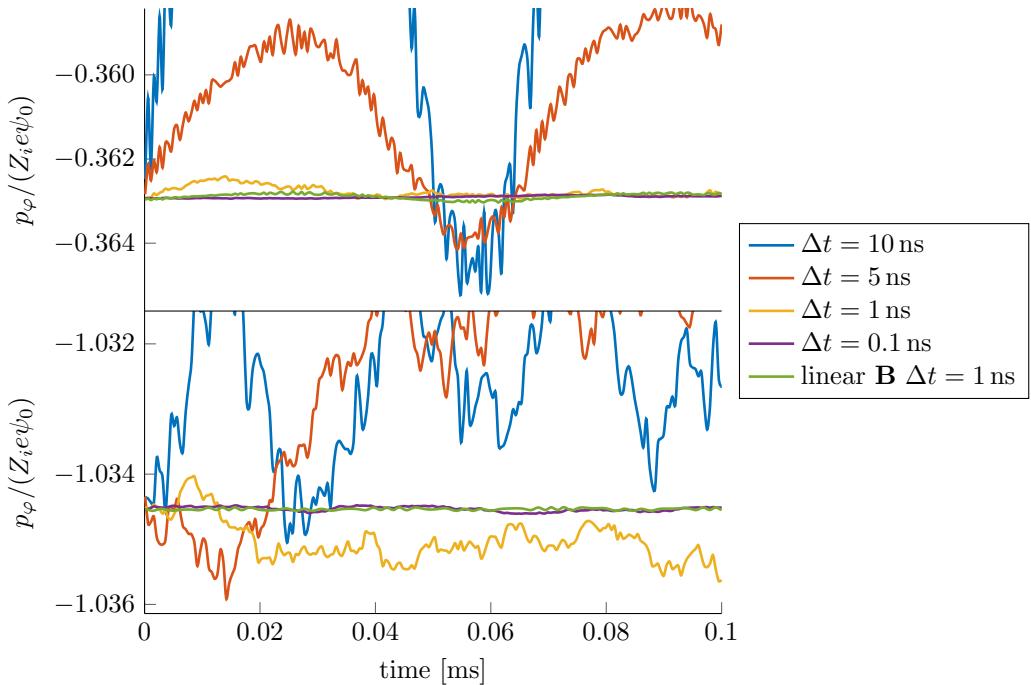


Figure C.4: Behaviour of p_φ for 2 particles $E_{kin} = 20 \text{ keV}$ using the \mathbf{A} based interpolation in 2D equilibrium in comparison with the linear \mathbf{B} interpolation scheme (#3). Though $\nabla \cdot \mathbf{B} = 0$ is ensured by interpolation scheme #0, p_φ is not conserved (as well as the other interpolation scheme). Evidently the particle ‘experiences’ some discontinuity / divergence and thus this interpolation didn’t perform as good as the default linear interpolation.

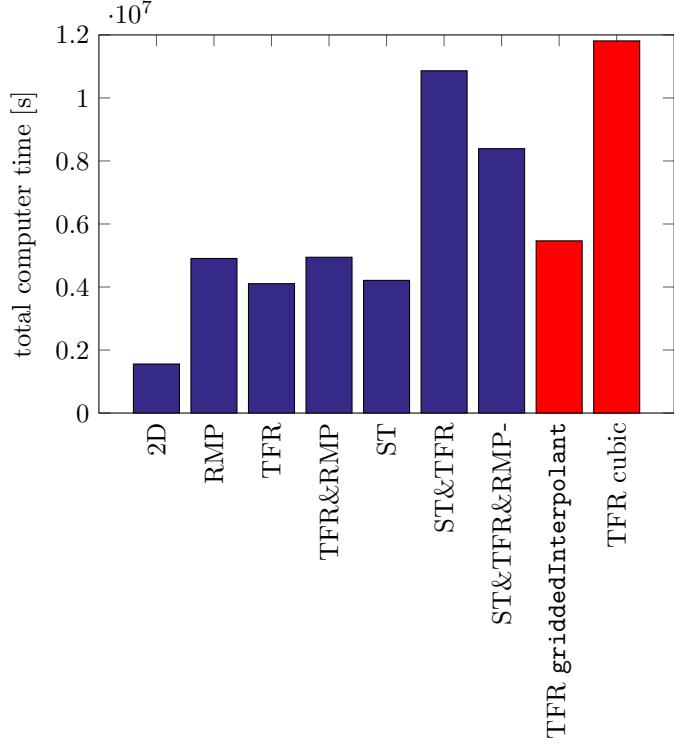


Figure C.5: Comparison of total computation time for the NBI distribution (see section 4.1) of all ‘boxes’ combined. There is little difference between sawtooth (ST), RMP ($n = 2$) and TFR simulations (TFR can do with a smaller 3D grid). Note that when particles are ejected, the computer time lowers, which might play a role here. Times shown here are with a linear interpolation by `ba_interp`. In red two different interpolation methods are shown, one based on linear interpolation using `griddedInterpolant` and also a cubic interpolation method by `ba_interp` (see appendix C.3.2).

in the fields at the grid point boundaries are experienced ‘heavier’ / are amplified when the time step when crossing such boundary is larger.

`EBdyna_go` requires an interpolation to acquire \mathbf{B} on each step. This is by far the most costly step and can be performed in different grids. With a 2D equilibrium, a 2D grid is sufficient R, Z . However, when a toroidal dependency is applied, an additional interpolation in a 3D grid is necessary (R, Z, φ). Instead of superimposing the result of both interpolations, `EBdyna_go` will superimpose the fields beforehand. This makes the 2D interpolation obsolete.

Unfortunately, this speedup cannot apply when the sawtooth is modeled as well. Particles which are affected by the sawtooth obtain a poloidal field \mathbf{B}_{pol} from the model, and require the 3D field to be added as a perturbation only. Storing both a superimposed 3D field and the 3D perturbation only has not been implemented due to foreseen memory issues. Thus, all particles require two interpolations, i.e. one in the 2D equilibrium and another in the 3D perturbation. An impression of the computer times is provided in figure C.5. Here we introduce the term ‘box’, used for parallelization, i.e. the particles are divided into parts called ‘boxes’.

C.3.3 Trajectory integration with BORIS

The equation of motion (equation (2.30)) is a differential equation that needs to be solved by `EBdyna_go`. The choice has been made to solve this differential equation with the `BORIS` algorithm [10, 88], which is the ‘de facto’ standard.

This algorithm is well known for its stability and performance for particle tracking in magnetic

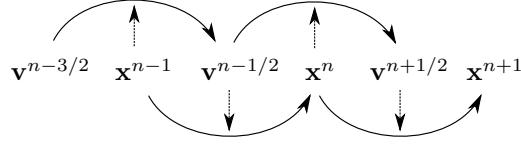


Figure C.6: Schematic representation of 2 leap-frog iterations. The velocity \mathbf{v} is known on half integer steps and used to find the positions at integer steps. Conversely, the integer positions are used for the velocity update, i.e. the \mathbf{E} - and \mathbf{B} -fields at the integer positions.

fields [71]. Extensive literature is available about this algorithm, e.g. [16, 65, 66, 88]. It is lightweight in computational cost and second-order accurate in time. We cannot straightforwardly assume that higher order schemes are beneficial [99]. Hence they aren't investigated in this work.

The BORIS integration method is a ‘leap-frog’ method. This means that the velocity is known at half time steps, whilst the position is known at integer time steps. The current time step is indicated with n . Determining the electric and magnetic fields at time step n provides an estimate for the acceleration between $n - 1/2$ and $n + 1/2$. Hence the change in velocity from $\mathbf{v}^{n-1/2}$ to $\mathbf{v}^{n+1/2}$ can be determined. The velocity $\mathbf{v}^{n+1/2}$ is then used to ‘push’ the particle from \mathbf{x}^n to \mathbf{x}^{n+1} as sketched in figure C.6.

The BORIS method consists (in our case) of 5 steps to incorporate the electric field \mathbf{E} and the magnetic field \mathbf{B} :

1. Apply the electric field \mathbf{E} for the initial half of the step:

$$\mathbf{v}^- = \mathbf{v}^{n-1/2} + \frac{Z_i e}{m} \mathbf{E} \frac{\Delta t}{2}. \quad (\text{C.18})$$

2. Apply a rotation of the velocity according to the magnetic field:

$$\mathbf{v}^+ = \mathbf{v}^- + \frac{2f}{1 + f^2 B^2} (\mathbf{v}^- \times \mathbf{B} - f B^2 \mathbf{v}^- + f(\mathbf{v}^- \cdot \mathbf{B}) \mathbf{B}), \quad (\text{C.19})$$

with $f = \tan\left(\frac{q}{m} \frac{\Delta t}{2} B\right)/B$.

3. Apply the electric field \mathbf{E} for the second half of the step:

$$\mathbf{v}^{n+1/2*} = \mathbf{v}^+ + \frac{Z_i e}{m} \mathbf{E} \frac{\Delta t}{2}. \quad (\text{C.20})$$

4. Update the positions of the particles accordingly:

$$R^{n+1} = \sqrt{\left(R^n + v_R^{n+1/2*}\right)^2 + \left(v_\varphi^{n+1/2*} \Delta t\right)^2}, \quad (\text{C.21})$$

$$Z^{n+1} = Z^n + v_Z^{n+1/2*}, \quad (\text{C.22})$$

$$\varphi^{n+1} = \varphi^{n+1} + \arcsin\left(v_\varphi^{n+1/2*} \Delta t / R^n\right). \quad (\text{C.23})$$

5. Note the * symbol used for the velocity components. This has been done to indicate the components are expressed for position \mathbf{x}^n . Since the coordinate system is toroidal (R, Z, φ) the unit vectors change. Thus after the position update the components need to be have a *cylindrical correction* to match the unit vectors at position \mathbf{x}^{n+1} . The change of \mathbf{e}_φ and \mathbf{e}_R change during the motion is expressed with the rotation:

$$v_R^{n+1/2} = \cos(\alpha) v_r^{n+1/2*} + \sin(\alpha) v_\varphi^{n+1/2*}, \quad (\text{C.24})$$

$$v_\varphi^{n+1/2} = -\sin(\alpha) v_r^{n+1/2*} + \cos(\alpha) v_\varphi^{n+1/2*}, \quad (\text{C.25})$$

with $\sin(\alpha) = \frac{v_\varphi^{n+1/2*} \Delta t}{R^{n+1}}$ and $\cos(\alpha) = \frac{R^n + v_r^{n+1/2*} \Delta t}{R^{n+1}}$.

As mentioned in appendix A, help from ir. D.C. (Daan) van Vugt from Eindhoven University of Technology, has been indispensable. The following explanation of the BORIS-scheme presented are to his credit. Let's start with the equation of motion (equation (2.30)), here repeated for convenience:

$$m\dot{\mathbf{v}} = q[\mathbf{E} + \mathbf{v} \times \mathbf{B}],$$

When discretized, the equation of motion has the following form, known as the ‘centered difference form’

$$\frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\Delta t} = \frac{q}{m} \left[\mathbf{E} + \frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{2} \times \mathbf{B} \right], \quad (\text{C.26})$$

with \mathbf{E} and \mathbf{B} at time step n and position \mathbf{x}^n .

With the expressions for the half-steps \mathbf{v}^- and \mathbf{v}^+ we find

$$\frac{\mathbf{v}^+ - \mathbf{v}^-}{\Delta t} = \frac{q}{2m} (\mathbf{v}^+ + \mathbf{v}^-) \times \mathbf{B}^n, \quad (\text{C.27})$$

showing that \mathbf{v}_\perp^+ is a rotation from \mathbf{v}_\perp^- . This rotation β is given by the equation:

$$\frac{|\mathbf{v}_\perp^+ - \mathbf{v}_\perp^-|}{|\mathbf{v}_\perp^+ + \mathbf{v}_\perp^-|} = |\tan(\beta/2)| = \frac{qB}{2m} \Delta t, \quad (\text{C.28})$$

One can now find equation (C.19) with $f = \frac{q}{m} \frac{\Delta t}{2}$ to solve the equation of motion. However, the angle of rotation β *should* be $\omega_c \Delta t$, thus having the correct gyration velocity. To correct for this error, we can just use an alternative time step during the iteration, i.e. $f = \tan(\frac{q}{m} \frac{\Delta t}{2} B) / B$ [65].

C.3.4 Simulation modes

In order to analyse particle distributions in the chosen fields, several ‘modes’ are available.

1. A ‘test’ mode, to simulate a small number of particles.
2. A ‘precession’ mode, which is used to ‘try out’ the distribution in an equilibrium. Characteristics can be obtained from this, like which one is trapped and passing, or the values of ω_b and ω_d . This requires a frequent storing of parameters and a sufficiently long simulation. It was found sufficient to sample every $\sim 2 \mu\text{s}$ for 1 ms long.
3. A ‘full’ simulation, where the time and sampling of choice is used.
4. A fourth obsolete mode
5. A Poincaré mode, in which one can make either a Poincaré plot or make a figure showing the orbit of particles during a sawtooth crash as used for figure C.13.

One can, by setting a parameter, let `EBdyna.go` first execute mode (2), followed by mode (3). Of course, the precession needs to be done only once for each distribution. How to execute this has been described in appendix C.6.4.

Furthermore, one can (with some manual postprocessing) determine where the midplane ‘actually’ lies. Since the shaping of the plasma is asymmetric (‘X’-point at the bottom), particles can be trapped without crossing $Z = 0$.

