UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Physics and Technology

Numerical Simulations of Turbulence at the Boundary of Fusion Plasma

Henrik Jäntti

FYS-3900 Master's thesis in physics - 60 ECTS
May 2022

"*If I have seen further it is by standing on the shoulders of giants, and a mountain of caffeine.*" - Newton, augmented by me

Abstract

Numerical simulations of reduced fluid models describing magnetized plasmas have previously been used to study the boundary of fusion plasmas extensively. A common approximation applied to these models is the so-called Boussinesq approximation. One of the requirements for the Boussinesq approximation to be applicable is that the background particle density is significantly higher than the fluctuations in the particle density. Experimental measurements have revealed that at least in the scrape-off layer (SOL) this is not the case, we will therefore investigate the effects of the Boussinesq approximation in this thesis. In addition to this, we also investigate the effects of linearizing the logarithmic density in the drift wave term of the models. To do this we derive four sets of reduced fluid equations describing the edge and SOL regions of a tokamak plasma, one for each combination of approximations. These models are then simulated numerically using the BOUT++ framework. Single point time-series measurements from the simulations are then analysed using the stochastic filtered Poisson process in addition to conventional statistical methods. From the analysis, we observe that there are noticeable differences between the different models, where the largest difference appears when the Boussinesq approximation is used. We found that the logarithmic particle density in the drift wave term generated increased shear flow in the edge region increasing the confinement of plasma to the edge region reducing the overall particle density in the SOL. By relaxing the Boussinesq approximation we observed that structures moving radially outward through the domain have smaller sizes and that the structures in the Boussinesq approximated case, and that these structures decrease in size at faster rates. This indicates that the use of the Boussinesq approximation influences the results of numerical simulation, and deeper studies into the differences between the two cases need to be done.

Acknowledgements

First I would like to thank Professor Odd Eric Garcia who introduced me to this wonderful and diverse field of physics and who throughout the year gave ideas, insight and feedback for my thesis.

Secondly, I would like to thank my Co-Supervisor PhD Gregor Decristoforo whose knowledge of numerics and programming in general, has helped me get through both the running of the simulations and the analysis afterwards. I also need to thank him for taking the time to have regular meetings even with a six-hour time difference.

Thirdly I would also thank Co-Supervisor Associate Professor Markus Held who introduced me to gyro-fluids and who spend many hours discussing and helping me with FELTOR.

I would also like to thank Professor Rune Graversen who generously let me use his computation time on FRAM, allowing for my simulations to be run and finished in time.

Finally thanks to all my friends and family for support throughout the process of writing this thesis, especially my office mates who helped keep me sane for the last couple of weeks of writing.

Contents

List of Figures

List of Tables

1

Introduction

In a time when the need for energy increases while we still try to tackle the ongoing climate crisis, the need for greener and more efficient energy sources is higher than ever. One of the more promising candidates for such an energy source is nuclear fusion power. Fusion energy is achieved by fusing lighter elements creating heavier elements and releasing energy. This is the process that takes place in the sun and other stars. If one wants to sound poetic it will be like capturing a star in a box.

One of the major advantages of a potential fusion power reactor is the extreme energy density of the fuel that is used and its abundance. The choice of fuel is between different isotopes of hydrogen and helium. The most common fuel used is a mix between the hydrogen isotopes deuterium H^2 (D) and tritium N^3 (T) known as a D-T reaction, this would have an energy density of around $338 \cdot 10^6$MJ/kg. Deuterium can be extracted from water as it naturally occurs in ocean water and if one used all the deuterium in the ocean there would be enough fuel to create energy for around $2 \cdot 10^9$ years [1]. While there is no naturally occurring tritium on earth it is possible to extract from the lithium isotope Li^6 so we are only limited by the total amount of lithium available. The total amount of lithium should last for around 20 000 years [1], in which time the development of D-D or other types of reactions could be achieved. Another advantage of the fusion power plant is the fact that the only byproduct that would be generated by this process would be radioactive elements with a half-life [1] much smaller than what is created by the conventional fission reaction that has byproducts such as Plutonium-239 with a half-life of up to 24

000 years [2]. The safety of the fusion reactor is also much greater than the fission reactor, a meltdown event of a fission reactor could be catastrophic as demonstrated by the Chernobyl accident. In comparison, the most damaging event that could happen to a fusion reactor would be a loss of the plasma to the wall which would just ruin the machine, as in a fusion reactor as there is no possibility for a runaway nuclear process.

The big question now is how we achieve this great act of creating energy through fusion. The temperature needed to achieve fusion is extreme, as the natural occurrence of fusion is within the core of a star. Our closest star the Sun has a core temperature of around $15.7 \cdot 10^6$ Kelvin. This turns out to be relevant as an estimated temperature for D-T fusion to occur is around $15 \cdot 10^6$ K. To be able to contain something of this temperature we exploit the fact that at these temperatures the D-T mixture will become ionized and form a plasma. This allows us to exploit the fact that plasmas can be contained within magnetic fields. The exploitation of this property is the basis for most fusion reactor designs. The currently most developed fusion reactor concept is the tokamak. Developed by Igor Tamm and Andrei Sakharov the name is a translation of the Russian acronym for **to**roidal **cha**mber with **ma**gnetic **c**oils [3]. The basics of this concept is in the name, a reactor shaped like a torus (donut) using a combination of current bearing coils to create magnetic fields that contain the plasma. A schematic of a general tokamak design can be found in figure 1.1. The biggest project in fusion reactor research, the ITER experiment, is based on the tokamak design [4]. And it is with this design of reactor that this thesis concerns itself.

Figure 1.1 illustrates the electric current and magnetic field configurations of a tokamak plasma. The blue coils in figure 1.1 induce the toroidal magnetic field and are capable of creating fields with strengths of several teslas. They achieve this using superconductors, which in the case of ITER must be cooled to around 4 K. A newer superconductor technology that is currently in testing can operate at higher temperatures and created a record-breaking field strength of 20 T in the autumn of 2021 [6]. For comparison the Earths magnetic field is around $25 - 64\mu$T [7]. In addition to the magnetic coils, the green central beam in figure 1.1 is a solenoid inducing a toroidal current in the plasma creating a poloidal magnetic field. These two components create the main parts of the magnetic field for the reactor. The gray magnets in figure 1.1 create vertical magnetic fields which limit the plasmas' tendency to move outwards toward the wall of the reactor. This configuration of coils splits the plasma into three main regions: the core, the scrape-off layer (SOL) and the wall shadow. The core and SOL regions can is depicted in figure 1.2. Firstly, the core region contains most of the plasma, the magnetic field lines lie on flux surfaces of constant pressure and are closed in on themselves creating the main body of the magnetic field. Secondly, the SOL contains magnetic field lines that do not

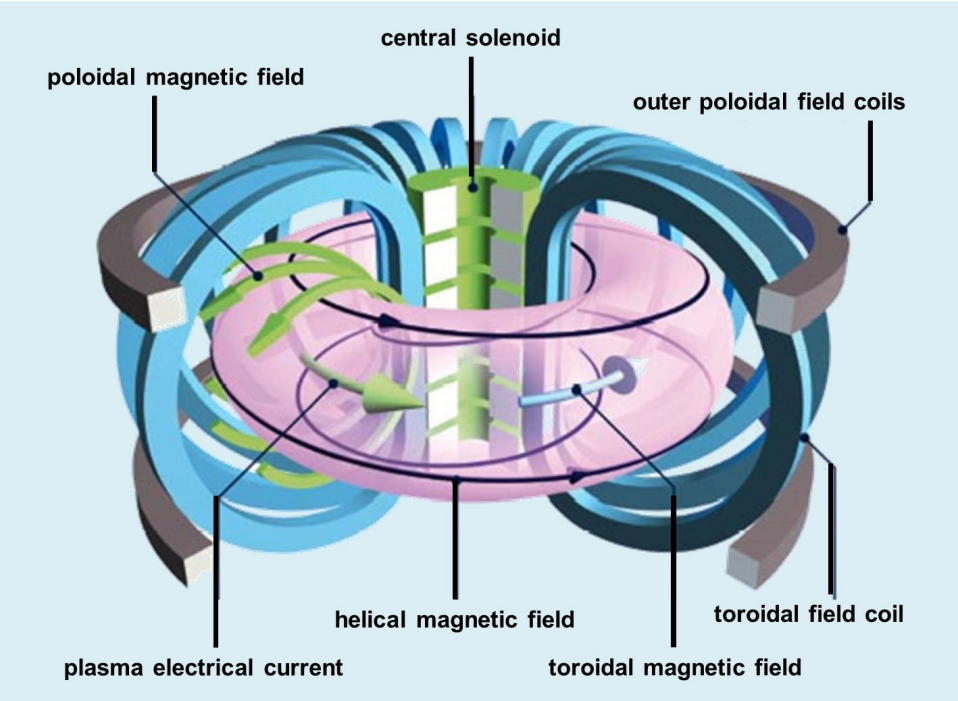

Figure 1.1: The basic concept behind a tokamaks magnetic fields. [5]

close in on themselves but intersect material surfaces called divertor plates. Thirdly, the wall shadow in which the magnetic fields intersect with the main chamber walls of the reactor. The boundary between the core and the SOL is indicated by the commonly called last closed flux surface (LCFS) or separatrix and ideally, all the plasma would be contained within the core. In the event of plasma exiting the LCFS, the magnetic fields in the SOL would then ideally redirect the plasma towards the divertor plates which are specifically made to handle the heat exhaust from the plasma in order to handle plasma interacting with the wall.

It turns out from both experimental measurements and numerical simulations [9–12] that not all the plasma leaving the core is diverted. There have been observed coherent structures of plasma, with particle densities much higher than the mean density of the SOL, moving out of the core region and reaching the reactor walls. This does not only damage the walls by erosion and melting, it also introduces impurities into the plasma making fusion harder to achieve. As these events will have an important effect on both the longevity and viability of any reactor, there has been put major efforts into understanding the mechanisms behind these so-called filaments. In figure 1.3 we can see images from a fusion reactor where a filament of plasma leaves the last closed flux surface and move through the SOL.

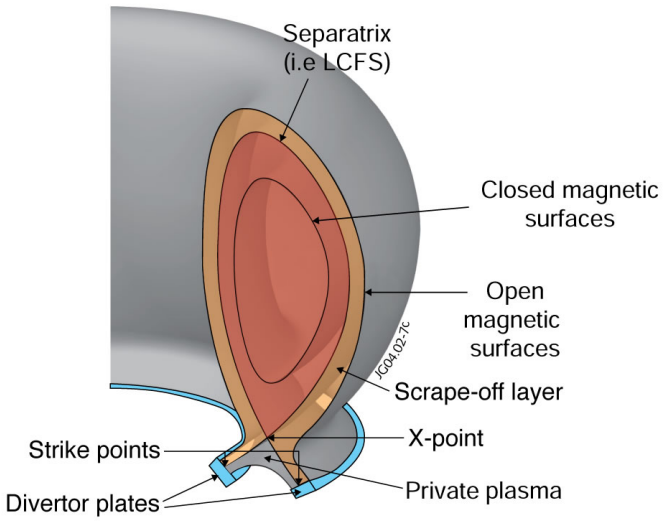

Figure 1.2: Illustration of the basic magnetic field structure of a tokamak. With the core region marked in red and the scrape of layer marked in yellow. [8]

The main mechanism behind the transport is attributed to the interchange instability [11]. This instability is driven by the occurrence of curvature and gradient-B drifts due to the non-uniform magnetic field of the reactor. If there is a local build-up of plasma density these drifts will create a region with charge separation in the plasma. The separation of electrons and ions will create an electric field which then induces an electric drift of the plasma. This will then make the filament move out towards the walls of the reactor. A schematic representation of this process is depicted in figure 1.4.

As these filaments lead to transport of plasma throughout the SOL there have been multiple attempts to model this transport. The first examples of this was trying to model the transport by using a simple diffusion model [15]. Using just a diffusive process gave results that did not compare well to what was observed in experiments. This led to the introduction of an advective-diffusive model where in addition to a diffusive process an advection term was added. This as well does not give results in agreement with experimental measurements. [16]. Therefore another model has been introduced where instead of modeling the transport as a diffusion or advection process the transport is modeled using a stochastic statistical model to describe the fluctuations of plasma caused by the plasma filaments [17]. This statistical model is referred to as a Filtered Poisson Process (FPP) and we will be using this model for the basis of our statistical analyses.

The data that will be analysed in this thesis will all be from numerical sim-

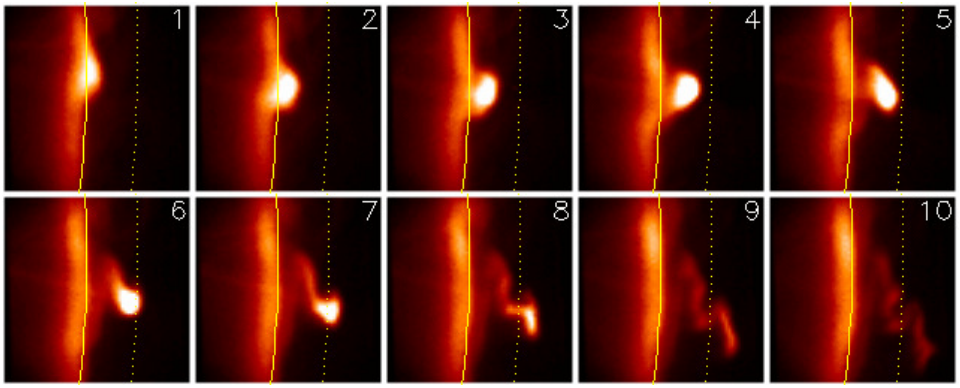

Figure 1.3: Plasma moving trough the SOL of the NSTX experiment in the form of blob-like structures. The image captures the cross field view of the rector near the outer midplane, where the solid line is the estimated position of the separatrix and the dotted line shows the shadow of a RF antenna. Camera captures area of $(23cm)^2$ with $10\mu s$ between each frame. Image from Princeton Plasma Physics Laboratory [13]

ulations. To describe the transport of the plasma trough the SOL we will be doing full turbulence simulations of the SOL and the edge regions. The edge region is the part of the core closest to the separatrix, and will therefore have an effect on the turbulence in the SOL. Doing full turbulence simulations the accuracy of our results will be mostly determined by the accuracy of the model and the simulation code. As it is not feasible to run numerical models for an indefinite amount of time, approximations are used to lower the computational cost. One such approximation that is in common use in plasma physics is the Boussinesq approximation. We will therefore investigate if the use of this approximation has significant effects on the results from analysis. This will be tested by applying different approximations to the simulations we are doing.

As it is hard to distinguish between differences in turbulence visually, the method of comparison between different simulations will be the statistics properties of the fluctuations. While mean, relative fluctuations, skewness and flatness moments at different radial positions would be the most know statistics, there are others, mainly fluctuation statistics, which are the basis of the FPP model. This focuses on time series measured over long periods of time and allows other insight into the statistical properties of the turbulence. Fluctuation statistics consist of power spectra, conditionally averaged waveform and probability density functions. This will give even more ways of comparing the numerical simulations and might give greater insight into the potential differences between the models and dependence on model parameters

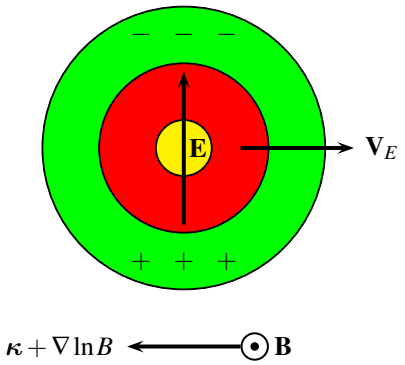

Figure 1.4: Illustration showing buildup of density leading to charge separation by gradient and curvature drifts, inducing an electric field and therefore an electric drift. Here $\boldsymbol{\kappa} = \boldsymbol{b} \cdot \nabla \boldsymbol{b}$ is the curvature vector with $\boldsymbol{b}$ being a unit vector pointing in the direction of the magnetic field and $\nabla \ln B$ show the gradient of the magnetic field [14]

The rest of the thesis is structured as follows. In Chapter 2 we will derive model equations describing the edge and SOL regions, discuss the Boussinesq approximation, and present the final equation that will be the basis for our numerical simulations. Then in Chapter 3 we will introduce and give an overview of how the numerical simulations of the equations are done and how the output of the data are analyzed. In Chapter 4 we will present the statistical analysis methods and present the results of these methods applied to the data from the simulations. We will then in Chapter 5 discuss the results for the previous chapters and in Chapter 6 we will present the conclusions and outlook for future work.

2

Model equations

In this chapter, we are going to derive the reduced fluid equations that are the basis for the numerical simulations later in this thesis.

2.1 Fluid equations

The fundamental movement of a charged particle in a magnetic field is its perpendicular gyration around what is called its guiding centre. This gyration is described by a particle's gyro-frequency ω_s, which for a singly charged particle s is given as $\omega_s = eB/m_s$, where e is the elementary charge, m_s is the and mass of the particle and B is the magnetic field it is exposed to. The radius of this gyrating motion is captured by the gyro-radius ρ_s given as $\rho_s = m_s v_\perp / eB$, where $v_\perp$ is the velocity in which the particle gyrates around the magnetic field lines. From the definitions of the gyro-radius and gyro-frequency, it is apparent that an increase in the strength of the magnetic field will increase the gyro-frequency and decrease the gyro-radius. We can for strong enough magnetic fields the following will hold

$$\rho_s \ll \lambda_{\text{mfp}}, \tag{2.1}$$

and

$$\omega_s \gg \nu_s. \tag{2.2}$$

Here λ_{mfp} is the mean free path for the particle, or in other words, the length a particle travels before it collides with another particle and ν_s is the collision frequency i.e. the time between each collision. These are the conditions for a fully magnetized plasma and with these conditions the movement of a single particle becomes less significant to the entire system, thus using single particle motion to describe the entire system becomes an exercise of patience and extreme computing power. A system that also has the property where the motion of a single particle is insignificant to the wider system is fluids. With the similarities in mind, one can therefore construct equations for a plasma resembling the equations for a fluid.

To construct the equations for a fluid out of the particle motion of a collection of charged particle we transform the usual kinematic description by considering a 6-dimensional phase space of position and momentum. One can then by the use of the Boiltzmann equation derive an infinite set of equations that describes the plasma as a fluid. The first two moments of the kinetic equation are the particle continuity equation and the momentum equation for any species s given as

$$\frac{\partial n_s}{\partial t} + \nabla \cdot (n_s \boldsymbol{u}_s) = 0 \tag{2.3a}$$

$$m_s n_s \left(\frac{\partial}{\partial t} + \boldsymbol{u}_s \cdot \nabla \right) \boldsymbol{u}_s = -\nabla \cdot \boldsymbol{\pi}_s - \nabla p_s + q_s n_s \left(\boldsymbol{E} + \boldsymbol{u}_s \times \boldsymbol{B} \right) + \sum_{s'} m_s n_s \nu_{ss'} (\boldsymbol{u}_{s'} - \boldsymbol{u}_s) \tag{2.3b}$$

where $n_s, m_s, \boldsymbol{u}_s, q_s$ is the particle density, mass, velocity and charge for the respective species s. $\boldsymbol{E}$ and $\boldsymbol{B}$ are the electric and magnetic fields; $\boldsymbol{\pi}_s$ is the viscous stress tensor, p_s is the pressure and $\nu_{ss'}$ is the collisional frequency between species s and s', where s' denotes all other particle species present in the plasma. We denote electrons by a subscript e and ions by subscript i such that $q_e = -e$ is the charge for an electron and $q_i = e$ is the charge for a singly charged ion. As the magnetic field is an important part of all the dynamics of a plasma it is usual to define the coordinate system with this in mind, thus we introduce the unit vector $\boldsymbol{b} = \boldsymbol{B}/B$. We will also use the subscripts $\parallel$ and $\perp$ such that a general vector $\boldsymbol{A}_\parallel = (\boldsymbol{b} \cdot \boldsymbol{A})\boldsymbol{b}$ is parallel to the magnetic field and $\boldsymbol{A}_\perp = \boldsymbol{b} \times (\boldsymbol{A} \times \boldsymbol{b}) = \boldsymbol{A} - \boldsymbol{A}_\parallel$ to indicate vector components perpendicular to the magnetic field, respectively.

As an infinite set of equations are hard to do anything with, it is necessary to impose a closure to the fluid equations. This will allow us to reduce the number of equations in use, thus allowing for analytical treatment. One of the more common closures is the drift ordering, which will be the closure that we will utilize.

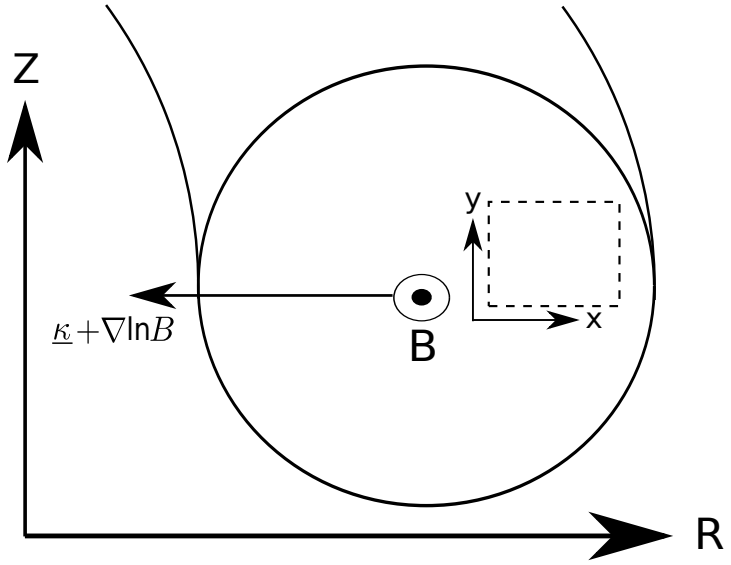

Figure 2.1: Illustration of the cross section of a tokamak plasma with the main toroidal magnetic field pointing out of the figure. The rectangular area illustrates where the the slab approximation is applied. $\boldsymbol{\kappa} = \boldsymbol{b} \cdot \nabla \boldsymbol{b}$ is the curvature of the magnetic field and $\nabla \ln B$ show the gradient of the magnetic field. [18]

The geometry of the system we are modelling plays an important role in how the equation will turn out. As a tokamak is in the shape of a torus it is then natural to apply a toroidal geometry to the equations, this however can be quite tricky to deal with. A system with a cartesian coordinate system would be easier to deal with and we will therefore impose the so-called slab geometry on our equations. In the slab geometry, we only consider a small part of the entire torus, namely a rectangular area around the outboard mid-plane. In a small domain around this point, toroidal geometry is approximately cartesian such that the x-coordinate points along the R axis and y-coordinate points along the bi-normal $\boldsymbol{r} \times \boldsymbol{b}$. A not-to-scale illustration of this can be seen in figure 2.1. The arrow at the centre of the figure illustrates the direction of the curvature $\boldsymbol{\kappa} = \boldsymbol{b} \cdot \nabla \boldsymbol{b}$ and the gradient of the magnetic field $\nabla \ln B$. For the slab geometry, both of these will point inwards towards the centre of the tokamak. For the derivations of the model, we will first look at more general coordinates, and then later introduce this specific slab geometry.

