

INTERNSHIP REPORT

Pre-sawtooth crash q-profile reconstruction

UNIVERSITY OF TECHNOLOGY EINDHOVEN

and

DIFFER

Supervisors:

Author:

Karel VAN DE PLASSCHE

Dr. Roger JASPERS

Fabien JAULMES MSc

December 2015

Abstract

The global demand for energy is ever increasing. One of the energy solutions proposed to fulfill this demand is fusion energy. However, more research is needed before fusion energy can be successfully implemented in a reactor. One of the subjects of this research is Sawtooth crashes. A problem currently faced by researchers is a mismatch between experimental data, in this case from ASDEX Upgrade, and a software package used to reconstruct plasma equilibrium before a crash, FINESSE. The safety factor profile, also known as q -profile, was used to compare the ASDEX and FINESSE data. Another software package, PF2q, was designed in this report to solve this mismatch. PF2q, using the Grad-Shafranov equation and Finite Element Methods, was able to quickly estimate the q -profile from an input pressure- and F -profiles. This was then used to provide the experimenter with a visual tool to make the matching of data easier. PF2q calculates the q -profile within 1 % relative error, and was able to estimate the q -profile given input pressure and F radial profiles within 1 ms.

Contents

Abstract	ii
Contents	iv
1 Introduction	1
1.1 Fusion Energy	1
1.2 The sawtooth instability	2
1.3 Reconstruction of equilibria	6
1.4 Design requirements	7
1.5 Approach	8
2 Theory	10
2.1 The Grad-Shafranov equation	10
2.2 The q -profile	12
2.3 FINESSE	12
2.3.1 Inputs	12
2.3.2 Outputs	13
3 Methods	15
3.1 Definition of the grid	15
3.2 Rescaling the pressure	17
3.3 Rescaling the poloidal magnetic field	17
3.4 Estimating the q -profile	19
4 Results and Discussion	21
4.1 PF2q description	21
4.2 Calculation of the q -profile from input parameters	22
4.3 Estimation of the q -profile by varying input parameters	26
4.4 Flux-surface shape change by varying input parameters	30
5 Conclusion	32
6 Outlook	33
Appendix A <i>Description of PF2q code</i>	35
A.1 asdex.py	35

A.2	fem.py	36
A.3	tools.py	37
A.4	finesse.py	38
A.5	pf2qvis.py	39

Bibliography	41
--------------	----

Chapter 1

Introduction

1.1 Fusion Energy

We live in a world where the demand for energy is ever-increasing, because of the growing world population and the increasing energy use per person. The global energy consumption is expected to grow from 5.598×10^{20} J in 2014 by 37 % in 2040 [1, 2]. A combination of innovative new energy sources are needed to fulfill this demand with minimal load on the environment. Nuclear fusion, or fusion energy, is one strong candidate to fulfill part of this demand. Nuclear fusion is the process that fuels our sun and is a clean way to produce energy. In the sun, nuclear fusion is made possible by the huge gravitational forces caused by the mass of the sun. It is impossible to recreate these forces on earth, so fusion is achieved by the use of strong magnetic forces in a device called a tokamak, see figure 1.1.

A tokamak is a device using a magnetic field to confine a plasma in the shape of a torus. One tokamak currently being build is ITER, which will be the first tokamak to produce net power. In ITER, a mixture of deuterium (D) and tritium (T) is heated to 150 000 000 K, creating a plasma and causing the ions to fuse to produce helium, neutrons and energy. Since the kinetic energy of the neutrons is converted to energy as well, helium is the only waste product. Combining this with the abundance of the materials to make deuterium and tritium (water and lithium respectively) makes fusion a very attractive solution [3]. However, as one can imagine, controlling such a superheated plasma is challenging. And if the plasma is controlled, its even more difficult to make a sustainable cost effective power plant out of it, so there is still a lot of research needed to make a fusion reactor reality. Instabilities are a factor contributing to the difficulty of controlling the plasma. This report will focus on one of these instabilities expected to appear in ITER: The sawtooth instability.

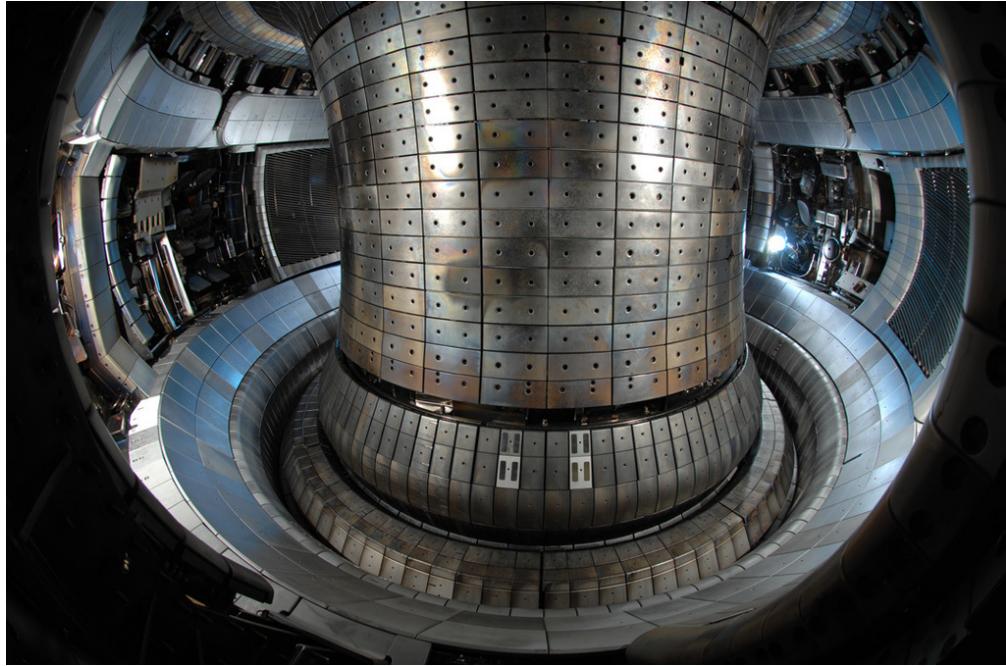


FIGURE 1.1: The ASDEX-Upgrade tokamak. [4].

1.2 The sawtooth instability

Different modes of operation are foreseen for ITER. These modes of operation can be characterized by their radial safety factor profiles, shown in figure 1.2.

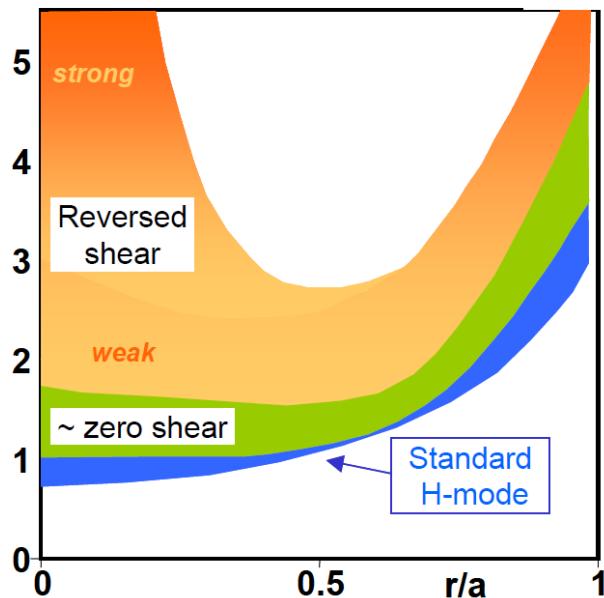


FIGURE 1.2: q -profiles of various scenarios expected in ITER. This report focuses on the blue and bottom of the green scenarios, in which sawtooth behavior is expected.

The safety factor, or q , is defined as in equations (1.1) to (1.3) [5, pp. 249–250].

$$q \equiv \frac{d\Phi}{d\Psi} \quad (1.1)$$

In which Ψ and Φ are the poloidal and toroidal flux respectively:

$$\Psi \equiv \frac{1}{2\pi} \iint B_p dS_\Psi \quad (1.2)$$

Where B_p is the magnetic field in the poloidal direction and dS_Ψ is a surface element in the horizontal cross-sectional plane. The integral is taken over a surface for which Φ is constant.

$$\Phi \equiv \frac{1}{2\pi} \iint B_\varphi dS_\Phi \quad (1.3)$$

Where B_φ is the magnetic field in the toroidal direction and dS_Φ is a surface element in the vertical cross-sectional plane. The integral is taken over a flux surface, which is a surface inside the plasma for which the poloidal magnetic flux Ψ is constant.

Each mode of operation shows different physical phenomena, which results in a different performance and thus in a different net power generated by the fusion reactor [6]. One of the instabilities associated with the $q = 1$ surface, matching the blue and bottom of the green scenario in figure 1.2, is the sawtooth instability. An example of the magnetic fields during pre-sawtooth equilibrium can be found in figure 1.3.