A particle distribution of 6×10^6 has been used, of which 1.3×10^6 were trapped (and confined). Of each orbit, the position with the maximum Z and minimum Z value were stored. The average between these positions is shown in figure C.7.

A linear fit is made, to determine the coefficients a, b which define the new midplane $\hat{Z} = aR + b$. The fit can be encoded in `EBdyna.go`, which would ‘count’ for each particle the number of crossings. Particles which do not cross the newly defined midplane are identified as ‘ripple’ trapped.

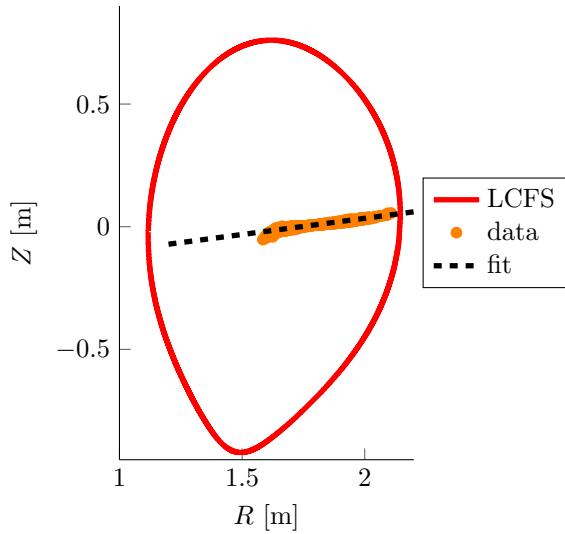


Figure C.7: Result of equilibrium simulation where the maximum and minimal vertical points have been ‘averaged’ for trapped particles. A total of 6×10^6 particles are used of which 6000 are shown here. The average position forms the basis for a linear fit, determining the actual midplane.

Particles that bounce but never cross this new midplane are identified as *ripple* trapped.

C.3.5 Input and output

The quantities used by `EBdyna.go` are listed in appendix D.2.2. Here we discuss which of these quantities need to be defined in the input file. Secondly we discuss the output files that `EBdyna.go` creates.

Input

Within the `particles_equilibrium` directory one finds the two directories: `input` and `output`. The input `input` directory would need to hold the initial distribution. This `.mat`-file can be made scripts from the `build_intial_distributions`-directory or by hand. The following fields are read from this file if the file is an older distribution file:

- `Nalphas_simulated` (*integer*) (if present) containing the number of particles.
- `alphas_Ekin` (*vector*) Kinetic energy
- `mHe` and `ZHe` (*scalars*) The mass and charge state of the particle respectively.
- `alphas_pos_x`, `alphas_pos_z`, `alphas_pos_phi` (*vectors*), *gyrocenter* of the particles, in (X, Z, φ) -coordinates, in which $X = R - R_0$.
- `v_X`, `v_Z`, `v_phi` OR `alphas_vp11` (*vectors*), components of velocity vector of particles, where $v_X = v_R$ OR one specifies the parallel velocity $v_{||}$ component (negative for counter parallel).

In case the parameter `Nalphas_simulated` is not specified in the distribution file, but the parameter `x` is, the file is identified as a newer distribution file. Fields that can be processed from such an input file are:

- `input` (*struct*), containing mass (`m`), charge state (`Z`)), kinetic energy (`E_kin`) and any optional input quantities.
- `x` (*matrix: 2D*), containing the *gyrocenter* of particles. The three columns represent the (R, Z, φ) -components respectively.

- `v` OR `vp11` (*matrix: 2D OR vector*), containing the velocity of the each particles OR the parallel components.
- `rare` (*vector*), optional quantity with the indexes of particles that are requested. Once a ‘rare’ field is found, the simulation will extract the indicated particles from the distribution and simulate the trajectory of these.

The simulation stored data in the `output`-struct. Originally, these are 3D-matrices of `x` and `v`, in which the 3rd dimension is the time stamp. The `evaluate_output` function is able to add physical quantities (e.g. p_φ) to this.

Output

`EBdyna.go` can produce four types of output files. Since `EBdyna.go` is parallelized, the name of these files contains the process number, here denoted as `<NR>`. Note that each simulation should have an unique `<ID>`. The first two should be related to the distribution and thereon one is free to use any string. One could for instance make a ‘precession’ simulation (`PREC`-file) with ID `19xx`. Any following simulation with an ID starting with `19` could use this precession file for information about particle orbits in 2D. Possible output files of `EBdyna.go` are:

- `G_eq_<ID>_raw_process<NR>.mat`-files, which are either stored intermediately or after a simulation. It holds the `output`-struct with the `x`- and `v`-fields and also the `par`-struct with parameters. At the end of a simulation the `input`-struct is added. The `RAW`-files are can be used to study orbits, for debugging or to add additional data to the output file in a later stage.⁷
- `G_eq_<ID>_prec_process<NR>.mat`-files, produced simulation mode 2. This simulation tracks particles in 2D to categorize particles and find equilibrium quantities, e.g. ω_b . These are stored in the `prec`-struct.
- `G_eq_<ID>_prec_process<NR>.matprec`-files, which are rarely produced. These files are leftovers if the categorization failed. This might occur when all particles are lost in the 2D equilibrium. Any other reason indicates an failure to execute the `extract_precession_information_struct`-function.
- `G_eq_<ID>_full_process<NR>.mat`-files, which holds the output *with* quantities derived from `x` and `v`. Numerous derived quantities could be stored, e.g. evolution of E_{kin} .

C.4 Sawtooth

Implementation of the sawtooth model is intertwined in `EBdyna.go`. Every necessary code for the sawtooth crash can be found within `G.eq.m`, `initialize_maps.m` and `B.interpolation.m`.

When the parameter are chosen correctly, `EBdyna.go` will simulate a sawtooth crash using the following steps (each time step)

1. Interpolate in the 2D equilibrium field (from before the sawtooth crash). This equilibrium is used for particles with $r > 1.1r_{mix}$ and only the toroidal component B_φ is used for $r \leq 1.1r_{mix}$ (only \mathbf{B}_{pol} is altered by the sawtooth model). In figure C.8 this boundary is shown in the poloidal cross section.
2. Use a combination of interpolation and equations are used to determine the parameters of the sawtooth, e.g. r_1, κ_r, κ .

⁷ Note that the `FULL`-file, which is intended for evaluation, does not contain `x` and `v` in order to reduce the file size. When `EBdyna.go` encounters a `RAW`-file present, it will (try to) continue from this point on. A warning is issued, but might be useful during debugging. Continuing from a `RAW`-file is only possible if the ID, the ‘mode’, the `time_scale`, the ‘box’ number and total amount of boxes are equal.

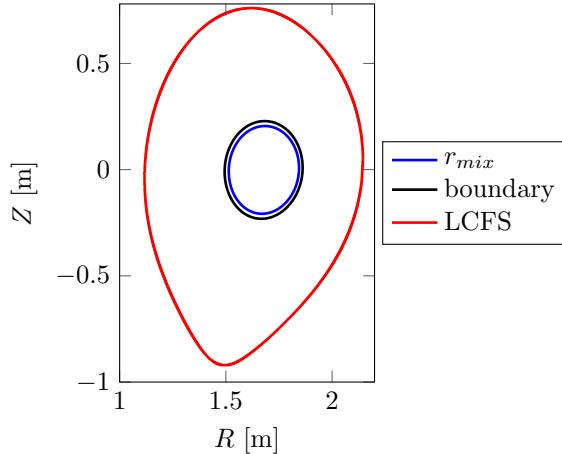


Figure C.8: Position of r_{mix} in R, Z space and the boundary separating when particles make use of the sawtooth algorithm or interpolation in the ‘regular’ 2D equilibrium.

3. Determine the potential Φ and flux $\tilde{\psi} = \psi_* + \chi$ in the three regions at the particle location R, Z and at $R + \epsilon, Z$ and $R, Z + \epsilon$, with $\epsilon, \epsilon \sim 1 \times 10^{-9} \text{ m} \ll 1$
4. Determine for each particle with $r < 1.1r_{mix}$ the fields: $\mathbf{E}_{\text{pol}} = -\nabla\Phi$ and $\mathbf{B}_{\text{pol}} = \frac{1}{R}\nabla\tilde{\psi} \times \mathbf{e}_\varphi$.
5. Acquire E_φ from $\mathbf{E} \cdot \mathbf{B} = 0$.
6. If a 3D fields applies, find its perturbation by interpolation in the ‘regular’ 3D grid⁸.

Sawtooth parameters can be differently determined. Here the choice is to use a linearly decreasing $r_1(t)$. Based on r_1 and ψ_- one finds $r_2(t)$. Its time derivative $\dot{r}_2(t)$ is acquired by considering the conservation of helical flux ψ_* , which implies

$$\dot{r}_2 = \dot{r}_1 \frac{\psi'_1}{\psi'_2}, \quad (\text{C.29})$$

where ψ'_i is found with equation (3.11).

Evolution of r_2 can be found by sampling r_1 on ~ 1000 points ‘a priori’ for the reconnection phase. Based thereon, `EBdyna.go` will determine the evolution of the following sawtooth parameters a priori:

- r_1, r_2 and $r_3 = \sqrt{r_2^2 - r_1^2}$
- q_1 and q_2
- ψ'_1 and ψ'_2
- $\dot{r}_1, \dot{r}_2 = \dot{r}_1 \psi'_1 / \psi'_2$ and $\dot{r}_3 = \frac{\dot{r}_2 r_2 - \dot{r}_1 r_1}{r_3}$
- $\kappa_r \equiv \frac{r_2 - r_1}{r_1 + r_2}$
- $\dot{\kappa}_r = 2 \frac{r_1 \dot{r}_2 - r_2 \dot{r}_1}{(r_1 + r_2)^2}$

⁸If no sawtooth is applied, the 2D equilibrium is superimposed on the 3D grid to save computer time. However, this isn’t applicable for the particles within the sawtooth boundary. Therefore, particles outside the sawtooth boundary will experience a different numerical scheme. Instead of a single interpolation in the 3D grid (complete field), there will be an interpolation in the 2D grid (equilibrium) and another interpolation in the 3D grid (perturbation). These two, superpositioned, would in theory result in the same field. However, small numerical deviations in combination with the stochasticity did influence some particles.

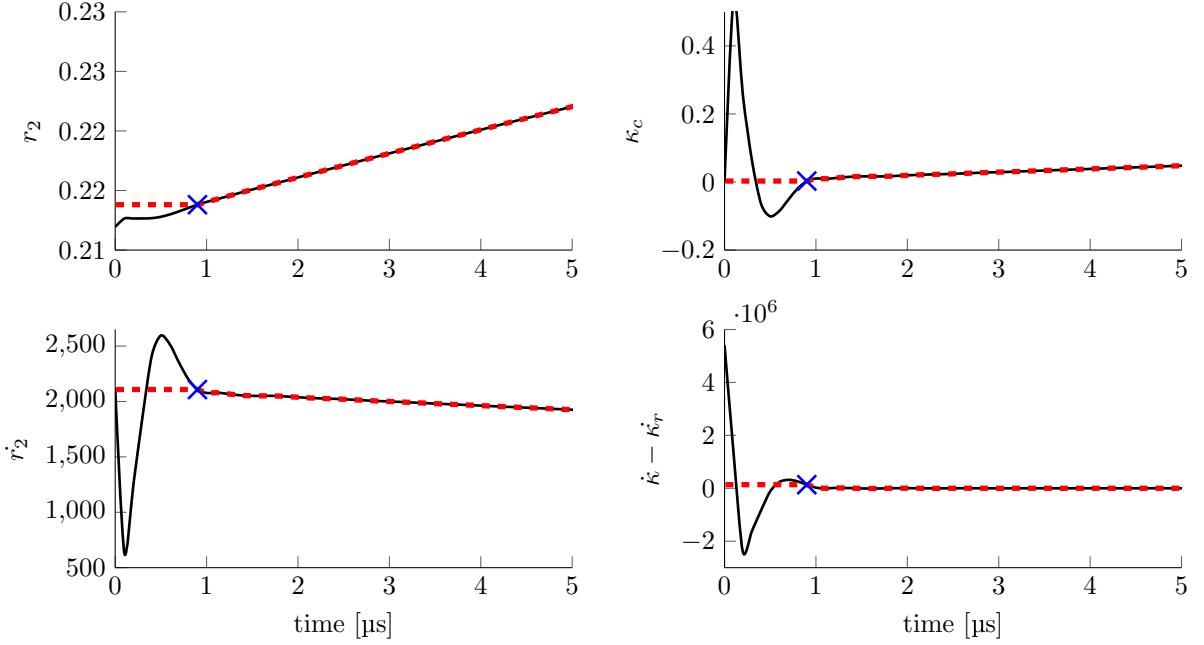


Figure C.9: Numerical instability at start of the sawtooth crash result in high inaccuracies of $\kappa, \dot{\kappa} - \kappa_c$ and r_2 . The black lines show the parameters according to the a priori determined reconnection and the equations. The red line shows the path of the model. The model uses the sawtooth fields at 9 μs (linearly interpolating with \mathbf{B}_{eq}) after which the sawtooth fields are used according to the model.

- $\kappa_c = \frac{2}{r_1+r_2} \frac{\psi'_1+\psi'_2}{q_1-q_2}$
- κ , which is κ_c during reconnection and linearly decreasing to zero afterwards.
- $\dot{\kappa} - \dot{\kappa}_r$, where $\dot{\kappa}$ is determined numerically.