2.2 Deriving the drift velocities

We also for the sake of simplicity only consider a two-species plasma, with electrons and singly charged ions. We will also be neglecting viscosity and

collisionality for now, as dealing with them self-consistently can be quite finicky at best, they will be introduced later on in the derivations. Starting form the momentum equation for a particle species s, we can rewrite the perpendicular fluid velocity by crossing the equation with the unit vector along the magnetic field $\boldsymbol{b}$

$$\boldsymbol{b} \times \left| m_s n_s \left(\frac{\partial}{\partial t} + \boldsymbol{u}_s \cdot \nabla \right) \boldsymbol{u}_s = -\nabla p + q_s n_s \left(\boldsymbol{E} + \boldsymbol{u}_s \times \boldsymbol{B} \right) \right. \tag{2.4}$$

$$\boldsymbol{b} \times m_s n_s \left(\frac{\partial}{\partial t} + \boldsymbol{u}_s \cdot \nabla \right) \boldsymbol{u}_s = -\boldsymbol{b} \times \nabla p + q_s n_s \boldsymbol{b} \times \boldsymbol{E} + q_s n_s \boldsymbol{u}_{s\perp} B \tag{2.5}$$

here we have used the fact that $\boldsymbol{b} \times (\boldsymbol{u}_s \times \boldsymbol{b}) = (\boldsymbol{u}_s - \boldsymbol{u}_{s\parallel}) = \boldsymbol{u}_{s\perp}$. Sorting the terms we get the equation for the velocity of the particle species s

$$\boldsymbol{u}_{s\perp} = \frac{\boldsymbol{E} \times \boldsymbol{b}}{B} + \frac{\boldsymbol{b}}{q_s n_s B} \times \nabla p + \frac{m_s}{q_s B} \boldsymbol{b} \times \left(\frac{\partial}{\partial t} + \boldsymbol{u}_s \cdot \nabla \right) \boldsymbol{u}_s. \tag{2.6}$$

From this equation we can isolate the different drifts that makes up the total velocity for the species

$$\boldsymbol{u}_E = \frac{\boldsymbol{E} \times \boldsymbol{b}}{B} = \frac{1}{B} \boldsymbol{b} \times \nabla \phi \tag{2.7a}$$

$$\boldsymbol{u}_{ds} = \frac{\boldsymbol{b}}{q_s n_s B} \times \nabla p \tag{2.7b}$$

$$\boldsymbol{u}_{ps} = \frac{m_s}{q_s B} \boldsymbol{b} \times \left(\frac{\partial}{\partial t} + \boldsymbol{u}_s \cdot \nabla \right) \boldsymbol{u}_s. \tag{2.7c}$$

From top to bottom these are the E cross B (ExB) drift, diamagnetic drift and polarisation drift, respectively.

The ExB drift is the main mechanism behind any sort of bulk movement in plasmas. It is also of interest to note that it is independent of the charge of the particle, thus any time there is an electric field present in the plasma this drift will be present for all particle species. The second equality in the ExB drift comes from using the electrostatic approximation ,$\boldsymbol{E} = -\nabla \phi$, where ϕ is the electrostatic potential.

The diamagnetic drift occurs due to a pressure difference in the plasma. It is important to understand that this is an apparent drift resulting from the fact that there are more gyrating particles in the area with higher pressure and therefore the plasma fluid will appear to move, even if there is no actual movement of the guiding centres of the particles. There is however a relationship between this drift and the gradient and curvature drifts of a single particle that occur due to the unfavourable magnetic field present in tokamaks. There

are also additional drifts such as the drift of particles along the magnetic field which will get further exploration later.

In the polarisation drift given by equation (2.7c) we observe the dependence of the entire species velocity. To solve this problem we introduce the drift ordering. We do this by first analysing the scale of the different drifts. Introducing a reference scale for our equations $\delta \equiv \rho_s/L$, where $\rho_s = mC_s/eB$ is the gyroradius and L is the typical length scale of the plasma parameters. Here C_s is the thermal velocity also called the acoustic speed defined as $C_s = (T_s/m_s)^2$. We also introduce the normalisation $\hat{n} = n/n_0, \hat{\boldsymbol{u}} = \boldsymbol{u}/U, \hat{p} = p/mn_oC_s^2$, and $\hat{\boldsymbol{x}} = \boldsymbol{x}/L, \hat{t} = tL/U$. Where n_0 is the background density, U is the typical velocity of the system. Using this normalisation into our equation we get

$$\hat{n}_s \left(\frac{\partial}{\partial \hat{t}} + \hat{\boldsymbol{u}}_s \cdot \nabla \right) \hat{\boldsymbol{u}}_s = -\frac{C_s^2}{U^2} \hat{\nabla} \hat{p} + \hat{n}_s \frac{L}{\rho_c} \frac{C_s}{U} \left(\frac{\boldsymbol{E}}{UB} + \hat{\boldsymbol{u}}_s \times \boldsymbol{b} \right) \tag{2.8}$$

$$\tag{2.9}$$

Now we can consider two different sizes to decide the ordering. The ratio between the thermal velocity and the fluid velocity as well as their relation to the scale of the system. In the case of drift ordering, we assume that the fluid velocity is smaller than the thermal velocity and we get

$$\frac{U}{C_s} \sim \frac{\rho_s}{L} \ll 1. \tag{2.10}$$

This relation tells us that to lowest order we have just the pressure and the Lorentz force terms left. Taking the cross product of these terms with $\boldsymbol{b}$ we get the drifts

$$\hat{n}_s \hat{\boldsymbol{u}}_{s\perp} = \hat{n} \hat{\boldsymbol{u}}_E + \boldsymbol{b} \times \hat{\nabla} \hat{p}_s.$$

The first term on the RHS of the expression can be identified as the normalized ExB drift $\hat{\boldsymbol{u}}_E = \boldsymbol{u}_E/U$ and the second is the diamagnetic drift. As these are the lowest order drifts and therefore most important we use these as the velocity in the polarisation drift and equation (2.7c) becomes

$$\boldsymbol{u}_{ps} = \frac{m_s}{q_s B} \boldsymbol{b} \times \left[\frac{\partial}{\partial t} + \left(\boldsymbol{u}_E + \boldsymbol{u}_{ds} + \boldsymbol{u}_{s\parallel} \right) \cdot \nabla \right] \left(\boldsymbol{u}_E + \boldsymbol{u}_{ds} + \boldsymbol{u}_{s\parallel} \right).$$

We now have the velocities for both particle species and we insert them into our density equation,

$$\frac{\partial n_s}{\partial t} + \nabla \cdot \left[n_s \left(\boldsymbol{u}_E + \boldsymbol{u}_{ds} + \boldsymbol{u}_{ps} \right) \right] + \nabla \cdot \left[n_s \left(\boldsymbol{u}_{\parallel s} + \boldsymbol{u}_{vs} + \boldsymbol{u}_{\pi s} \right) \right] = 0 \tag{2.11}$$

where we have chosen to write the parallel drifts $\boldsymbol{u}_{\parallel s}$ on their own as the choice of closure for the drift is highly dependent on the chosen ordering. We have

also at this point added the collisional drift $\boldsymbol{u}_{vs}$ and the viscosity drift $\boldsymbol{u}_{\pi s}$, they are included together with the parallel drift as their closure is also highly involved.

The equation above will then be the basis for any further modelling we do.

2.3 Particle and charge conservation

We first need to derive an equation for the evolution of the particle densities. We start with the electrons, assuming that they are isothermal, also since the mass of an electron is orders of magnitude smaller than the mass of an ion the polarisation drift for electrons will have an inconsequential effect on the overall plasma dynamics and can therefor be neglected. Not considering the collisional, parallel and viscous terms, the equation for the electrons becomes

$$\begin{aligned}\frac{\partial n_e}{\partial t} + \nabla \cdot \left[n_e \left(\boldsymbol{u}_E + \boldsymbol{u}_{de}\right)\right] &= 0 \\ \frac{\mathrm{d}n_e}{\mathrm{d}t} + n_e \nabla \cdot \boldsymbol{u}_E + \nabla \cdot \left(n_e \boldsymbol{u}_{de}\right) &= 0\end{aligned}$$

where we have defined the advective derivative with the ExB drift as $d/dt = \partial_t + \nabla \cdot \boldsymbol{u}_E$.

Taking a closer look at the ExB term we can define a useful operator

$$\begin{aligned}\nabla \cdot \boldsymbol{u}_E &= \nabla \cdot \left(\frac{1}{B}\boldsymbol{b} \times \nabla\phi\right) \\ &= \frac{1}{B}\left(\nabla \times \boldsymbol{b} + \nabla \ln B \times \boldsymbol{b}\right) \cdot \nabla\phi \\ &= \mathcal{K}\left(\phi\right)\end{aligned}$$

where $\mathcal{K}\left(\cdot\right)$ indicates the curvature operator, see appendix A.1 for further detail. Observe also that the diamagnetic term can be written in terms of the curvature operator

$$\nabla \cdot \left(n_e \boldsymbol{u}_{de}\right) = \frac{T_e}{q_e}\nabla \cdot \left(\frac{1}{B}\boldsymbol{b} \times \nabla n_e\right) = \frac{T_e}{q_e}\mathcal{K}\left(n_E\right)$$

and we end up with an equation for the electron density that becomes

$$\frac{\mathrm{d}n_e}{\mathrm{d}t} + n_e \mathcal{K}\left(\phi\right) + \frac{T_e}{q_e}\mathcal{K}\left(n_e\right) = 0 \tag{2.12}$$

$$\frac{\mathrm{d}n_e}{\mathrm{d}t} + n_e \mathcal{K}\left(\phi\right) + \frac{T_e}{q_e}\mathcal{K}\left(n_e\right) = -\nabla \cdot \left[n\left(\boldsymbol{u}_{e\parallel} + \boldsymbol{u}_{ve}\right)\right]. \tag{2.13}$$

where we in the last step have added the parallel and colissional terms, neglecting electron viscosity as it is approximately zero.

To get an equation for the evolution of both plasma species we invoke the quasi-neutrality condition of a plasma. As long as we look at spatial scales larger than one Debye-length this is valid, as Debye-shielding will neutralize any charge buildup. This then gives us $n = n_e \approx n_i$ and we only need one equation for the evolution of the density.

To get an evolution equation for the velocities we introduce the equation of charge conservation given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \boldsymbol{j} = 0 \tag{2.14}$$

where ρ is the spatial charge density summed over all species. Given the quasi-neutrality condition ρ is neglected and we get an equation for conservation of charge. We then have $\nabla \cdot \boldsymbol{j} = 0$ where $\boldsymbol{j} = en(\boldsymbol{u}_i - \boldsymbol{u}_e)$ and this gives us

$$\begin{aligned} &\nabla \cdot \left[en \left(\boldsymbol{u}_E + \boldsymbol{u}_{di} + \boldsymbol{u}_{pi} - \boldsymbol{u}_E - \boldsymbol{u}_{de} - \boldsymbol{u}_{pi} \right) \right] \\ = &-\nabla \cdot \left[en \left(\boldsymbol{u}_{i\parallel} - \boldsymbol{u}_{e\parallel} + \boldsymbol{u}_{vi} - \boldsymbol{u}_{ve} + \boldsymbol{u}_{\pi i} \right) \right], \end{aligned}$$

where we have included the parallel, collisional and viscosity drifts for them selves. Again neglecting electron inertia and also invoking the cold ion approximation we can reduce this equation into

$$\begin{aligned} \nabla \cdot \left[en \left(\boldsymbol{u}_{pi} - \boldsymbol{u}_{de} \right) \right] &= -\nabla \cdot \left[en \left(\boldsymbol{u}_{i\parallel} - \boldsymbol{u}_{e\parallel} + \boldsymbol{u}_{vi} - \boldsymbol{u}_{ve} - e\nabla \left(n\boldsymbol{u}_{\pi i} \right) \right) \right] \\ e\nabla \cdot \left(n\boldsymbol{u}_{pi} \right) + T_e \mathcal{K}(n) &= -e\nabla \cdot \left(n\boldsymbol{u}_{i\parallel} - n\boldsymbol{u}_{e\parallel} \right) - e\nabla \left(n\boldsymbol{u}_{\pi i} \right). \end{aligned} \tag{2.15}$$

here we have removed the collsional term as the momentum conservation implies that the two cancel each other out. While using the cold ion approximation there technically is no viscosity the ion viscous term is needed for numerical stability in our simulations and is therefor included.

We again get the curvature operator of the diamagnetic drift for the electrons. The polarisation term as well as the terms on the RHS requires some more investigation.

2.4 Polarisation equation

In this section, we treat the polarisation equation and also discuss and derive a requirement for the Boussinesq approximation to be applicable.

The polarisation term can be expanded as follows

$$\nabla \cdot (n\boldsymbol{u}_{pi}) = \nabla \cdot \left(\frac{nm_i}{q_i B} \hat{\boldsymbol{b}} \times \frac{\mathrm{d}\boldsymbol{u}_E}{\mathrm{d}t} \right) \tag{2.16}$$

$$= \frac{m_i}{q_i} \left(\frac{\nabla n}{B} \cdot \hat{\boldsymbol{b}} \times \frac{\mathrm{d}\boldsymbol{u}_E}{\mathrm{d}t} - \frac{n \nabla \ln B}{B} \cdot \hat{\boldsymbol{b}} \times \frac{\mathrm{d}\boldsymbol{u}_E}{\mathrm{d}t} \right) + \frac{m_i n}{q_i B} \left[(\nabla \times \boldsymbol{b}) \cdot \frac{\mathrm{d}\boldsymbol{u}_e}{\mathrm{d}t} - \boldsymbol{b} \cdot \left(\nabla \times \frac{\mathrm{d}\boldsymbol{u}_e}{\mathrm{d}t} \right) \right]. \tag{2.17}$$

For tokamak plasmas the main magnetic field varies on a much larger scale than for the dynamic scale we are interested in, such that $O(\nabla \ln B) \sim O(\nabla \times B) \ll O(\nabla \phi)$. Therefore we can neglect the second and third terms in equation 2.17. Also using the vector identity $\hat{\boldsymbol{b}} \times \left(\hat{\boldsymbol{b}} \times \nabla \phi \right) = -\nabla_\perp \phi$ we can transform the equations into

$$\nabla \cdot (n\boldsymbol{u}_p) = -\frac{m_i n}{q_i B^2} \left(\nabla \ln n \cdot \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\perp \phi + \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\perp^2 \phi \right). \tag{2.18}$$

We can simplify the polarisation drift even further by applying the Boussinesq approximation. The Boussinesq approximation [19] first appeared in the context of fluid dynamics in a paper by Oberbeck [20], and so the approximation is also called the Oberbeck-Boussinesq approximation. Its original use was to reduce the complexity of fluid equations by assuming that the particle density of a fluid only had a spacial variance. This allows one to break the particle density into a background density such that the spatial variance is a small fluctuation around this background. We will use this same argument to reduce the polarisation equation. Applying the Boussinesq approximation to equation 2.18 the $\nabla \ln n$ term will become negligibly small and we thus reduce the polarisation drift to be

$$\nabla \cdot (nu_p) = -\frac{m_i}{q_i B^2} \left(n \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\perp^2 \phi \right). \tag{2.19}$$

This approximation is useful in multiple ways. This neglects a cubic nonlinearity which makes analytical and numerical treatment of the term simpler. This reduction of complexity makes simulations run faster which is why it is in common use.

In order to find when the Boussinesq approximation is applicable we expand the density into a background n_0 and a fluctuation part $\tilde{n}$ as $n = n_0 + \tilde{n}$. This allows us to expand the first term of equating 2.18 into

$$\left[\nabla \ln n_0 + \nabla \ln \left(1 + \frac{\tilde{n}}{n_0} \right) \right] \cdot \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\perp \phi \tag{2.20}$$

where the two terms in the square bracket scale as

$$\nabla \ln n_0 \sim \frac{1}{L_n}$$
$$\nabla \ln \left(1 + \frac{\tilde{n}}{n_0}\right) \sim k_\perp \frac{\tilde{n}}{n_0}$$

where $k_\perp$ is a characteristic perpendicular wave number for the particle density fluctuations.

Writing out the order of magnitude for each of the terms in the polarisation equation 2.45 and requiring that the last term is greater than the two others we get

$$\frac{1}{L_n} + k_\perp \frac{\tilde{n}}{n_0} \ll k_\perp$$
$$\frac{1}{k_\perp L_n} + \frac{\tilde{n}}{n_0} \ll 1.$$

From here we get the requirements that need to be fulfilled. We need to be in a scenario where the characteristic length of the equilibrium particle density is larger than the perpendicular wavelength of the particle density fluctuations. Moreover, the particle density fluctuation amplitude $\tilde{n}$ has to be smaller than the background density n_0. Neither of the requirement is shown to hold in the SOL as there have been observed order unity fluctuations in density in that region [11, 13], and thus this approximation should not hold in our case.

This will be part of the investigation done in this thesis, to see which effect including this approximation has on the statistical properties of the plasma fluctuations.

2.5 Parallel and collisional closure

We will now look at the collisional and parallel closures for the model. Let us start by considering the parallel closure.

In the SOL the plasma will interact with the walls and divertor targets of the fusion reactor. We describe this interaction with the divertors in our parallel closure. As the fusion reactor is starting up the wall has neutral charge, further on in the process it will start to receive an influx of particles. There will be a larger amount of electrons that will interact with the wall first owing to their greater velocity thus making the walls gain negative charge. This will

create a potential barrier called the sheath, giving us the sheath boundary condition [21]:

$$\left\langle \nabla \cdot \left(n \boldsymbol{u}_{e\parallel} \right) \right\rangle_{\parallel} = \frac{n}{L_{\parallel}} u_{e\parallel} = \frac{n}{L_{\parallel}} C_s \exp\left(\Lambda_{sh} - \frac{e\phi}{T_e} \right), \tag{2.21}$$

where we take the parallel average over the field lines such that $L_{\parallel}$ is the parallel connection length defined as the distance between the divertors following the magnetic field lines. C_s is the acoustic speed and $\Lambda_{sh} = \ln \sqrt{m_i / 2\pi m_e}$ is the potential that is induced by the sheath, called the sheath potential. As the wall is negatively charged this does not impede the ions which is assumed to move with acoustic speed

$$\left\langle \nabla \cdot \left(n \boldsymbol{u}_{i\parallel} \right) \right\rangle_{\parallel} = \frac{n}{L_{\parallel}} C_s. \tag{2.22}$$

This assumes that we have flute modes, meaning that there is no perpendicular change in plasma parameters while following the magnetic field lines from divertor to divertor.

As we also need to model the edge region we need to handle a parallel closure here as well. In the edge region we assume that ions are stationary compared to the fast moving electrons and following the magnetic field lines electrons are isothermal and in force balance such that

$$\boldsymbol{u}_{i\parallel} = 0 \tag{2.23}$$

$$\frac{\mathrm{d}\boldsymbol{u}_e}{\mathrm{d}t} = 0 \tag{2.24}$$

this then allows us to get an expression for the parallel electron velocity:

$$\begin{aligned} \boldsymbol{b} \cdot \left| m_e n_e \left(\frac{\partial}{\partial t} + \boldsymbol{u}_e \cdot \nabla \right) \boldsymbol{u}_e = -T_e \nabla n + n_e q_e \left(\boldsymbol{E} + \boldsymbol{u}_e \times \boldsymbol{B} \right) \right. \\ - m_e n_e \nu_{ei} \boldsymbol{u}_e \\ \boldsymbol{u}_{e\parallel} = \frac{e \nabla_{\parallel} \phi - T_e \nabla_{\parallel} \ln n}{m_e \nu_{ei}}, \end{aligned} \tag{2.25}$$

notice the inclusion of the collisional term in the momentum equation. By including the collisional term we get an equation for parallel velocity, by not including it we would get an equation relating density with potential. Using this velocity in our expression for the parallel term we get:

$$\begin{aligned} \nabla \cdot \left(n \boldsymbol{u}_{e\parallel} \right) &= \nabla \cdot \left(\frac{n T_e}{m_e \nu_{ei}} \nabla_{\parallel} \left(\frac{e\phi}{T_e} - \ln n \right) \right) \\ &= \frac{n T_e}{m_e \nu_{ei}} \nabla_{\parallel}^2 \left(\frac{e\phi}{T_e} - \ln n \right), \end{aligned} \tag{2.26}$$

here the collision frequency defined as

$$\nu_{ei} = \frac{\log \Lambda_c e^4 n}{6\sqrt{2}\pi^{3/2}\epsilon_0^2\sqrt{m_e}T_e^{3/2}} \tag{2.27}$$

where $\log \Lambda_c$ is the Coulomb logarithm and ϵ_0 is the vacuum permittivity [22].

We will now consider the effect of perpendicular friction. The collisional drift between two fluid species can be derived to be:

$$\boldsymbol{u}_{vs\perp} = \frac{m_s}{q_s B}\boldsymbol{b} \times \nu_{ss'}(\boldsymbol{u}_s - \boldsymbol{u}_{s'}) \tag{2.28}$$

where s' is the particle spices the particles s collides with. As we can see the collisional drift dependents on the velocities both particle species. To tackle this we consider only the lowest order drifts, again using the cold ion approximation and neglecting polarization drifts to lowest order we get for the electron collisions

$$\begin{aligned}\boldsymbol{u}_{ve\perp} &= \frac{m_e\nu_{ei}}{q_e B}\boldsymbol{b} \times (\boldsymbol{u}_e - \boldsymbol{u}_i) \\ &= \frac{m_e\nu_{ei}}{q_e B}\boldsymbol{b} \times (\boldsymbol{u}_E + \boldsymbol{u}_{de} - \boldsymbol{u}_E) \\ &= \frac{m_e\nu_{ei}}{q_e B}\boldsymbol{b} \times \boldsymbol{u}_{de}.\end{aligned} \tag{2.29}$$

Using this as our collisional drift we then get that the collisional diffusion is given by

$$\begin{aligned}\nabla \cdot (n\boldsymbol{u}_{ve}) &= \nabla \cdot \left(n\frac{m_e\nu_{ei}}{q_e B}\boldsymbol{b} \times \frac{T_e}{q_e n_e B}\boldsymbol{b} \times \nabla n \right) \\ &\approx -\frac{\nu_{ei}m_e T_e}{q_e^2 B^2}\nabla_\perp^2 n = -\frac{\nu_{ei}m_e C_s^2}{\omega_c^2 m_i}\nabla_\perp^2 n.\end{aligned} \tag{2.30}$$

where ω_c is the ion gyro-frequency. To preserve momentum the collisional drifts must cancel each other out an is therefore not present in the current conservation equation.

The viscosity term is one of the most complicated terms to deal with, we will therefore use the approximation

$$\nabla \cdot \boldsymbol{\pi} \approx \eta_\perp \nabla_\perp^2 \boldsymbol{u}_e + \eta_\parallel \nabla_\parallel^2 \boldsymbol{u}_e$$

where $\eta_\perp$ and $\eta_\parallel$ is the perpendicular and parallel viscosity constants. A more proper discussion of viscosity is highly involved and can be found in [22].

Using the above mentioned approximation we can find an expression for the viscous diffusion term

$$\nabla \cdot (n\boldsymbol{u}_{\pi i}) = \nabla \cdot \left(\frac{m_i n}{q_i B} \boldsymbol{b} \times \nabla \cdot \boldsymbol{\pi}_i \right) \approx \frac{m_i n}{q_i B} \eta_\perp \nabla_\perp^4 \phi, \tag{2.31}$$

where we again use flute modes to neglect the parallel part of the viscous term.