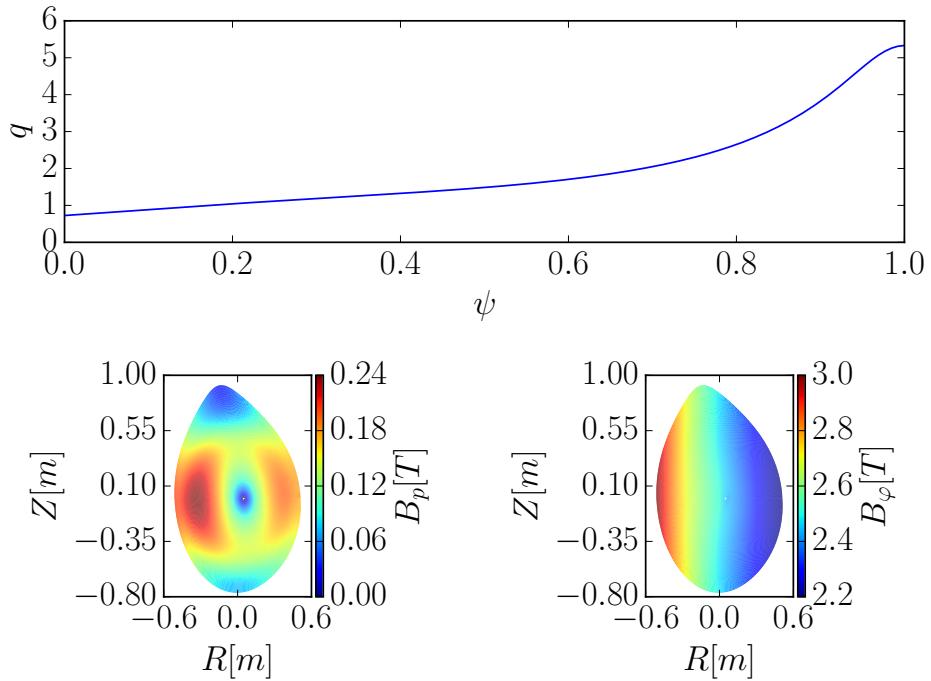


FIGURE 1.3: Example of the magnetic fields inside a tokamak. In the top plot the q -profile as function of the normalized poloidal field, as in equation (1.7). The shape of the q -profile is similar to the blue scenario in figure 1.2. On the bottom a plot of the poloidal field (on the left) and toroidal field (on the right) on a vertical cross-section of the tokamak.

The sawtooth instability is named after the characteristic drop in core pressure, which results in a saw-like shape. An example of this instability in JET is shown in figure 1.4. A few examples of effects influenced by sawteeth are Toroidicity-induced Alfvén eigenmodes (TAEs), and with it both energetic and thermal particle transports [7]. They can also be used to control neo-classical tearing modes (NTMs), which in turn influence the confinement of particles in a plasma and might even result in a catastrophic loss of the fusion plasma [8, 9]. Finally, they can be used to flush the impurities accumulated in the core out of the plasma [8]. All these effects influence the final performance of the fusion reactor, thus control and knowledge of sawtooth oscillations will result in a better fusion reactor.

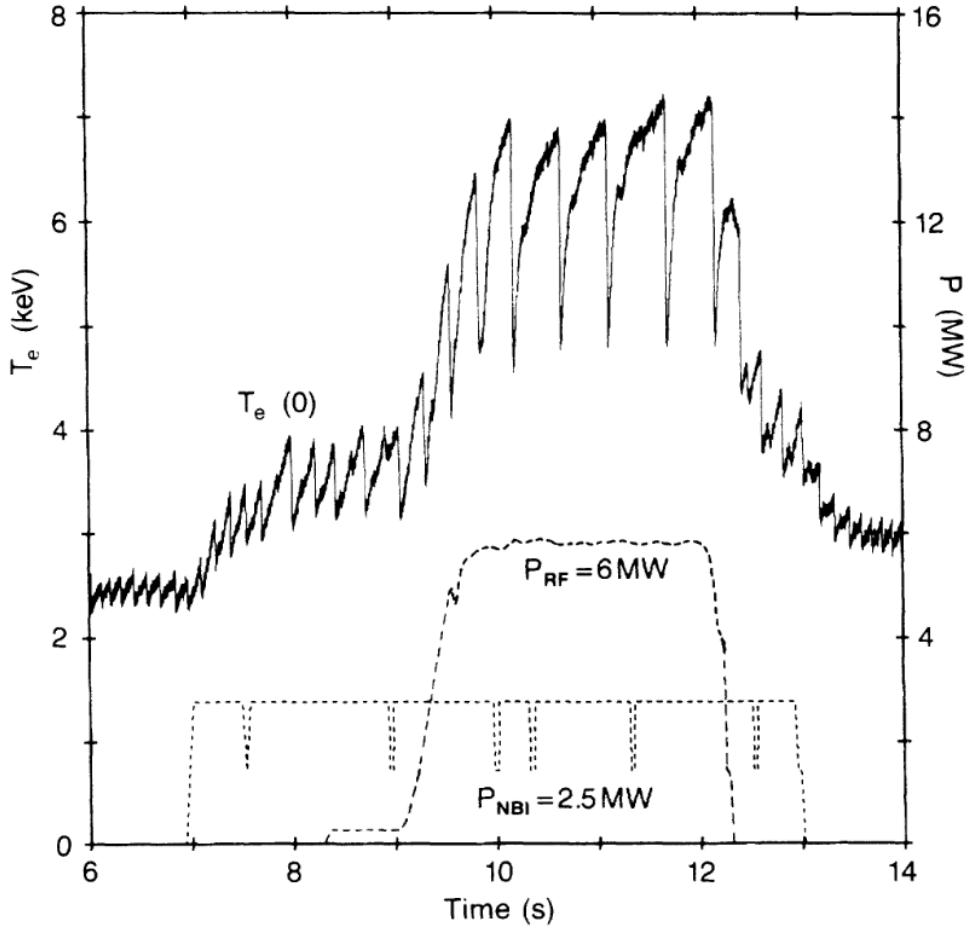


FIGURE 1.4: Usual behavior of sawtooth activity in JET with the increase in amplitude and period of sawteeth during additional heating ($I_p = 2.2$ MA, $q_{\Psi_1} = 5.2$) [10, pp. 2148]. Note the characteristic saw-like shape.

This report focuses on one of the tools used to gain insight in the behavior of high-energy particles during a sawtooth. `EBdyna_go` is a tool written in MATLAB specifically for this purpose [7, 11]. It starts with a plasma in (quasi-)equilibrium and it simulates the evolution of the electromagnetic fields during a sawtooth crash. After the magnetic and electric fields are known, it takes an initial distribution of charged particles and simulates their trajectories over time. Because these particles are charged, their paths are strongly dependent of the electromagnetic fields. Furthermore, the resulting electromagnetic fields are strongly dependent of their initial configuration, thus it is important to know the fields before the crash [12]. This is done by reconstructing the equilibrium before the crash using the `FINESSE` tool, as explained in the next section.

1.3 Reconstruction of equilibria

A program called **FINESSE** is used to reconstruct the magnetic fields during the pre-sawtooth equilibrium. **FINESSE** has as input the scaled pressure profile as function of the scaled poloidal magnetic flux ($\tilde{p}(\psi)$) and so called scaled F-profile as function of the scaled poloidal magnetic flux ($\tilde{F}(\psi)$), which is a measure for the toroidal current, and the scaling constants α and ϵ , see equations (1.4) to (1.7) [13]. It is also needed to give the shape of the outermost flux-surface (Ψ_1).

$$\epsilon \equiv \frac{a}{R_0} \quad (1.4)$$

Where ϵ is the inverse aspect ratio, R_0 is the distance from the geometrical center of the plasma to the symmetry axis and a is the half-width of the plasma

$$\alpha \equiv \frac{a^2 B_0}{\Psi_1} \quad (1.5)$$

Where α is the inverse flux parameter, B_0 is the external vacuum magnetic field at R_0 at the outermost flux surface and Ψ_1 is the poloidal flux at the outermost flux surface.

$$F \equiv R B_\varphi \quad (1.6)$$

$$\psi \equiv \tilde{\Psi} \equiv \Psi / \Psi_1 \quad (1.7)$$

The outputs include the q -profile and the magnetic fields, which we need as input for **EBdyna_go**. The exact inputs and outputs can be found in chapter 3. Data from the ASDEX Upgrade tokamak is used to provide input for **FINESSE**, as this is the same data as was used in the thesis by Jaulmes [7]. This was the thesis **EBdyna_go** was originally designed for. Included in this data are the pressure profile as function of the normalized radial coordinate ($p(\rho)$), the q -profile as function of the normalized radial coordinate ($q(\rho)$) and the total current at the plasma edge ($I_{\psi=1}$). These profiles are reconstructed from measurement data using the **TRANSP** transport solver [14, 15].

$$\rho \equiv \sqrt{\frac{\Phi}{\Phi_1}} \quad (1.8)$$

In which Φ_1 is the toroidal flux at the outermost flux surface.

An overview of all different in- and outputs can be found in table 1.1. Clearly, there is a mismatch between the ASDEX data set and the **FINESSE** input. Firstly, even though ρ depends solely on ψ , ψ itself is not known before running **FINESSE**. Secondly, q is an output from **FINESSE** and depends on $\tilde{F}(\psi)$ and $\tilde{p}(\psi)$ in a non-trivial way, making it impossible to recover $\tilde{F}(\psi)$ and $\tilde{p}(\psi)$ from q . Finally, some uncertainty remains in

the value of the plasma parameters as obtained from the transport solver. This report provides a way to fix this mismatch.

ASDEX constants	ASDEX data	FINESSE in	FINESSE out	EBdyna_go in
a	$p(\rho)$	$\tilde{F}^2(\psi)$	$\tilde{x}(\Psi, \vartheta)$	$x(\Psi, \vartheta)$
B_0	$q(\rho)$	$\tilde{p}(\psi)$	$\tilde{y}(\Psi, \vartheta)$	$y(\Psi, \vartheta)$
ϵ	$I(\psi = 1)$	α	$\rho(\Psi, \vartheta)$	$\mathbf{B}(\Psi, \vartheta)$
$\Psi_1(R, Z)$		ϵ	$\tilde{\mathbf{B}}(\Psi, \vartheta)$	
		$\Psi_1(a_m, b_m, \Theta)$	$\tilde{p}_{out}(\Psi)$	
			$q(\Psi)$	

TABLE 1.1: All different in- and outputs for **FINESSE** and **EBdyna_go**, as well as the supplied ASDEX-Upgrade data. The exact definitions can be found in chapter 3. Note that the $\tilde{F}(\psi)$ and $\tilde{p}(\psi)$ are given as a polinomal of maximum 10 coeficients.