However, once r_2 and \dot{r}_2 are known, we do not need to find q_i and ψ'_i anymore. At least, if we rewrite equation (3.12) to:

$$\kappa_c = -2r_1r_2 \frac{\dot{r}_1 + \dot{r}_2}{r_3\dot{r}_3(r_1 + r_2)}, \quad (\text{C.30})$$

leaving ‘only’ r_2, \dot{r}_2 and $\dot{\kappa} - \dot{\kappa}_r$ to be determined by interpolating in the a priori determined profile, saving computation time.

Some numerical inaccuracies arose in this model. Since the original core ‘suddenly’ starts moving at $t = 0$, it experiences a large acceleration. This is present in the form of numerical instabilities in κ_c and its time derivative during the first 9 μs as shown in figure C.9.

The ‘workaround’ is to linearly interpolate \mathbf{E} and \mathbf{B} between the original 2D equilibrium and the values of the sawtooth at a given ‘stable time’. These need to be determined by the operator of the code before the simulation. A suitable ‘stable time’ is chosen, here indicated with the blue cross.

Another problem arose at the end of the reconnection, as $\dot{r}_3 \rightarrow 0$. Small deviations due to the discrete interpolation (to determine r_2 and \dot{r}_2 and thus \dot{r}_3) cause κ_c determined by equation (C.30) to deviate from the a priori determined profile as shown in figure C.10.

This has been solved by switching from the way κ_c is determined. Instead of equation (C.30), it will determine κ_c by interpolation in the a priori determined profile at a certain time (indicated

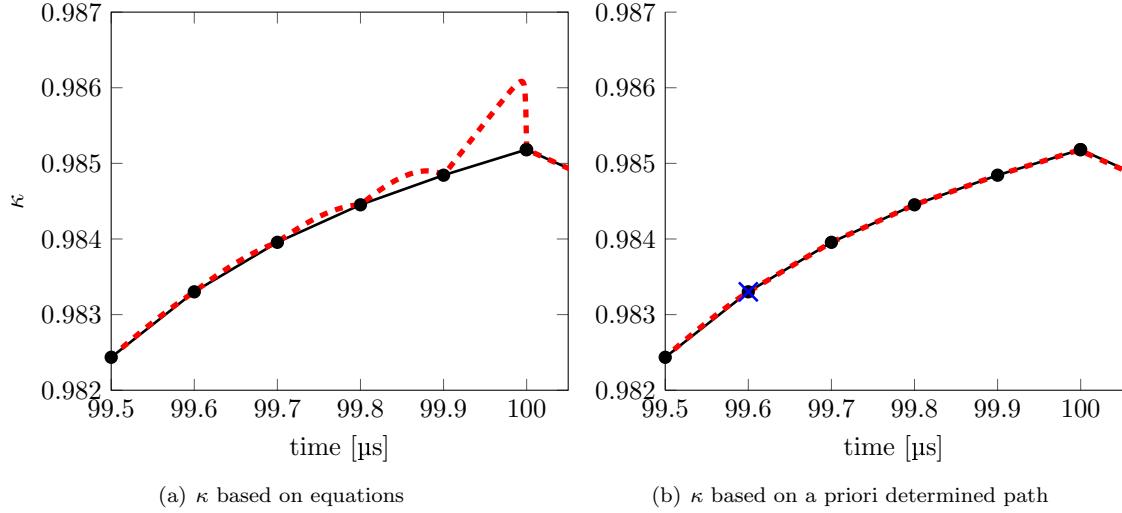


Figure C.10: Numerical instability at end of reconnection result in high inaccuracies in determining κ (which is κ_c until 100 μs). The black dots show where κ has been determined a priori. The red line is κ used by the model. The blue cross indicated where the model switches from using equation (C.30) to a linear interpolation in the a priori determined profile.

with a blue cross). An extra point is added in the a priori determined profile at $t_{reconnection} - \Delta t/2$ to ensure an accurate interpolation.

C.5 Verification and convergence study

C.5.1 EBdyna.go

Conservation of energy and canonical toroidal angular momentum p_φ can verify simulations of `EBdyna.go` in an axisymmetric equilibrium. When no electric fields apply, i.e. $\mathbf{E} = \mathbf{0}$, the BORIS scheme ensures conservation of energy. In these cases the maximum relative change in E_{kin} is in the order of 10^{-13} .

Conservation of p_φ (in 2D fields) is less straightforward, since the ‘quality’ of \mathbf{B} will play a role. In theory, p_φ is perfectly conserved when $\nabla \cdot \mathbf{B} = 0$. However, as discussed in appendix C.3.2, of interest is the divergence the particle ‘experiences’ (figure C.4).

To determine p_φ at time step n one needs the velocity at the integer time \mathbf{v}^n . Since \mathbf{x}^n and $\mathbf{v}^{n-1/2}$ are stored, an estimation of \mathbf{v}^n needs to be made. For this we have chosen to use a half BORIS iteration with a time step of $\frac{1}{2}\Delta t$. Note that this is slightly different from the average of $\mathbf{v}^{n-1/2}$ and $\mathbf{v}^{n+1/2}$. The cylindrical correction from equations (C.21) to (C.23) is made with position $\mathbf{x}^{n-1/4}$. This half-step ‘problem’ causes p_φ to oscillate, but does not result in a drift⁹.

A distribution of 128 particles positions in a cross positioned is used to test the convergence (cross on $R = R_0$ and $Z = 0$ lines). These particles were given a high energy of 90 keV to prove robustness. Let’s first discuss some 1 ms simulations in the 2D equilibrium.

When $\Delta p_\varphi = p_\varphi(t) - p_\varphi(t=0)$ is plotted one observes oscillating signals with different amplitudes (due to the half-step having a variable magnitude). These oscillations do not say anything about the convergence, since they might have to do with the estimation of \mathbf{v}^n as previously argued. However, the drift of p_φ is an interesting quantity. The root mean square of the drift is

⁹ The loop of `EBdyna.go` starts with a time step, after which the position \mathbf{x}^{n+1} and $\mathbf{v}^{n+1/2}$ could be stored. One determined p_φ^{n+1} , in accordance to the ‘analytic’ expression, where $p_\varphi^n + \Delta p_\varphi^n = p_\varphi^{n+1}$. Merely to avoid confusion, here we use n instead of $n + 1$. Most importantly, the time and n counter are used consistently.

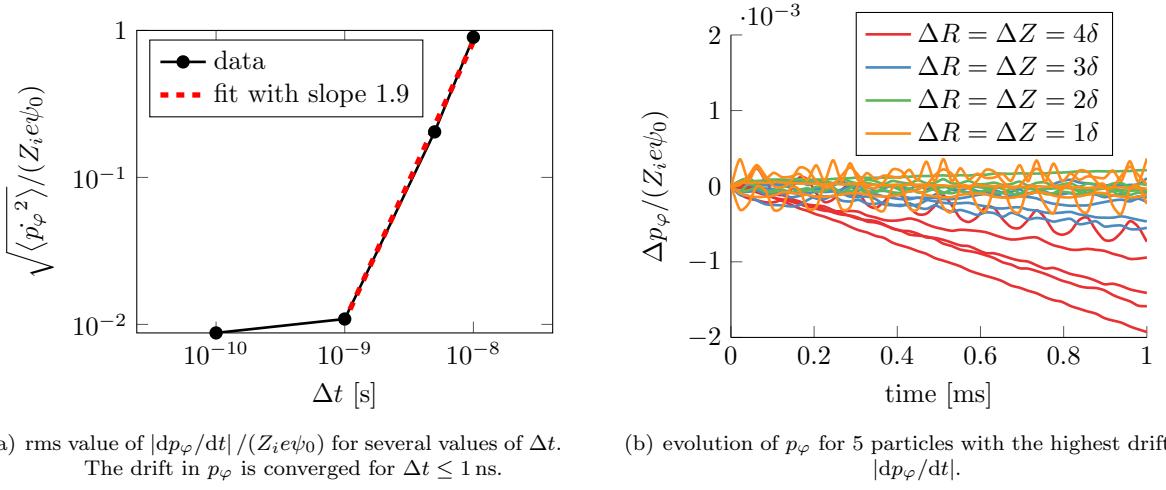


Figure C.11: Convergence test based on p_φ with 128 particles of $E_{kin} = 90$ keV.

proportional to $\Delta t^{1.9}$ up to $\Delta t = 1$ ns as shown in figure C.11(a)¹⁰. Here we have used 4 values for Δt , namely 10, 5, 1 or 0.1 ns. Smaller time steps do not result in lower drifts, so convergence is considered to be achieved with $\Delta t = 1$ ns.

When looking at the grid size, it is convenient to define the internode distance in the finest grid as $\delta \approx 0.963$ mm (in R or Z direction). The drift of p_φ is shown for four cases in figure C.11(b).

Although the grid is quite fine, some drift is still visible. However, the magnitude is much lower than the excursions due to the sawtooth or 3D fields ~ 0.1 in $t_{sim} = 2$ ms. Therefore, it complies with our demands and (maybe more importantly) the grid does show convergence for presented statistically averaged results.

Let's consider the 3D grid and its convergence. The 3D field with the highest mode number is the Toroidal Field Ripple. the evolution of p_φ for one particle has already been shown in section 3.3 (figure 3.3). Here the excursion of p_φ was $\sim 0.1 Z_i e \psi_0$ and the two expressions $p_{\varphi,kin}$ and $p_{\varphi,an}$ were very similar.

We compare $p_{\varphi,kin} - p_{\varphi,an}$, similarly to the previous quantity Δp_φ in equilibrium. The finest toroidal grid used had 512 unique toroidal points, thus $\delta_\varphi = \frac{2\pi}{512}$ radians. Other directions in this grid R, Z are equal to the 2D field, i.e. $\Delta R = \Delta Z = \delta$.

Errors in $\delta p_\varphi \equiv p_{\varphi,kin} - p_{\varphi,an}$ grow larger in time. Despite this, the deviation δp_φ can be kept relatively small. Note that Δp_φ (the change over time since the start of the simulation) is in the order of $0.1 Z_i e \psi_0$. Hence, the deviation $\sim 10^{-3} Z_i e \psi_0$ is deemed acceptable, which is shown in figure C.12(a).

The root mean square of the deviation δp_φ is shown in figure C.12(b). Here we note that with the most fine δ_φ grid, the time derivative of δp_φ is of the same order as the time derivative of Δp_φ in the 2D equilibrium. We therefore conclude that the simulation needs 512 toroidal points over the tokamak (32 per coil) in order to simulate a TFR with comparable precision. Finer grid have not been tested since one would not expect to obtain a lower value for (the rms of) δp_φ than in the 2D equilibrium (figure C.11(a)).

C.5.2 Sawtooth implementation

For the sawtooth implementation we can resort to three methods to verify the implementation:

1. Check if particles indeed tend to ‘follow the field lines’ with $\mathbf{E} = \mathbf{0}$. Note that field lines now lie on contours of constant helical flux ψ_* instead of ψ , since the field is perturbed. Thus particles should only move along a contour of constant ψ_* .

¹⁰The slope is nearly ~ 2 , the scaling one expects with by the BORIS scheme.

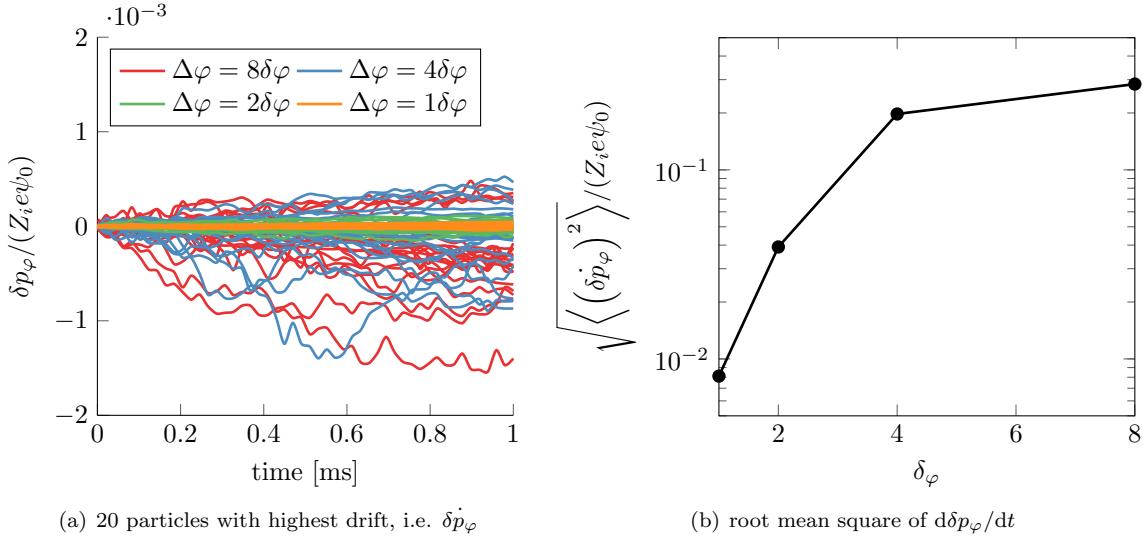


Figure C.12: Convergence test based on $\delta p_\varphi \equiv p_{\varphi,kin} - p_{\varphi,an}$ with 128 particles of $E_{kin} = 90$ keV in TFR. For imaging only the 20 particles with largest ‘drift’ in δp_φ are shown. Here the time steps were $\Delta t = 0.1$ ns to avoid large disturbances due to the determination of p_φ .