2.6 Normalization

Combining the equations 2.13, 2.15, 2.30, 2.31 and the parallel closure from 2.26 we get the set of equations for the edge region:

$$\frac{\mathrm{d}n}{\mathrm{d}t} + n\mathcal{K}(\phi) - \frac{T_e}{e}\mathcal{K}(n) = \frac{T_e n}{m_e \nu_{e,i}} \nabla_\parallel^2 \left(\ln n - \frac{e\phi}{T_e} \right) + \frac{\nu_{ei} m_e C_s^2}{\omega_c^2 m_i} \nabla_\perp^2 n \tag{2.32a}$$

$$\frac{m_i n}{eB^2} (\nabla \ln n + \nabla) \cdot \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\perp \phi - \frac{T_e}{e}\mathcal{K}(n) = \frac{T_e n}{m_e \nu_{e,i}} \nabla_\parallel^2 \left(\ln n - \frac{e\phi}{T_e} \right) + \frac{m_i n}{eB} \eta_\perp \nabla_\perp^4 \phi, \tag{2.32b}$$

Again using the same equations except for now utilizing the parallel closure from equations 2.21 and 2.22 we get the set of equations describing the SOL:

$$\frac{\mathrm{d}n}{\mathrm{d}t} + n\mathcal{K}(\phi) - \frac{T_e}{e}\mathcal{K}(n) = -\frac{n}{L_\parallel} C_s \exp\left(\Lambda_{sh} - \frac{e\phi}{T_e} \right) + \frac{\nu_{ei} m_e C_s^2}{\omega_c^2 m_i} \nabla_\perp^2 n \tag{2.33a}$$

$$\frac{m_i n}{eB^2} (\nabla \ln n + \nabla) \cdot \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\perp \phi - \frac{T_e}{e}\mathcal{K}(n) = \frac{n}{L_\parallel} C_s \left[1 - \exp\left(\Lambda_{sh} - \frac{e\phi}{T_e} \right) \right] + \frac{m_i n}{eB} \eta_\perp \nabla_\perp^4 \phi \tag{2.33b}$$

As it stands the equations above are given in any arbitrary coordinate system. As mentioned earlier we are looking at at the outboard mid-plane with slab coordinates. To transform our equations into this geometry let us first consider a purely radial magnetic field in toroidal coordinates:

$$\boldsymbol{B} = \frac{B_0 R_0}{R} \boldsymbol{b}. \tag{2.34}$$

Here B_0 and R_0 are the magnetic field strength at the midpoint of the plasma and the distance form the center of the torus to the middle of the confined plasma. Putting this into equation for the curvature operator we then get for the outboard mid-plane:

$$\mathcal{K}(f) = -\frac{2}{B_0 R_0}\frac{\partial f}{\partial y}. \tag{2.35}$$

We will also introduce a commonly used quantity in fluid dynamics called vorticity. This is used to describe the rotation of fluids, and for a plasma, it is commonly defined as:

$$\begin{aligned}\Omega &= \boldsymbol{b}\cdot\nabla\times\boldsymbol{u}_E \\ &= \boldsymbol{b}\cdot\left[\frac{\boldsymbol{b}}{B}\left(\nabla\cdot\nabla_\perp\phi\right) - \nabla_\perp\phi\left(\nabla\cdot\frac{\boldsymbol{b}}{B}\right) + \left(\nabla_\perp\phi\cdot\nabla\right)\frac{\boldsymbol{b}}{B} - \left(\frac{\boldsymbol{b}}{B}\cdot\nabla\right)\nabla_\perp\phi\right] \\ &\approx \frac{1}{B}\nabla_\perp^2\phi,\end{aligned} \tag{2.36}$$

where the last approximation holds in cases where the magnetic field changes on larger scales than the electrostatic potential giving us $O(\nabla B) \ll O(\nabla\phi)$ allowing us to neglect the terms involving the gradient of the magnetic field.

With these two changes the equations 2.32 and 2.33 becomes

$$\frac{\mathrm{d}n}{\mathrm{d}t} - \frac{2n}{B_0R_0}\frac{\partial\phi}{\partial y} + \frac{2T_e}{eB_0R_0}\frac{\partial n}{\partial y} = \frac{T_e n}{m_e\nu_{e,i}}\nabla_\parallel^2\left(\ln n - \frac{e\phi}{T_e}\right) + \frac{\nu_{ei}m_eC_s^2}{\omega_c^2 m_i}\nabla_\perp^2 n \tag{2.37a}$$

$$\frac{m_i n}{eB^2}\left(\nabla\ln n\cdot\frac{\mathrm{d}}{\mathrm{d}t}\nabla_\perp\phi + \frac{\mathrm{d}\Omega}{\mathrm{d}t}\right) + \frac{2T_e}{eB_0R_0}\frac{\partial n}{\partial y} = \frac{T_e n}{m_e\nu_{e,i}}\nabla_\parallel^2\left(\ln n - \frac{e\phi}{T_e}\right) + \frac{m_i n}{eB}\eta_\perp\nabla_\perp^2\Omega, \tag{2.37b}$$

and

$$\frac{\mathrm{d}n}{\mathrm{d}t} - \frac{2n}{B_0R_0}\frac{\partial\phi}{\partial y} + \frac{2T_e}{eB_0R_0}\frac{\partial n}{\partial y} = -\frac{n}{L_\parallel}C_s\exp\left(\Lambda_{sh} - \frac{e\phi}{T_e}\right) + \frac{\nu_{ei}m_eC_s^2}{\omega_c^2 m_i}\nabla_\perp^2 n \tag{2.38a}$$

$$\frac{m_i n}{eB^2}\left(\nabla\ln n\cdot\frac{\mathrm{d}}{\mathrm{d}t}\nabla_\perp\phi + \frac{\mathrm{d}\Omega}{\mathrm{d}t}\right) + \frac{2T_e}{eB_0R_0}\frac{\partial n}{\partial y} = \frac{n}{L_\parallel}C_s\left[1 - \exp\left(\Lambda_{sh} - \frac{e\phi}{T_e}\right)\right] + \frac{m_i n}{eB}\eta_\perp\nabla_\perp^2\Omega. \tag{2.38b}$$

For the purposes of numerical simulation we would like to have the equations in a non-dimensional form we will therefore apply the so-called Bohm

normalization to the equations. Using the following normalization:

$$\omega_c t \to \hat{t}, \qquad \frac{T_e}{T_0} \to \hat{T} \qquad \frac{e\phi}{T_0} \to \hat{\phi}, \qquad \frac{x}{\rho_s} \to \hat{x}, \qquad \frac{n}{n_0} \to \hat{n},$$

where $\rho_s = \sqrt{m_i T_0}/eB$, then the the above equation become

$$\frac{\mathrm{d}n}{\mathrm{d}t} + g\left(\frac{\partial n}{\partial y} - n\frac{\partial \phi}{\partial y}\right) = \chi_0\left(\phi - \ln n\right) + D_\perp \nabla_\perp^2 n \tag{2.39a}$$

$$\nabla \ln n \cdot \frac{\mathrm{d}}{\mathrm{d}t}\nabla_\perp \phi + \frac{\mathrm{d}\Omega}{\mathrm{d}t} + \frac{g}{n}\frac{\partial n}{\partial y} = \chi_0\left(\phi - \ln n\right) + \nu_\perp \nabla_\perp^2 \Omega, \tag{2.39b}$$

and

$$\frac{\mathrm{d}n}{\mathrm{d}t} + g\left(\frac{\partial n}{\partial y} - n\frac{\partial \phi}{\partial y}\right) = -\sigma_0 n \exp(\Lambda_{sh} - \phi) + D_\perp \nabla_\perp^2 n \tag{2.40a}$$

$$\nabla \ln n \cdot \frac{\mathrm{d}}{\mathrm{d}t}\nabla_\perp \phi + \frac{\mathrm{d}\Omega}{\mathrm{d}t} + \frac{g}{n}\frac{\partial n}{\partial y} = \sigma_0\left[1 - \exp(\Lambda_{sh} - \phi)\right] + \nu_\perp \nabla_\perp^2 \Omega. \tag{2.40b}$$

where we have dropped the hat as all values are normalized and defined $g = 2\rho_s/R$, commonly referred to as effective gravity. This is because the curvature of the magnetic field will in this case act as something similar to a gravitational field wanting to drag particles inward. In addition we have introduced the sheath parameter σ_0 as

$$\sigma_0 = \frac{\rho_s}{L_\parallel}, \tag{2.41}$$

and the conductivity χ_0 as

$$\chi_0 = \left(\frac{\rho_s}{L_\parallel}\right)^2 \frac{m_i}{m_e}\frac{\omega_c}{\nu_{ei}}, \tag{2.42}$$

where the we have used $\nabla_\parallel^2 \to -k_\parallel^2 \approx -L_\parallel^{-2}$. We also defined the collisional diffusion of $D_\perp = \nu_{ei} m_e/\omega_c \rho_s^2 m_i$ and collisional diffusion due to viscosity as $\nu_\perp = \eta_\perp/\omega_c \rho_s^2$.

2.7 Model Equations

To combine the two sets of equations into one set of equation we redefine the sheath parameter and the conductivity parameter to have a radial dependence such that

$$\sigma(x) = \frac{\sigma_0}{2}\left\{1 + \tanh\left[w\left(x - x_{\mathrm{SOL}}\right)\right]\right\}, \tag{2.43}$$

$$\chi(x) = \chi_0\left(1 - \frac{1}{2}\left\{1 + \tanh\left[w\left(x - x_{\mathrm{SOL}}\right)\right]\right\}\right). \tag{2.44}$$

This will create two regions separated by the LCFS whose location is defined by x_{SOL}, where the width of the transition between the two regions is decided by the w parameter. Adding this transition to the parameters allows us to define one set of equations, this will be one of the main models used in our simulations and throughout the thesis, we will refer to it as the **Full n** model given as:

$$\frac{\mathrm{d}n}{\mathrm{d}t} + g\left(\frac{\partial n}{\partial y} - n\frac{\partial \phi}{\partial y}\right) = \Sigma_n(x) + D_\perp \nabla_\perp^2 n - \sigma(x) n \exp\left(\Lambda_{sh} - \phi\right) + \chi(x)\left(\tilde{\phi} - \ln(\tilde{n})\right) \tag{2.45a}$$

$$\frac{\mathrm{d}\nabla_\perp^2 \phi}{\mathrm{d}t} + \nabla \ln n \cdot \frac{\mathrm{d}}{\mathrm{d}t}\nabla_\perp \phi + \frac{g}{n}\frac{\partial n}{\partial y} = \nu_\perp \nabla_\perp^4 \phi + \sigma(x)\left[1 - \exp\left(\Lambda_{sh} - \phi\right)\right] + \chi(x)\left(\tilde{\phi} - \ln(\tilde{n})\right). \tag{2.45b}$$

The first term on the right hand side of equation 2.45a is the plasma source term defined as

$$\Sigma_n(x) = \Sigma_0 \exp\left(-\frac{(x - x_0)^2}{\lambda_s^2}\right), \tag{2.46}$$

where Σ_0 is the amplitude of the source, x_0 the location and λ_s is the e-folding length of the source. The values $\tilde{\phi}$ and $\tilde{n}$ is the fluctuating part of the plasma potential and electron density defined by $\tilde{\psi} = \psi - \langle\psi\rangle_y$ where $\langle\psi\rangle_y$ is the flux surface average.

In addition to this base model, we will also be exploring some variations on this model. One such variation is linearizing the density in edge term, such that the driftwave term now becomes:

$$\chi(x)\left(\tilde{\phi} - \tilde{n}\right). \tag{2.47}$$

Applying the linearization we then have the model we will refer to as the **non Bussinesq** model:

$$\frac{\mathrm{d}n}{\mathrm{d}t} + g\left(\frac{\partial n}{\partial y} - n\frac{\partial \phi}{\partial y}\right) = \Sigma_n(x) + D_\perp \nabla_\perp^2 n - \sigma(x) n \exp\left(\Lambda_{sh} - \phi\right) + \chi(x)\left(\tilde{\phi} - \tilde{n}\right) \tag{2.48a}$$

$$\frac{\mathrm{d}\nabla_\perp^2 \phi}{\mathrm{d}t} + \nabla \ln n \cdot \frac{\mathrm{d}}{\mathrm{d}t}\nabla_\perp \phi + \frac{g}{n}\frac{\partial n}{\partial y} = \nu_\perp \nabla_\perp^4 \phi + \sigma(x)\left[1 - \exp\left(\Lambda_{sh} - \phi\right)\right] + \chi(x)\left(\tilde{\phi} - \tilde{n}\right), \tag{2.48b}$$

The second variation we can do to the equations is applying the Boussinesq approximation. This will neglect the term

$$\nabla \ln n \cdot \frac{\mathrm{d}}{\mathrm{d}t} \nabla_\perp \phi, \tag{2.49}$$

that originates from the polarization equation. Applying just the Boussinesq approximation give us the **Boussinesq log dw** model:

$$\begin{aligned} \frac{\mathrm{d}n}{\mathrm{d}t} + g\left(\frac{\partial n}{\partial y} - n\frac{\partial \phi}{\partial y}\right) &= \Sigma_n(x) + D_\perp \nabla_\perp^2 n - \sigma(x) n \exp\left(\Lambda_{sh} - \phi\right) \\ &\quad + \chi(x)\left(\tilde{\phi} - \ln(\tilde{n})\right) \end{aligned} \tag{2.50a}$$

$$\begin{aligned} \frac{\mathrm{d}\nabla_\perp^2 \phi}{\mathrm{d}t} + \frac{g}{n}\frac{\partial n}{\partial y} &= \nu_\perp \nabla_\perp^4 \phi + \sigma(x)\left[1 - \exp\left(\Lambda_{sh} - \phi\right)\right] \\ &\quad + \chi(x)\left(\tilde{\phi} - \ln(\tilde{n})\right), \end{aligned} \tag{2.50b}$$

this uses the Boussinesq approximation but keeps the logarithm of the density in the drift wave term.

Finally if we apply the Boussinesq approximation and linearize the density term inside the drift wave term we end up with a model that has previously been used to simulate an edge plus SOL plasma by Sarazin *et al.* [23–25], Garcia *et al.* [26–28], Myra *et al.* [29–31], Bisai *et al.* [32–34], Nielsen *et al.* [35–37], and Decristoforo *et al.* [38]. This will be referred to as the **Boussinesq** model:

$$\frac{\mathrm{d}n}{\mathrm{d}t} + g\left(\frac{\partial n}{\partial y} - n\frac{\partial \phi}{\partial y}\right) = \Sigma_n(x) + D_\perp \nabla_\perp^2 n - \sigma(x) n \exp\left(\Lambda_{sh} - \phi\right) + \chi(x)\left(\tilde{\phi} - \tilde{n}\right) \tag{2.51a}$$

$$\frac{\mathrm{d}\nabla_\perp^2 \phi}{\mathrm{d}t} + \frac{g}{n}\frac{\partial n}{\partial y} = \nu_\perp \nabla_\perp^4 \phi + \sigma(x)\left[1 - \exp\left(\Lambda_{sh} - \phi\right)\right] + \chi(x)\left(\tilde{\phi} - \tilde{n}\right). \tag{2.51b}$$

These will be the four main models used throughout the thesis and will be referred to by the same name as given above.

3 Numerical simulations

In this chapter, we will introduce and discuss the different numerical simulations done for this thesis.

3.1 The BOUT++ code

In this section, we will introduce the numerical framework used for the simulations that have been done for this thesis. All the simulations that will be presented in this thesis have used BOUT++ [39]. BOUT++ (BOUndary Turbulence in C++) is a framework for solving non-linear equations in 2D and 3D curvelinear coordinate systems. It is primarily used for plasma fluid simulations but can be used to solve general time-dependent equations as well. The code is open source and can be found at `https://github.com/boutproject/BOUT-dev`. The advantage of using a preexisting library is the reduction of significant development time that has to go into creating a new and usable code for simulation purposes. BOUT++ already contains various methods and solvers that already have been tested for parallel processing, numerical stability and convergence. Adding on the fact that it has a large user base most potential issues have likely been identified and solved.

The usage of BOUT++ is also quite simple, once you have a compiled version of the software you require one main file in which the code initialisation and equations of the simulation are given and a directory */data* with a file that

gives the input parameters. This is also where the output from the simulations is stored. The syntax for equations in the main file is made so that is as easy as possible to implement. For example:

```
ddt(vort) = -bracket(phi,vort,bm) + (1.0/n) * g0 * DDZ(n);
```

describes the equation

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t} = -\{\phi, \Omega\} + \frac{g_0}{n}\frac{\partial n}{\partial z} \tag{3.1}$$

where the last input in the bracket function can be used to decide which solver should be used for the poisson bracket.

This independent choice of solver also extends to solving the entirety of equations, as BOUT++ allows for a choice in spatial solver, time integrator, and even how the scheme for differentiation works independent of coordinates. This allows for systems in which there is a centre-difference scheme of order 2 in one coordinate direction and an order 4 scheme in another direction. This freedom of choice is quite useful as different problems require different solvers. For instance, in our simulations to solve for the electric potential using Laplacian inversion, we use the BOUT++ cyclic solver for the Bousinessq approximated cases and the BOUT++ implementation of PETSc solvers for the non-Bousinessq case. Also changing which solver you use only requires a change in the input file so there is no need to rewrite the whole code if the initially chosen solvers are unfit. The model used in our simulations is a version of the STORM module for BOUT++, it has previously been used to simulate turbulence in an edge plus SOL domain [38, 40]. To solve our specific system we use a finite difference method in the radial direction and a spectral method in the poloidal direction.

As an example of how a setup for a BOUT++ main file looks, we will take a look at the main file for the simulations that we are going to run.

```
1 #include <bout/physicsmodel.hxx>
2 #include <derivs.hxx>
3 #include <invert_laplace.hxx>
4 #include <bout/constants.hxx>
5 #include <initialprofiles.hxx>
6 #include "fast_output.hxx"
```

This first part imports all the necessary libraries that will be in use. The first line includes the main functionality of the BOUT++ library, such as the **PhysicsModel** class. The rest of the lines include useful functionality. Of note here is the *fast_output.hxx* which is a nonstandard library but part of the STORM

model of BOUT++. This defines the functionality for the virtual probes that are implemented.

The next part will initialize the physics model as a child of the **PhysicsModel** class. Lines 12 and 13 initialise the functions that will be used in the class, while 14-24 does error handling on the probe method. The lines of interest in this first section come at 27-54, here we define the variables that are going to be used in the simulation. Lines 56-62 are some switches to change the physics that are included in the model, of interest to us is the *Boussinesq* and *dw_logn* terms. These change whether we use the Boussinesq approximation in our model or not and if the resistive term contains a linear or logarithmic density.

```
 8 class DriftFluidModel : public PhysicsModel
 9 {
10
11 protected:
12   int init(bool restarting);
13   int rhs(BoutReal t);
14   int timestepMonitor(BoutReal simtime,
15                       BoutReal UNUSED(dt))
16   {
17     int ret = 0;
18     if (fast_output.enable_timestep)
19     {
20       ret = fast_output.monitor_method(simtime);
21       if (ret)
22         // return if ret is non-zero (indicating error)
23         return ret;
24     }
25     return ret;
26   }
27
28 private:
29   // Evolving density, vorticity and potential
30   Field3D n, vort, logn, phi;
31   //Calculating radial and poloidal velocity
32   Field3D vx,vz;
33   // Density sources
34   Field2D S;
35   Field3D dw;
36   // Communicated variables
37   FieldGroup comms;
38   // Bracket method for advection terms
39   BRACKET_METHOD bm;
```

```
40    // sheath dissipation parameter
41    BoutReal lambda_n, lambda_vort, L;
42    // Seed for turbulence initialization
43    int turbulence_seed;
44    // Sets location of separatrix
45    BoutReal ixsep;
46    // represents size of dissipation terms
47    Field2D loss;
48    // width of transition region between core and SOL
49    BoutReal trans_width;
50
51    // Variables and functions used for parallel loss terms
52    BoutReal mu_n, mu_vort, g0, m_i, u, m_e, mu_0, alpha_dw;
53    // floating potential for the nonisothermal sheath loss terms
54    BoutReal mu, V_float;
55    // switch to add random noise to density and vorticity
56    bool initial_noise;
57    // switch to include term representing drift-wave physics in core
58    bool driftwaves;
59    // use the Boussinesq approximation in vorticity
60    bool boussinesq;
61    // use logn instead of n in Hasegawa-Wakatani term
62    bool dw_logn;
63
64    // Laplacian solver for vort -> phi
65    class Laplacian *phiSolver;
66
67    // Object to handle fast output
68    FastOutput fast_output;
69 };
```

In this code block, we define what will happens when the simulation is initialized. The **OPTION** keyword in lines 77-90 looks up the values of the given name in the input file, if none is found the default value given as the last parameter is used. Lines 105-128 set the transition between the edge region and the scrape-off layer. Then in lines 137-145, we define first the values that are going to be solved for, in this case: logarithmic density and vorticity, then we also set up for which other values are going to be saved into the output files. **SAVE_REPEAT** writes every time step while **SAVE_ONCE** writes the values at initialization. Lines 148-160 set the solver used for getting the potential from the vorticity depending on whether the Boussinesq approximation is used or not. The rest of the lines initialize the density noise for instability and the handling of the probes.

```
71  int DriftFluidModel::init(bool UNUSED(restart))
72  {
73
74    Options *options = Options::getRoot()
75                         ->getSection("model");
76
77    OPTION(options, mu_n, -1.0);
78    OPTION(options, mu_vort, -1.0);
79    OPTION(options, g0, -1.0);
80    OPTION(options, m_i, 2); // Atomic Units
81    OPTION(options, lambda_n, 5e-4);
82    OPTION(options, lambda_vort, 5e-4);
83
84    OPTION(options, ixsep, -1);
85    OPTION(options, initial_noise, false);
86    OPTION(options, driftwaves, false);
87    OPTION(options, dw_logn, false);
88    OPTION(options, boussinesq, true);
89    OPTION(options, alpha_dw, 1e-4);
90    OPTION(options, turbulence_seed, 0);
91
92    // Specify Constants
93    u = 1.66053892e-27;   // kg
94    m_e = 9.10938291e-31; // kg
95    mu_0 = 4. * PI * 1.e-7;
96
97    // Convert Parameters to SI units
98    m_i = m_i * u; // kg
99
100   // Floating potential (only used for nonisothermal cases)
101   mu = m_i / m_e;
102   V_float = 0.5 * log(TWOPI / mu);
103
104   // Include position of separatrix by weighting effect of SOL closure terms
105   if (ixsep > -1)
106   {
107     loss.allocate();
108     // Set width of transition region from core to SOL
109     trans_width = 2.0 * mesh->getCoordinates()
110                           ->dx(mesh->xstart, mesh->ystart);
111     for (int ix = 0; ix < mesh->LocalNx; ix++)
112     {
113       for (int iy = 0; iy < mesh->LocalNy; iy++)
```

```
114         {
115           for (int iz = 0; iz < mesh->LocalNz; iz++)
116           {
117             loss(ix, iy, iz) = (0.5 * (1.0
118                                 + tanh(trans_width
119                                 * (mesh->getGlobalXIndex(ix)
120                                 - ixsep - 0.5))));
121           }
122         }
123       }
124     }
125     else
126     { // Whole domain is SOL
127       loss = 1.0;
128     }
129
130     initial_profile("S", S);
131
132     // Poisson brackets: b_hat x Grad(f) dot Grad(g) / B = [f, g]
133     // Method to use: BRACKET_ARAKAWA
134     // Choose method to use for Poisson bracket advection terms
135     bm = BRACKET_ARAKAWA;
136
137     SOLVE_FOR2(logn, vort);
138     SAVE_REPEAT(n);
139     SAVE_REPEAT(phi);
140     SAVE_REPEAT(vx) ;
141     SAVE_REPEAT(vz) ;
142     SAVE_ONCE(S);
143     SAVE_ONCE(loss);
144     SAVE_ONCE(mu_n);
145     SAVE_ONCE(mu_vort);
146
147
148     if (boussinesq)
149     {
150       // BOUT.inp section "phiBoussinesq"
151       phiSolver = Laplacian::create(Options::getRoot()
152                                     ->getSection("phiBoussinesq"));
153     }
154     else
155     {
156       // BOUT.inp section "phiSolver"
```

```
157     phiSolver = Laplacian::create(Options::getRoot()
158                                   ->getSection("phiSolver"));
159   }
160   phi = 0.0; // Starting guess for first solve (if iterative)
161
162   // Initialise the fields
163   initial_profile("n", n);
164
165   comms.add(logn);
166   comms.add(vort);
167
168   // Seed turbulence with random noise
169   if (initial_noise)
170   {
171     output << "\tSeeding random noise
172                for triggering turbulent instabilities\n";
173     output.write("\tAdded %i as seed\n", turbulence_seed);
174     srand(turbulence_seed);
175     //mpisrand (turbulence_seed);
176     for (int i = 0; i < mesh->LocalNx; i++)
177     {
178       for (int k = 0; k < mesh->LocalNz; k++)
179       {
180         n(i, 0, k) += 2. * (((double)rand()
181                         /(RAND_MAX)) - 0.5) * 0.0001;
182         //vort(i,0,k) += 2.*(((double) rand()/(RAND_MAX)) - 0.5)*0.00001;
183       }
184     }
185   }
186
187   // Read boundary conditions
188   logn.setBoundary("n");
189
190   logn = log(n);
191
192   // Set up output for synthetic Langmuir probe trace
193   if (fast_output.enabled)
194   {
195
196     // Add monitor if necessary
197     if (fast_output.enable_monitor)
198     {
199       solver->addMonitor(&fast_output);
```

```
200     }
201
202     // Add points from the input file
203     int i = 0;
204     BoutReal xpos, zpos;
205     int ix, iy, iz;
206     Options *fast_output_options = Options::getRoot()
207                                    ->getSection("fast_output");
208     while (true)
209     {
210       // Add more points if explicitly set in input file
211       fast_output_options->get("xpos" + std::to_string(i), xpos, -1.);
212       fast_output_options->get("zpos" + std::to_string(i), zpos, -1.);
213       if (xpos < 0. || zpos < 0.)
214       {
215         output.write("\tAdded %i fast_output points\n", i);
216         break;
217       }
218       ix = int(xpos * mesh->GlobalNx);
219       iy = mesh->ystart;a
220       iz = int(zpos * mesh->GlobalNz);
221
222       fast_output.add("n" + std::to_string(i), n, ix, iy, iz);
223       fast_output.add("phi" + std::to_string(i), phi, ix, iy, iz);
224       fast_output.add("vx"+std::to_string(i), vx, ix, iy, iz);
225       fast_output.add("vz"+std::to_string(i), vz, ix, iy, iz);
226       i++;
227     }
228   }
229
230   return 0;
231 }
```