1.4 Design requirements

This mismatch used to be solved by manually trying and adjusting **FINESSE** input parameters, running **FINESSE**, and comparing the output to the ASDEX data set. The equilibrium is successfully reconstructed when the q -profile given by **FINESSE**, the input p -profile and the current $I_{\psi=1}$ match the ASDEX-data. This process is time and labor intensive, as there are 22 parameters to be considered in general, and because running **FINESSE** is computationally expensive. Moreover, it is hard to predict exactly how the q -profile given by **FINESSE** will change with a change in input, because of the aforementioned non-trivial relation between **FINESSE** input and output. This means that it is generally hard to match the **FINESSE** data with the ASDEX Upgrade data, and thus to reconstruct the equilibrium.

To make matching of this data easier, a software package called **PF2q** was created. This software package provides the experimenter with a visual interface which uses quick estimation procedures to estimate the q -profile, thus working around the non-trivial relation between **FINESSE** input and output. To correctly use this estimation, it has to be precise enough as in some cases the influence of the q -profile on resulting particle orbits can be quite drastic. For example, three simulations based on an ASDEX Upgrade sawtooth are shown in figure 1.5.

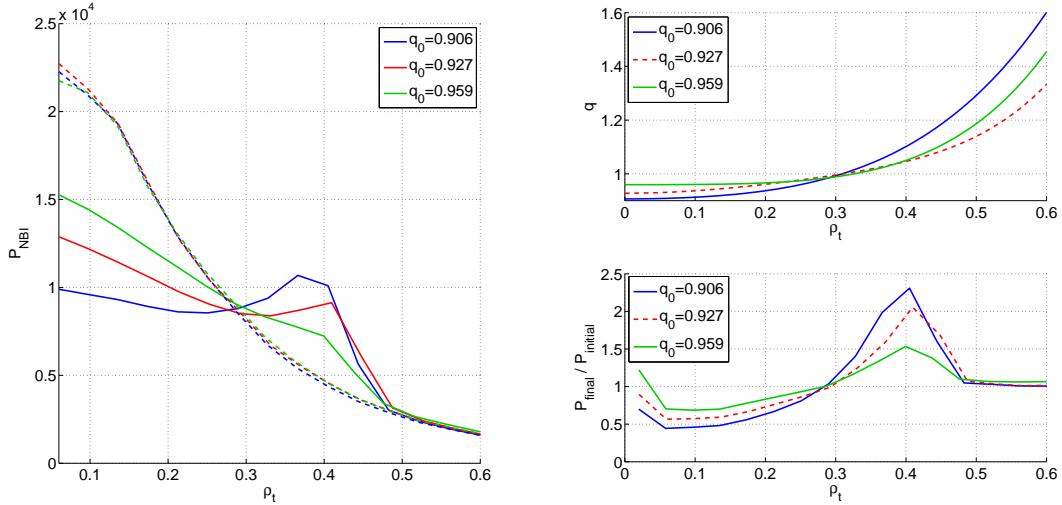


FIGURE 1.5: At the left the pre- and post-sawtooth NBI pressure as function of the normalized radial coordinate in dashed and solid lines respectively. On the upper right the pre-sawtooth q -profiles and on the lower left the relative change in pressure as function of the normalized radial coordinate. Note the influence of the q -profiles on the resulting pressure, most noticeably at $\rho = 0.4$.

It can be seen that a change of 6 % in q_0 results in a change of 50 % in post-sawtooth core pressure. And, as the $q = 1$ surface triggers the sawtooth crash, this point in the profile is particularly important. To try to prevent misrepresentation of the q -profile, the following design requirements are implemented:

1. Error in estimation of the $q(\rho)$ -profile (using minimal assumptions) from FINESSE input data ($\tilde{p}(\psi)$, $\tilde{F}(\psi)$, α and ϵ) at max 2 %.
2. Error in estimation during quick estimation procedure at max 5 % overall and 2 % around $q = 1$.

1.5 Approach

The following approach is proposed to answer these questions:

1. Determine how to go from FINESSE scaled pressure profile $\tilde{p}(\psi)$ to the physical profile $p(\psi)$. How to go from $\tilde{F}(\psi)$ to $F(\psi)$ is known from equation (1.6).
2. Calculate $q(\psi)$. If this calculated q -profile matches the q -profile from FINESSE's output within 2%, we can assume that our underlying methods to calculate q are correct.

3. Make predictions of q as function of $\tilde{p}(\psi)$ and $\tilde{F}(\psi)$. ψ is an output from **FINESSE** (or similar code), so we use **FINESSE** to get an initial ψ -profile. By then making only small adjustments to the input profiles, we can assume that ψ also does not change much. Afterwards, it is possible to estimate q as long as this assumption holds.
4. If we are able to estimate q reasonably well in a short enough time, we can make a visual tool. This tool can then be used to manually adjust the profiles and directly see the resulting q -profile, greatly reducing the time in which a profile can be matched manually. It can also be used to gain the needed insight for designing an algorithm that is able to match profiles automatically.

Chapter 2

Theory

2.1 The Grad-Shafranov equation

The Grad-Shafranov equation is the equilibrium equation in ideal magnetohydrodynamics (MHD) for a two dimensional plasma, for example the axisymmetric toroidal plasma in a tokamak. This equation is a two-dimensional, nonlinear, elliptic partial differential equation obtained from the reduction of the ideal MHD equations to two dimensions, often for the case of toroidal axisymmetry (the case relevant in a tokamak) [16]. The static ideal MHD equations are given by equations (2.1) to (2.3).

$$\mathbf{j} \times \mathbf{B} = \nabla p \quad (2.1)$$

$$\mu_0 \mathbf{j} = \nabla \times \mathbf{B} \quad (2.2)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.3)$$

Where p is the pressure, \mathbf{j} is the current density and \mathbf{B} is the magnetic field. The resulting Grad-Shafranov equation is given by equation (2.4) [5, pp. 270].

$$R \frac{\partial}{\partial R} \left(\frac{\partial \Psi}{\partial R} \right) + \frac{\partial^2 \Psi}{\partial Z^2} = -FF' - \mu_0 R^2 p' \equiv \mu_0 R j_\varphi \quad (2.4)$$

with: (equation (1.6) repeated for convenience)

$$F \equiv RB_\varphi \quad (2.5)$$

Where j_φ the toroidal current density, and (R, Z, φ) are the ordinary geometric coordinates. R is the distance to the symmetry axis, e.g. the distance to the center of the tokamak, Z the vertical coordinate and φ the toroidal coordinate. It is often convenient

to work in straight field line coordinates $(\Psi, \vartheta, \varphi)$ to simplify calculations and to increase stability of numerical solutions. Magnetic field lines appear straight in the (ϑ, φ) plane, and planes with constant Ψ are commonly called flux-surfaces. Note that going from (R, Z, φ) to $(\Psi, \vartheta, \varphi)$ is non-trivial, because $\Psi = \Psi(R, Z)$ and $\vartheta = \vartheta(\mathbf{B})$, thus the solution of the Grad-Shafranov equation is needed to calculate these coordinates. This does not pose a problem, as the Grad-Shafranov equation has to be solved in an iterative way in the first place because it is nonlinear. Unfortunately this makes solving the equation computationally expensive, and thus time expensive.

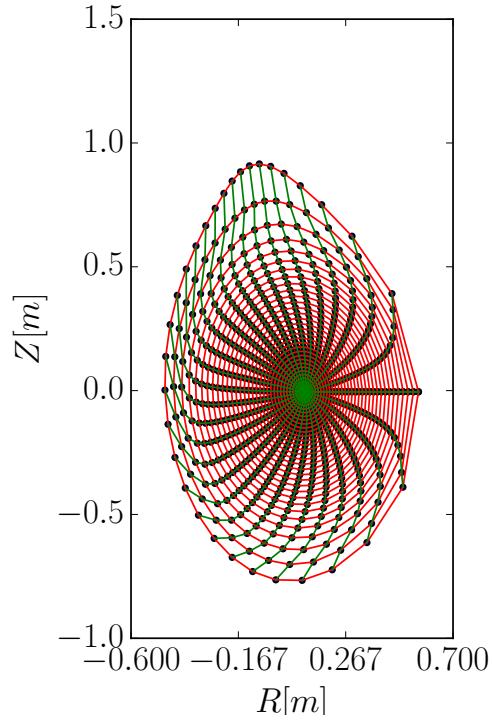


FIGURE 2.1: Picture of a straight field-line grid. Lines with constant ϑ are green, and lines with constant Ψ are red. The axes are in ordinary geometric coordinates.

The toroidal current density (j_φ) is related to the poloidal magnetic field (B_p) as in equation (2.6).

$$\oint_{\mathcal{B}(\Psi)} B_p d\ell = \iint_{\Psi} j_\varphi dS_\Phi \quad (2.6)$$

Where the left-hand side of the equation is taken over the boundary $\mathcal{B}(\Psi)$ of a flux-surface. Note that information about the local value of B_p is lost, because B_p is inside the integral.