2. Check if particles with low velocity tend to stay on contours of constant Φ when $\partial\mathbf{E}/\partial t = \partial\mathbf{B}/\partial t = \mathbf{0}$ (Lorentz force of the magnetic field is low due to the low velocity). Particle drift should be directed according to the $\mathbf{E} \times \mathbf{B}$ drift.
3. Check if particles near the magnetic axis follow the original core. The r coordinate ($= \sqrt{\tilde{x}^2 + \tilde{y}^2}$) should be similar to $r_2 - r_1$.

For all test cases we have chosen for low energy particles of 0.04 eV.

In the first two cases, 144 particles have been positioned on a rectangular grid in (R, Z) -space. To avoid radial excursions, the velocity is directed purely parallel in the first case ($|\mathbf{v}| = v_{||}$) and purely perpendicular in the second ($|\mathbf{v}| = v_{\perp}$). Orbits followed the expected behavior as shown in figure C.13.

In the last test case, 128 particles have been distributed randomly near the magnetic axis. The random distribution has been uniform between 0 and 1 mm. Position in (\tilde{x}, \tilde{y}) -space is indicated by r in figure C.14.

In addition we can compare the present work to previous modelling of the sawtooth crash, i.e. Jaulmes et al. [41]. Here the redistribution of thermal helium is computed. Some parameters might not be the same (e.g. equilibrium, temperature). Here we used a temperature of 2.8 keV for the helium and a similar sawtooth with a 100 μ s reconnection phase and a 200 μ s recovery. Both models show similar characteristics, having outward redistribution up to $r \approx 0.6r_{mix}$, inward redistribution from there up to r_{mix} and no significant redistribution from thereon. The velocity component is responsible for the ‘thickness’ of the curve. Other figures in this thesis avoid this by using p_φ as radial coordinate.

C.5.3 BS3D

Two cases with an analytic solution have been used to validate the BS3D code and estimate the error dependency on the filament size. Firstly an infinite straight wire has been used and secondly a square coil. Take note that the square coil has an analytic solution for the field perpendicular to the enclosed surface and on the axis piercing through it. The results of these cases are shown in figure C.16.

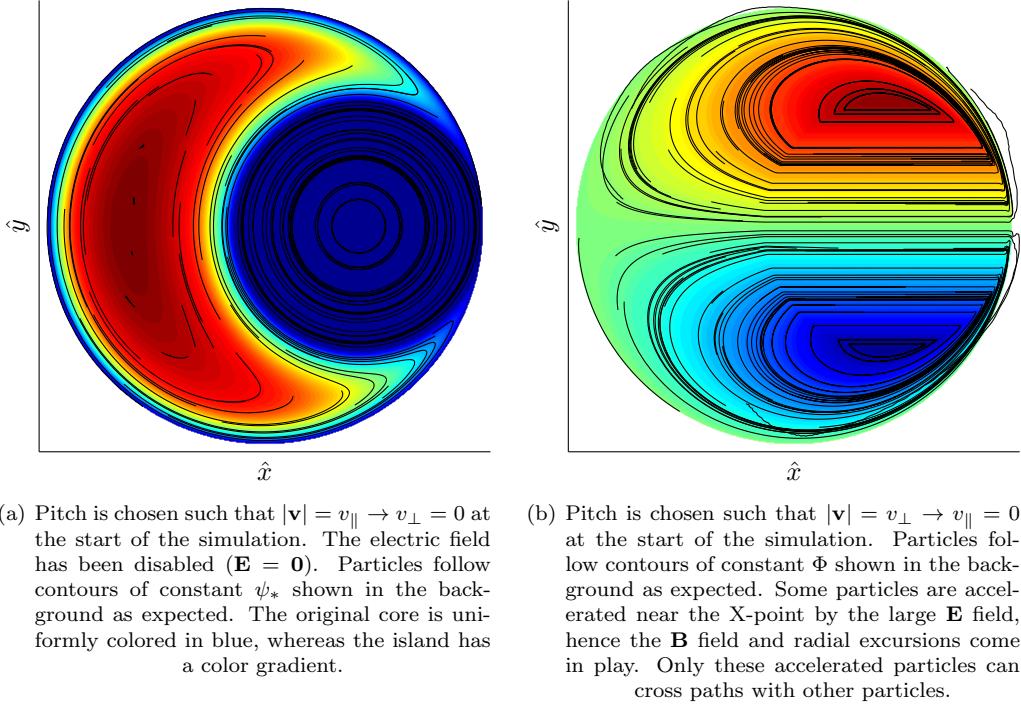


Figure C.13: Particle orbits (black) in \tilde{x}, \tilde{y} space with a ‘frozen’ field at $\kappa_c \approx 0.27$ for two cases to verify the sawtooth simulations. In both cases particles with little energy (0.04 eV) have been used to avoid drifts. These cases agree well with the expected behavior of the particles.

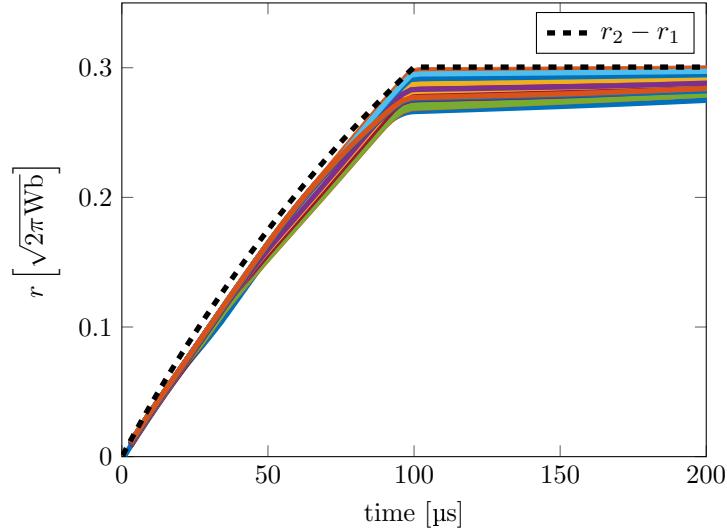


Figure C.14: Position of 128 low-energy particles initialized near the magnetic axis (distance 1 mm, with random θ). All follow the movement of the original core ($r_2 - r_1$) as expected.

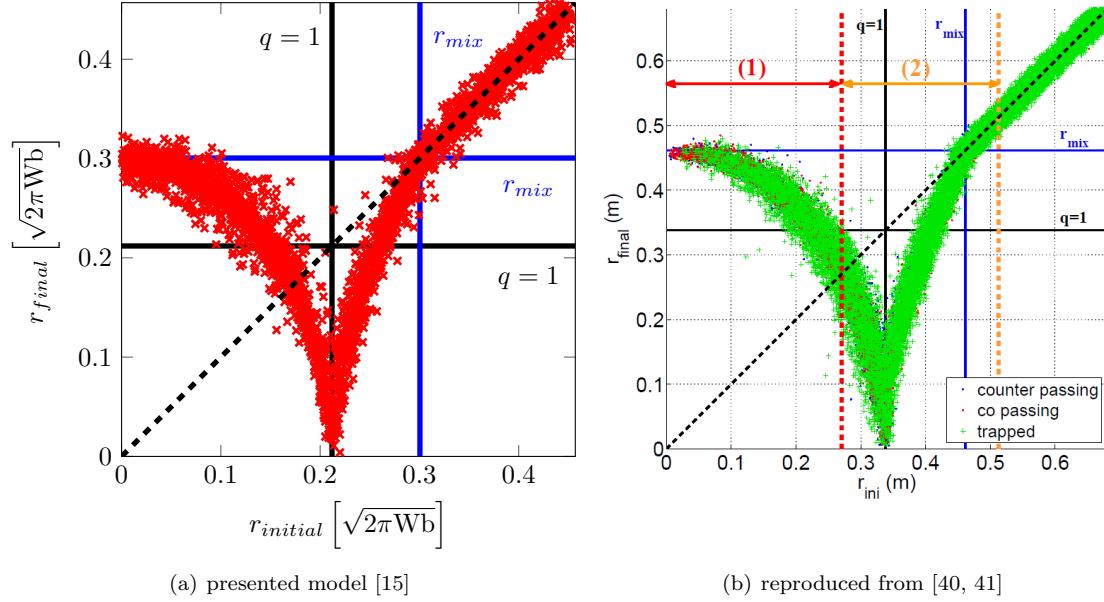


Figure C.15: Redistribution of thermal helium by the presented sawtooth model in comparison with the published model by Jaulmes et al. [41]. Note that a different equilibrium, # particles and the pitch (or μ) might cause differences independent of the model. Nevertheless, the models show similar behavior. The width of the redistribution is caused by the particle trajectory (radial excursions due to velocity), hence even outside r_{mix} the r coordinate does not remain constant.

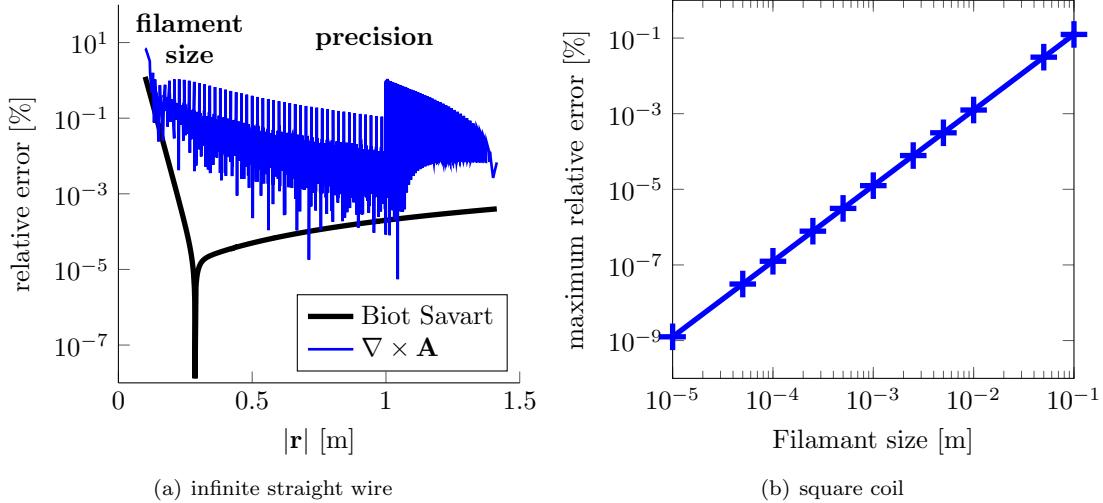
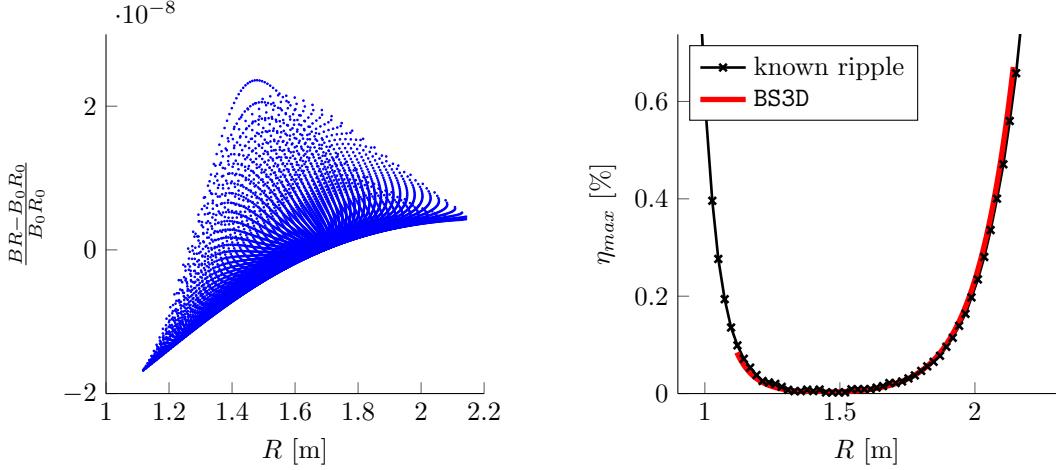


Figure C.16: Validation of BS3D code with an infinite long straight wire and a square coil. The error in the $\nabla \times \mathbf{A}$ depends on the grid size. Nevertheless the figure shows that the vector field is determined properly. The error using the sum in Biot Savart (equation (3.3)) consist of 2 ‘regions’. One numerical error is due to the finite filament size, which is more pronounced close to the wire. The other error is due to numerical precision (adding more values, each with a finite precise values).



(a) A relatively low error in determining the TF vacuum field is found when the BS3D output is averaged over the φ coordinate. For the straight field line-coordinates this figure shows that B scaled very well with R^{-1} (B_0 is the magnetic field on axis). The toroidal grid performed equally well, having a maximum relative error of $\sim 1 \times 10^{-7}$.

(b) The toroidal maximum of the ripple as function of R . Note the ‘known data’ was send in a graphic format, from which data is extracted by manually selecting the marks (which might have been somewhat inaccurate).

Figure C.17: Two validations of BS3D for the Toroidal Field Ripple, showing an accuracy that complies with the requirements without any problem.

From these figures we can see a steady, but small ($\sim 0.1\%$) error when the magnetic field is determined by the rotation of the vector field \mathbf{A} . This turns out to be grid dependent, i.e. a finer grid results in a smaller error. Errors in this case are made when calculating the rotation. The low error increases confidence in the vector potential calculation.

For the Biot-Savart law we can see a small error ($\sim 1 \times 10^{-3}\%$) with two error ‘regions’. Evaluation points nearby the coil/wire suffer from the finite size of filaments, whilst errors due to machine precision and superposition dominate at further distances from the wire (note B becomes smaller and hence the relative error increases).