In this code block, we set up the right-hand side, more precisely the equations that will be evolved by the solvers. This function is called every time step. To avoid any problems with potential negative density from numerics we evolve the logarithm of the density so in line 239 we regain the true density. Then lines 241-254 we calculate the potential from the vorticity, using different solvers which again is dependent on if we use the Boussinesq approximation or not. Then lines 259-297 set up all the equations that will be evolved, the equations themselves are written using the intuitive syntax specific to BOUT++. Lines 273-286 change if the resistivity term is linear or logarithmic. In the last two lines 295 and 296, we calculate the velocities from the potential, this is done

as the fast outputs lack the spacial resolution to calculate the velocities from the output data. The last line calls the BOUT++ main routine that handles the actual backend of the simulation.

```
233 int DriftFluidModel::rhs(BoutReal UNUSED(time))
234 {
235
236   // Communicate variables
237   mesh->communicate(comms);
238
239   n = exp(logn);
240
241   if (!boussinesq)
242   {
243     // Including full density in vorticit inversion
244     phiSolver->setCoefC(n);
245     phi = phiSolver->solve(vort / n, phi); // Use previous solution as guess
246   }
247   else
248   {
249     // Background density only (1 in normalised units)
250     phi = phiSolver->solve(vort, phi);
251   }
252
253   // Communicate phi
254   mesh->communicate(phi);
255
256   //****************** Equations of the model ***************************
257
258   // Continuity equation:
259   ddt(logn) = -bracket(phi, logn, bm) - g0 * DDZ(phi)
260               + g0 * DDZ(n) / n + S / n;
261
262   // Particle diffusion
263   ddt(logn) += mu_n * Delp2(n) / n;
264
265   // Parallel density loss
266   ddt(logn) -= lambda_n * loss * n * exp(V_float - phi) / n;
267
268   // Vorticity equation:
269   ddt(vort) = -bracket(phi, vort, bm) + (1.0 / n) * g0 * DDZ(n);
270
271   // Drift-wave adiabaticity term (from SOLT)
272   if (driftwaves)
```

```
273   {
274     if (dw_logn)
275     {
276       dw = alpha_dw * (1 - loss)
277             * ((phi - DC(phi)) - (logn - DC(logn)));
278     }
279     else
280     {
281       dw = alpha_dw * (1 - loss)
282             * ((phi - DC(phi)) - (n - DC(n)));
283     }
284     ddt(logn) += dw / n;
285     ddt(vort) += dw;
286   }
287
288   // Momentum diffusion
289   ddt(vort) += mu_vort * Delp2(vort);
290
291   // Choice of parallel loss terms for vorticity
292   ddt(vort) -= lambda_vort * loss * (exp(V_float - phi) - 1);
293
294   //velocity in x and z direction
295   vx = DDZ(phi);
296   vz = -DDX(phi);
297   return 0;
298 }
299
300 BOUTMAIN(DriftFluidModel);
```

Note that BOUT++ uses by default the y-coordinate as the coordinate along the magnetic field whereas the standard coordinate in the literature is z.

For further detail, BOUT++ is documented with examples at `https://bout-dev.readthedocs.io/en/latest/index.html`

3.1.1 Data analysis tool

As BOUT++ dumps all the outputs from the simulations into multiple files, one file for each core it runs on, running a simulation over multiple cores gives many files to deal with. A tool to easily handle the output files has been developed called xBOUT [41]. It is a python package built on the xarray [42] package, with features specific for handling BOUT++ data. Throughout this thesis, all

the data is collected from the output files using xBOUT, which allows retrieval of the data in a fast and efficient way for further analysis using python.

3.2 Numerical setup

In this section will go over the setup for the simulations done in the thesis.

The simulations will be done using the different variations of equation 2.51 in a domain of lengths $L_x = 200$ and $L_y = 100$, where the domain is discretized into a grid of 512×256. The location of the separatrix x_{SOL} is set to $x_{\mathrm{SOL}} = 50$ where the width of the transition w is set to $w = 25/32$. Using the same numerical values used in [38] we get $D_\perp = \nu_\perp = 10^{-2}$, $g = 10^{-3}$, $\chi_0 = 6 \cdot 10^{-4}$, $\Sigma_0 = 11/2000$, $\sigma_0 = 5 \cdot 10^{-4}$ and $\Lambda = (1/2)\ln(2\pi m_i/m_e)$, where the ions are deuterium such that $m_i = 3.34449469 \cdot 10^{-27}$kg and $m_e = 9.109383632 \cdot 10^{-31}$kg, and finally $x_0 = 20$ and $\lambda_s = 10$. We apply Neumann boundary condition on both the inner and outer boundary for the density and vorticity, while for the potential we use a Dirichlet boundary condition for the outer but a Neumann condition for the inner. Such that the boundary conditions are given as

$$\left.\frac{\partial n}{\partial x}\right|_{x=0,L_x} = \left.\frac{\partial \Omega}{\partial x}\right|_{x=0,L_x} = \left.\frac{\partial \phi}{\partial x}\right|_{L_x} = 0, \tag{3.2}$$

$$\phi(x=0) = 0. \tag{3.3}$$

For the poloidal directions, we apply periodic boundary conditions for all the fields. A schematic of the simulation domain is shown in figure 3.1, it shows the location of the plasma source, separatrix, and plots of the functions $\chi(x)$ and $\sigma(x)$. The radial positions of the probes where data time series of the plasma parameters are recorded are also shown.

3.2.1 Measurements

We are doing long turbulence simulations to be able to get the required amount of data points for the statistical analysis. This makes the handling of the full field of the simulation domain for full time resolution extremely heavy both in storage and memory. Therefore the physical values from a given simulation are written out in two parts: the full field and a grid of virtual probes. The full field is written out at a slower temporal resolution while for the higher time resolution output we have a grid of 9x9 probes evenly distributed throughout the simulation domain. This means that in addition to the 9 probes shown in

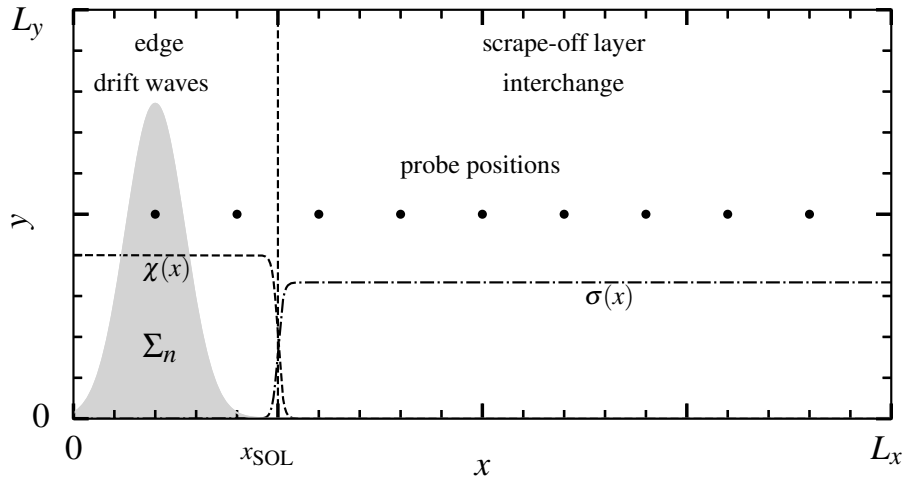

Figure 3.1: Schematic showing the domain used in the simulations. The plasma source Σ_n is shown as the gray shaded area, the radial variation in conductivity χ and sheath parameter strength σ, the location of the separatrix x_{SOL} and the radial positions of the virtual probes. [38]

figure 3.1 there are 8 other such rows evenly distributed among the poloidal direction. The virtual probes write out all the same data as the full field, but at one specific point in the simulation domain and at a higher temporal resolution.

3.2.2 Simulations

The main simulations that will be done for this thesis consist of the four main sets of equations introduced in chapter 2. These simulations will be done to compare how the different models affect the statistical properties of the fluctuations in the turbulence. There will be simulations done where we set $x_{\mathrm{SOL}} = 0$, these simulations will then be pure SOL simulations. This is done so that one can get a clearer picture of which effects occur because of the edge region and which are pure SOL effects. As mentioned above we use parameters form [38], since our model is an extension of the one used in that paper it will allow us to compare results easier. In table 3.1 we have listed some more realistic machine parameters for comparison with the ones chosen for the simulations. We will also be doing a scan for different sheath parameter strengths σ_0 to check more realistic sheath parameter strengths.

In figure 3.2 we show contour plots of the logarithmic electron particle density for the Full n and Boussinesq model.

Table 3.1: Estimates for input parameters for different tokamaks [43].

	g	σ_0	χ_0
MAST	$2.4 \cdot 10^{-3}$	$6.1 \cdot 10^{-5}$	$2.7 \cdot 10^{-4}$
C-Mod	$3.4 \cdot 10^{-3}$	$7.7 \cdot 10^{-6}$	$1.0 \cdot 10^{-5}$
TCV	$8.8 \cdot 10^{-4}$	$3.3 \cdot 10^{-5}$	$1.9 \cdot 10^{-4}$
KSTAR	$3.7 \cdot 10^{-4}$	$1.6 \cdot 10^{-5}$	$6.8 \cdot 10^{-4}$
AUG	$3.4 \cdot 10^{-4}$	$1.5 \cdot 10^{-5}$	$7.7 \cdot 10^{-5}$
JET	$1.3 \cdot 10^{-4}$	$1.1 \cdot 10^{-5}$	$3.1 \cdot 10^{-5}$
NSTX	$2.7 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$

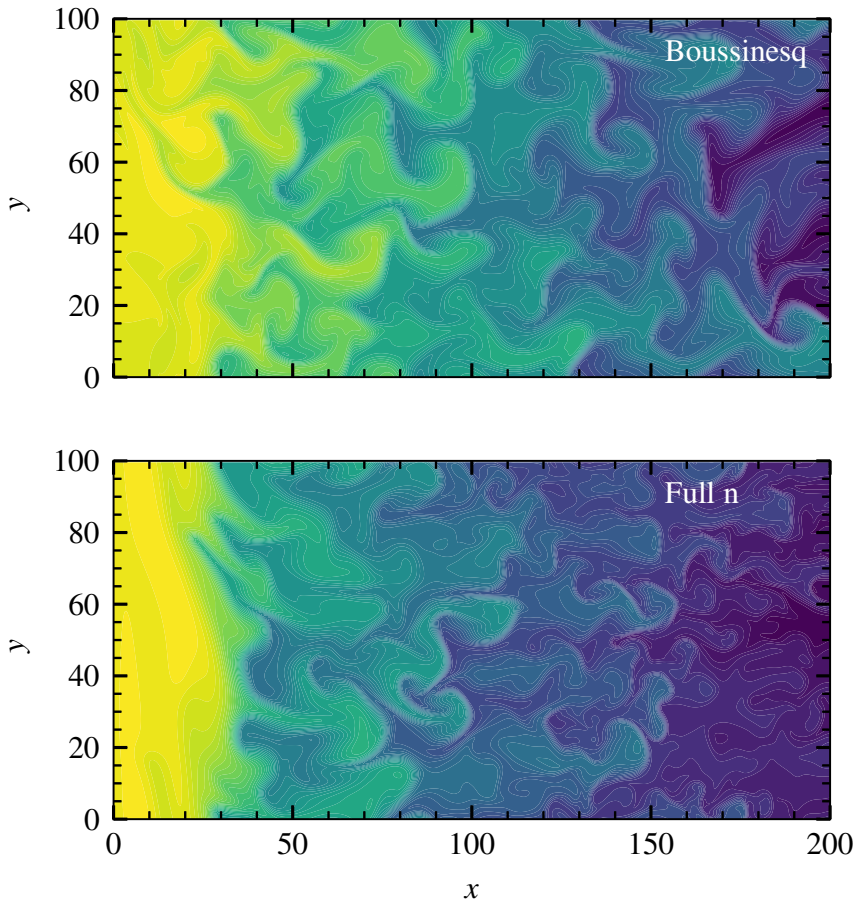

Figure 3.2: Contour plots of $\log(n)$ in the turbulent state for the Boussinesq model (top) and the Full n model (bottom).

4

Statistical analysis

In this chapter, we will introduce methods we will use for statistical analysis. These methods will then be applied to the data from the simulations described in the previous chapter.

4.1 Consepts

We start by introducing some statistical concepts that we are going to utilize.

For this analysis we are going to utilize the absolute and normalized time series of the physical quantities that we are investigating. For a given time series $\Psi(t)$ we introduce the normalized time series $\tilde{\Psi}(t)$ by

$$\tilde{\Psi} = \frac{\Psi - \langle\Psi\rangle}{\Psi_{\text{rms}}} \tag{4.1}$$

where $\langle\Psi\rangle$ denotes the time average and Ψ_{rms} indicates the root mean square (RMS) value defined as

$$\Psi_{\text{rms}} = \left\langle (\Psi - \langle\Psi\rangle)^2 \right\rangle^{1/2}. \tag{4.2}$$

Following this definition, the time series $\tilde{\Psi}$ has zero mean as well as a standard deviation of one. This will allow us to compare time series from different

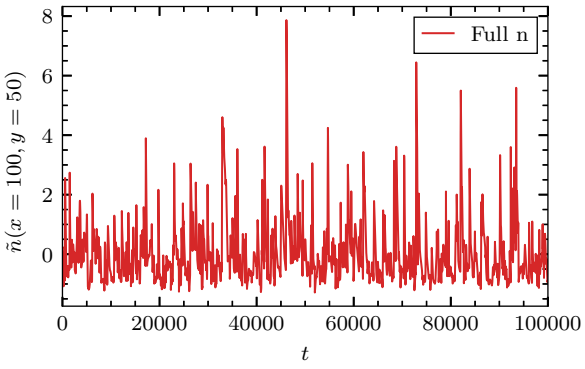

Figure 4.1: A part of the normalized electron density time series recorded at $x = 100$ and $y = 50$

simulations better. A normalized time series of electron density is given in figure 4.1.

We will model each of the time series we analyse as a superposition of uncorrelated pulses with fixed shape and constant duration given by

$$\Psi_K(t) = \sum_{k=1}^{K(T)} A_k \psi\left(\frac{t - t_k}{\tau_\mathrm{d}}\right) \tag{4.3}$$

where $K(T)$ is the total amount of pulses in our time series over its duration T, ψ is some pulse function, τ_d is the duration of the pulses, t_k is the arrival time and A_k is the amplitude for each pulse k. This stochastic model is often called a filtered Poison process (FPP) [44]. The mean of the random variable Ψ_K is then $\langle\Psi\rangle = (\tau_\mathrm{d}/\tau_w)\langle A\rangle$, where $\langle A\rangle$ is the average pulse amplitude and τ_w is the average waiting time between each pulse. As the name eludes we further assume that pulses arrive according to a Poisson process. This means that the waiting times will be independent and exponentially distributed while the arrival times are independent and uniformly distributed. Then we also assume that the amplitudes are independent and exponentially distributed such that

$$P_A(A) = \frac{1}{\langle A\rangle} \exp\left(\frac{1}{\langle A\rangle}\right). \tag{4.4}$$

This then also give us that

$$\Psi_\mathrm{rms}^2 = \frac{\gamma}{2}\langle A\rangle, \tag{4.5}$$

where γ relationship between τ_d and τ_w and describes the intermittency of the prosess and is called the intermittency parameter deffined as:

$$\gamma = \frac{\tau_\mathrm{d}}{\tau_w}. \tag{4.6}$$

The intermittency parameter can be used to estimate different statistical moments for the time series it describes. The estimates for the skewness and the flatness moments for the time series are given as

$$S_\Psi = \frac{\left\langle (\Psi - \langle\Psi\rangle)^3 \right\rangle}{\Psi_{\text{rms}}^3} = \frac{2}{\gamma^{1/2}}, \tag{4.7}$$

$$F_\Psi = \frac{\left\langle (\Psi - \langle\Psi\rangle)^4 \right\rangle}{\Psi_{\text{rms}}^4} = 3 + \frac{6}{\gamma}. \tag{4.8}$$

We can also use these moments for estimating the intermittency of the time series. The relative fluctuation level given as $\Psi_{\text{rms}}/\langle\Psi\rangle$ can also estimate the intermittency by

$$\frac{\Psi_{\text{rms}}}{\langle\Psi\rangle} = \frac{1}{\gamma^{1/2}}. \tag{4.9}$$

Two-sided exponential pulses will be considered as the main pulse shape ϕ_e as they have been shown to be in agreement with results from experimental measurements [45–48]. They are defined as:

$$\phi_e(\theta, \lambda_e) = \begin{cases} \exp\left(-\frac{\theta}{1-\lambda_e}\right), & \theta \geq 0 \\ \exp\left(\frac{\theta}{\lambda_e}\right), & \theta < 0 \end{cases} \tag{4.10}$$

where θ is a nondimensional variable which in our case is time normalized by the duration time t/τ_e, and λ_e is called the asymmetry parameter. The asymmetry parameter describes the rise and fall time for each side of the exponential tails, where $\lambda_e = 1/2$ will be a symmetric pulse shape.

The probability distribution function or PDF describes the probability that a certain outcome of an event happens. For our case it would be the probabilities for different fluctuation amplitudes we will be looking at. The shape of the distribution function tells us how likely the different amplitudes are to occur. For a symmetric PDF like the normal distribution, there is an equal distribution of fluctuation amplitudes around the mean, while for a gamma-distributed normalized time series fluctuation amplitudes will have an uneven distribution where there the most probable fluctuations still happen around the mean but there is a possibility for large positive fluctuations but not for negative fluctuations. For a normalized two-sided exponential pulse shape the PDF is a gamma distribution given as:

$$P_{\tilde{\Phi}_e,\phi_e}(\tilde{\Phi}_e) = \frac{\gamma^{\gamma/2}}{\Gamma(\gamma)} \left(\tilde{\Phi}_e + \gamma^{1/2}\right)^{\gamma-1} \exp\left(-\gamma^{1/2}\tilde{\Phi}_e - \gamma\right) \tag{4.11}$$

where γ the intermittency parameter is the shape parameter.

To describe the shape of a distribution we can use the previously defined skewness and flatness. Skewness describes the asymmetry of the distribution, a distribution with positive skewness will have longer tails in the right direction and vice versa. Flatness describes the tails of the distribution, a higher flatness indicates that a distribution has longer tails, while a lower value indicates the opposite. For a common point of reference, a normal distribution will have a skewness of 0 as it is symmetric, and has a flatness of 3.

The power spectral density or PSD of a time series describes how the the energy of the time series is distributed among different frequencies. This allows us to describe where the energy distribution of the signal is, where low frequencies will describe the background of the system and high frequencies describe the energy present in the fluctuations. In the context where we are given a waveform we can calculate the PSD as the Fourier transform of the waveform multiplied by it complex conjugate. The Fourier transform of a pulse form ψ is given by

$$\hat{\psi}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \psi(\theta) \exp(i\theta\omega) \mathrm{d}t, \tag{4.12}$$

where ω is the angular frequency. This gives us the PSD $\Omega_{\tilde{\Psi},\psi}(\omega)$ for the pulse shape ψ by

$$\Omega_{\tilde{\Psi},\psi}(\omega) = \frac{1}{I_2} \left| \hat{\psi}(\omega) \right|, \tag{4.13}$$

where I_2 is a normalization factor defined by

$$I_n = \int_{-\infty}^{\infty} [\psi(t)]^n \, \mathrm{d}t. \tag{4.14}$$

From this relationship between the pulse function and the PSD we can use a fit function on one and estimate parameters that should fit both the PSD and the waveform.

For the two-sided exponential pulse, the PSD is given by:

$$\Omega_{\tilde{\Phi}_e,\phi_e}(\omega; \lambda_e) = \frac{2\tau_e}{\left[1 + (1-\lambda_e)^2 \tau_e^2 \omega^2\right] \left[1 + \lambda_e^2 \tau_e^2 \omega^2\right]} \tag{4.15}$$

where the asymmetry parameter λ_e for the two-sided exponential shows up as the shape parameter.