2.2 The q -profile

The q -profile can now be calculated using equation (2.7). Due to toroidal symmetry, the ratio between toroidal and poloidal revolutions of the field lines can be found with an integral of $d\phi/d\theta$ over just one poloidal revolution [17]. Using that the pitch of the field lines is locally given by the ratio between the toroidal and poloidal field components, gives us:

$$q(\psi) = \frac{1}{2\pi} \int_0^{2\pi} \frac{d\phi}{d\theta} d\theta = \frac{1}{2\pi} \oint \frac{B_\varphi}{RB_p} d\ell \quad (2.7)$$

2.3 FINESSE

The FINESSE code (FINite Element Solver for Stationary Equilibria) computes axisymmetric magnetohydrodynamic equilibria in poloidal elliptic flow regimes for a variety of astrophysical and laboratory plasma configurations [13]. Only a small part of the code is used, as the focus of this report is on reconstructing static equilibria for tokamaks. An equilibrium is uniquely defined by the F - and p -profile (equation (1.6)), the scaling constants α and ϵ (equations (1.4) and (1.5)) as well as the shape of the outermost flux-surface (equation (2.12)) [13, Ch. 16.3][5, Ch. 2], so these are taken as input for FINESSE. However, the F - and p -profile are arbitrarily scaled internally until a solution is found. Consequently not the absolute value, but the shape is the important aspect, thus the shape of the input profile is generally conserved.

2.3.1 Inputs

The F - and p -profile are given to FINESSE as 9th order polynomials. The first polynomial is the scaled squared F -profile.

$$\sum \tilde{F}_i^2 \psi^i \equiv \tilde{F}^2 \equiv a_{\tilde{F}} F^2 + b_{\tilde{F}} \quad (2.8)$$

Where \tilde{F} is the scaled F -profile, F is the F -profile and $a_{\tilde{F}}, b_{\tilde{F}}$ are unknown constants. These can be determined with equation (2.9) and equation (2.10).

$$a_{\tilde{F}} = \frac{F_0^2 - F_1^2}{\tilde{F}_0^2 - \tilde{F}_1^2} \quad (2.9)$$

Where F_0 and F_1 are the physical values of pressure at the center and edge respectively, and \tilde{F}_0 and \tilde{F}_1 are the scaled values of the F -profile at the center and edge respectively. The physical values can be calculated with equation (1.6). The shift $b_{\tilde{F}}$ is then given by

equation (2.10).

$$b_{\tilde{F}} = F_0^2 - a_{\tilde{F}} \tilde{F}_0^2 \quad (2.10)$$

The second polynomial is the scaled pressure profile.

$$\tilde{p}_{in} \equiv \sum \tilde{p}_i \psi^i \equiv \tilde{p} \equiv a_{\tilde{p}} p + b_{\tilde{p}} \quad (2.11)$$

Where \tilde{p} is the scaled pressure profile, p is the pressure, and $a_{\tilde{p}}, b_{\tilde{p}}$ are unknown constants. A method to determine these constants will be presented in section 3.2.

Finally, the shape of the outermost flux surface (Ψ_1) has to be given. This is most conveniently done by specifying it through the Fourier decomposition in polar coordinates. The Fourier decomposition of the equation of Ψ_1 around a center R_0 in polar coordinates is given by equation (2.12).

$$\Psi_1(\Theta) = a \left[\frac{c_0}{2} + \Re \left(\sum_{m=1}^{m=N_p} (c_m \exp(im\Theta)) \right) \right] \quad (2.12)$$

Where $c_m = a_m - ib_m$ is complex and $\Re(x)$ is the real value of x . The parameters a_m and b_m are given to FINESSE. These parameters can be determined using equation (2.13).

$$c_m = \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{1}{a} (\Psi_1(\Theta) \cos(m\Theta) - i\Psi_1(\Theta) \sin(m\Theta)) d\Theta \quad (2.13)$$

The ordinary geometric coordinates can be recovered by equations (2.14) and (2.15).

$$R(\Theta) = R_0 + \Psi_1(\Theta) \cos(\Theta) \quad (2.14)$$

$$Z(\Theta) = R_0 + \Psi_1(\Theta) \sin(\Theta) \quad (2.15)$$

2.3.2 Outputs

FINESSE calculates the scaled geometric coordinates \tilde{x} and \tilde{y} , the density ρ (not used in this report), the scaled magnetic fields \tilde{B}_φ and \tilde{B}_p , the scaled pressure profile \tilde{p} , and the q -profile. All these values are calculated on N_p points of constant poloidal flux (Ψ) and N_R points of constant toroidal flux (ϑ). The scaled coordinates and fields are related to their physical values as in equations (2.16) to (2.21).

$$R \equiv R_0 + a\tilde{x} \quad (2.16)$$

Where \tilde{x} is the scaled dimensionless horizontal coordinate, R_0 the distance from the symmetry axis to the magnetic axis, and a the minor radius of the tokamak. R_0 and a

are generally known for a specific tokamak configuration.

$$Z \equiv a\tilde{y} \quad (2.17)$$

Where \tilde{y} is the scaled dimensionless vertical coordinate.

$$B_\varphi \equiv B_{\varphi,0}\tilde{B}_\varphi \quad (2.18)$$

Where \tilde{B}_φ is the scaled dimensionless toroidal magnetic field, and $B_{\varphi,0}$ is the toroidal magnetic field at the magnetic axis. $B_{\varphi,0}$ is generally known for a specific experiment. Note that this field is related to B_0 in equation (1.5) as in equation (2.19).

$$B_0 = \frac{B_{\varphi,0}}{d} \quad (2.19)$$

Where d is the on-axis diamagnetism of the plasma.

$$B_p \equiv B_{\varphi,0}\tilde{B}_p \quad (2.20)$$

Where \tilde{B}_p is the scaled dimensionless poloidal magnetic field.

$$p \equiv a_{\tilde{p}_{out}}\tilde{p}_{out} \quad (2.21)$$

Where \tilde{p} is the scaled pressure profile, p is the pressure, and $a_{\tilde{p}_{out}}$ is an unknown constant. A method to determine this constant will be presented in section 3.2. Note that in general $a_{\tilde{p}_{out}} \neq a_{\tilde{p}}$. The normalized toroidal flux ψ can be rescaled using equations (1.5) and (1.7).

Chapter 3

Methods

3.1 Definition of the grid

The code written in this report, called `PF2q`, uses a finite element method to solve integrals. As `FINESSE` is also a finite element solver, the grid of `PF2q` is chosen to coincide with that of `FINESSE`. The amount of points are defined by the parameters N_p and N_R , which are respectively the amount of points in the poloidal and radial direction. As mentioned in section 2.3, `FINESSE`s grid is defined on points of constant poloidal and toroidal flux. A quadrilateral grid is then uniquely defined by connecting these points. To increase precision, especially for grids with a small amount of points, these quadrilaterals are then divided into two triangles. An example of the grid with $N_p = N_R = 33$ can be found in figure 3.1. Note that in `FINESSE`, the first and last poloidal point overlap and are thus the same point.

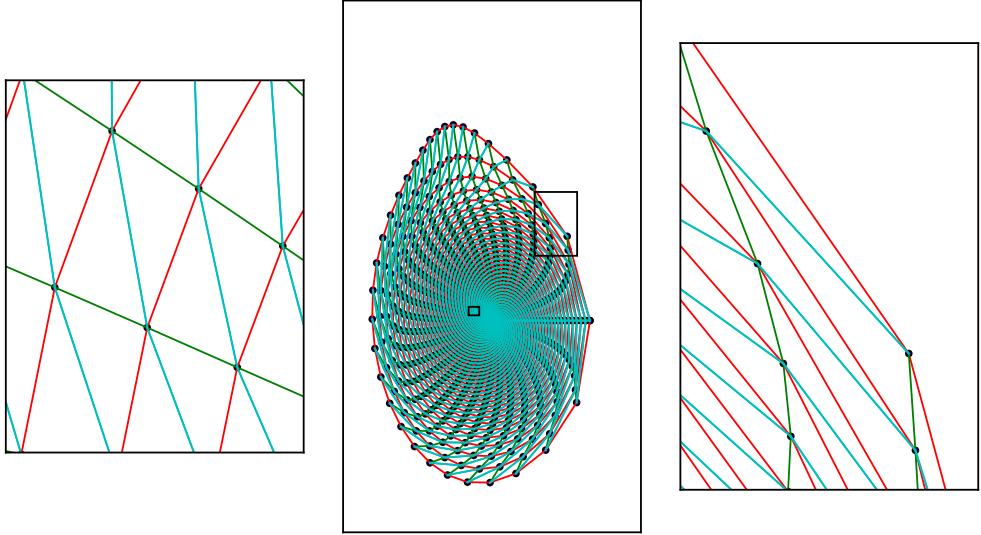


FIGURE 3.1: Picture of the grid. Lines with constant ϑ are green, lines with constant Ψ are red, and the triangulation is blue. To the left a zoom-in of the center of the grid, where the red contours are more or less oval and the green lines are more or less straight. To the right a zoom-in of the outside of the grid, where red contours trace the characteristic D-shape of the outermost flux surface and the green lines are bent.