A square coil with dimensions 0.2×0.2 m (same order of magnitude as RMP coil) has been used to examine the required filament size of the coils. For an accurate result with reasonable computation time the minimum filament size for BS3D is set to 1×10^{-3} m. The relative error of $10^{-5}\%$ meets requirements without problems. Indeed it is known that a Biot-Savart solver is more precise than other methods (e.g. finite element methods) [3].

The vacuum field determined by BS3D has been compared to the theoretical $B \propto R^{-1}$ scaling and the documented TFR of AUG, presented in figure C.17. These have been made using the ‘flux’ based coordinate system¹¹. The toroidal coordinate system performed comparably good.

Additionally one can examine the divergence of the magnetic field as an error indicator [81]. In theory the magnetic field should be divergence free, i.e. $\nabla \cdot \mathbf{B} = 0$ [33]. However, numerical inaccuracies can give rise to some deviations. As seen when comparing interpolation methods based on p_φ evolution, divergence of the magnetic field does not ensure a perfect field (appendix C.3.2). Interpolation of this field is also relevant for the field. So, do not think a more divergence free field (in these figures) result in a better confinement, but only use them as a ‘check’ of the BS3D error.

Here we have used a finite difference method to determine the divergence of both the 2D and 3D field. A finite difference method uses the surrounding grid points to approximate derivatives of

¹¹A small error has later been found, since the Z coordinate is not exactly 0 at the magnetic axis. This ‘bug’ has been fixed with no noticeable difference in the output. Simulations in this work have used 3D fields with this small bug.

Table C.1: Finite difference coefficients for approximating the first derivative based on surrounding points [25]. This has been used both to estimate the divergence of the magnetic field $\nabla \cdot \mathbf{B}$ and the φ -derivatives of components of the vector potential.

accuracy	-4	-3	-2	-1	0	1	2	3	4
2	0	0	0	-1/2	0	1/2	0	0	0
4	0	0	1/12	-2/3	0	2/3	-1/12	0	0
6	0	-1/60	3/20	-3/4	0	3/4	-3/20	1/60	0
8	1/280	-4/105	1/5	-4/5	0	4/5	-1/5	4/105	-1/280

the field. The coefficient used to multiply the surrounding points are given by table C.1 for several ‘accuracies’ [25]. A higher accuracy uses more surrounding nodes. After the surrounding grid points have been multiplied with the given coefficients the divergence is determined by dividing by the internode distances (δ and δ_φ).

From these coefficients we compute $\nabla \cdot \mathbf{B}$ as shown in figure C.18. Here we used an accuracy of 4 and 6. Since the poloidal field \mathbf{B}_{pol} is based on the finite difference of ψ (with accuracy=4) and the toroidal field B_φ , no divergence is observed with the same accuracy. Conversely, when the accuracy is increased to 6, the divergence of the 3D field, especially the ripple, disappears. Thus the determination of divergence results in an indication of the errors and sensitive areas, but not in any concrete criteria. However we conclude that the divergence appears to be very low in the 2D grid, whilst it might be experienced larger by particles. Since the 2D grid is also used in previous studies, shows good p_φ conservation and is sufficient, we assume the comparable divergence of the 3D fields is ‘low enough’.

C.6 Manual

The directory structure of the `EBdyna.go` repository devises several functionalities for the `EBdyna.go` code. In this context, `BS3D` is seen as a functionality of `EBdyna.go`, and has thus been housed in a separate directory. Equilibrium data from `FINESSE` is stored in the `data_tokamak` directory, which is used by both `EBdyna.go` and `BS3D`. Due to the (historically) intertwined directory structure, some names and places of scripts might be counter intuitively.

Since the equilibrium reconstruction has already been documented by van de Plassche [93], we do not discuss `FINESSE` and `PF2q`.

C.6.1 Parallelization on SurfSara Lisa

Both `BS3D` and `EBdyna.go` are parallelized codes. For efficiency one should ‘compile’ the codes, producing ‘binaries’ from the `.m`-scripts. For example, the main file `G_eq.m` for `EBdyna.go`, will be compiled producing a `G_eq` binary. Shared memory is loaded with (3D) fields before starting `EBdyna.go`. The binary `load_RAM` is compiled for this purpose.

MATLAB® does have internal options for parallelization, which requires an additional toolbox (extra costs). On top of that, the toolbox isn’t available in older MATLAB®-versions and syntax isn’t consistent. Therefore it is chosen to manually parallelize the code.

A module called `stopos` is available on `SurfSara Lisa`, which keeps track of the parallelization. This module makes use of so-called ‘pools’. Different pools can be used for different parameters of `BS3D` or `EBdyna.go`. For instance, one can simultaneously let `BS3D` compute the Toroidal Field and the RMP-fields, using `pool_TFR` and `pool_RMP`. We will come back to how to operate the pools later, but first let’s connect to `SurfSara Lisa`.

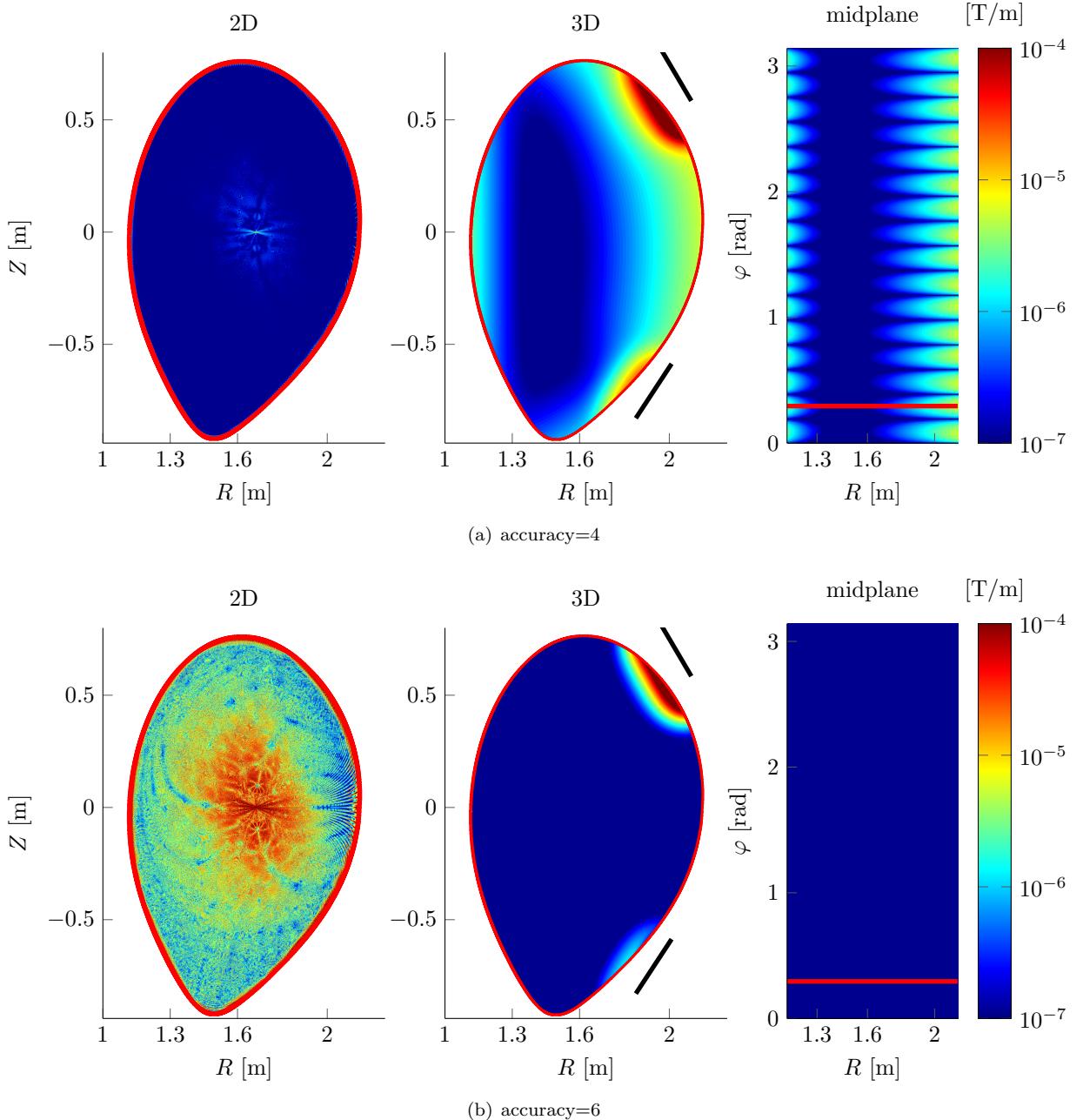


Figure C.18: Divergence of magnetic field with and without 3D fields. The addition of the 3D field ($n = 2$ with even parity and TFR) increases $\nabla \cdot \mathbf{B}$. The location of the saddle coils is indicated with black in the ‘3D’ figure, its toroidal location is indicated with a red line in the ‘midplane’ figure.

C.6.2 First use of SurfSara Lisa

First one needs to ‘make contact’ with **SurfSara Lisa**. Next the custom functions need to compiled in order for **EBdyna_goto** work (**sharedmatrix**, **ba_interp2** and **ba_interp3**).

1. Use the **scp**-protocol to upload the repository files to **SurfSara Lisa**.
2. Upload the 2D fields from **FINESSE** to the **data_tokamak** directory.
3. Use an SSH client, connect to **lisa.surfsara.nl** (port 22)
4. Make sure all **.sh**-files are executable (type **chmod +rwx *.sh** in the SSH-client).
5. Go to the directory **/EBdyna_go**.
6. Open MATLAB® interactively (on a login node) by typing **matlab -nodisplay**¹²

Make sure MATLAB® is in the directory **EBdyna_go** (the main directory). Check this with **pwd** and if incorrect, correct it with **cd**. Next type following in the MATLAB® command window:

```
addpath(genpath(pwd)) % Add all functions in subdirectories to the 'path'. Now ...
Matlab can use them.

make_sharedmatrix_function
make_ba_interp3_function
make_ba_interp2_function
```

Check that three **.mexa64** files have appeared in the **01_general_functions** directory. Compilation on other systems is done similarly. Windows makes use of the function called **SharedMemory**, which is almost identical. The command **make_sharedmatrix_function** will compile **SharedMemory** on Windows systems. Operation of **EBdyna_go** is identical, i.e. it will automatically use **SharedMemory**, thus one might not even notice the difference.

C.6.3 Producing 3D fields with BS3D

Calculation of 3D fields is almost identical for the TFR and RMP fields. It is recommended to determine the vacuum field for a single coil, from which BS3D can determine the total field with periodically rotating this field¹³.

We will talk the reader through the Toroidal Field calculation first. Make sure to set the parameters in **BS_AUG_TFR.main.m** For the first use we recommend the following settings:

```
TF_map.I=0.3e6; par.I=TF_map.I; % Coil current [At] (arbitrary),
par.determine_vector_potential=false; % Determines derivatives in phi of ...
% vector potential. Needs axillary points for nice finite difference calculation.
par.example=0; % No plotting / example
par.coord_sys='flux'; % Switch between coordinate systems
par.nr_coils=1; % Switch between 1 or 16 coils
```

Next set the grid parameters in **BS_AUG_get_grid** to:

```
nr_psi =2; % Number of radial points
nr_theta=513; % Number of poloidal points
nr_phi =1025; % Number of toroidal points
```

¹²MATLAB® access needs to be requested by the **SurfSara Lisa** helpdesk. **SurfSara Lisa** has a student license, so the software is for educational purpose only.

¹³Periodicity is applied when only the vacuum field of a single coil has been determined (per type). Modelling of the ‘A’ coils should be revised, since the size and toroidal position (in respect to other coils) varies.

These settings will produce a magnetic field on the magnetic axis and the LCFS. Here an arbitrary current I_{BS3D} can be chosen. One can choose to determine the field without **SurfSara Lisa**, since the required time is relatively low. However it is likely that one needs to parallelize by hand to avoid memory problems. Let's therefore assume we do this on **SurfSara Lisa**.

First determine the number of parallel ‘boxes’. Points on the grid will be evenly divided among the boxes. Do this by changing the value of m in `reset_pool_TFR.sh` (16 would do for now). Next type in the SSH-client connected to **SurfSara Lisa** the following:

```
./compile_BS_AUG_TFR.sh
qsub ./run_BS_AUG_TFR.sh
```

Here the last command can be submitted multiple times to speed up the calculation by using more nodes / processors. However, if only 16 ‘boxes’ have been chosen, one node could handle them all (no memory problem are expected on **SurfSara Lisa**).

After the calculation, the output file of each parallel process is found in `output_TF`. Either choose to copy these locally or to submit a new job to combine these files in one output file. Change the parameter `save_folder` in `BS_TFR_combine_output.m` accordingly Submission of the job on **SurfSara Lisa** can be done with:

```
qsub ./combine_TFR.sh
```

This produces the Toroidal Field(and the Toroidal Field Ripple) on the two radial positions for all θ, φ positions from all TF coils. Feel free to throw the Toroidal Field Ripple away, it can be easily reproduced from the Toroidal Field-file. Similar to the previous steps, we determine the field on the toroidal R, Z, φ grid with the following settings. Change the settings in `BS_AUG_TFR_main.m` to:

```
TF_map.I=0.3e6; par.I=TF_map.I; % Coil current [At] (arbitrary),
par.determine_vector_potential=true; % Determines derivatives in phi of ...
% vector potential. Needs axillary points for nice finite difference calculation.
par.example=0; % No plotting / example
par.coord_sys='toroidal'; % Switch between coordinal systems
par.nr_coils=1; % Switch between 1 or 16 coils
```

Make sure to use the *same current* I_{BS3D} as before! The scaling factor for the current of the straight field line-coordinates will be used later on. Since a large number of points will be present (if the settings are kept the same), make sure to increase the number of ‘boxes’ to ~ 300 (tip: use 304, a multiple of 16). First delete any remains in the `output_TF` directory, but not the file with `sca_fac_TF*`. Then then compile and submit the job. Ideally submit the number of jobs that cover the number of boxes (19 in the example). Also combine the field to obtain the TF and TFR. Note that the date when the scaling factor is obtained is used in the script `TF_to_TFR.m`. Also convince yourself that the file with scaling data is located in the directory of the correct script. If done correctly, an TF and TFR file will be saved in the output directory.