A two-sided exponential pulse has a discontinuous peak, for high sampling frequencies we expect the peak of the pulses in the simulation to be continuous. As a Lorentzian pulse has a continuous peak we will in addition to the two-sided exponential pulse shape also consider a convolution between the two-sided

exponential and a skewed Lorentzian pulse. The Fourier transform of a two-sided exponential is given by:

$$\hat{\phi}_e = \frac{\tau_e}{(1 - i\lambda_e\tau_e\omega)\,(1 + i\,(1 - \lambda_e)\,\tau_e\omega)}, \tag{4.16}$$

and the Fourier transform of a skewed Lorentzian is given as [49]

$$\hat{\phi}_l = \tau_l e^{-\tau_l|\omega|}|\tau_l\omega|^{-i2\lambda_l\tau_l\omega/\pi}, \tag{4.17}$$

where λ_l again is the asymmetry parameter and τ_l is the duration time. Then a convolution of the two gives us:

$$\hat{\phi}_c = \frac{\tau_e\tau_l e^{-\tau_l|\omega|}|\tau_l\omega|^{-i2\lambda_l\tau_l\omega/\pi}}{(1 - i\lambda_e\tau_e\omega)\,(1 + i\,(1 - \lambda_e)\,\tau_e\omega)} \tag{4.18}$$

where subscript e and l refers to the parameters associated with the two sided exponential and the skewed Lorentzian respectively. The waveform would be an inverse Fourier transform of the $\hat{\phi}_c$ but no closed analytical expression is known so it will be calculated by a numerical inverse Fourier transform. The PSD for this waveform will then be the calculated using this numerical waveform, and normalized by the integral of itself.

A numerical way to retrieve the waveform of the fluctuations that occur in a time series is through conditional averaging. As the name implies we apply a condition C to the time series, like requiring a certain magnitude for the fluctuations. Using this condition we will then pick out time intervals of a time series $\tilde{\Psi}(t)$. The ensemble average of all these subintervals of the time series then gives us the conditional average

$$\tilde{\Psi}_C = \left\langle \tilde{\Psi}|C \right\rangle. \tag{4.19}$$

The subintervals are selected around the maxima of $\tilde{\Psi}$ that fulfill the condition C, such that the interval is $t \in (t_m - \Delta, t_m + \Delta)$. Here t_m is the time where the maxima occurred and Δ is a window around t_m such that the total length of the interval is 2Δ. In addition, we also require that none of the subintervals overlaps each other. With these requirements in mind, we define the conditional average of a time series as:

$$\tilde{\Psi}_C = \frac{1}{M}\sum_{m=1}^{M} \tilde{\Psi}(t - t_m), t - t_m \in (-\Delta, \Delta) \tag{4.20}$$

where M is the total amount of sub intervals. One other use for the conditional average is the cross conditional average. The cross conditional average of a time series $\tilde{\Phi}(t)$ would not be defined around when its maxima fulfills the condition, but rather when the time series $\tilde{\Psi}$ reaches a maxima that fulfills the condition. So the cross conditional average would be defined as

$$\tilde{\Phi}_{CC} = \left\langle \tilde{\Phi}|\tilde{\Psi}_C \right\rangle. \tag{4.21}$$

4.2 Time averaged radial profiles

In this section, we will present radial profiles of the lowest order moments for the different simulations. All of these profiles are calculated by first taking the appropriate moment for each of the time series measured with the artificial probes. Then for each radial position we average over the poloidal direction, this gives us a set of radial profiles for each of the moments. For the four main cases, each probe has time series of length $2 \cdot 10^5$time steps.

The time-averaged electron density profile of our main simulations is shown in the top left of figure 4.2. The density decreases exponentially moving outwards through the SOL. We can observe that there is a notable difference between the particle densities of the full n model and the rest of the simulations. The fit to an exponential function to the full n model and the Boussinesq approximated model with logarithmic density in the drift wave term shows that there is a difference in both amplitude and scale length of the density profiles. Where the full n model has a scale length of $L_n = 35.59$, and the Boussinesq approximated model with log density has a scale length of $L_n = 33.76$. The biggest difference between the full n and the other models is the magnitude of the density. At $x = 40$ the measured mean density for the full n model is $\langle n \rangle = 1.28$ while for the Boussinesq log dw model it measures to be $\langle n \rangle = 2.77$.

The top plots in figure 4.2 hint at some interesting physics happening. For the mean particle density, we see that in the full n case there is a higher peak density than for the rest of the simulation inside the edge region. Then in the SOL, the density of the full n model is smaller than the rest of the simulation cases, hinting at higher confinement of the plasma. One interesting thing to note is that this high containment only shows up in the model where the Boussinesq approximation is not present and the drift wave term is logarithmic, not for either of the cases where just one of them is present hinting at some type of interplay between these two terms.

The relative electron density fluctuation profile of the four main simulations is presented in the top right of figure 4.2. For all simulations, we see relative fluctuation levels of order unity throughout the simulation domain. The two simulations done with the Boussinesq approximations follow each other increasing outwards in the SOL, while the non-Boussinesq simulations diverge from them. The non-Boussinesq models seem to follow at the start with a peak at the approximate location of the separatrix before starting to reduce in magnitude further out in the SOL. There even is a divergence between the relative fluctuations of the non-Boussinesq and the full n models.

While the relative fluctuation level for all four simulations seams about the same in the edge region there is a noticeable difference near the separatrix.

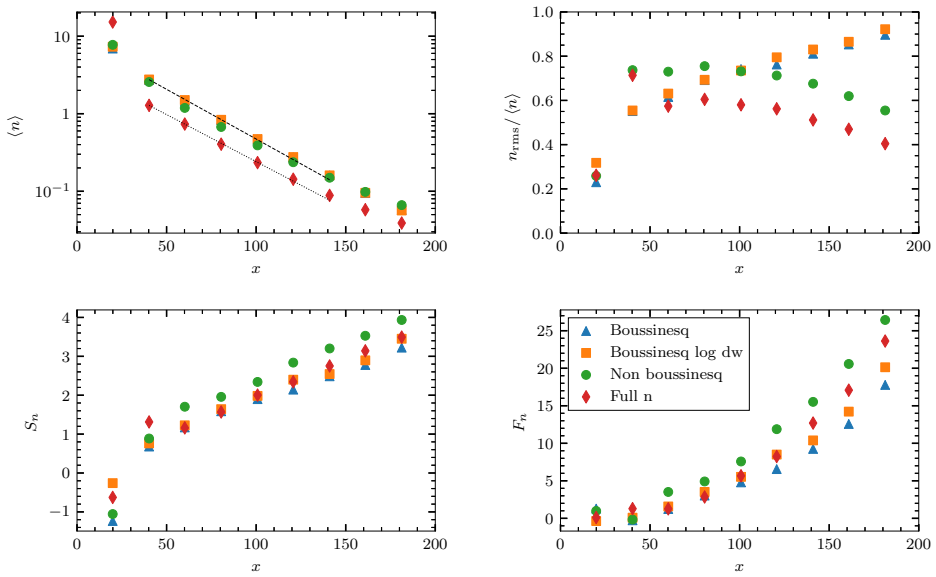

Figure 4.2: The radial profile of the mean (top left), relative fluctuation (top right), skewness (bottom right), and flatness (bottom right) of the density for different simulations. The dashed and the dotted lines represent an analytical fit to an exponential function for the Boussinesq log dw and full n simulations.

Here the non-Boussinesq approximated models have an increase much larger than the Boussinesq approximated models, implying that the area around the separatrix has more fluctuations in the non-Boussinesq cases. Then after the initial increase in relative fluctuation levels, the full n model drops in amplitude creating a difference in fluctuation magnitude between the two non Boussinesq approximated cases, again hinting at the increase of confinement in the full n model. After this initial increase, and drop for the full n case, the two non-Boussinesq approximated models start to curve downwards, decreasing in relative fluctuation level, while the Boussinesq approximated models increase. The decrease in relative fluctuation level outward through the SOL indicates that the term normally neglected in the Boussinesq approximation has a dampening effect on the fluctuations.

As for the higher-order moments, skewness and flatness, they are plotted in figure 4.2 bottom left and right respectively. The higher-order moments seem to follow each other in shape, but with a slight spread. The shape follows what has previously been observed in pure Boussinesq approximated simulations of the same model [38].

In figure 4.3 the radial profile of RMS values for the radial velocity have been

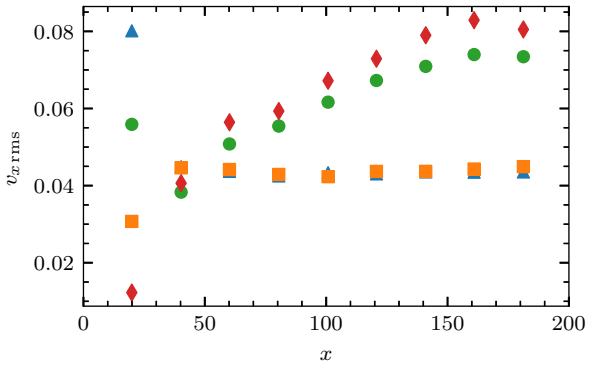

Figure 4.3: The radial profile of $v_{x\mathrm{rms}}$ for the four main simulation cases.

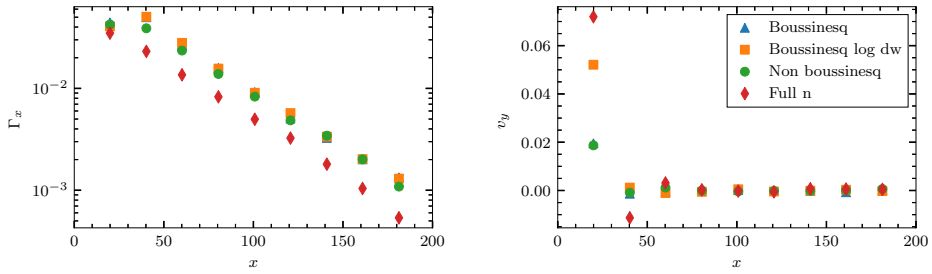

Figure 4.4: The radial profile of particle flux (left) and the poloidal velocity (right) for the four main simulation cases.

plotted. From the plot, we see that the fluctuations of the two non-Boussinesq approximated cases increase moving radially outwards, while the Boussinesq approximated cases the has a more stable value throughout the domain. This indicates an increase in the peak velocity for the fluctuations moving radially outward meaning as filaments move radially outwards through the domain they increase in radial velocity.

To see the effect that the edge region has on the profile we plot the radial profiles for the radial electron flux and the mean poloidal velocity in figure 4.4. In the particle flux, we see an increase around the area where the source is located in both the Boussinesq approximated cases. In the poloidal velocity, we observe that there is a strong gradient in the velocities in the edge region. This strong gradient indicates the presence of zonal flows, and the strength of the gradient indicates the strength of the flow. Observe the cases with the logarithmic density in the drift wave term have a greater gradient than the ones in the cases with the linear drift waves. The fact that for the non-Boussinesq case we see no increase indicates that there is an important interplay between both the drift wave term and the polarisation equation.

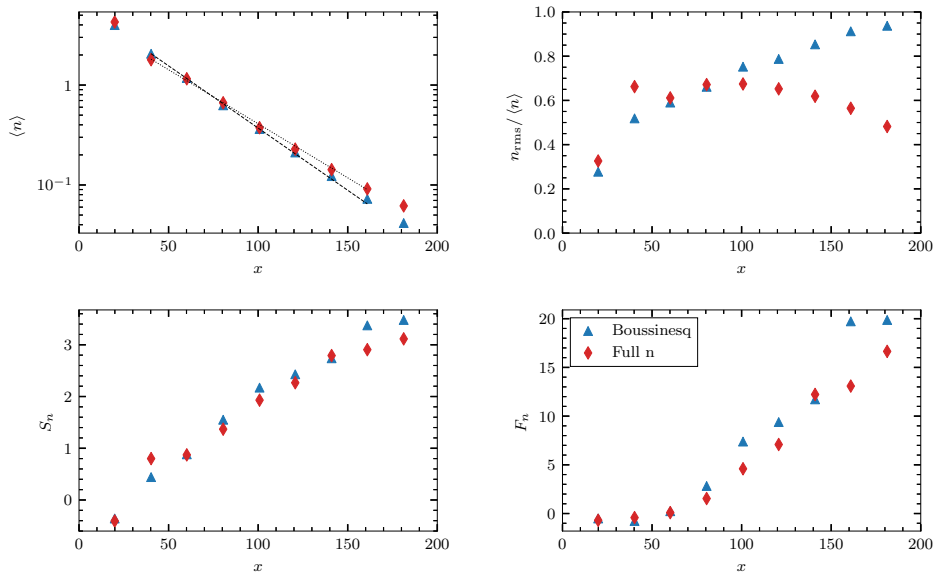

Figure 4.5: The radial profile of the mean (top left), relative fluctuation (top right), skewness (bottom right), and flatness (bottom right) comparing the different models in a pure SOL simulation.

To try to get an even better understanding of the physics underlying the complex dynamics going on in these simulations we also did runs with the SOL through the entire simulation domain by setting $x_{\mathrm{SOL}} = 0$. There are only two models in this simulation as the difference between two of them is a change in the drift wave term which only occurs in the edge region and therefore does not apply in these simulations. The profiles are shown in figure 4.5. The mean density profile for the two simulations is plotted in the top-left frame. Comparing the two scale lengths we have for the Boussinesq model $L_n = 34.87$, and for the full n model $L_n = 40.16$. For the skewness and the flatness shown in the bottom two panels there seem to be no significant difference between the two simulations, the exception being the last two data points in the flatness. As higher-order moments need more data to converge this might be down to too short simulations.

While all the other statistics look fairly similar the one standing out is the relative fluctuation seen in the top right of figure 4.5. Here we observe a deviation from the Full n model and the Boussinesq approximated models. The relative fluctuation level decreases with radius for the first after reaching $x = 100$, whereas the latter keeps increasing with radius. The similarity between the behaviour of the relative fluctuation in the pure SOL and the edge and SOL simulations indicates that the effect of the Boussinesq approximation in the SOL is mainly in the fluctuation levels.

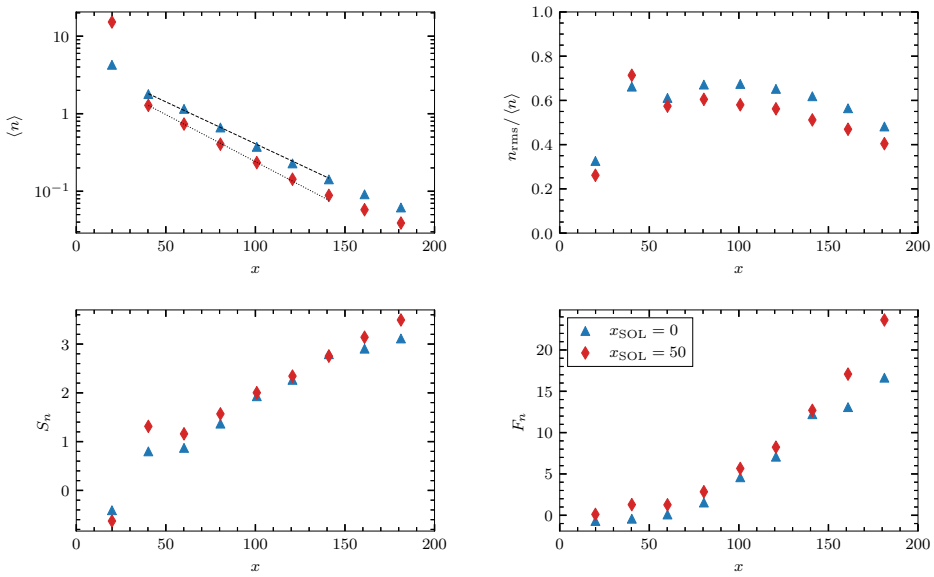

Figure 4.6: The radial profile of the mean (top left), relative fluctuation (top right), skewness (bottom right), and flatness (bottom right) comparing difference between a pure SOL simulation and a edge plus SOL simulation in a full n model.

To compare the differences in the full n model with and without the edge region we plot the profiles from the simulations done of the full n model in the edge plus SOL and the pure SOL case against each other in figure 4.6. The third and fourth-order moments coincide for both simulations. The scale length of the two density profiles as found previously is $L_n = 40.15$ and $L_n = 35.59$ for the SOL and edge plus SOL simulations respectively. The density profiles differ also as the simulation with the edge region has a slightly higher mean density at the start but lower mean densities going outwards. This is also seen in the relative fluctuations where there is a higher relative fluctuation level in the case with just the SOL throughout the entire simulation domain, except for one point close to where the separatrix would be.

This indicates that the separatrix as one would expect plays a role in the confinement of the plasma. Comparing it with the results form [38] where we do not see this same difference would indicate that the Boussinesq approximation changes both how the separatrix and the fluctuation in the SOL behave.

In the left part of figure 4.7 we compare the electron flux and the poloidal velocities of the full n model with the Boussinesq approximated in a pure SOL simulation domain. There is not a notable difference between the different models for the particle flux. We also plot the pure SOL and the edge plus SOL

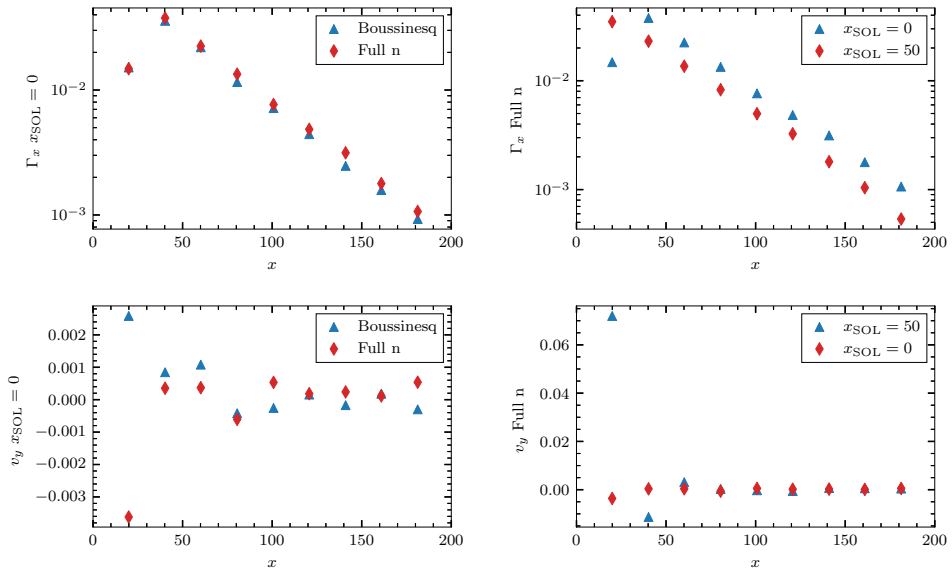

Figure 4.7: The radial profile of particle flux and poloidal velocity. Comparing the different models in a pure SOL domain (left), and comparing the full n model in a pure SOL with the edge and SOL case.

full n models against each other. Here we again observe the steady decrease in flux for the edge plus SOL case while in the pure SOL case there is an increase around the source is. In the plot for the poloidal velocities, we observe a peak near the source, and further outwards into the domain the points lay around zero. Where the difference in the sign of the velocities can be attributed to slight differences in how the source term manifests itself. For the plot comparing the SOL with the edge plus SOL we observe how much of a difference it makes for the poloidal velocity to have the edge region, this also shows how small the poloidal velocities are in the pure SOL case compared to the velocities of the zonal flow in the edge plus SOL case. The plots in figure 4.7 again describe the importance of the zonal flow barrier between the source of the plasma and the SOL for sake of confinement.

For completeness, a scan of different sheath parameters has been done for the full n model the profiles of which are shown in figure 4.8. The profiles for the densities have a quite pronounced brake for the lower sheath parameters giving us the broadening of the far SOL seen in experiments for change in plasma densities [16]. Universally the relative fluctuation levels all decrease outwards through the simulation domain as we have seen for all the non-Boussinesq cases. The mean electron density measured at $x = 80$ as well as the length scale for each of the cases is given in table 4.1. We also plot the scale length against the sheath parameter strength in figure 4.10, where we have fitted it to

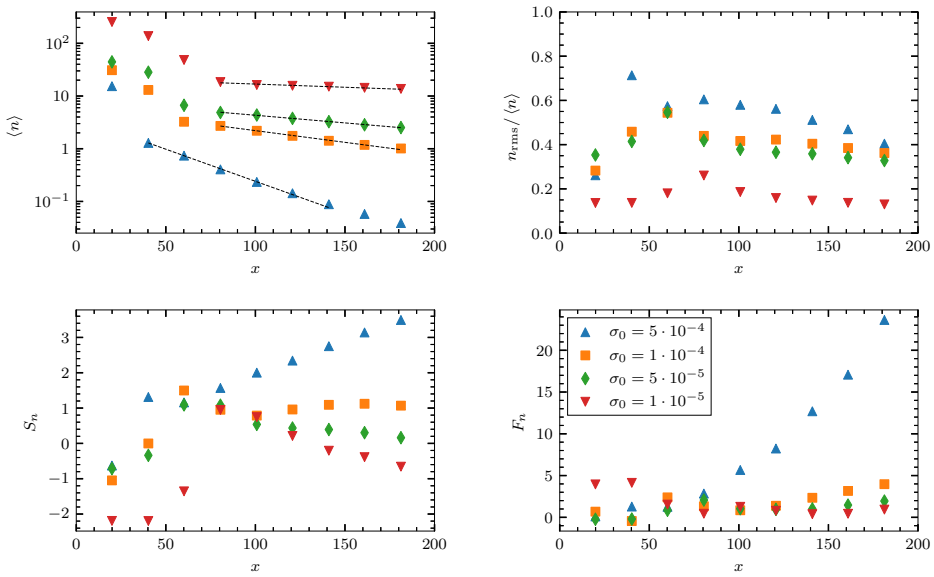

Figure 4.8: The radial profile of the mean (top left), relative fluctuation (top right), skewness (bottom right), and flatness (bottom right) of the density for different sheath parameter σ_0 strength.

Table 4.1: The fit parameters for different sheath parameter strengths for the full n model.

σ_0	$\langle n \rangle\ (x = 80)$	L_n
$5 \cdot 10^{-4}$	0.40	35.77
$1 \cdot 10^{-4}$	2.71	97.72
$5 \cdot 10^{-5}$	4.87	149.75
$1 \cdot 10^{-5}$	18.40	359.01

a function of $L_n(\sigma_0) = a\sigma_0^k$. The values of the fit parameters is then estimated to be $a = 0.54, k = -0.57$.

We have in figure 4.9 plotted the time series for the start of the simulations of different sheath parameter strengths. The time series is measured at the middle of the domain, $x = 100, z = 50$. As is quite clear from the plot the transient time of the simulations increases with lower σ_0, the amplitude of the entire time series also increases. We can also see this in the amplitude of the densities in figure 4.8.

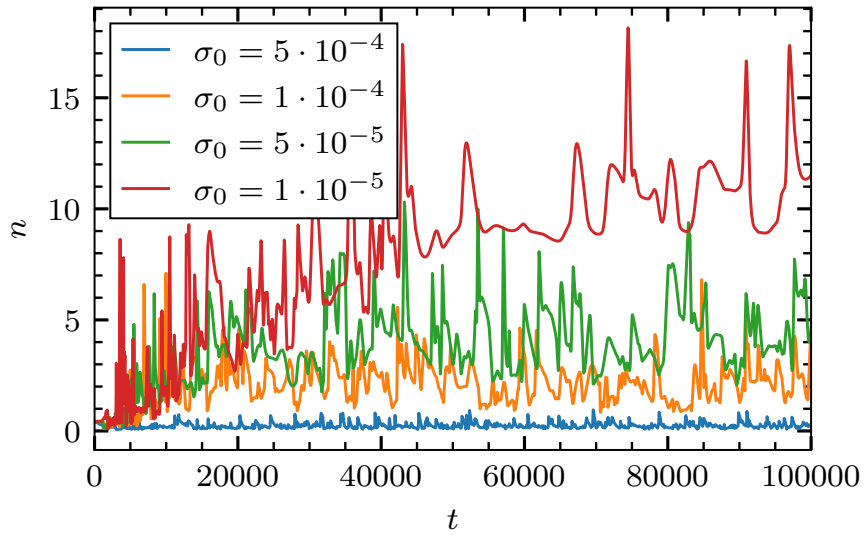

Figure 4.9: Time series showing the evolution of the density for different sheath parameters measured at $x = 100, z = 50$.