In general, the surface of an triangle $\triangle ABC$ can then be calculated using equation (3.1).

$$dA = \frac{1}{2} |A_x(B_y - C_y) + B_x(C_y - A_y) + C_x(A_y - B_y)| \quad (3.1)$$

Where X_x and X_y are the x- and y-coordinates of the vertex respectively. The value of a quantity everywhere on this triangle can then be approximated by taking the average value of the three vertices. This then gives the framework to approximate integrals on the grid. These integrals converge to the real value for an infinite amount of points.

$$\int d\ell \xrightarrow{\text{code}} \sum dl \quad (3.2)$$

Where $d\ell$ is an infinitesimal small length and dl is a length on the PF2q grid.

$$\int dS \xrightarrow{\text{code}} \sum dA \quad (3.3)$$

Where dS is an infinitesimal small surface and dA is a surface on the PF2q grid. Also, by using toroidal symmetry of values on the grid.

$$\iiint dV = 2\pi \iint R dS \xrightarrow{\text{code}} 2\pi \sum R dA \quad (3.4)$$

Where dV is an infinitesimal small volume. Note that the grid cannot be determined a priori, as it uses both Ψ and Φ . For this reason, FINESSE is used to determine the grid.

3.2 Rescaling the pressure

The unknown constants in equation (2.11) need to be determined before the q -profile can be estimated. One way of rescaling \tilde{p} to its physical value p is by using β and β_p . β and β_p are defined by equation (3.5) and equation (3.6), and are outputs of FINESSE.

$$\beta \equiv 2\mu_0 \frac{\iiint p \, dV}{\iiint B_\varphi^2 \, dV} \xrightarrow{\text{code}} \frac{\sum_A R p_i \, dA_i}{\sum_A R B_{\varphi,i}^2 \, dA_i} \quad (3.5)$$

$$\beta_p \equiv 2\mu_0 \frac{\iiint ddV}{\iiint B_p^2 \, dV} \xrightarrow{\text{code}} \frac{\sum_A R p_i \, dA_i}{\sum_A R B_{p,i}^2 \, dA_i} \quad (3.6)$$

B_φ can be directly calculated with equation (2.18), B_p with equation (3.15). The sums are taken over all triangular elements. Note that all these a values are taken from the FINESSE output. The real pressure (p) can then be calculated using equation (2.21).

$$a_{\tilde{p}_{out}} \equiv \frac{1}{2} \left(\frac{\beta_{out}}{\beta} + \frac{\beta_{p,out}}{\beta_p} \right) \quad (3.7)$$

Where β_{out} and $\beta_{p,out}$ are the output β s from FINESSE, and β and β_p are calculated with $p = \tilde{p}_{out}$, where \tilde{p}_{out} is the scaled output pressure. This pressure is an output of FINESSE. The scaling factor $a_{\tilde{p}}$ in equation (2.11) is then given by equation (3.8).

$$a_{\tilde{p}} = \frac{p_0 - p_1}{\tilde{p}_0 - \tilde{p}_1} \quad (3.8)$$

Where p_0 and p_1 are the physical values of pressure at the center and edge respectively, and \tilde{p}_0 and \tilde{p}_1 are the scaled values of pressure at the center and edge respectively. The shift $b_{\tilde{p}}$ is then given by equation (3.9).

$$b_{\tilde{p}} = p_0 - a_{\tilde{p}} \tilde{p}_0 \quad (3.9)$$

3.3 Rescaling the poloidal magnetic field

Because B_p is inside the left-hand integral of equation (2.6), information about the local value of B_p along the flux-surface is lost. This information is important to accurately determine B_p and thus the q -profile. The local information can be calculated using purely geometric means [18]. However, an easier method is possible if the magnetic fields are already known beforehand, for instance by running FINESSE. It can then be assumed that the fluctuation of B_p does not change too much if the shape of the flux-surfaces these fields create does not change too much. The variation along the flux-surface can

be described by a scaling factor a_{B_p} using equation (3.10).

$$a_{B_p}(l) = \frac{B_p}{\oint_{\mathcal{B}(\Psi)} B_p d\ell} \xrightarrow{\text{code}} \frac{B_p}{\sum_p B_p dl} \quad (3.10)$$

Where B_p is the poloidal magnetic field given by FINESSE and dl is the distance between two neighboring grid points on the same flux surface. The sum is taken over all poloidal points excluding the last point, since this point is the same as the first point by FINESSE convention. The poloidal field can then be estimated by first assuming B_p is constant and calculate it with equation (3.11).

$$B_{p,const} = \frac{I_{encl}}{L} \quad (3.11)$$

Where L is the total length of the boundary of the flux surface and I_{encl} the enclosed current.

$$L \equiv \oint_{\mathcal{B}(\Psi)} dl \xrightarrow{\text{code}} \sum_p dl \quad (3.12)$$

$$I_{encl} \equiv \iint j_\varphi dS_\Phi \xrightarrow{\text{code}} \sum_A j_{\varphi,i} dA_i \quad (3.13)$$

Where j_φ can be calculated with equation (2.4), which is rewritten in equation (3.14). The real B_p can then be found by rescaling it as in equation (3.15).

$$j_\varphi = -\frac{1}{2\mu_0 R} \frac{dI^2}{d\Psi} - R \frac{dp}{d\Psi} \quad (3.14)$$

$$B_p = a_{B_p} B_{p,const} \quad (3.15)$$

An example of the variation of B_p along a flux surface can be found in figure 3.2.

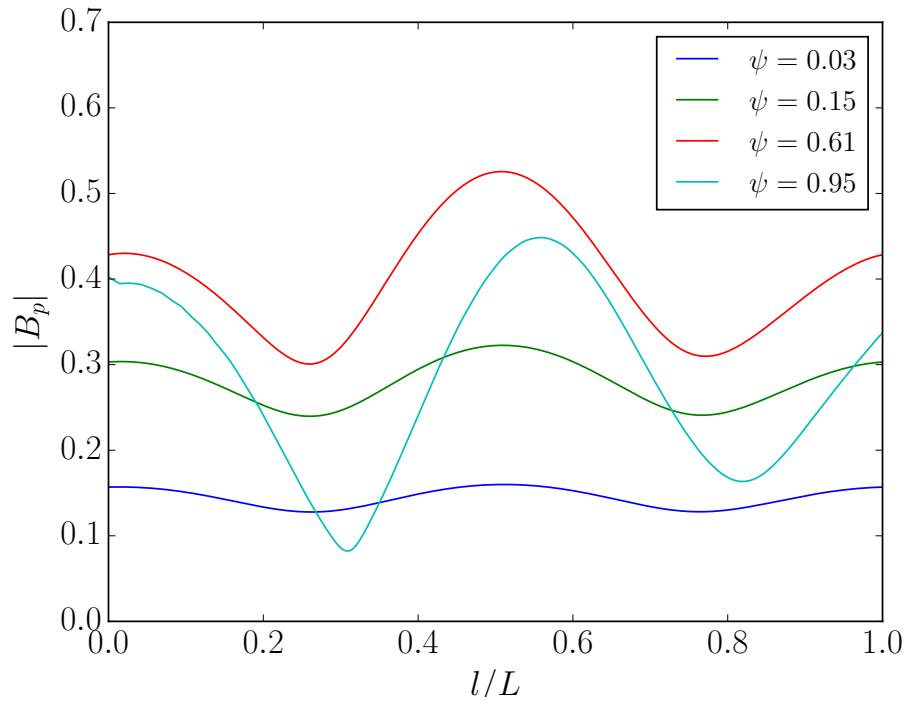


FIGURE 3.2: The value of B_p along a flux-surface contour. The horizontal axis displays the normalized distance along the contour. The fluctuation depends on the flux surface ψ .

3.4 Estimating the q -profile

Finally, the q -profile can be calculated using the discretization of equation (2.7), which is shown in equation (3.16):

$$q = \oint \frac{B_\varphi}{RB_p} d\ell \xrightarrow{\text{code}} \sum_p \frac{B_\varphi}{RB_p} dl \quad (3.16)$$

The q -profile can be calculated using the following steps:

1. Run FINESSE for the initial grid.
2. Calculate the real p - and F -profile using equations (1.6) and (2.21) from FINESSE output.
3. Calculate the B_p fluctuation factor using equation (3.10).
4. Calculate the input scaling factors using equations (2.9), (2.10), (3.8) and (3.9).
5. Rescale the input profiles to their physical values using the input scaling factors using equations (2.8) and (2.11).

6. Calculate B_p with equations (3.11) and (3.15) and B_φ with equation (2.18).
7. Calculate the q -profile with equation (3.16).

The q -profile can now be predicted after a change in input profiles \tilde{p} and \tilde{F} if the following assumptions hold by repeating step 5 – 7:

1. Assume F_0 , F_1 , p_0 , and p_1 stay the same. This means the input scaling factors stay the same too.
2. Assume the shape of the flux surfaces does not change. This means the grid and B_p fluctuation factor stay the same too.

The process has to be started from the first step if the assumptions break. This has to be repeated until the q -profile given by FINESSE, the input p -profile and the current $I_{\psi=1}$ match the ASDEX data.

Chapter 4

Results and Discussion

4.1 PF2q description

The main result of this report is the software tool `PF2q`. It is written in Python and contains all the methods described in this report. `PF2q` is set up in a modular way, separating functions per subject. An overview of all the modules can be found in table 4.1.

Name	Description
<code>asdex</code>	Read in and manipulate ASDEX Upgrade data sets
<code>fem</code>	Create the triangular FEM grid and calculate surface and contour integrals as described in section 3.1
<code>finesse</code>	Read and write FINESSE in- and output files, carry out rescaling and estimations described in sections 3.2 to 3.4
<code>tools</code>	General tools used in other modules
<code>pf2qvis</code>	Draw the visual interface

TABLE 4.1: List of modules designed for `PF2q`.

All these modules can be used separately, and they are defined keeping the visual interface from `pf2qvis` in mind. A picture of the default visual interface can be found in figure 4.1.