For the RMP, similarly use `combine_RMP.m` and `BS_AUG_RMP_main.m`, the latter with the following settings:

```
%% Parameters
RMP_map.I=4.8e3; par.I=RMP_map.I; % Coil current [kAt] (default),
par.determine_vector_potential=true; % Determines derivatives in phi ...
% of vector potential. Needs axillary points for nice finite difference ...
% calculation.
par.example=0; % No plotting / example
par.coord_sys='toroidal'; % Switch between coordinate ...
% systems (flux: theta,psi,phi OR toroidal: R,Z,phi)

%% Load Coils
```

C.6. MANUAL

```
[ RMP_coils] = BS_AUG_RMP.load_coils(1,1,0);
```

Here feel free to change the coil current as intended for the simulation. The latter line will only load the geometry of the first and last coil. The script `combine_RMP.m` will rotate the coil periodically for each chosen configuration and produce the output for the Bu and Bl coils separately. Afterwards, these two are superimposed for the total RMP field. Note that configurations are chosen in `combine_RMP.m` and defined in `BS_AUG_RMP_add_signs.m`. Also take ~ 300 boxes and call

```
./compile_BS_AUG_RMP.sh
qsub ./run_BS_AUG_RMP.sh
```

and call the latter multiple times to your liking.

C.6.4 Particle tracking with EBdyna.go

`EBdyna.go` requires an input file. Either upload a distribution file, or create one. To create one type in the MATLAB® command window:

```
cd('../particles_equilibrium') % Any directory would do for now, but this ...
                                % directory will required later on
run('../build_initial_distributions/build_homogeneous_dist.m') % This creates a ...
                                % homogeneous distribution file in EBdyna.go/input
exist('./input/G_eq_19xx_particles.mat','file') % Check whether the creation of ...
                                % the file is successful. If so, the number 2 is returned

exit % Go back to the SSH client
```

The last two lines aren't really necessary, but that's assuming everything went right. Now we will start a 2D simulation first by the ID of `19xx`. Make sure the parameters are set correctly in `G_eq.m` and `sim_parameters.m`. Presumably you'll set:

```
%% Settings for G_eq (others will not really matter)
par.ID='19xx'; % ID-name
par.interpscheme=3; % ba_interp linear
par.APPLY_RMP=false; % No RMP
par.APPLY_TFR=false; % No TFR
par.APPLY_SAWTOOTH=false; % No ST

par.GET_PREC_INFO=true; % Since this is true and PREC-files aren't ...
                        % present yet, the above settings don't matter
par.SAVE_DATA_FILE=true; % You want to save the result of this ...
                        % computationally intensive operation
par.DISTNAME='./input/G_eq_19xx_particles.mat'; % Name of file in ...
                                % input-directory

par.step.R=1; % Finest possibly resolution in X
par.step.Z=1; % Finest possibly resolution in Z
par.step.phi=1; % Finest possibly resolution in phi

%% Settings for sim.parameters
par.NR_FUND_IN_LOOP=100; % Check every 100 step the loss-position, number ...
                            % of losses etc.
par.dt=1e-9; % Fundamental time step
t_sim=1e-3; % Total simulated time
par.NB_TIME_STAMPS=500; % Sampling (frequent for correct orbital data)
PC_SAVE=Inf; % Save at the end, not intermediately
```

Now set the preferred number of ‘boxes’ (e.g. ~ 6000 per box) in the file `reset_pool_i.sh`. Next go to the SSH client and use:

```

ID=19xx # Repeat the ID here, this is not linked with the par.ID parameter of ...
EBdyna_go self.

# Now we compile and submit the job. Take care to add the ID as input for these ...
# .sh scripts
./compile_G_eq.sh $ID # Compile, which will reset the pool, create the output ...
# folder etc.
qsub -N $ID -v ID=$ID ./run_G_eq_i.sh # Submit to a single node. Repeat this ...
# line as often as you would like to have multiple nodes working in parallel.

```

One can choose to use the last line more often to speedup the calculation (use more nodes). After execution, we will want to combine the output of all files and save them in the distribution file. Use an interactive session with MATLAB® and type:

```

combine_output_eq('19xx','prec') % Combination script, with the ID and the ...
# `type' of simulation

```

One can also submit this as a job with the file `run_combine_output.sh`. First make sure that this script copies the required files to the scratch file system and copied the proper files back.

Alternatively, one can keep the separate PREC-files. These should then be copied along to the scratch when submitting a job. For this it is advised to remove the `output-struct` from the PREC-files. Another disadvantage is that one is obliged to use the number of ‘boxes’ consistently, otherwise `EBdyna.go` can provide an error message or worse, produce incorrect output.

Now it’s easy to start a simulation. Make sure the required 3D fields are present in the `data_tokamak` directory. Alter the settings in `G_eq.m` (especially the ID) and do the same as before. However, in the SSH window, type a different ID¹⁴. Submit a job with an ID that does not end with ‘xx’ this time.

Each core will run `EBdyna.go` individually, thus several instances of `EBdyna.go` coexist. Therefore it is most efficient to compile MATLAB® with the option `singleCompThread`. Secondly it is worth mentioning that one could submit a new job midway the simulation if, for example, the simulation isn’t as fast as the user had hoped. However, one is advised to assure that `G_eq<ID>` (the compiled version) hasn’t changed since.

Once the simulation is done, two files will be saved for each ‘box’ in the output directory. Note that the `FULL`-file has only the output stored according to the saving at the end of `GT_eq.m`. To combine the output of individual ‘boxes’ one can run `combine_output_eq.m` now with the new ID and the input argument ‘full’. The positions stored in the `RAW`-files are required to obtain the energy profile as shown in figure 4.12 after the `FULL`-file is made (use the ‘raw’ input argument).

When a sawtooth is applied, one should examine the parameters beforehand locally. To do this, use a 2D field and any test distribution. Make sure to set `par.st.show_parameters` in `G_eq.m`. When now running `G_eq.m` MATLAB® will plot the profile a priori determined along with the parameters used by `EBdyna.go`. See appendix C.4 for the possible numerical instabilities and how to work around these. Profiles of ψ_* and Φ can be made with `complete_sawtooth_struct.m` in the `reconnection_mapping` directory. Once all sawtooth parameters are set to satisfaction one can add 3D fields and run the code on `SurfSara Lisa`.

To make Poincaré plots, uncomment the line in `G_eq.m` saying:

```

% par.mode=5; par.Poincare_plot=true; % Approximately line 161, just below the ...
# list of modes

```

In `sim_parameters.m` one should define the time steps Δt and the `poincare_type`. Three possible figures have been implemented. One can choose to plot in straight field line-coordinates (with `sf1`), or toroidal coordinates (with `tor`) or in the sawtooth \tilde{x}, \tilde{y} -space (with `xy`). A nested function in

¹⁴The ID in the SSH window is used for the storage directory, the compiled files, the pools and the name of the submitted jobs. It has no direct relation to the ID `EBdyna.go` uses itself. Thus, the file names (based on `par.ID_name`) do not depend on the ID in the SSH environment.

`GT_eq.m` can be used to plot respectively radial lines at resonant positions (sfl), the LCFS (tor) or a background with either ψ_* or Φ (xy). In the latter case the corresponding background image should be made with `complete_sawtooth_struct.m` and the `B_interpolation` function should be altered to produce only the required about (e.g. for a fixed time, with or without `E` etc.).

C.6.5 Evaluating output

The script `plot_output.m` can be found in the `particles_equilibrium` directory. It is well commented, but also subject to change. Therefore we will only describe the steps lightly.

First load the file to analyse. If this is a `FULL`-file one should be able to use most plot routines without problems. However, plot routine #1 requires the positions of particles, hence a `RAW`-file should be loaded. In this case (or in case 2), the `plot_output` function will try to load the magnetic fields. Then the `evaluate_output` will be performed to obtain some parameters. Note that this could be problematic if (large) 3D fields have been applied.

If one would like to load 3D fields, please execute the following commands:

```
global par dim maps const qom % Define global parameters
load('./output/G_eq_19xx.full','par') % Load in the parameters of the selected ...
simulation
reset_data_analysis_environment % Execute this script (which is also backwards ...
compatible with the old EBdyna-go
```

Appendix D

List of functions and parameters

For documentation and openness we list here all functions relevant for the `BS3D` calculations and `EBdyna.go` simulations. Also the most important paramaters and their definition is provided.

D.1 Function list

D.1.1 General

Some functions are used by both `EBdyna.go` and `BS3D`. Other functions are quite general or other reasons exist to store these in the `01 general functions` directory. These are listed in table D.1.

Table D.1: Function list of general functions in `EBdyna.go` repository. These are housed in `01 general functions`.

function name	description
<code>remove_fields</code>	Elaborate variant on <code>rmfield</code> function.
<code>combine_structs</code>	Combine the fields of two structs.
<code>get_expr_job</code>	Devises an array based on the ‘box’ number and the total number of ‘boxes’.
<code>Fourier_3D_field</code>	Determine the Fourier components of a field (m and n) and show the nomenclature. Note that this requires the 3D grid in straight field line-coordinates.
<code>divergence_poloidal</code>	Determine the divergence of the 2D and/or 3D field based on the finite difference method (see appendix C.5.3).
<code>histndim</code>	Function to bin particles in 2D on older MATLAB® versions. Newer MATLAB® versions have the <code>histogram2</code> function, which is preferred. (Written by Mike Brookes.)
<code>imagescnan</code>	Shows figure with the NaN values coloured white.
<code>initialize_folder_names_struct</code>	Loads in the names of directories, e.g. the <code>data_tkamak</code> directory.
<code>make_ba_interp2.function</code>	Function to compile the <code>ba_interp2</code> function.
<code>make_ba_interp3.function</code>	Function to compile the <code>ba_interp3</code> function.
<code>make_shared_matrix.function</code>	Function to compile the <code>shared_matrix</code> (UNIX) or the <code>SharedMemory</code> (WINDOWS) function.
<code>reinitialize_data</code>	Script which can reload the parameters when evaluating the data based on the parameters stored with the output file (e.g. <code>maps</code>).
<code>interp_index_list_2D</code>	Finds indexes in 2D grid necessary for interpolation method # 1.

D.1. FUNCTION LIST

<code>interp_index_list_3D</code>	Finds indexes in 3D grid necessary for interpolation method # 1.
<code>interp2_XZ</code>	Self-written interpolation in 2D grid used by interpolation method # 1.
<code>lininterp3</code>	Self-written interpolation in 3D grid used by interpolation method # 1.

D.1.2 BS3D

Functions specifically used by `BS3D` are listed in table table D.2.

Table D.2: Function list of `BS3D`.

function name	description
<code>BS_AUG_RMP_main</code>	Main function to determine RMP field
<code>BS_AUG_RMP_load_coils</code>	Loads RMP coil geometry
<code>BS_AUG_get_grid</code>	Loads in the flux or toroidal grid (both for the RMP and TFR). Also loads the vertical distance between $z = 0$ and $Z = 0$ in meters.
<code>BS_AUG_RMP_calc_coils_individual</code>	Corrects for the difference in $z = 0$ and $Z = 0$, loops over each coil to determine \mathbf{A} and \mathbf{B} (by <code>BS_calc_B.m</code>) and determines the φ -derivatives of \mathbf{A} . The latter is done with an auxiliary grid with a finite difference method (accuracy=4). More information about this method is found in table C.1.
<code>BS_AUG_TFR_calc_coils_individual</code>	Does the same as <code>BS_AUG_RMP_calc_coils_individual</code> , but for the TFR.
<code>BS_calc_B</code>	Determines \mathbf{B} and/or \mathbf{A} for a single coil and a number of evaluation points. Based on input, it will use linear interpolation within the coil to meet the minimum filament element length. Shows a unit test with a long straight wire when used without input arguments.
<code>BS_RMP_combine_output</code>	Combine the output of the parallel ‘boxes’. For each type of coil, i.e. ‘Bu’, ‘Bl’, or ‘A’, it will apply the chosen modes n and parities $\Delta\Omega$. The three types are stored separately, after which a superposition of the three is performed.
<code>BS_AUG_RMP_add_signs</code>	Add the <code>sign</code> parameter to the RMP coils, depending on the chosen configuration $n, \Delta\Omega$.

D.1.3 EB

Functions specifically used by `EBdyna.go` are listed in table table D.3.

Table D.3: Function list of `EBdyna.go`. These are housed in `particles_equilibrium`. When `.../` is written in front, it indicates the function is usually called within the previous described function.

function name	description
<code>G_eq</code>	Main script of <code>EBdyna.go</code> . Most parameters are set here
<code>sim_parameters</code>	Defines additional simulation parameters, e.g. simulation length t_{sim} .