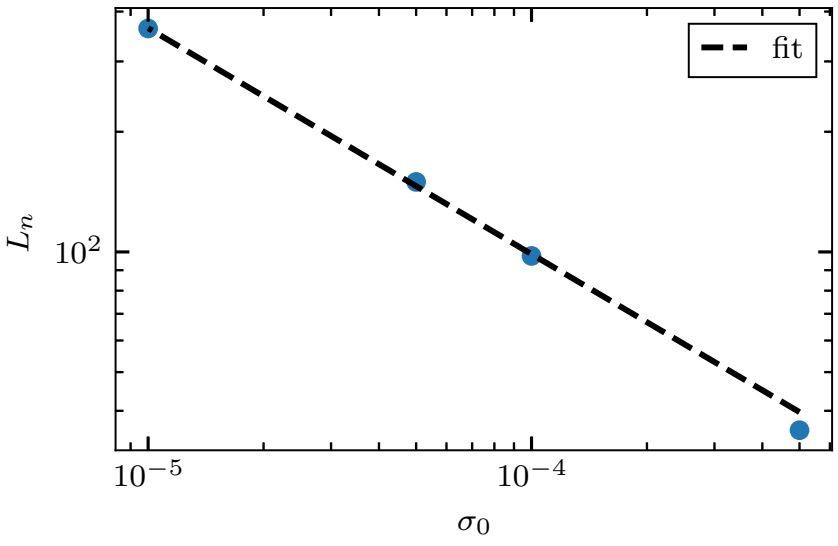

Figure 4.10: Plot of the scale length against sheath parameter strength.

4.3 Fluctuation statistics

In this section, we will go over the results from analysis using fluctuation statistics. Again we have multiple probes at different poloidal positions for each radial position. To be able to reduce the data to just radial positions we need to average over the time series at all poloidal positions. One subtle part of this is that we can not just average over the poloidal probes, this will average out the fluctuations for each of the time series. The trick is to calculate all the fluctuation statistics for each poloidal position and then average over them. As we expect the same statistical properties at all poloidal positions this will then give better convergence for each radial position. All our analysis is done with the normalized time series $\tilde{\Psi}$.

The PDF for all the different simulations at two radial positions is given in figure 4.11. The PDF of the different simulations seem to overlay except for the non Boussinesq approximated case has a higher peak. In figure 4.12 the PDF at different radial positions for each of the different simulations is presented. Universally for all of them, we observe a broadening of the distribution towards high amplitude fluctuations when we move further out into the simulation domain. We also observe that the peak of the Boussinesq approximated cases sharpen as we move radially outwards.

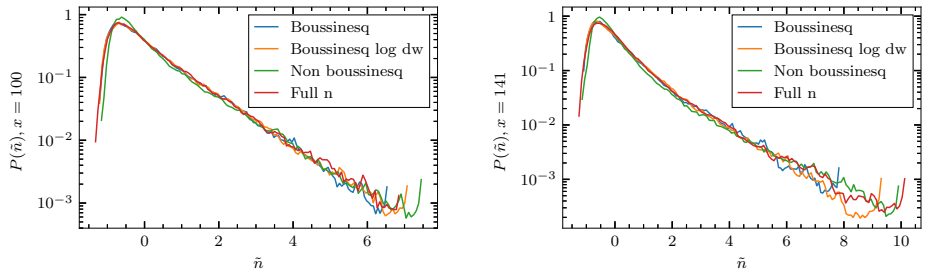

Figure 4.11: The probability distribution function for all the simulation. At position $x = 100$ (left), $x = 141$ (right).

Figures 4.13 and 4.14 shows plots of the PDF for the different simulations at $x = 100$ and $x = 141$ respectively. They are fitted to the PDF of a gamma distribution with additive Gaussian noise, where γ is the shape parameter, and ϵ is the noise parameter defined as the ratio of the squared root mean square value for the gamma distribution and the normal distribution. To add noise to the FPP process you augment the process Φ such that

$$\Psi(t) = \Phi(t) + \sigma N(t) \tag{4.22}$$

here σ is the strength of the noise and $N(t)$ is in our case a normally distributed process. The PDF of this process will then be the convolution of the PDF for the

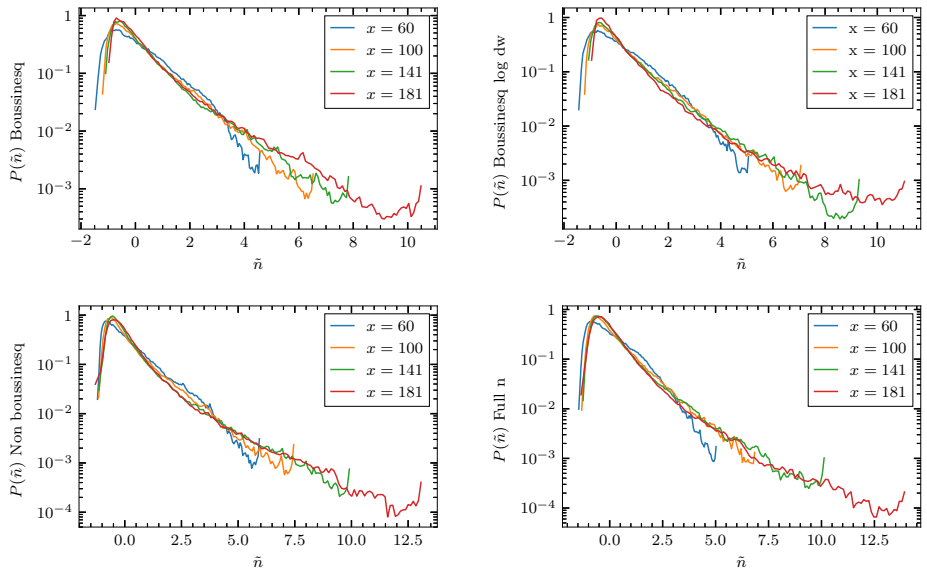

Figure 4.12: The probability distribution function for the four main simulation cases. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

two processes [50]. We have applied two fits to each of the distributions, one that captures the peak of the distribution but not the tail and one that captures the tail of the distribution but not the peak. The values of the fit parameters are given in table 4.2. The tail of the distribution is captured by the shape parameter, while the noise parameter fits the peak of the distribution. For the peak fit the shape parameter decreases moving radially outwards and the noise increases for all simulations. The same holds for the tail fit as well.

The power spectral density for all the simulations at two different radial positions is shown in figure 4.15. For the case where $x = 100$ there is a good correspondence between the different simulations, while further out into the domain at $x = 141$ we see a spread in the tail of the different simulations. Indicating that the fluctuations change in shape moving through the SOL. Figure 4.16 shows the PSD at different radial positions, here we observe that there is little to no change in the power spectra for the two Boussinesq approximated runs. On the other hand, there is an observable spread in the tails of the runs where the Boussinesq approximation is not present.

Figures 4.17 and 4.18 show the power spectral density of the different simulations at $x = 100$ and $x = 141$ where we have fitted the analytical expression for the PSD of a two-sided exponential pulse defined in equation 4.15. The pa-

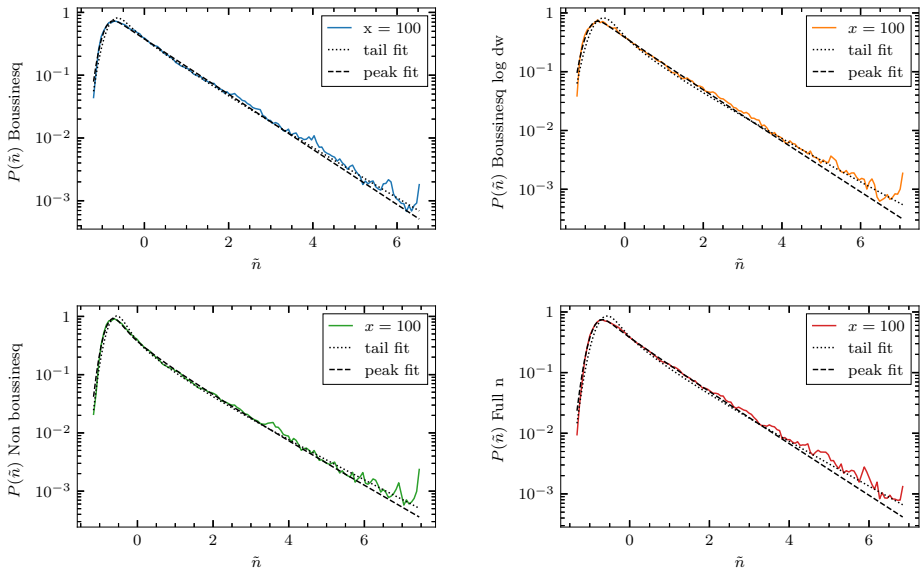

Figure 4.13: The probability distribution function for the four main simulation cases at $x = 100$. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

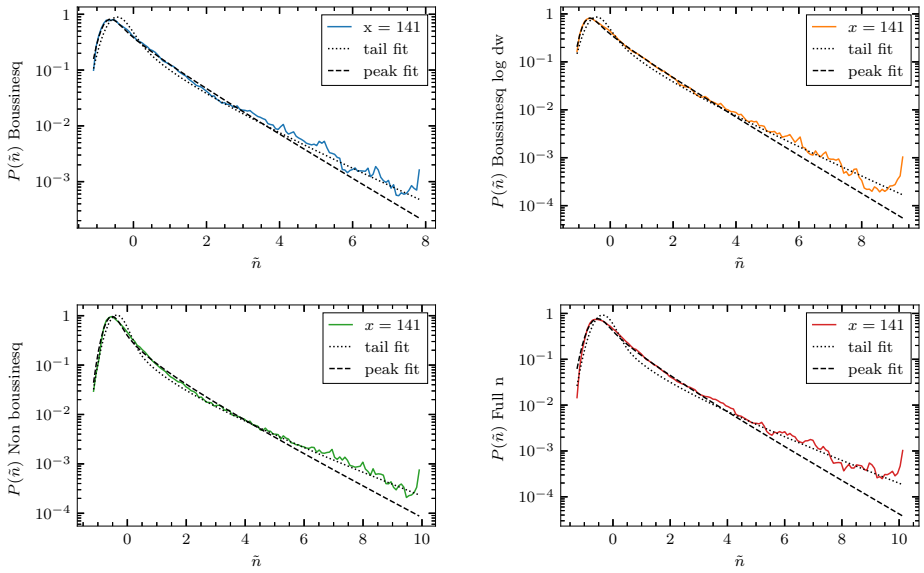

Figure 4.14: The probability distribution function for the four main simulation cases at $x = 141$. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

Table 4.2: The fit parameters for the PDF fits, one set for the peak fit and one for the tail fit.

Simulation	x	γ (peak)	ϵ (peak)	γ (tail)	ϵ (tail)
Boussinesq	100	0.99	0.03	0.74	0.04
	140	0.71	0.05	0.35	0.09
Boussinesq log dw	100	0.93	0.04	0.55	0.07
	140	0.74	0.04	0.35	0.10
Non Boussinesq	100	0.65	0.03	0.47	0.04
	140	0.43	0.04	0.20	0.09
Full n	100	0.89	0.04	0.57	0.06
	140	0.59	0.08	0.21	0.12

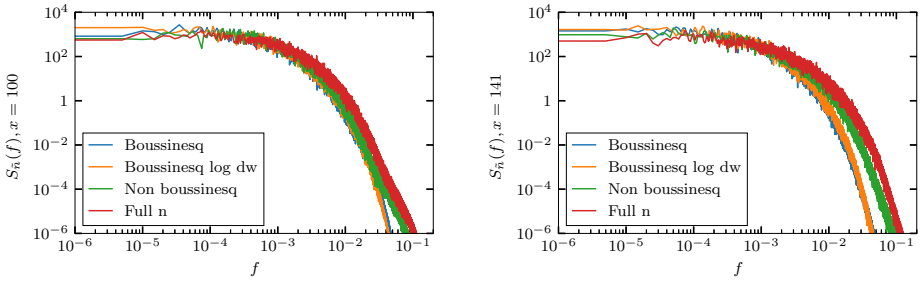

Figure 4.15: The power spectral density for the four main simulation cases. At position $x = 100$ (left) and $x = 141$ (right).

rameters from the fit are shown in table 4.3. We observe that the analytical fit is good for the first 4.5 decades, after which the power-law tail of the analytical fit diverges from the PSD of the simulations. If we then instead fit to a convolution of a two-sided exponential and a Lorentzian pulse we obtain a much better fit. The fit of the convolved pulse has been done using the values for τ_e and λ_e we estimated from the two-sided exponential, such that the only parameter we fit for is the duration time τ_l of the Lorentzian. The shape parameter of the Lorentzian pulse does not affect the shape of the PSD and is therefore set to $\lambda_l = 0$ assuming a symmetric Lorentzian function. The fit parameters for this case are presented in table 4.4.

From table 4.3 we see that the Boussinesq approximated simulations do not have a great change in their parameters when moving further out in the SOL, where both parameters for the simulations stay in about the same range with the more consistent model changes to a larger degree. Whereas in the simulations where the Boussinesq approximation is not used we observe a great change in both parameters when moving radially outwards. Note the large difference in duration time between the models where the Boussinesq approximation is present and where it is not. In table 4.4 the parameter of interest is the duration

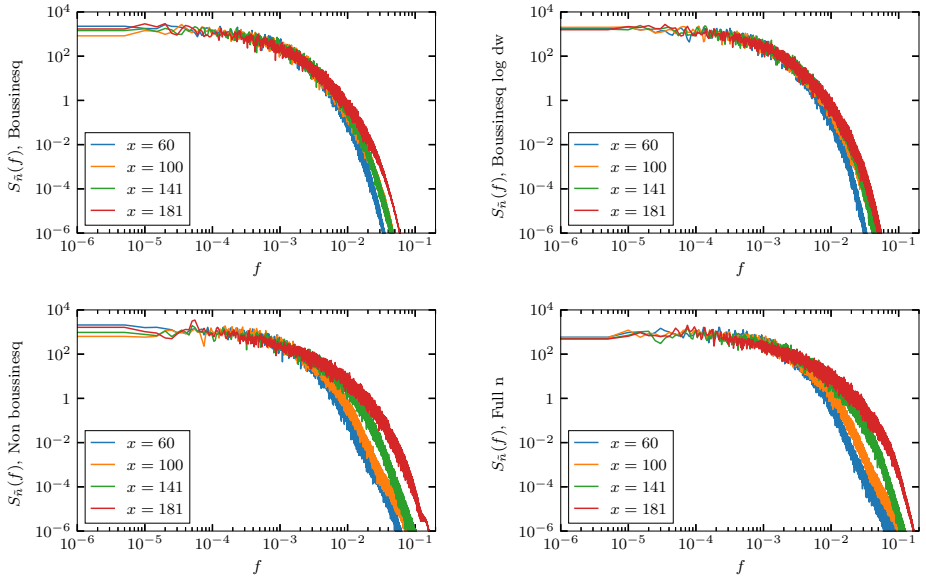

Figure 4.16: The power spectral density for the four main simulation cases. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

time τ_l for the Lorentzian pulse where the Boussinesq approximated cases have a much higher duration time than both the non-Boussinesq cases. The duration times for both the two-sided exponential pulse τ_e and the Lorentzian pulse τ_l are smaller for the two non Boussinesq approximated cases have smaller structures than the Boussinesq cases moving radially though the SOL.

For completeness, we have also plotted the compensated spectra for all simulations at $x = 100$ in figure 4.19. The compensated spectra allow us to check if there is power-law scaling present in the PSD, this would show up as a flat plateau over multiple decades in the compensated spectra. As can be observed in the figures there is no such plateaus present and we can therefore conclude the PSD is not power-law distributed. This can be attributed to the Lorentzian peak of the waveform as a Lorentzian pulse have exponential tails in the PSD [49].

The conditionally averaged waveform for all the simulations at $x = 100$ and $x = 141$ is plotted in figure 4.20, where they have been normalized by the maximum amplitude for easier comparison. At $x = 100$ we can observe that all the simulations have a similar pulse shape, the exception here being the full n model where the duration time of the pulse is noticeably shorter. For $x = 141$ we observe that the waveforms for non-Boussinesq approximated simulations

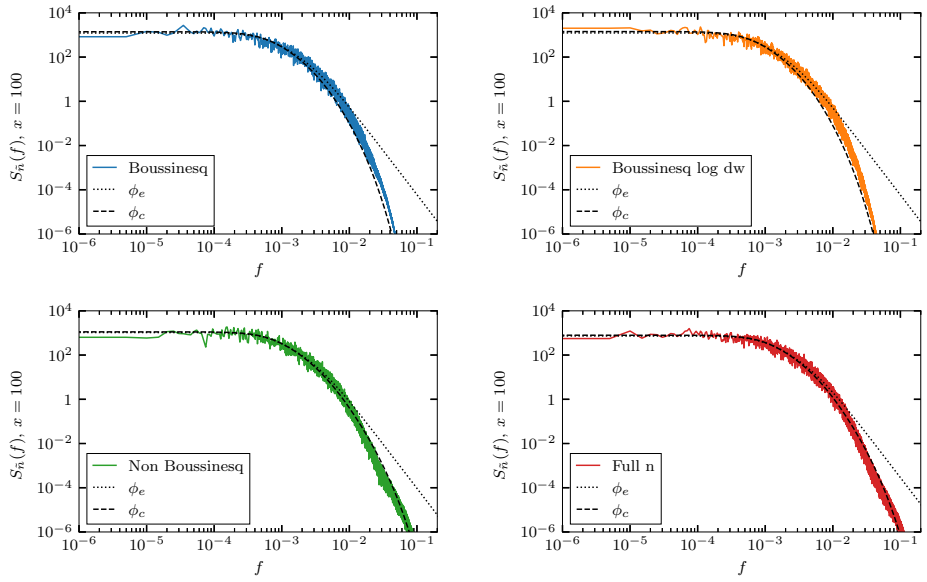

Figure 4.17: The power spectral density for the four main simulation cases at radial position $x = 100$. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

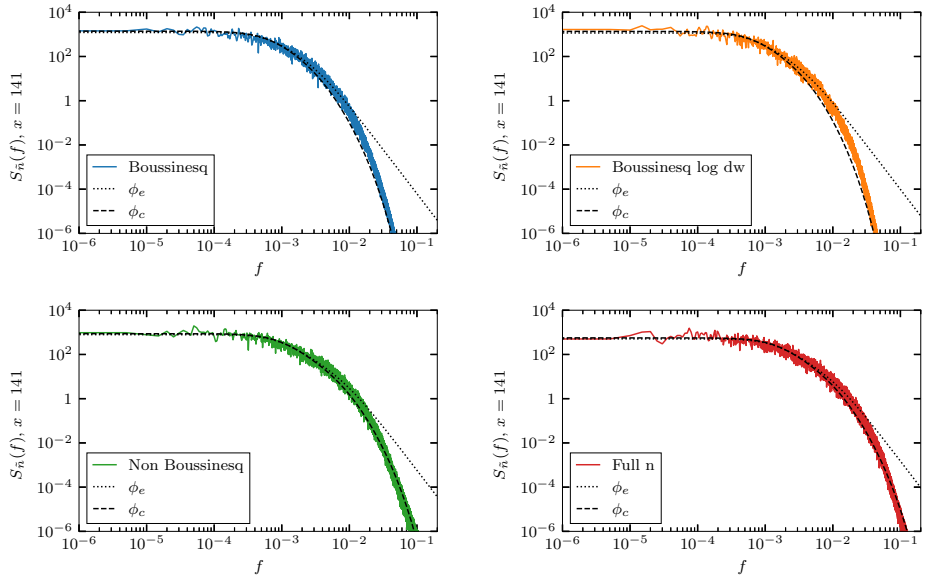

Figure 4.18: The power spectral density for the four main simulation cases at radial position $x = 140$. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

Table 4.3: The fit parameters for the PSD of a two-sided exponential fitted to the simulation data.

Simulation	x	τ_e	λ_e
Boussinesq	100	309.64	0.14
	140	307.81	0.14
Boussinesq log dw	100	313.03	0.14
	140	293.2	0.12
Non Boussinesq	100	264.76	0.14
	140	199.16	0.08
Full n	100	184.42	0.14
	140	128.64	0.09

Table 4.4: The fit parameters for the PSD of a two-sided exponential pulse convolved with a Lorentzian pule fitted to the simulation data.

Simulation	x	τ_e	λ_e	τ_l
Boussinesq	100	309.63	0.14	14.33
	140	307.80	0.14	14.58
Boussinesq log dw	100	313.02	0.14	15.83
	140	293.19	0.11	16.38
Non Boussinesq	100	264.77	0.14	5.85
	140	199.15	0.08	5.96
Full n	100	184.43	0.14	4.86
	140	128.63	0.10	4.12

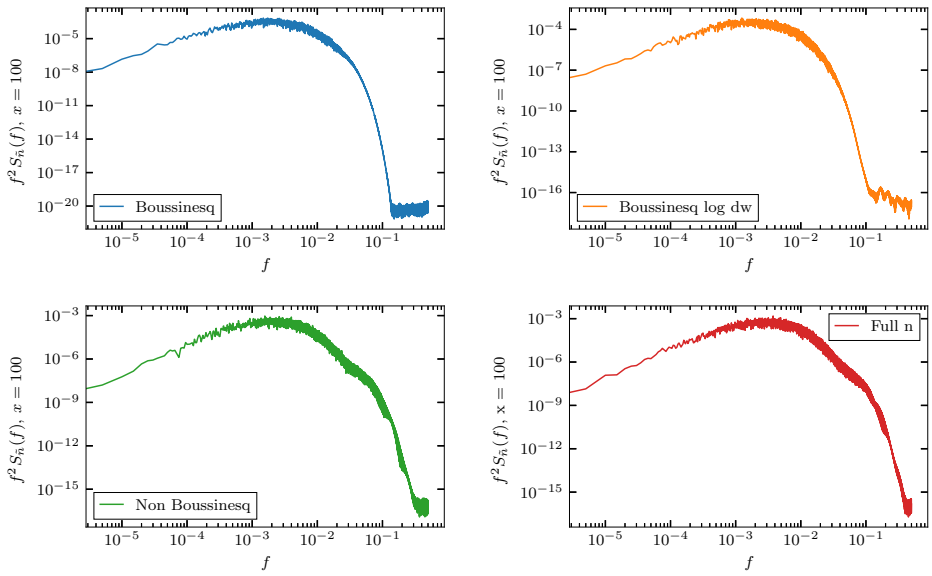

Figure 4.19: The compensated spectra for the four main simulation cases. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

diverge from the waveforms for the Boussinesq approximated simulations, which again seem to overlap each other.

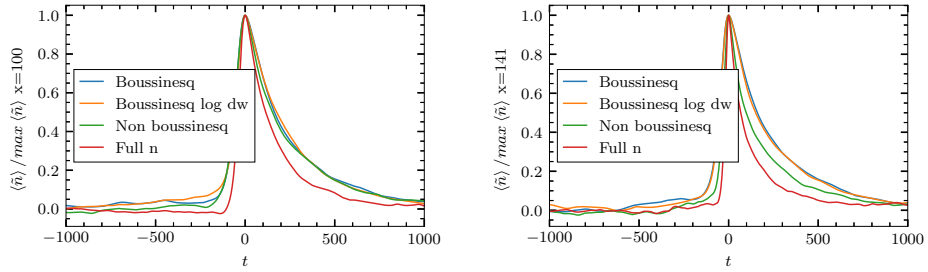

Figure 4.20: The conditionally averaged waveform for all the simulation. At position $x = 100$ (left), $x = 141$ (right).