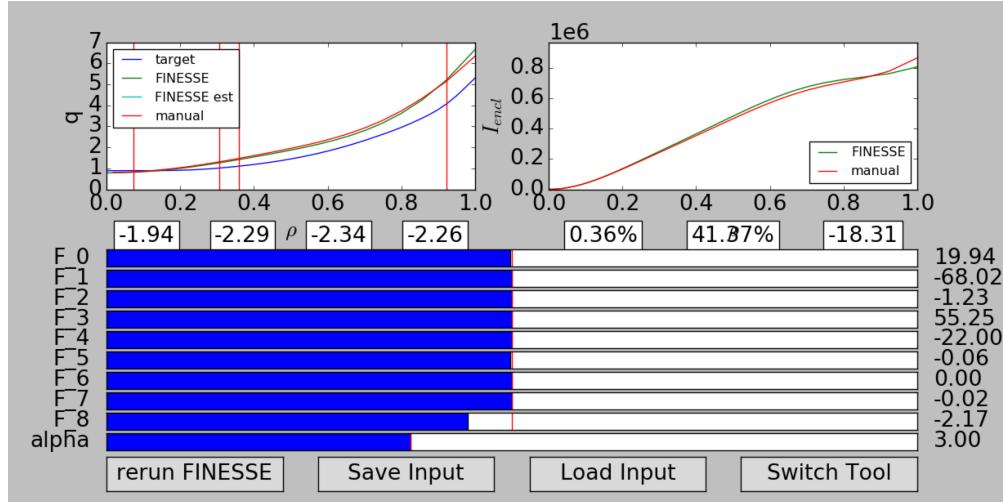


FIGURE 4.1: Screen capture of the visual interface of PF2q. On the top left, a plot of the q -profile as given by FINESSE, the target, and the quick estimate are shown. Under the plot the current relative error at the vertical lines in the plot are shown. On the top right a plot of the enclosed current of the quick estimate, and as given by FINESSE are shown. Under the plots the estimated β and β_p and the sum of the coefficients are shown. Under the plots are sliders to adjust the first 9 coefficients of the F -profile, as well as a slider for α . At the bottom there are buttons to re-run FINESSE for when the estimation starts to diverge to much from the FINESSE output, and button to save and load input files and switch the sliders so that the p -profile can be entered.

4.2 Calculation of the q -profile from input parameters

The magnetic field B_φ and B_p can now be calculated using the steps described in section 3.4. All results in this section use ASDEX Upgrade as base, so $B_{\varphi,0} = 2.64$, $a = 0.513$, and $\epsilon = 0.3145$. α is kept constant at 2.4, and the input profiles can be found in table 4.2. While this data set is one that was a result from manual matching, all results in this chapter are valid for any random input profile.

	A	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
\tilde{F}^2	1	20	-47	-1.5	-1	30.3	16	-3.7	-3.5	-12.5	-2.2
\tilde{p}	11	1	-5.73	20.7	-46.3	61.9	-44.8	13.2	0.0	0.0	0.0

TABLE 4.2: Used input profiles. In which x_i is \tilde{F}_i^2 for the \tilde{F}^2 -profile and \tilde{p}_i for the \tilde{p} -profile.

The relative error between FINESSE output and the calculated fields can be found in figure 4.2, where the relative error η of a quantity x is defined in equation (4.1).

$$\eta_x \equiv \left| \frac{x_{estimate}}{x_{real}} - 1 \right| = \left| \frac{x_{PF2q}}{x_{FINESSE,out}} - 1 \right| \quad (4.1)$$

And the average relative error $\bar{\eta}_x$ is simply the arithmetic average.

$$\bar{\eta}_x = \frac{\sum \eta_x}{N} \quad (4.2)$$

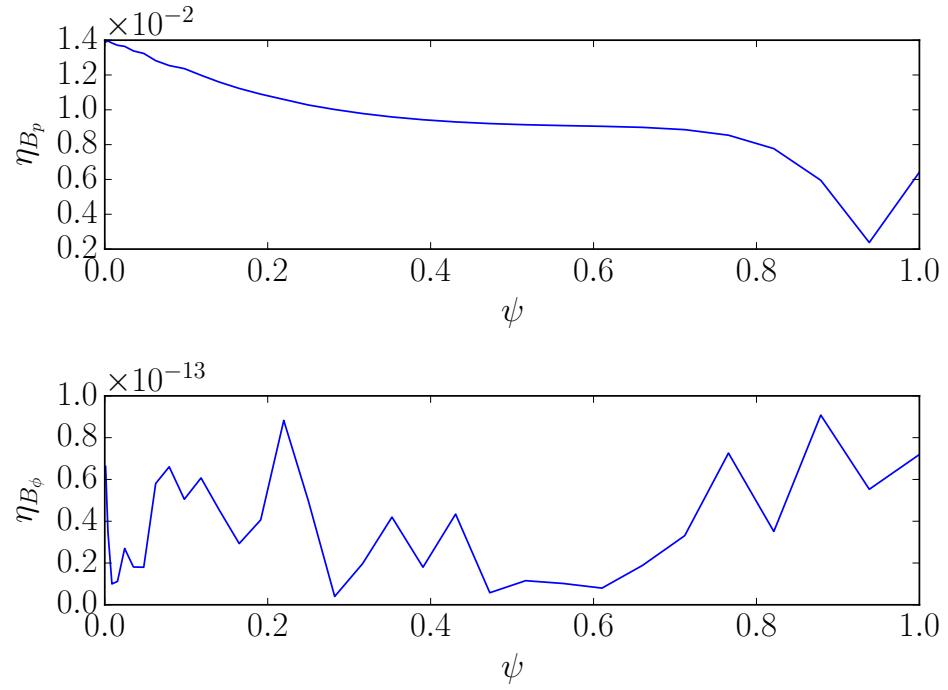


FIGURE 4.2: Relative error of the poloidal field η_{B_p} and the toroidal field η_{B_φ} dependent on ψ . In this plot $N_R = N_t = 33$. Note that $\mathcal{O}(\eta_{B_p}) = 10 \times 10^{-2}$ and $\mathcal{O}(\eta_{B_\varphi}) = 10 \times 10^{-13}$, both below the design requirement for q of 2 %. The edge effect in η_{B_p} is not further investigated.

It is expected that the relative error in B_p arises from numerical effects, given the extremely low value and irregular shape. The error in B_p is 11 orders of magnitude bigger, and will thus be the main source of error in the estimation of the q -profile. Still, both errors are below 2 %, giving confidence that that the goal of an average error of 2 % in the q -profile can be reached. The relative error of the q -profile can be found in figure 4.3.

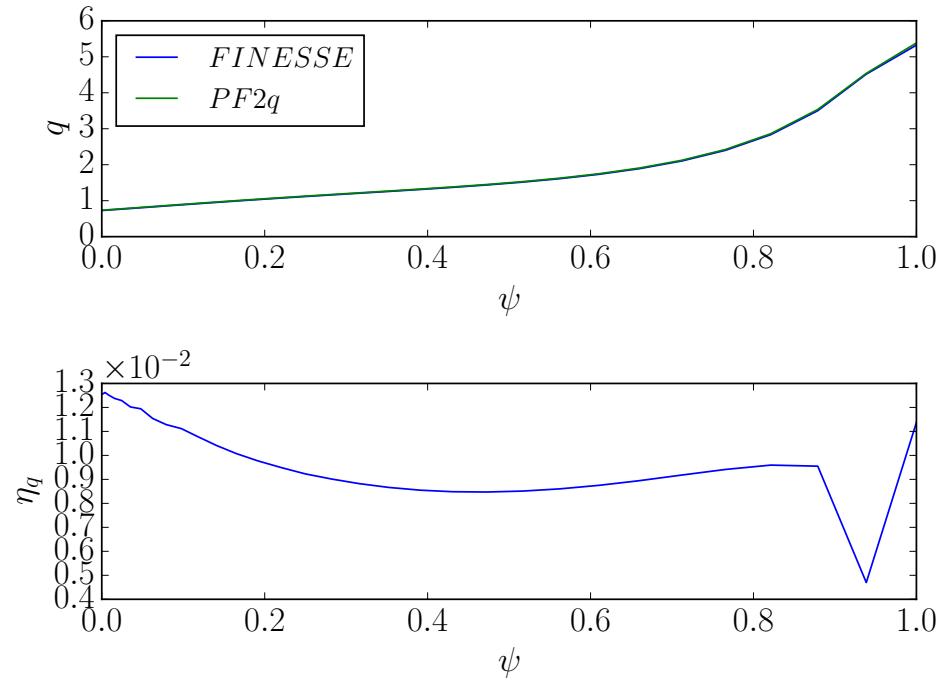


FIGURE 4.3: q -profile and its relative error η_q . In this plot $N_R = N_t = 33$. The relative error of the whole profile is below the design requirement of 2 %. The edge effect in η_q is not further investigated.

As can be seen, the relative error is below 2 % for the whole q -profile. As PF2q uses FEM to make its estimations, it is expected that there is an effect of the number of points on the error. This effect can be seen in figure 4.4.