<code>initialize_maps</code>	Initializes the quantities on 2D and 3D grid. To be specific, the quantities <code>const</code> , <code>dim</code> and <code>maps</code> are initialized.
<code>../split_theta_map</code>	Store an additional 2D grid with θ to avoid interpolation errors near the boundary between 0 and 2π . ¹
<code>initialize_particles</code>	Initializes particles (either performs <code>make_test_dist</code> or loads the distribution file).
<code>../make_test_dist</code>	Initializes test particles (various distributions can be chosen)
<code>GT_eq</code>	Main simulation loop, executed once the initialization by <code>G_eq</code> is completed.
<code>../time_step_integration_GT_eq_struct</code>	Perform a time step of the BORIS scheme.
<code>../../B_interpolation</code>	Interpolates in the grid to obtain B and (if the sawtooth crash is occurring) E .
<code>../find_3D_Afield</code>	Interpolate to acquire quantities in the 3D grid other than components of B (e.g. components of A or the derivatives). Note the A field is only the component of the 3D field, i.e. the contribution of the 2D field is not included (in A_φ).
<code>../evaluate_output</code>	Determine output variables, e.g. p_φ
<code>../extract_precession_information_struct</code>	Determine equilibrium quantities, e.g. identify particles as trapped or passing.
<code>../reduce_time_stamps¹</code>	function to remove some time stamps to avoid too large data sets.
<code>plot_output</code>	Plots output of <code>EBdyna.go</code> with multiple options.

D.2 Parameter list

Here the parameters of the codes made in this work are listed, i.e. for `BS3D` and `EBdyna.go`.

D.2.1 BS3D

Table D.4: List of variables within the global `par` struct of `BS3D`. It holds all parameters.

variable	type	description
<code>PROCESS_NUMBER</code>	integer	'box' number
<code>N_PROCESS</code>	integer	total number of 'boxes'
<code>I</code>	double	current through coil(s)
<code>determine_vector_potential</code>	logical	switch to determine φ -derivatives from A (with auxiliary grid)
<code>example</code>	integer	switch to plot the output (0 is off, 1 to 3 give different representations).
<code>coord_sys</code>	string	switch between <code>flux</code> (sf) or <code>toroidal</code> evaluation grid
<code>nr_coils²</code>	integer	number of coils (1 or 8)

¹Actually housed in `01 general functions` directory.

²Only available for the TFR.

D.2. PARAMETER LIST

<code>size_total</code>	vector	size of 3D map
<code>indexes</code>	vector	the indexes that identify the points evaluated by the current ‘box’
<code>delta_phi2</code>	double	toroidal angle distance of auxiliary grid

Table D.5: List of variables within the `RMP_map/TF_map` struct of BS3D (called `n3D_map` within `BS_AUG_get_grid.m` script). It holds data about the RMP/TF coils.

variable	type	description
I	double	current through coil(s)
R	vector	R -coordinate of evaluation points of the normal grid
Z	vector	Z -coordinate of evaluation points of the normal grid
phi	vector	φ -coordinate of evaluation points of the normal grid
X	vector	x -coordinate of evaluation points of the normal grid (Cartesian)
Y	vector	y -coordinate of evaluation points of the normal grid (Cartesian)
$R2^3$	vector	R -coordinate of evaluation points of the auxiliary grid
$Z2^3$	vector	Z -coordinate of evaluation points of the auxiliary grid
$X2^3$	vector	x -coordinate of evaluation points of the auxiliary grid (Cartesian)
$Y2^3$	vector	y -coordinate of evaluation points of the auxiliary grid (Cartesian)
<code>TF_coil^{2,4}</code>	vector struct	Struct holding the fields determined by BS3D for each TF coil
<code>Bu^{4,5}</code>	vector struct	Struct holding the fields determined by BS3D for each Bu coil
<code>Bl^{4,5}</code>	vector struct	Struct holding the fields determined by BS3D for each Bl coil
<code>A^{4,5}</code>	vector struct	Struct holding the fields determined by BS3D for each A coil

Table D.6: List of variables within the `RMP_coils/TF_coils` vector struct of BS3D. It holds the geometry of the RMP/TF coils.

variable	type	description
X^2	vector	x -coordinate of TF coil (Cartesian)
Y^2	vector	y -coordinate of TF coil (Cartesian)
Z^2	vector	z -coordinate of TF coil (Cartesian)
<code>Bu⁵</code>	vector struct	Information for each the Bu RMP coil
<code>Bl⁵</code>	vector struct	Information for each the Bl RMP coil
<code>A⁵</code>	vector struct	Information for each the A RMP coil
<code>../coil</code>	matrix	R, Z, φ -coordinates of Bu/Bl/A RMP coil
<code>../X</code>	vector	x -coordinates of Bu/Bl/A RMP coil (Cartesian)
<code>../Y</code>	vector	y -coordinates of Bu/Bl/A RMP coil (Cartesian)
<code>../Z</code>	vector	z -coordinates of Bu/Bl/A RMP coil (Cartesian)
<code>../phase</code>	double	Difference in average φ coordinate between the coil and the first coil of the same type (the reference coil)
<code>../ref_coil</code>	double	Reference coil used to determine the <code>phase</code> parameters. Note that the A-coils are not shaped consistently and therefore multiple reference coils are possible.
<code>../sign</code>	double	Multiplication factor (often -1,0 or 1) for the current through the current coil, depending on the chosen configuration. i.e. $n, \Delta\Omega$

³Only available when the auxiliary grid is used based on `par.determine_vector_potential`.

⁴Holds the same fields as the `fields` struct (see table D.7).

⁵Only available for the RMP

Table D.7: List of variables within the `fields` struct of `BS3D`. It holds the same values as e.g. `TF_map.TF_coil`. It holds data about the RMP/TF coils.

variable	type	description
BX	vector	x -component of the B field (Cartesian)
BY	vector	y -component of the B field (Cartesian)
BZ	vector	z/Z -component of the B field
AX	vector	x -component of the A field (Cartesian)
AY	vector	y -component of the A field (Cartesian)
AZ	vector	z/Z -component of the A field
dAR_dphi ³	vector	φ -derivative of A_R
dAR_dphi ³	vector	φ -derivative of A_R
dAR_dphi ³	vector	φ -derivative of A_R
Aphi	vector	φ -component of the A field
Bphi	vector	φ -component of the B field
AR	vector	R -component of the A field
BR	vector	R -component of the B field

D.2.2 EBdyna.go

Global variables

Some quantities in `EBdyna.go` are reoccurring and some are ‘global’, so they’re available within multiple functions. These global quantities are described in table D.8:

Table D.8: List of global variables of `EBdyna.go`.

variable	type	description
const	struct	Physical constants, e.g. elementary charge e , loaded from a file in the <code>data_tokamak</code> directory.
par	struct	Parameters, settings and switches for the simulation.
dim	struct	Dimensions and radial profiles. For instance the size of a grid, physical distance between grid points and some indexes like the index of the magnetic axis in the 2D matrices.
maps	struct	Quantities stored on a 2D and 3D grid (and historically the value of ψ_0). In addition to the magnetic fields, also ψ, θ and A are stored in this struct.
time	double	Time in seconds. Only the sawtooth simulation requires this parameter.
mach	string	Machine type, i.e. either ‘WINDOWS’ or ‘LINUX’ (which should have been ‘UNIX’). Only used when a ‘shared RAM’ is used (see appendix C.6.4)
map_data	struct	Information about the location of the read-only RAM. This is necessary to avoid any ‘Run-time errors’ of MATLAB®.
qom	double	Ratio of charge over mass.

Some of these global variables need further explanation. Here follows the complete list of most of these quantities in tables D.9 to D.11.

D.2. PARAMETER LIST

Table D.9: List of variables within the global `par` struct of `EBdyna.go`. It holds all type of parameters. Most are set in the main function `G_eq`, but some are set by `sim_parameters` and `initialize_maps`. Here we often encounter the X -coordinate used in `EBdyna.go` and defined by

$$X \equiv R - R_0$$

variable	type	description
PROCESS_NUMBER	integer	'box' number
NB_PROCESS	integer	total number of 'boxes'
TEST_DIST	logical	set true for selected number of particles according to <code>make_test_dist.m</code> function
shared_memory	logical	true if <code>maps</code> -struct is stored in 'shared RAM'
ID	string	unique ID ⁶
ID_NAME	string	start of each output file, default: <code>G_eq.<ID>*.mat</code> , see appendix C.3.5 ⁷
comment	string	describe simulation
interp_scheme	integer	interpolation scheme, see appendix C.3.2
scheme	string	particle pushing scheme, BORIS recommended (<code>FabienB</code> is an older version).
coord_syst	string	coordinate system used in 3D interpolation (<code>tor</code> (toroidal) or <code>flux</code> (sfl))
superimpose_2D_3D	logical	enable the superposition of any 3D fields (will only have effect if certain conditions apply), see section 3.3
APPLY_RMP	logical	enable RMP field
APPLY_TFR	logical	enable TFR
APPLY_3D	logical	true given <code>par.mode</code> is not 2 and given either <code>par.APPLY_RMP</code> or <code>par.APPLY_TFR</code> or both
CALCULATE_PPHI_3D	logical	determine Δp_φ^n with equation (2.34)
APPLY_SAWTOOTH	logical	enable a ST crash
RMP_file	string	location of RMP-file produced by BS3D
TFR_file	string	location of TFR-file produced by BS3D
GET_PREC_INFO	logical	determine if precession information should be produced/loaded
SAVE_DATA_FILE	logical	produce any output files
DISTNAME	string	location of input distribution ⁸
LOADNAME	string	location of file actually used by simulation (often <code>DISTNAME</code> or <code>par.SAVENAME_STATS</code>)
mode	integer	switch to determine the simulation mode, see appendix C.3.4
Poincare_plot	logical	switch to determine if a Poincaré plot should be made during simulation
poincare_type	string	determines the figure to plot ⁹
paths	struct	struct with directory names, e.g. the <code>data_tokamak</code> -directory
SAVENAME_RAW	string	name of the RAW-file to save, see appendix C.3.5
SAVENAME_STATS	string	name of the PREC-file to save, see appendix C.3.5
SAVENAME	string	name of the FULL-file to save, see appendix C.3.5
NR_FUND_IN_LOOP	integer	number of fundamental time steps without counting the ejected particles (i.e. a complete 'loop')

⁶ ID should be unique, needs to start with 2 numbers and cannot contain 'prec', 'raw' or 'full'.

⁷ `par.ID_NAME` needs to contain `par.ID` and can for example be used to include a repository version.

⁸The input distribution is used if `TEST_DIST` is false and not if a precession file has been found.

⁹One can choose to plot Poincaré plots in (ψ, θ) space with `sfl`, (R, Z) space with `tor` or \tilde{x}, \tilde{y} space with `xy` (see appendix C.6.4).

<code>dt</code>	double	fundamental time step Δt
<code>NB_TIME_STAMPS</code>	integer	number of recordings of \mathbf{x}^n and $\mathbf{v}^{n-1/2}$ for each particle
<code>NB_STAMPS_SAVED</code>	integer	number of stamps saved ¹⁰
<code>NB_TIME_STEPS</code>	integer	number of ‘loops’ in total
<code>TIME_STAMP_PRECISION</code>	integer	number of ‘loops’ in between a recording of \mathbf{x}^n and $\mathbf{v}^{n-1/2}$ is made for each particle
<code>RECORD_PRECISION</code>	integer	number of ‘loops’ in between the intermediate saving of an RAW-file (default Inf)
<code>time_scale</code>	double vector	time value coinciding with the recordings of \mathbf{x}^n and $\mathbf{v}^{n-1/2}$
<code>offset_midplane</code>	vector	vector with polynomial (a, b) to determine $\hat{Z} = aR + b$, where \hat{Z} defined as the midplane
<code>step</code>	struct	parameters to use a coarser grid
<code>.. / R</code>	integer	coarser grid in R -direction, e.g. <code>par.step.R=3</code> skips 2 grid points periodically
<code>.. / Z</code>	integer	coarser grid in Z -direction, e.g. <code>par.step.Z=3</code> skips 2 grid points periodically
<code>.. / phi</code>	integer	coarser grid in φ -direction, e.g. <code>par.step.Z=4</code> skips 3 grid points. Condition <code>par.st.phi</code> needs to be 2^L , with $L \in \mathbb{N}$.
<code>st</code>	struct	sawtooth parameters (see section 3.4)
<code>.. / t_reconnection</code>	double	time for the reconnection process of the sawtooth
<code>.. / t_relaxation</code>	double	time for the recovery process of the sawtooth
<code>.. / n_reconnection</code>	integer	number time points in the reconnection process, where the values of r_1 , r_2 and are determined based on interpolation of the ψ_- profile
<code>.. / n_reconnection</code>	integer	number time points in the recovery process, where κ decreases
<code>.. / DR</code>	double	distance [m] in R direction to determine $\partial\Phi/\partial R$ and $\partial\psi/\partial R$ with finite difference
<code>.. / DZ</code>	double	distance [m] in Z direction to determine $\partial\Phi/\partial R$ and $\partial\psi/\partial R$ with finite difference
<code>.. / stable_time</code>	integer	In order to avoid singularities in reconnection parameters, the E and B fields are linearly interpolation from the original equilibrium towards the sawtooth field with index <code>stable_time</code>
<code>.. / rec_end_time</code>	integer	Index before reconnection phase finished, from hereon $\kappa = \kappa_c$ is determined by interpolation of the profile instead of equation (3.12), to avoid the numeric singularity
<code>.. / show_parameters</code>	logical	Switch to plot a figure with the sawtooth parameters. Used to determine <code>stable_time</code> and <code>par.st.rec_end_time</code>

Table D.10: List of variables within the global `dim` struct of `EBdyna.go`. It holds *dimensions*, e.g. scales / size of `maps`-struct and data about the sawtooth collapse.

variable	type	description
----------	------	-------------

¹⁰One might choose for a large number of time stamps ~ 1000 to extract information about the orbit, but one cannot save that many positions / parameters for all particles. Thus one can set `NB_STAMPS_SAVED` to a lower value than `NB_TIME_STAMPS` in order to reduce data storage.