The conditional average waveform for each of the simulations at different radial positions is plotted in figure 4.21. For the two simulations with the Boussinesq approximation, the change in the waveform is small reducing the duration time by small amounts, the exception is the ones at low radial positions. The non-Boussinesq approximated models have a significant change in duration time, where the model with a linear drift wave term changes the most.

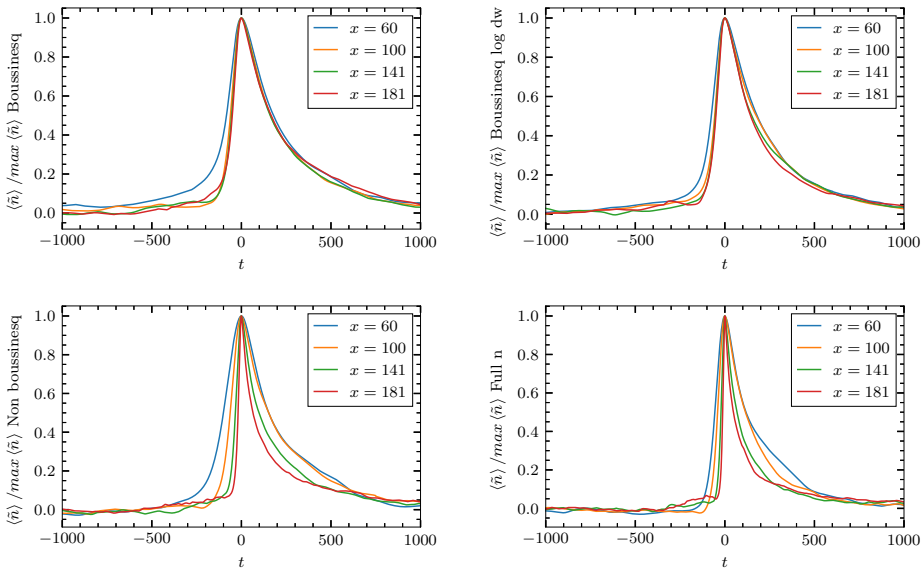

Figure 4.21: The conditionally averaged waveform for the four main simulation cases. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

In figures 4.22 and 4.23 we have plotted the conditional averaged waveform at $x = 100$ and $x = 141$, respectively. We have fitted them to a two-sided exponential using the values estimated from the PSD fit form table 4.3, as well as the convolution of a two-sided exponential with a Lorentzian using the values form table 4.4. For all cases, the convolved pulse gives a better fit for the rise and peak of the conditionally averaged waveform while the two-sided exponential pulse fits the tails better.

The cross conditional waveform of the radial velocity for all the simulation at $x = 60$, $x = 100$, $x = 141$, and $x = 181$ are shown in figure 4.24. The most distinct part of each of the plots is the noticeable difference in peak velocity. The difference in velocities seems to be more focused on whether the Boussinesq approximation is applied or not, with the two simulations where the Boussinesq approximation is used having the same peak velocity, while the peak velocity for the non-Boussinesq approximated cases seem to grow radially outwards. While both of the Boussinesq approximated cases have approximately the same peak amplitude, for the non-Boussinesq cases the full n model always has a greater peak than the pure non-Boussinesq case. Notice also that the peak in velocity is not centred at $t = 0$, this comes from the fact that the peak velocity is trailing after the density peak.

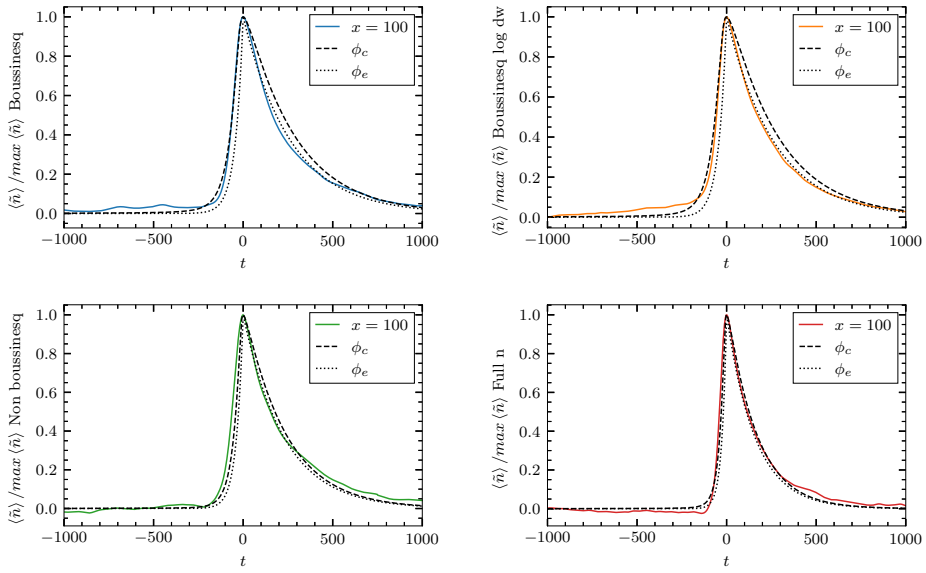

Figure 4.22: The conditional averaged waveform for the four main simulation cases at $x = 100$. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

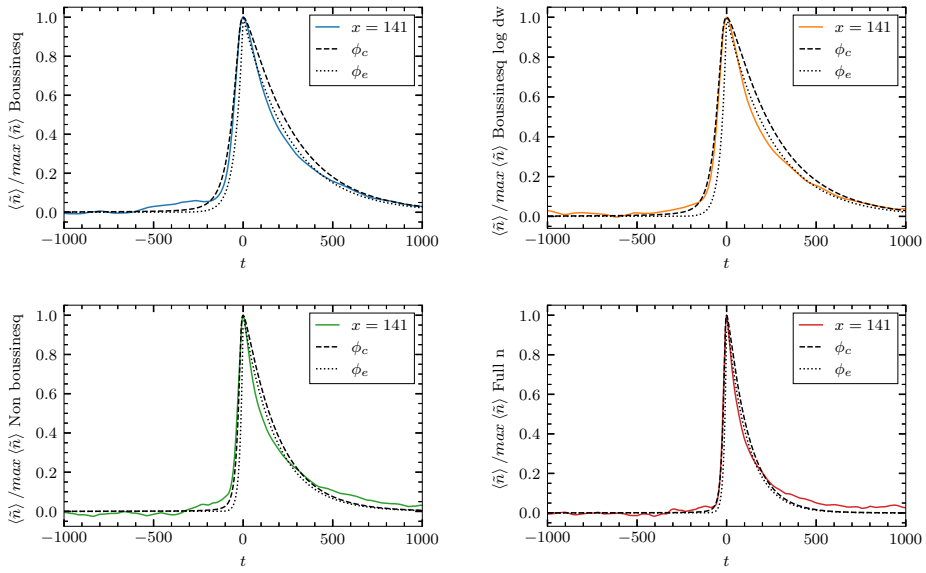

Figure 4.23: The conditional averaged waveform for the four main simulation cases at $x = 141$. Boussinesq approximated model (top left), Boussinesq approximated model with logarithmic drift waves (top right), non Boussinesq approximated model (bottom left), and the full n model (bottom right).

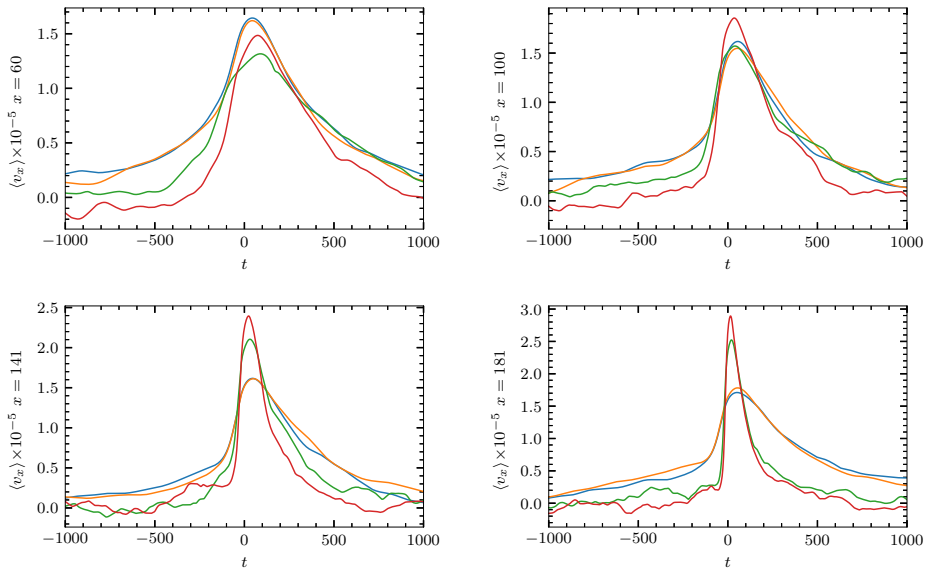

Figure 4.24: Cross conditional average waveform of velocities for all the simulations at different radial positions: $x = 60$ (top left), $x = 100$ (top right), $x = 141$ (bottom left), and $x = 181$ (bottom right)

As we previously also looked at the profiles for the different values of the sheath parameter we now in figure plot 4.25 the conditional averaged waveform as well as the cross conditional averaged waveform at $x = 100$ and $x = 141$. For the conditional averaged waveform we see that the lower the sheath parameter is the larger the duration time becomes. For the cross conditional waveform, the peak velocity decreases for weaker sheath parameters. The delay between the peak density and the peak velocity increases for weaker sheath parameters, this is most noticeable in the normalized cross conditional average.

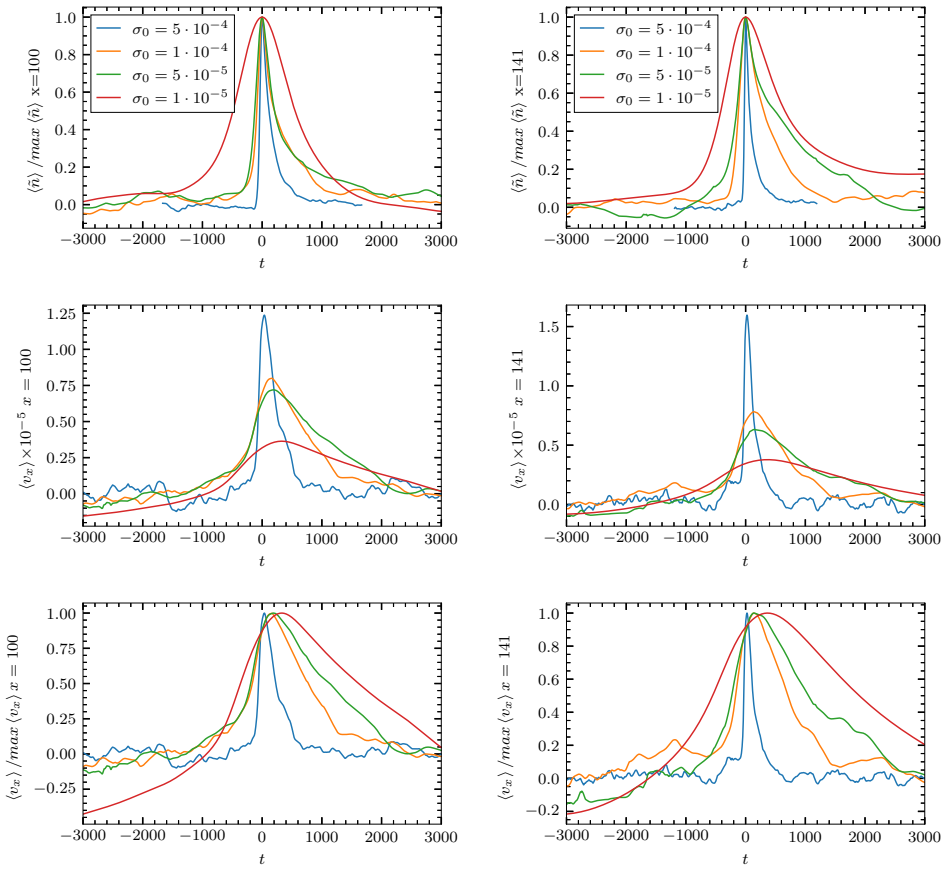

Figure 4.25: The full n model with four different sheath parameter strengths. The conditional average at $x = 100$ (top left) and $x = 141$ (top right). The cross conditional average at $x = 100$ (middle left) and $x = 141$ (middle right). The normalized cross conditional average at $x = 100$ (bottom left) and $x = 141$ (bottom right).

5 Discussion

In this chapter, we will discuss and interpret the results we got from the analysis of the different simulations.

5.1 The effect of non-linear drift wave term

The effect that the logarithmic drift wave term has is most notable in figure 4.4 where we plot both the electron flux and the poloidal velocity. The noticeable difference in the gradient of the poloidal velocity around the LCFS indicates the fact that the logarithmic term increases the shear flow. This higher shear flow will then decrease the amount of transport through the separatrix as it breaks up any structure wanting to move through it. This is confirmed by the profiles in figure 4.2 where there is a notable difference in the amplitudes between the two non Boussinesq approximated cases. The high peak amplitude for the mean density in the full n model in the edge region and the lower density throughout the rest of the simulation domain indicates that the logarithmic density term in the drift waves helps with the containment of the plasma. We also see this lower density in the relative fluctuation levels. While both non-Boussinesq cases follow the same shape as each other the amplitude difference between the two is again noticeable. The containment created by the switch between regions can also be observed in figure 4.6 where we plot the pure SOL case and the edge plus SOL case together. The mean density is higher at $x = 20$ for the case with the edge region while the mean density is higher

throughout the rest of the domain for the case without a separatrix.

Looking at the duration time τ_e of the two-sided exponential for the Boussinesq cases in tables 4.3 and 4.4, we observe for small x a higher value for the logarithmic case. Then moving radially outwards we observe a decrease in τ_e. This decrease is much larger than the decrease for the linear case. This decrease is also present in the asymmetry parameter λ_e as we have a decrease for the logarithmic case but no change in the linear case. Looking at the duration time τ_l for the Lorentzian pulse the value at $x = 100$ is higher for the logarithmic case, then moving radially outwards there is an increase for both Boussinesq cases with a much larger increase for the logarithmic case. For the non-Boussinesq cases, there is a noticeable difference in the duration time τ_e compared to the Boussinesq cases for the two-sided exponential at $x = 100$. The change in duration time τ_e when moving radially outward is larger for the linear case than for the logarithmic case, this is also the case for the change in asymmetry parameterλ_e. For the duration time λ_l of the Lorentzian pulse the linear case has higher values and increases when moving radially outwards, while for the logarithmic case the duration has a greater absolute change but decreases instead of increase. This indicates that including the logarithmic density in the drift wave term induces differences in how the duration times and asymmetry parameters change while moving radially outwards. This again is the product of the fact that the higher containment of plasma in the edge. As there is less plasma moving outwards through the domain small changes in properties are more likely to be noticed. This can also be observed in the parameters for the fit to the PDFs for the different cases in table 4.2. Here we see that the noise parameter for the cases with logarithmic density is generally higher than for the cases with linear density.

5.2 The effect of the Boussinesq approximation

The most pronounced difference between the different Boussinesq approximated cases and the non-Boussinesq approximated cases is the curve that appears in the radial profile of the relative fluctuations. The relative fluctuation levels for the non-Boussinesq approximated cases decrease moving radially outward from $x = 80$ while the Boussinesq approximated cases increase. This curve can be observed in figures 4.2, 4.5 and 4.6. The fact that this shows up in all the simulations shows that this curve is an effect of the physics of the SOL when the Boussinesq approximation is not present and not a consequence of the combination of the edge and SOL regions.

As shown in tables 4.3 and 4.4 there is a noticeable difference in duration time for both the two-sided exponential and Lorentzian pulses for the case

with and the case without the Boussinesq approximation. This indicates that the sizes of the filaments moving through the simulation domain are affected by the approximation. The sharper decrease in the duration times indicates that the structures also break up faster when radially moving outward. This can also be seen visually from figure 4.20 where both the non Boussinesq approximated cases have shorter duration time and sharper peaks. This can then explain the change in the relative fluctuation levels of the density observed in the non-Boussinesq case. The breaking up of the structures as they travel radially outwards gives the fluctuations less overall density equating to lower fluctuations.

The cross conditional averaged waveform for the velocities shown in figure 4.24 indicates a radial change of the peak velocities for each of the non-Boussinesq cases. Here the peak velocities increase moving radially outwards while for the Boussinesq approximated cases there is no discernible change in the peak velocity. This radial change in fluctuation velocity can also be seen in figure 4.3 where $v_{x\mathrm{rms}}$ is plotted. Here both the Boussinesq cases have a radially constant root mean square value, while both the non Boussinesq approximated cases increase radially outwards.

From the profiles in figures 4.2 and 4.4 there also seem to be an interplay between the logarithmic density and the term neglected in the Boussinesq approximation. For the mean poloidal velocity in figure 4.4, the gradient increases for both logarithmic cases, but the increase in gradient is larger for the full n case. We also see a deviation in the full n case from the other simulations in the particle flux where again there is a difference in the interaction with the drift wave term. The difference in shear flow strength is also clearly visible in the density profile of figure 4.2, where the density gradient over the separatrix is much larger for the full n case compared to the Boussinesq log dw case. This contribution to the zonal flow strength of the non-Boussinesq term has also been found in gyro-fluid simulations of a non Boussinesq approximated modified Hasegawa-Wakatani model [51].

5.3 The effect of the sheath parameter strength

From the scans done of different sheath parameter strengths, we see from figure 4.9 that for lower sheath parameter strength the signal becomes more intermittent. In figure 4.25 we also see that the duration time for the lower sheath strength increases, indicating that the structures broaden. Also from figure 4.25 we see from the cross conditional average for the lower sheath parameter strength we have lower fluctuation velocities, as the delay between the peak velocity and peak density increases this also indicates that the front of

the moving filament is not as steep as the for higher sheath parameters.

As seen in the radial mean profile in figure 4.8 the lower the sheath parameter is the broader the profiles become. This can be attributed to a reduction in parallel density dissipation allowing more of the density to propagate throughout the domain. A model describing radial profiles in relation to sheath parameter strength has been derived for the FPP model [43, 44]. As this model requires that the amplitudes and the velocities are constant we can not apply it directly, but it predicts that a lower sheath parameter strength will increase the density profile which agrees with our results.

6

Conclusion and outlook

In this thesis, we investigated how different approximations made to a 2D plasma fluid model influenced the statistical properties for numerical simulations of the model. In particular, we investigated the effect that the Boussinesq approximation has on the statistical properties of the turbulence produced, as well as what effect linearizing the particle density in the drift wave term has on the same statistics. We derive four sets of reduced fluid equations describing the evolution of electron density and plasma vorticity for a two-dimensional plane encapsulating the edge and SOL regions of a tokamak. Each set represented different combinations of the Boussinesq approximation and linearization of the drift wave particle density. We also gave a condition that needs to be fulfilled for the Boussinesq approximation to be applicable. We presented the simulation code using the BOUT++ framework that was utilized for the simulations. In addition to the simulations of the four main cases, we also did pure SOL simulations and a scan in sheath parameter strength. To collect the data from the simulations we used a grid of virtual probes measuring long time series with high temporal resolution at different radial and poloidal positions. We then introduced the FPP model which build the main statistical framework for the analysis we did. We presented plots showing differing radial profiles dependent on the approximations used. We observe higher confinement with lower particle density profiles throughout the simulation domain for the full n case. Additionally, the profiles show differing radial changes in the relative fluctuation levels of the particle density and differing radial changes for the RMS values of the velocity dependent on if the Boussinesq approximation was used or not. The presented radial profiles also show that the logarithmic

particle density in the drift wave term increased the shear flow in the edge region. In addition, the profile of varying sheath parameter strength indicates that the radial particle density profile scale length varies depending on this strength. We observed that PDFs of the particle density were best fitted by a gamma distribution with additive Gaussian noise. The plots of the PSDs show that they fit a convolution between a Lorentzian pulse and a two-sided exponential, where the duration times where highly effected by the relaxation of the Boussinesq approximation. We found that the conditionally averaged waveforms had larger radial change when the Boussinesq approximation was relaxed. Additionally, we found that the duration times of the conditionally averaged waveforms increased with lower sheath parameter strength. From the discussion, we found that having the logarithmic particle density in the drift wave term induces radial change in the parameters. We also found that the structures moving radially outwards in the non-Boussinesq case break into smaller structures at higher rates. We also found that there is an interplay between the term neglected in the Boussinesq approximation and the logarithmic density in the drift wave term as it increased the strength shear flow significantly compared to using one of the two terms.

Based on what is presented here it is clear that the Boussinesq approximation has an effect on the statistical properties of turbulence in the numerical simulations. But a more theoretical understanding of the differences between the models is desirable. This would give us a better understanding of the effects non-Boussinesq approximated models have on the radial transport of plasma and give a better explanation of the phenomena observed in the radial profiles presented in this thesis. For instance stability analysis of the different regions with differing degrees of approximations, or how the different approximations affect the energy integrals.

Some theoretical analysis of the effect of a non-Boussinesq approximated model [51, 52] has already been done using an altogether different approach than was presented here, namely using gyro-fluid equations. In a gyro-fluid model, you construct your equations with the fact that particles gyrate already in mind. This makes the equations harder to derive but gives innate handling of properties such as ion temperature, something we neglected due to the high complexity of dealing with it in a drift fluid model.

In addition to doing analytical studies, one could also investigate the gyro-fluid model through numerical simulations. The data from the numerical simulations could then be analysed in the same way as has been done in this thesis. For numerical simulations, there already exists a numerical framework developed for gyro-fluid simulations called FELTOR [53, 54]. Numerical investigation using FELTOR was originally also part of the thesis but could not be included due to numerical stability issues and time constraints. In appendix A.2 we therefore

present the reduction of a set of gyro-fluid equations into the Boussinesq approximated model from chapter 2. The advantages of using FELTOR over the BOUT++ would be that it can compile for bout CPU and GPU without needing to alter the code. This can then make use of the GPUs greater ability for parallel processing allowing for shorter simulation times. The FELTOR framework also handles non Boussinesq approximated models faster in a more efficient way allowing for longer time series in less time than the BOUT++ code.

In the last part of the discussion, we found that a simple stochastic model describing the density profile gave results in agreement with our findings. This model can and is being developed further to include the effects of time-dependent velocities, as we observed that the velocities of our fluctuations change over time this might give even better agreement with the results.

To simulate real tokamak plasmas one would like to use as consistent models as possible. The problem of full tokamak simulations is computation time, applying the Boussinesq approximation severely reduces this time. Having a greater understanding of what effects the Boussinesq approximation has on simulations will then possibly allow one to adjust for these effects afterwards. This will then still allow for manageable computation time while having more accurate results.