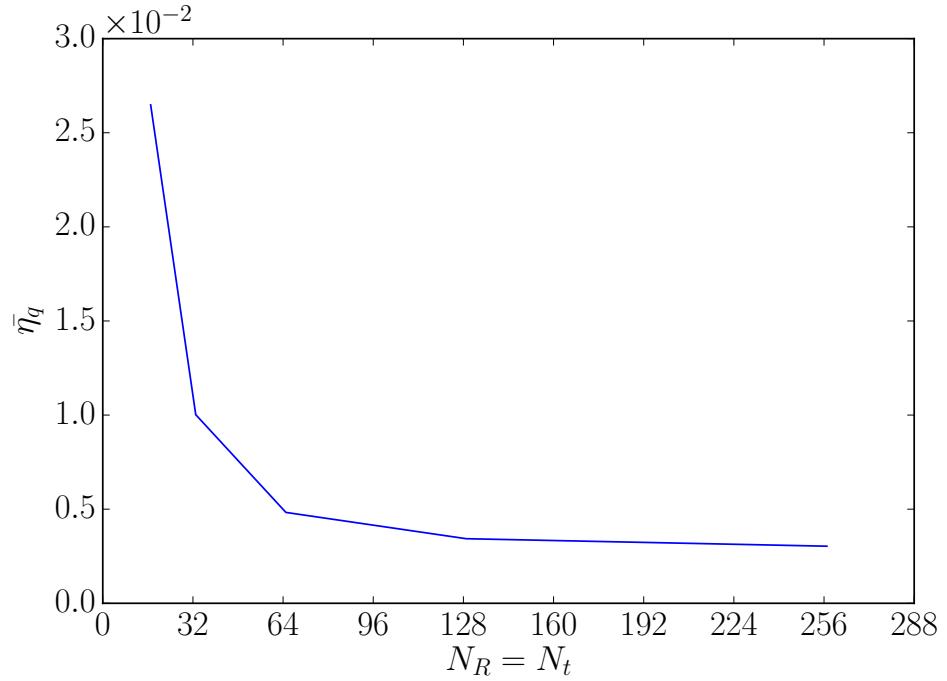


FIGURE 4.4: Average relative error of the q -profile versus number of points in grid.
Note that the total amount of points is $N = N_R N_t$.

This is expected, as an increase in elements gives a better approximation of the real shape of the flux surfaces, which in turn gives better precision in the integrals in equations (3.13) and (3.16). However, it is not possible to completely eliminate the error. One reason for this is that the method to calculate surface and contour integrals will always give an underestimation with the method proposed to calculate them. Another reason is that the value of a parameter on one triangular element is assumed to be the numerical average of its vertices. This is generally a simplification, a more accurate value can be obtained by using a more advanced method. However, as the goal of an average precision of 2 % is reached even for a low amount of points, these options are not further investigated. As usually the case in FEM, an increase in points comes at the cost of a longer computation time. This can be seen in figure 4.5.

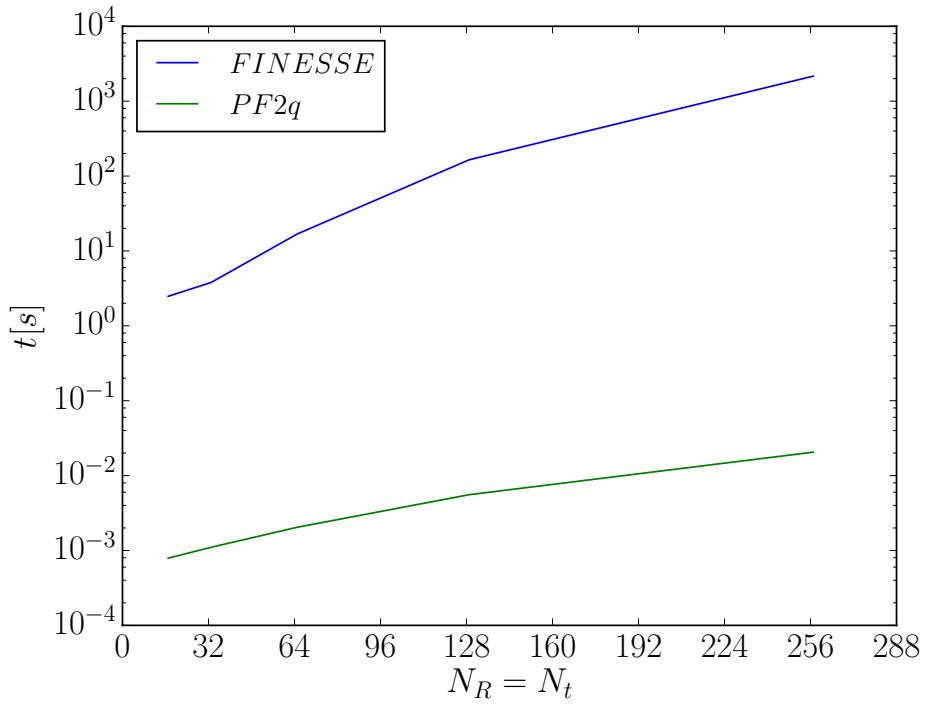


FIGURE 4.5: Time it takes to run FINESSE and make a prediction using PF2q versus number of points in grid. Note that the actual amount of points is $N = N_R N_t$.

Usually $N_R = N_t = 33$ points is sufficient, as the error in q -profile is below the goal of 2 %, but if more precision is required $N_R = N_t = 65$ gives a nice trade-off between precision and speed as well. However, a higher amount of points comes at a cost of a relatively large increase in computation time for a small increase in precision.

The estimation of the q -profile using PF2q takes a few milliseconds, about a factor 1000 less than running FINESSE itself.

4.3 Estimation of the q -profile by varying input parameters

The q -profile that results from a change in p - and F -profile can now be predicted using the methods described in section 3.4. The goodness of the prediction can then be determined by running FINESSE with the new input profile and comparing the output q -profile with the prediction. However, it is hard to give an absolute goodness of the estimation. This is because the goodness does not only depend on the change in coefficients in the profiles, but also on the absolute value of the coefficients, which coefficient is changed, interactions between coefficients and failure of convergence for certain combinations of coefficients. This report will not focus on these issues, because an absolute measure is not needed.

More important is an general understanding of the range over which the precision of the estimation is good enough according to the design requirements. To build this general understanding, an arbitrary subset of changes in input profile are tried.

There are multiple ways one could change the input profiles. One tactic is keeping all coefficients of the profile constant but one. For example, the second coefficient of the \tilde{F}^2 -profile (\tilde{F}_1^2) can be changed. An example of the original and new coefficients for a change of 9 can be found in table 4.3.

	A	\tilde{F}_0^2	\tilde{F}_1^2	\tilde{F}_2^2	\tilde{F}_3^2	\tilde{F}_4^2	\tilde{F}_5^2	\tilde{F}_6^2	\tilde{F}_7^2	\tilde{F}_8^2	\tilde{F}_9^2	$\sum \tilde{F}_i^2$
$\tilde{F}^2(\Delta F^2 = 0)$	1	20	-47	-1.5	-1	30.3	16	-3.7	-3.5	-12.5	-2.2	-4.1
$\tilde{F}^2(\Delta F^2 = 9)$	1	20	-36	-1.5	-1	30.3	16	-3.7	-3.5	-12.5	-2.2	4.9

TABLE 4.3: Example of a changed input profile by keeping all coefficients of the profile constant but one.

The resulting average relative error in the q -profile ($\bar{\eta}_q$) and the resulting error at the $q = 1$ point ($\eta_{q=1}$) of a change in the F - and p -profile (ΔF , Δp) can be found in figures 4.6 and 4.7 respectively.

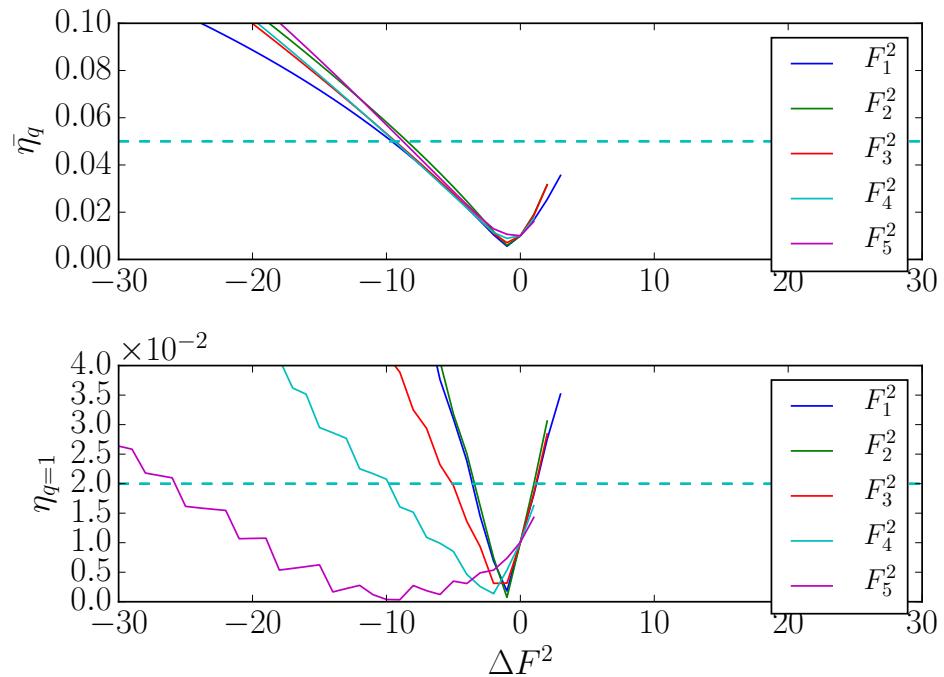


FIGURE 4.6: The average relative error and relative error as function of the change in input profile. Only one coefficient is changed per line. It is not known what causes the step-like behaviour for F_4^2 and F_5^2 , however, in general the larger the change in input, the bigger the error. The domain below the dashed line is the domain for which the design requirements of 5 % overall and 2 % around $q = 1$ are met.