<code>scale_X</code>	vector	unique values of $R - R_0$ in toroidal 2D grid
<code>scale_Z</code>	vector	unique values of Z in toroidal 2D grid
<code>mid_X</code>	integer	rounded index of magnetic axes in <code>scale_X</code> ($R \approx R_{axis}$)
<code>midX_zero</code>	integer	index of center of device in <code>scale_X</code> ($R = R_0$)
<code>mid_Z</code>	integer	rounded index of magnetic axes in <code>scale_Z</code> ($Z = 0 \approx Z_{axis}$)
<code>Z_axis</code>	double	Z -coordinate of magnetic axis ($Z = Z_{axis} = z_{axis} - z_{Z=0}$)
<code>X_axis</code>	double	X -coordinate of magnetic axis ($X_{axis} = R_{axis} - R_0$) (the Shafranov shift)
<code>R0</code>	double	R -coordinate of position $R = R_0$
<code>size_X</code>	integer	number of unique R positions in (toroidal) grid
<code>size_Z</code>	integer	number of unique Z positions in (toroidal) grid
<code>DX</code>	double	distance between two grid points in horizontal direction of the cross section
<code>DX</code>	double	distance between two grid points in vertical direction of the cross section
<code>DX_inv</code>	double	inverse of <code>DX</code> , i.e. $1/DX$, to avoid division when interpolating
<code>DZ_inv</code>	double	inverse of <code>DZ</code> , i.e. $1/DZ$, to avoid division when interpolating
<code>NB_PSI</code>	integer	number of radial positions (based on $\rho = \sqrt{\psi}$)
<code>psi_scale</code>	vector	value of ψ in for <code>NB_PSI</code> points given by FINESSE (based on equidistant $\rho \equiv \sqrt{\psi}$)
<code>psi_scale_correct</code>	double	value of ψ corresponding with <code>maps.psi_global</code> . ¹¹
<code>n3D</code>	struct	Contains dimensions / information about 3D fields
<code>../size_3D</code>	vector	three integers with the size of the 3D field
<code>../symm</code>	integer	symmetry in toroidal direction used to store 3D field (e.g. 16 for TFR)
<code>st</code>	struct	Stores quantities relevant for the sawtooth modelling
<code>../time</code>	vector	vector with times used in sawtooth modelling
<code>../psi_star</code>	vector	radial profile of ψ_- (length of <code>NB_PSI</code>)
<code>../psi_plus</code>	vector	radial profile of ψ_+ (length of <code>NB_PSI</code>)
<code>../r</code>	vector	radial coordinate $r \equiv \sqrt{2\chi}$
<code>../ind_rmix</code>	double	index of mixing radius, used to determine which particles are (possibly) influenced
<code>../r1</code>	vector	r_1 according to interpolation of fluxes during both phases of the sawtooth
<code>../r2</code>	vector	r_2 according to interpolation of fluxes during both phases of the sawtooth
<code>../r3</code>	vector	r_3 according to interpolation of fluxes during both phases of the sawtooth
<code>../r1_dot</code>	vector	r'_1 which is r_0/t_{rec} during reconnection phase and 0 during recovery phase
<code>../r2_dot</code>	vector	r'_2 according to equation (C.29)
<code>../r3_dot</code>	vector	value of r'_3 according to interpolation of fluxes during both phases of the sawtooth
<code>../k_c</code>	vector	value of κ_c determined by equation (3.12) during reconnection and constant afterwards
<code>../k</code>	vector	value of κ , which is equal to κ_c during reconnection phase and linear decrease during relaxation phase
<code>../kr</code>	vector	value of κ_r according to equation (3.12) during the reconnection phase and 1 afterwards
<code>../kr_inv</code>	vector	value of <code>dim.st.kr⁻¹</code> .

¹¹During the modelling of the sawtooth, it was found that two ψ -profiles were present in the `data_tokamak` directory. This one has been used in previous sawtooth modelling and is therefore considered the ‘correct’ one. Which one has been used to construct \mathbf{B}_{pol} is not clear.

<code>../kr_dot</code>	vector	time derivative of κ_r see appendix C.4
<code>../k_dot</code>	vector	time derivative of κ evaluated numerically during the reconnection phase and given by $-\max(\kappa_c) / \text{par.st.t_relaxation}$ during the relaxation phase.
<code>../k_m_kr_dot</code>	vector	time derivative of $(\kappa - \kappa_r)$ determined by <code>dim.st.k_dot - dim.st.kr_dot</code>
<code>../r1_t</code>	function	function to (quickly) determine r_1 during the reconnection phase
<code>../k_t</code>	function	function to (quickly) determine κ during the recovery phase

Table D.11: List of variables within the global `maps` struct of `EBdyna.go`. It holds quantities on 2D and 3D grids.

variable	type	description
<code>psi_global</code>	double	the maximum of <code>maps.psi_XZ</code> and equal to the maximum of <code>dim.psi_scale_correct</code>
<code>psi_XZ</code>	2D-matrix	value of ψ on rectangular R, Z grid
<code>psi_norm_XZ</code>	2D-matrix	radial index based on a equidistant ρ
<code>theta_normal_XZ</code>	2D-matrix	value of θ placed on rectangular R, Z grid
<code>theta_phase_shift_XZ</code>	2D-matrix	value of $\theta - \pi$ placed on R, Z grid. This grid is used to interpolate particles near the mid plane and avoid artefacts (e.g. interpolation in a grid cell with values $2\pi - \epsilon$ and $0 + \epsilon$ with $\epsilon, \varepsilon \ll 1$)
<code>chi_XZ</code>	2D-matrix	(available if <code>par.APPLY_SAWTOOTH</code>). Values of χ placed on rectangular R, Z grid
<code>B_2D</code>	3D-matrix	(available if <code>par.interp_scheme</code> is 3 or 6). Values of the magnetic field in the 2D equilibrium. The 3 rd dimension is reserved for different components: B_R, B_Z, B_φ
<code>B_3D</code>	4D-matrix	(available if <code>par.interp_scheme</code> is 3 or 6). Values of \mathbf{B} on toroidal (R, Z, φ) or straight field line-grid (θ, ψ, φ). The 4 th dimension is reserved for different components: B_R, B_Z, B_φ
<code>B_2D</code>	struct	(available if <code>par.interp_scheme</code> is not 3 or 6)
<code>../BR</code>	2D-matrix	values of B_R for the 2D equilibrium
<code>../BZ</code>	2D-matrix	values of B_Z for the 2D equilibrium
<code>../Bphi</code>	2D-matrix	values of B_φ for the 2D equilibrium
<code>n3D</code>	struct	Values in the 3D grid due to the perturbation
<code>../BR</code>	3D-matrix	(available if <code>par.interp_scheme</code> is not 3 or 6). Values of B_R for the 3D fields
<code>../BZ</code>	3D-matrix	(available if <code>par.interp_scheme</code> is not 3 or 6). Values of B_Z for the 3D fields
<code>../Bphi</code>	3D-matrix	(available if <code>par.interp_scheme</code> is not 3 or 6). Values of B_φ for the 3D fields
<code>../AR</code>	3D-matrix	(available if <code>par.CALCULATE_TRUE_PPHI</code>). Values of A_R for the 3D field (used to determine p_φ)
<code>../AZ</code>	3D-matrix	(available if <code>par.CALCULATE_TRUE_PPHI</code>). Values of A_Z for the 3D field (used to determine p_φ)
<code>../Aphi</code>	3D-matrix	Values of A_φ for the 3D field (used to determine p_φ)
<code>../dAR_dphi</code>	3D-matrix	(available if <code>par.CALCULATE_TRUE_PPHI</code>). Values of A_φ for the 3D field (used to determine p_φ)
<code>../dAZ_dphi</code>	3D-matrix	(available if <code>par.CALCULATE_TRUE_PPHI</code>). Values of A_φ for the 3D field (used to determine p_φ)

D.2. PARAMETER LIST

<code>../dAphi_dphi</code>	3D-matrix	Values of A_φ for the 3D field (used to determine p_φ)
<code>../others</code>	3D-matrices	Other 3D grid values can be loaded by the <code>initialize_maps.m</code> function when the <code>par.interp_scheme</code> is set to 0, i.e the A based interpolation is selected. These fields correspond to quantities described by equations (C.3) to (C.16)

Input, output and other re-occurring quantities

Along with these global quantities, there are other quantities which are quite straightforward, e.g. **x**, **v**. In addition we have the `ejected` parameter, a logical describing which particles have crossed the LCFS. On top of that also the `process_time` is stored, i.e. the duration of the simulation by this ‘box’.

Others are described in tables D.12 to D.14.

Table D.12: List of variables within the `input` struct of `EBdyna.go`. It holds the input of the current simulation.

variable	type	description
<code>m</code>	double	Mass of particles.
<code>Z</code>	double	Charge state of particles.
<code>Ekin</code>	vector	Kinetic energy at start.
<code>N_total</code>	integer	Total amount of particles in all ‘boxes’.
<code>particle_nr</code>	vector	Unique number assigned to each particle.
<code>N_job</code>	integer	Number of particles used by current ‘box’.
<code>x_gc</code>	matrix	Position of gyro center x_{gc} , i.e. the position defined in the distribution file.
<code>x</code>	matrix	Position of the particle at the start of the simulation x .
<code>v</code>	matrix	Velocity of the particle at the start of the simulation v .
<code>mm</code>	vector	Magnetic moment of the particle at the start of the simulation μ .
<code>pphi</code>	vector	Canonical angular toroidal momentum of the particle at the start of the simulation p_φ .

Table D.13: List of variables within the `output` struct of `EBdyna.go`. It holds the output of the current simulation.

variable	type	description
<code>x</code>	matrix	position x stored, where the three dimensions devide particles, R, Z, φ components and the time stamp.
<code>v</code>	matrix	velocity v stored, where the three dimensions devide particles, R, Z, φ components and the time stamp.
<code>x_ej</code>	matrix	position x in the last time step before the particle got lost.
<code>v_ej</code>	matrix	velocity v in the last time step before the particle got lost.
<code>time_step_loss</code>	vector	time step of particle when it has been lost.
<code>nr_midplane_crossing</code>	vector	number of times a particle crossed the midplane.
<code>nr_vpll_crossing</code>	vector	number of times a particle changed sign of v_{\parallel} .
<code>pphi_kin</code>	matrix	Kinetic value of p_φ (equation (2.33)).
<code>pphi_an</code>	matrix	Analytic value of p_φ (equation (2.34)).
<code>Delta_pphi</code>	vector	Difference of the final value of the kinetic p_φ since the start of the simulation.

<code>mm</code>	matrix	Magnetic moment μ .
<code>dpphi_dt</code>	vector	linear fit of the p_φ evolution.
<code>dmm_dt</code>	vector	linear fit of the magnetic moment evolution.
<code>ST_interaction</code>	vector	Logical describing if the particle ever has been within $r < 1.1r_{mix}$.
<code>wb</code>	double	Value of ω_b based on the number of bounces.
<code>wd</code>	double	Value of ω_d based a linear fit of φ .
<code>nr_profile</code>	matrix	Evolution of the number of particles within ψ and $\psi+D\psi$ (the flux surfaces in the profile).
<code>2D_profile</code>	3D-matrix	Evolution of the 2D profile in R, Z -space (binned).
<code>loss</code>	vector	number of particles that are lost at certain time stamp.

Table D.14: List of variables within the `prec` struct of `EBdyna.go`. It holds precession information acquired in a 2D equilibrium simulation.

variable	type	description
<code>psi_avg</code>	double	Average value of ψ in the 2D equilibrium simulations
<code>q_avg</code>	double	Average value of q in the 2D equilibrium simulations
<code>full_orbit</code>	logical	Obsolete parameter describing which of the trapped particles have more than 3 bounces, i.e. the simulation is longer then an orbit period.
<code>wb</code>	double	Value of ω_b based on the number of bounces.
<code>wd</code>	double	Value of ω_d based a linear fit of φ .
<code>par</code>	struct	parameters which, in combination with the <code>input</code> struct, can reproduce the 2D precession simulation.
<code>../dt</code>	double	Fundamental time stamp Δt .
<code>../NB_TIME_STAMPS</code>	integer	Number of time stamps saved in the 2D precession simulation.
<code>../end_time</code>	double	Simulation length of the precession simulation t_{sim} .
<code>pop</code>	struct	Describing the particle in the following categories:
<code>../ALL_TRAPPED</code>	logical	identifying trapped particles.
<code>../ALL_PASSING</code>	logical	identifying passing particles.
<code>../TRAPPED_MINUS</code>	logical	identifying trapped particles precessing in the co-current, counter-field direction.
<code>../TRAPPED_PLUS</code>	logical	identifying trapped particles precessing in the counter-current, co-field direction.
<code>../POTATOES</code>	logical	identifying trapped particles which have a inner leg crossing the midplane at the HFS.
<code>../STAGNATION</code>	logical	identifying passing particles which R coordinate is always smaller or always larger than R_0 , i.e. they stagnate at either the HFS or LFS of the plasma.