A

Appendix

A.1 Curvature operator

Deriving the curvature operator using the ExB drift

$$
\begin{aligned}
\nabla \cdot \boldsymbol{u}_E &= \nabla \cdot \left(\frac{1}{B} \boldsymbol{b} \times \nabla \phi \right) \\
&= \frac{1}{B} \left[\nabla \cdot (\boldsymbol{b} \times \nabla \phi) \right] + \boldsymbol{b} \times \nabla \phi \cdot \nabla \frac{1}{B} \\
&= \frac{1}{B} \left(\nabla \times \boldsymbol{b} \cdot \nabla \phi \right) - \frac{1}{B^2} \nabla B \cdot \boldsymbol{b} \times \nabla \phi \\
&= \frac{1}{B} \left(\nabla \times \boldsymbol{b} \cdot \nabla \phi - \nabla \ln B \cdot \boldsymbol{b} \times \nabla \phi \right) \\
&= \frac{1}{B} \left(\nabla \times \boldsymbol{b} + \nabla \ln B \times \boldsymbol{b} \right) \cdot \nabla \phi \\
&= \mathcal{K} \left(\phi \right)
\end{aligned}
$$

this then gives us that for a general scalar-field f we have

$$
\mathcal{K} \left(f \right) = \nabla \cdot \left(\frac{1}{B} \boldsymbol{b} \times \nabla f \right) = \frac{1}{B} \left(\nabla \times \boldsymbol{b} + \nabla \ln B \times \boldsymbol{b} \right) \cdot \nabla f \tag{A.1}
$$

A.1.1 Slab approximation

In toroidal geometry $\left(\hat{\boldsymbol{r}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\zeta}}\right)$ the gradient and curl operators are given as

$$\nabla\phi = \frac{\partial\phi}{\partial r}\hat{\boldsymbol{r}} + \frac{1}{r}\frac{\partial\phi}{\partial\theta}\hat{\boldsymbol{\theta}} + \frac{1}{R}\frac{\partial\phi}{\partial\zeta}\hat{\boldsymbol{\zeta}}, \tag{A.2}$$

$$\begin{aligned}\nabla \times \boldsymbol{F} = &\frac{1}{rR}\left(\frac{\partial}{\partial\theta}\left(RF_\zeta\right) - \frac{\partial}{\partial\zeta}\left(rF_\theta\right)\right)\hat{\boldsymbol{r}} \\ &+ \frac{1}{R}\left(\frac{\partial}{\partial\zeta}F_r - \frac{\partial}{\partial r}\left(RF_\zeta\right)\right)\hat{\boldsymbol{\theta}} \\ &+ \frac{1}{r}\left(\frac{\partial}{\partial r}\left(rF_\theta\right) - \frac{\partial}{\partial\theta}F_r\right)\hat{\boldsymbol{\zeta}},\end{aligned} \tag{A.3}$$

where $R = R_0 + r\cos\theta$. In a purly toroidal magnetic field

$$\boldsymbol{B} = \frac{B_0 R_0}{R}\hat{\boldsymbol{\zeta}}. \tag{A.4}$$

the curvature operator then becomes

$$\mathcal{K}\left(f\right) = \frac{1}{B}\left(-\frac{1}{R}\sin\theta\hat{\boldsymbol{r}} - \frac{1}{R}\cos\theta\hat{\boldsymbol{\theta}} - \frac{1}{R}\hat{\boldsymbol{\theta}}\right)\cdot\nabla f \tag{A.5}$$

then in the outboard mid-plane where $\theta \approx 0$ and $\hat{\boldsymbol{\theta}} \approx \hat{y}$ we get

$$\mathcal{K}\left(f\right) = \frac{1}{B}\left(-\frac{1}{R}\hat{\boldsymbol{y}} - \frac{1}{R}\hat{\boldsymbol{y}}\right)\cdot\nabla f \tag{A.6}$$

$$= -\frac{2}{BR}\frac{\partial f}{\partial y} \tag{A.7}$$

A.2 Gyro-Fluid

A set of gyro-fluid equations in slab coordinates describing the evolution of the electron density, ion gyro-center density and the polarization equation

$$\frac{\partial n}{\partial t} + \boldsymbol{u}_E\cdot\nabla n = \kappa n\frac{\partial\phi}{\partial y} - \kappa\frac{\partial n}{\partial y} + \Lambda_{n,\parallel} - \nu\Delta_\perp^2 n \tag{A.8}$$

$$\frac{\partial N}{\partial t} + \boldsymbol{U}_E\cdot\nabla N = \kappa N\frac{\partial\psi}{\partial y} + \tau\kappa\frac{\partial N}{\partial y} + \Lambda_{N,\parallel} - \nu\Delta_\perp^2 N \tag{A.9}$$

$$\nabla\cdot\boldsymbol{P}_2 = \Gamma_1 N - n, \tag{A.10}$$

where n is the electron density N is the ion gyro-centre density, ϕ is the potential ψ is the gyro-fluid potential,Γ_1 is the gyro-average operator, τ is

the ratio between ion and electron temperature $T_i/(T_e)$, κ is the curvature parameter, $\Lambda_{n,\parallel}$ and $\Lambda_{N,\parallel}$ is the parallel closure terms for the electrons and ion gyro-centre density respectively and ν is the hyper-diffusion coefficient. The gyro-centre velocity is defined as

$$\boldsymbol{U}_E = \frac{1}{B}\boldsymbol{b} \times \nabla\psi, \tag{A.11}$$

where the ψ is given to be

$$\psi = \Gamma_1\phi + \psi_2, \tag{A.12}$$

and Γ_1 is

$$\Gamma_1 = \frac{1}{1 - \frac{\tau}{2}\Delta_\perp}. \tag{A.13}$$

For the polarisation equation $\nabla \cdot \boldsymbol{P}$ the value of the term is highly dependent on the model used, for the long wave length limit (lwl) i.e $\rho_s k_\perp \ll 1$ we then have two values

$$\nabla \cdot \boldsymbol{P}_2 = \begin{cases} -\nabla \cdot \left(\frac{N}{B^2}\nabla_\perp\phi\right), & \text{N-OB} \\ -\frac{N_0}{B_0^2}\Delta_\perp\phi, & \text{OB} \end{cases}$$

where OB signifies the use of the (Oberbeck)-Boussinesq approximation and N-OB the lack thereof. ψ_2 is the polarization part of the gyro-fluid potential and also as two values in the lwl depending on the use of the Boussinesq approximation

$$\psi_2 = \begin{cases} -\frac{1}{2}\frac{(\nabla\phi)^2}{B^2}, & \text{N-OB} \\ 0, & \text{OB} \end{cases} \tag{A.14}$$

To start deriving our equations we set $\tau = 0$ and we get the cold ion approximation and our equations become

$$\frac{\partial n}{\partial t} + \boldsymbol{u}_E \cdot \nabla n = \kappa n\frac{\partial\phi}{\partial y} - \kappa\frac{\partial n}{\partial y} + \Lambda_{n,\parallel} - \nu\Delta_\perp^2 n \tag{A.15}$$

$$\frac{\partial N}{\partial t} + \boldsymbol{U}_E \cdot \nabla N = \kappa N\frac{\partial\psi}{\partial y} + \Lambda_{N,\parallel} - \nu\Delta_\perp^2 N \tag{A.16}$$

$$\nabla \cdot \boldsymbol{P}_2 = N - n \tag{A.17}$$

we want to invert the polarization equation so that we get $\boldsymbol{P}_2$ as a function of electron density.

$$N - n = -\nabla \cdot \left(\frac{N}{B^2}\nabla_\perp\phi\right) \tag{A.18}$$

$$N = \frac{n}{1 + \nabla_\perp\phi \cdot \nabla\frac{1}{B^2} + \frac{1}{B^2}\nabla_\perp^2\phi} \tag{A.19}$$

now by assuming the terms in the denominator are close to zero we can do a first order Taylor-expansion and get

$$N = n - \nabla \cdot \left(\frac{n}{B^2} \nabla_\perp \phi \right) \tag{A.20}$$

Putting this new definition for the gyro-center density into our evolution equation we get

$$\begin{aligned} \frac{\partial n}{\partial t} - \frac{\partial}{\partial t} \left[\nabla \cdot \left(\frac{n}{B^2} \nabla_\perp \phi \right) \right] + \boldsymbol{u}_E \cdot \nabla n - (\boldsymbol{u}_E \cdot \nabla) \nabla \cdot \left(\frac{n}{B^2} \nabla_\perp \phi \right) \\ + \boldsymbol{u}_{\psi_2} \cdot \nabla n - \left(\boldsymbol{u}_{\psi_2} \nabla \right) \nabla \cdot \left(\frac{n}{B^2} \nabla_\perp \phi \right) \\ = \kappa n \frac{\partial \phi}{\partial y} + \kappa n \frac{\partial \psi_2}{\partial y} - \kappa \nabla \cdot \left(\frac{n}{B^2} \nabla_\perp \phi \right) \frac{\partial \phi}{\partial y} - \kappa \nabla \cdot \left(\frac{n}{B^2} \nabla_\perp \phi \right) \frac{\partial \psi_2}{\partial y} \\ + \Lambda_{N,\parallel} - \nu \Delta_\perp^2 N \end{aligned}$$

where we also expanded the ψ term and defined $\boldsymbol{u}_{\psi_2} = \boldsymbol{b}/B \times \nabla \psi_2$.

Now inserting the terms from the evolution equation for the electron density and define our vorticity as $w = \nabla \cdot (n/B^2 \nabla_\perp \phi)$ we obtain

$$\begin{aligned} \frac{\mathrm{d}w}{\mathrm{d}t} + \kappa \frac{\partial n}{\partial y} - \boldsymbol{u}_{\psi_2} \cdot \nabla n \\ = -\kappa n \frac{\partial \psi_2}{\partial y} + \kappa w \frac{\partial \phi}{\partial y} + \Lambda_{w,\parallel} - \nu \Delta_\perp^2 w \end{aligned}$$

where we introduce the advective derivative and define $\Lambda_{w,\parallel} \equiv \Lambda_{n,\parallel} - \Lambda_{N,\parallel}$, as well as use the relation $w = n - N$. The cross terms between $\boldsymbol{u}_{\psi_2}$ and w is neglected as they brake the ordering of our model, and we now have a vorticity density equation.

We can reduce this to a vorticity equation by invoking the Boussinesq approximation ($w = n_0/B^2 \nabla_\perp^2 \phi$), where we all the terms involving ψ_2 have to be removed for energy conservation. This reduces our equation to

$$\frac{\mathrm{d}\nabla_\perp^2 \phi}{\mathrm{d}t} + B^2 \kappa \frac{\partial \ln n}{\partial y} = \kappa \nabla_\perp^2 \phi \frac{\partial \phi}{\partial y} + \frac{B^2}{n} \Lambda_{w,\parallel} - \nu \Delta_\perp^2 \nabla_\perp^2 \phi$$

The parallel closure term $\Lambda_{w,\parallel}$ is the sum of each of the closures that come from the combination of each of the edge and SOL closures. The parallel closure terms are given by

$$\begin{aligned} \Lambda_{n,\parallel} &= \alpha \left[\tilde{\phi} - \widetilde{\ln(n)} \right] - n\lambda \exp(\lambda_{sh} - \phi) \\ \Lambda_{N,\parallel} &= \lambda n \\ \Lambda_{w,\parallel} &= \alpha \left[\tilde{\phi} - \widetilde{\ln(n)} \right] - n\lambda \left[1 - \exp(\Lambda_{sh} - \phi) \right] \end{aligned}$$

where $\alpha = T_e k_\parallel^2 / \eta e^2 n \Omega_0 = \left(\rho_0 k_\parallel\right)^2 (m_i/m_e)\,(\Omega_0/\nu_e)\,(n/n_0)$ is the adiabaticity parameter, $\lambda = \rho_0/L_\parallel$ is the sheath parameter and Λ_{sh} is the sheath potential.

This gives combined with the electron continuity equation gives us our model equations

$$\frac{\mathrm{d}n}{\mathrm{d}t} + \kappa\left(\frac{\partial n}{\partial y} - n\frac{\partial \phi}{\partial y}\right) = \alpha\left[\tilde{\phi} - \widetilde{\ln(n)}\right] - n\lambda\exp(\lambda_{sh} - \phi) - \nu\Delta_\perp^2 n$$

$$\frac{\mathrm{d}\nabla_\perp^2\phi}{\mathrm{d}t} + B^2\kappa\frac{\partial \ln n}{\partial y} = \kappa\nabla_\perp^2\phi\frac{\partial\phi}{\partial y} + \frac{B^2}{n}\alpha\left[\tilde{\phi} - \widetilde{\ln(n)}\right] - B^2\lambda\left[1 - \exp(\Lambda_{sh} - \phi)\right] - \nu\Delta_\perp^2\nabla_\perp^2\phi.$$

This then shows that one can reduce a gyro-fluid model into the model used in in this thesis.

Bibliography

[1] Jeffrey P. Freidberg. *Plasma Physics and Fusion Energy*. Cambridge University Press, 2007.

[2] Backgrounder on radioactive waste. https://www.nrc.gov/reading-rm/doc-collections/fact-sheets/radwaste.html. accessed: 04.04.2022.

[3] T. Kenneth Fowler. Nuclear power—fusion. *Reviews of Modern Physics*, 71(2), 1999.

[4] Iter. https://www.iter.org/mach. Accessed: 31.01.2022.

[5] Searching for the perfect shape. https://www.iter.org/newsline/-/3037. Accessed: 01.02.2022.

[6] David Chandler | MIT News Office. Mit-designed project achieves major advance toward fusion energy. https://news.mit.edu/2021/MIT-CFS-major-advance-toward-fusion-energy-0908. Accessed: 31.01.2022.

[7] Geomagnetism frequently asked questions. https://www.ngdc.noaa.gov/geomag/faqgeom.shtml. Accessed: 31.01.2022.

[8] Differ. https://www.differ.nl/research/plasma-material-interactions. Accessed: 02.02.2022.

[9] O.E Garcia, J Horacek, R.A Pitts, A.H Nielsen, W Fundamenski, V Naulin, and J. Juul Rasmussen. Fluctuations and transport in the tcv scrape-off layer. *Nuclear Fusion*, 47(7):667–676, 2007.

[10] O E Garcia, J Horacek, R A Pitts, A H Nielsen, W Fundamenski, J P Graves, V Naulin, and J Juul Rasmussen. Interchange turbulence in the tcv scrape-off layer. *Plasma Physics and Controlled Fusion*, 48(1), 2005.

[11] D. A. D'Ippolito, J. R. Myra, and S. J. Zweben. Convective transport by intermittent blob-filaments: Comparison of theory and experiment.

Physics of Plasmas, 18(6):060501, 2011.

[12] A. H. Nielsen, H. L. Pécseli, and J. Juul Rasmussen. Turbulent transport in low- plasmas. *Physics of Plasmas*, 3(5):1530–1544, 1996.

[13] R.J. Maqueda, D.P. Stotler, and S.J. Zweben. Intermittency in the scrape-off layer of the national spherical torus experiment during h-mode confinement. *Journal of Nuclear Materials*, 415(1, Supplement):S459–S462, 2011. Proceedings of the 19th International Conference on Plasma-Surface Interactions in Controlled Fusion.

[14] Ralph Kube. *Dynamics and statistical properties of blob structures in scrape-off layer plasmas*. PhD thesis, 2014.

[15] J.W Connor, G.F Counsell, S.K Erents, S.J Fielding, B LaBombard, and K Morel. Comparison of theoretical models for scrape-off layer widths with data from compass-d, jet and alcator c-mod. *Nuclear Fusion*, 39(2):169–188, 1999.

[16] O.E. Garcia, R.A. Pitts, J. Horacek, A.H. Nielsen, W. Fundamenski, J.P. Graves, V. Naulin, and J. Juul Rasmussen. Turbulent transport in the tcv sol. *Journal of Nuclear Materials*, 363-365:575–580, 2007. Plasma-Surface Interactions-17.

[17] O. E. Garcia. Stochastic modeling of intermittent scrape-off layer plasma fluctuations. *Phys. Rev. Lett.*, 108:265001, Jun 2012.

[18] Gregor Decristiforo. Numerical computations of turbulent motions in magnetized plasmas, 2016.

[19] Boussinesq J. *Theorie Analytique De La Chleur*, 2, 1903.

[20] Oberbeck A. *Ann. Phys. Chem.*, page 271, 1879.

[21] P. C. Stangeby. *The plasma boundary of magnetic fusion devices*. Institute of Physics Pub., 2000.

[22] SI Braginskii. Transport processes in plasma. *Reviews of Plasma Physics*, 1965.

[23] Y. Sarazin and Ph. Ghendrih. Intermittent particle transport in two-dimensional edge turbulence. *Physics of Plasmas*, 5(12):4214–4228, 1998.

[24] Ph Ghendrih, Y Sarazin, G Attuel, S Benkadda, P Beyer, G Falchetto, C Fi-

garella, X Garbet, V Grandgirard, and M Ottaviani. Theoretical analysis of the influence of external biasing on long range turbulent transport in the scrape-off layer. *Nuclear Fusion*, 43(10):1013–1022, sep 2003.

[25] Y. Sarazin, Ph. Ghendrih, G. Attuel, C. Clément, X. Garbet, V. Grandgirard, M. Ottaviani, S. Benkadda, P. Beyer, N. Bian, and C. Figarella. Theoretical understanding of turbulent transport in the sol. *Journal of Nuclear Materials*, 313-316:796–803, 2003. Plasma-Surface Interactions in Controlled Fusion Devices 15.

[26] O. E. Garcia, V. Naulin, A. H. Nielsen, and J. Juul Rasmussen. Computations of intermittent transport in scrape-off layer plasmas. *Phys. Rev. Lett.*, 92:165003, Apr 2004.

[27] O. E. Garcia, V. Naulin, A. H. Nielsen, and J. Juul Rasmussen. Turbulence and intermittent transport at the boundary of magnetized plasmas. *Physics of Plasmas*, 12(6):062309, Jun 2005.

[28] O E Garcia, V Naulin, A H Nielsen, and J Juul Rasmussen. Turbulence simulations of blob formation and radial propagation in toroidally magnetized plasmas. *Physica Scripta*, T122:89–103, Jan 2006.

[29] J. R. Myra, D. A. Russell, and D. A. D'Ippolito. Transport of perpendicular edge momentum by drift-interchange turbulence and blobs. *Physics of Plasmas*, 15(3):032304, 2008.

[30] D. A. Russell, J. R. Myra, and D. A. D'Ippolito. Saturation mechanisms for edge turbulence. *Physics of Plasmas*, 16(12):122304, 2009.

[31] J.R. Myra, W.M. Davis, D.A. D'Ippolito, B. LaBombard, D.A. Russell, J.L. Terry, and S.J. Zweben. Edge sheared flows and the dynamics of blob-filaments. *Nuclear Fusion*, 53(7):073013, jun 2013.

[32] Nirmal Bisai, Amita Das, Shishir Deshpande, Ratneshwar Jha, Predhiman Kaw, Abhijit Sen, and Raghvendra Singh. Simulation of plasma transport by coherent structures in scrape-off-layer tokamak plasmas. *Physics of Plasmas*, 11(8):4018–4024, Aug 2004.

[33] Nirmal Bisai, Amita Das, Shishir Deshpande, Ratneshwar Jha, Predhiman Kaw, Abhijit Sen, and Raghvendra Singh. Edge and scrape-off layer tokamak plasma turbulence simulation using two-field fluid model. *Physics of Plasmas*, 12(7):072520, Jul 2005.

[34] N. Bisai, A. Das, S. Deshpande, R. Jha, P. Kaw, A. Sen, and R. Singh.

Formation of a density blob and its dynamics in the edge and the scrape-off layer of a tokamak plasma. *Physics of Plasmas*, 12(10):102515, 2005.

[35] A.H. Nielsen, G.S. Xu, J. Madsen, V. Naulin, J. Juul Rasmussen, and B.N. Wan. Simulation of transition dynamics to high confinement in fusion plasmas. *Physics Letters A*, 379(47):3097–3101, 2015.

[36] A H Nielsen, J Juul Rasmussen, J Madsen, G S Xu, V Naulin, J M Olsen, M Løiten, S K Hansen, N Yan, L Tophøj, and et al. Numerical simulations of blobs with ion dynamics. *Plasma Physics and Controlled Fusion*, 59(2):025012, 2016.

[37] Jeppe Olsen, Anders Henry Nielsen, Jens Juul Rasmussen, Jens Madsen, Thomas Eich, Bernhard Sieglin, and Volker Naulin. Scrape-off layer power fall-off length from turbulence simulations of asdex upgrade l-mode. *Plasma Physics and Controlled Fusion*, 60(8):085018, 2018.

[38] G. Decristoforo, A. Theodorsen, J. Omotani, T. Nicholas, and O. E. Garcia. Numerical turbulence simulations of intermittent fluctuations in the scrape-off layer of magnetized plasmas. *Physics of Plasmas*, 28(7):072301, Jul 2021.

[39] B.D. Dudson, M.V. Umansky, X.Q. Xu, P.B. Snyder, and H.R. Wilson. Bout++: A framework for parallel plasma fluid simulations. *Computer Physics Communications*, 180(9):1467–1480, 2009.

[40] Thomas Nicholas. *Reduced simulations of scrape-off-layer turbulence*. PhD thesis, University of York, March 2021.

[41] xbout. `https://github.com/boutproject/xBOUT`. Accessed: 15.05.2022.

[42] Xarray. `https://github.com/pydata/xarray`. Accessed: 015.05.2022.

[43] Gregor Decristoforo. *Numerical simulations and stochastic modeling of intermittent fluctuations in magnetized plasmas*. PhD thesis, 2021.

[44] O. E. Garcia, R. Kube, A. Theodorsen, and H. L. Pécseli. Stochastic modelling of intermittent fluctuations in the scrape-off layer: Correlations, distributions, level crossings, and moment estimation. *Physics of Plasmas*, 23(5):052308, 2016.

[45] A. Theodorsen, O. E. Garcia, R. Kube, B. LaBombard, and J. L. Terry. Universality of poisson-driven plasma fluctuations in the alcator c-mod scrape-off layer. *Physics of Plasmas*, 25(12):122309, 2018.

[46] O.E. Garcia, R. Kube, A. Theodorsen, J.-G. Bak, S.-H. Hong, H.-S. Kim, the KSTAR Team, and R.A. Pitts. Sol width and intermittent fluctuations in kstar. *Nuclear Materials and Energy*, 12:36–43, 2017.

[47] R Kube, O E Garcia, A Theodorsen, D Brunner, A Q Kuang, B LaBombard, and J L Terry. Intermittent electron density and temperature fluctuations and associated fluxes in the alcator c-mod scrape-off layer. *Plasma Physics and Controlled Fusion*, 60(6):065002, 2018.

[48] O. E. Garcia, R. Kube, A. Theodorsen, B. LaBombard, and J. L. Terry. Intermittent fluctuations in the alcator c-mod scrape-off layer for ohmic and high confinement mode plasmas. *Physics of Plasmas*, 25(5):056103, 2018.

[49] O. E. Garcia and A. Theodorsen. Skewed lorentzian pulses and exponential frequency power spectra. *Physics of Plasmas*, 25(1):014503, 2018.

[50] A Theodorsen, O E Garcia, and M Rypdal. Statistical properties of a filtered poisson process with additive random noise: distributions, correlations and moment estimation. *Physica Scripta*, 92(5):054002, apr 2017.

[51] M. Held, M. Wiesenberger, R. Kube, and A. Kendl. Non-oberbeck–boussinesq zonal flow generation. *Nuclear Fusion*, 58(10):104001, Jul 2018.

[52] M. Held, M. Wiesenberger, and A. Kendl. The collisional drift wave instability in steep density gradient regimes. *Nuclear Fusion*, 59(2):026015, jan 2019.

[53] Matthias Wiesenberger. *Gyrofluid computations of filament dynamics in tokamak scrape-off layers*. PhD thesis, 2014.

[54] Markus Held. *Full-F gyro-fluid modelling of the tokamak edge and scrape-off layer*. PhD thesis, 2016.