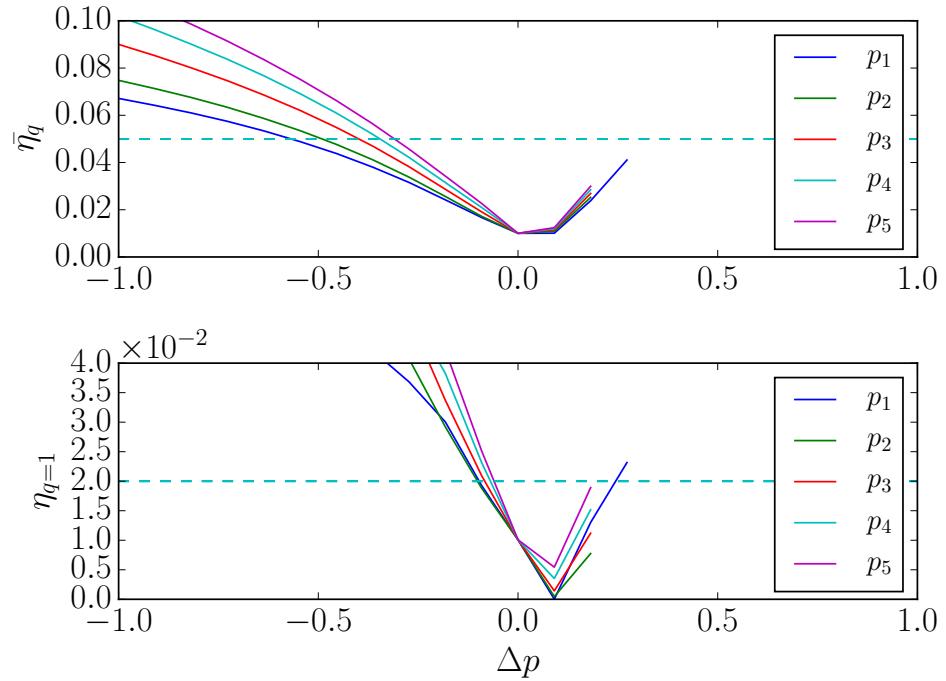


FIGURE 4.7: The average relative error and relative error as function of the change in input profile. Only one coefficient is changed per line. In general the larger the change in input, the bigger the error. The domain below the dashed line is the domain for which the design requirements of 5% overall and 2% around $q = 1$ are met.

As can be seen in both figures, the domain for which the estimation meets the requirements set in section 1.4 is strongly dependent on which coefficient is being changed. Also, it is not guaranteed that **FINESSE** finds a solution for each prediction PF2q makes, causing the cutoff for positive Δ . It is not known what causes the step-like behaviour for F_4^2 and F_5^2 . This makes it hard to quantify, and predict, exactly how good the q -profile estimation is. However, the general trend is that a too large change in p - and F -profile gives a bad estimation of q -profile. This is to be expected, as the more these profiles change, the more the assumptions 1 and 2 in section 3.4 get violated.

Another tactic can be used to better satisfy assumption 1 in section 3.4. The sum of all coefficients ($\sum \tilde{F}_i^2$) directly influences the value at the edges, as $0 \leq \psi \leq 1$, so $\tilde{F}_{\psi=1} = \sum \tilde{F}_i^2$. Thus trying to keep this sum constant results in a better agreement with the assumption, which in turn results in a better prediction. An example of the original and new coefficients can be found in table 4.4. The result in error of the predicted q -profile can be found in figures 4.8 and 4.9.

	A	\tilde{F}_0^2	\tilde{F}_1^2	\tilde{F}_2^2	\tilde{F}_3^2	\tilde{F}_4^2	\tilde{F}_5^2	\tilde{F}_6^2	\tilde{F}_7^2	\tilde{F}_8^2	\tilde{F}_9^2	$\sum \tilde{F}_i^2$
$\tilde{F}^2(\Delta F^2 = 0)$	1	20	-47	-1.5	-1	30.3	16	-3.7	-3.5	-12.5	-2.2	-4.1
$\tilde{F}^2(\Delta F^2 = 9)$	1	19	-36	-2.5	-2	29.3	15	-4.7	-4.5	-13.5	-3.2	-4.1

TABLE 4.4: Example of a changed input profile by keeping the sum of coefficients constant.

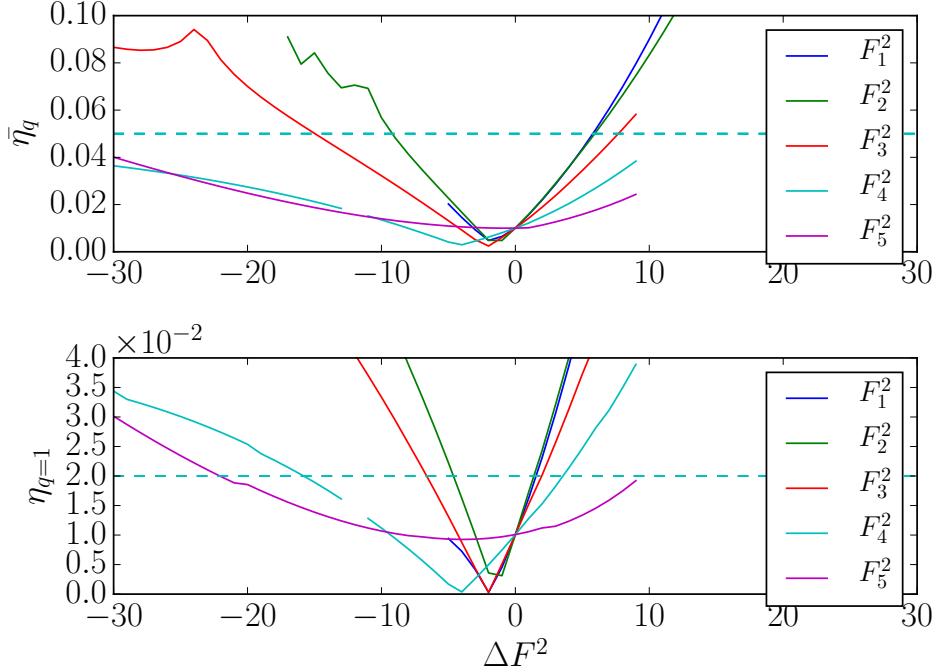


FIGURE 4.8: The average relative error and relative error as function of the change in input profile. The sum of coefficients is kept constant for each line. In general the larger the change in input, the bigger the error. The domain below the dashed line is the domain for which the design requirements of 5% overall and 2% around $q = 1$ are met. In general, this domain is larger than the domain when changing only one coefficient.

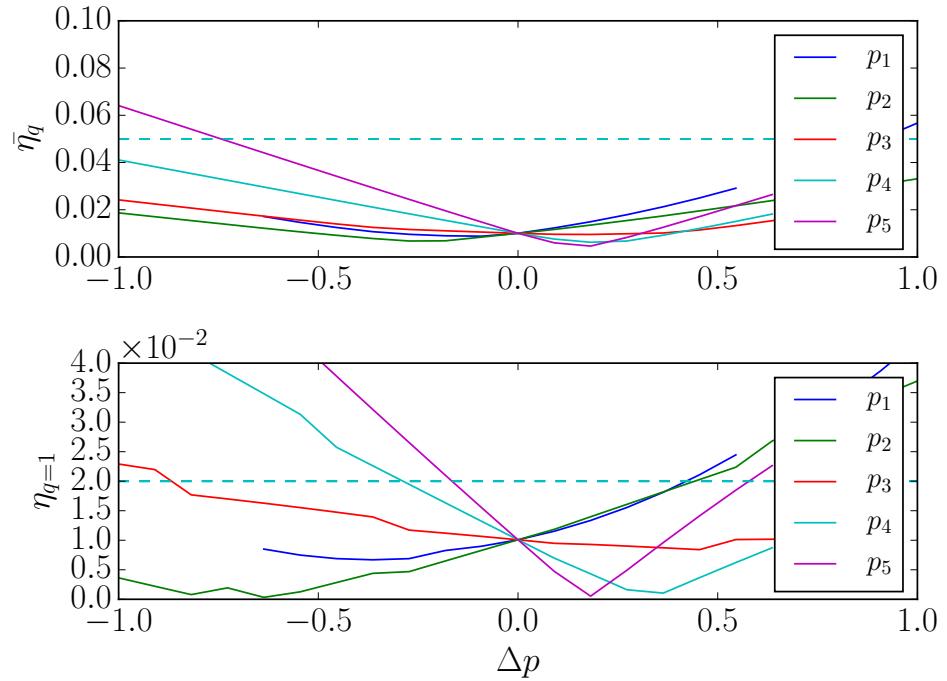


FIGURE 4.9: The average relative error and relative error as function of the change in input profile. The sum of coefficients is kept constant for each line. In general the larger the change in input, the bigger the error. The domain below the dashed line is the domain for which the design requirements of 5% overall and 2% around $q = 1$ are met. In general, this domain is larger than the domain when changing only one coefficient.

As before, the exact goodness of prediction is hard to quantify. However, by comparing the shapes of the graphs it can be seen that for most coefficients, keeping the sum of coefficients constant gives a larger domain of convergence and a larger domain over which the design requirements are met.

4.4 Flux-surface shape change by varying input parameters

As shown in section 4.3, the error in the estimation of the q -profile increases as a result of increased $\Delta\tilde{p}$ and $\Delta\tilde{F}^2$. This effect persists even when the sum of coefficients is kept constant. This suggests that assumption 2 of section 3.4 breaks down. This change in flux-surface shape as function of $\Delta\tilde{F}^2$ can be seen in figure 4.10.

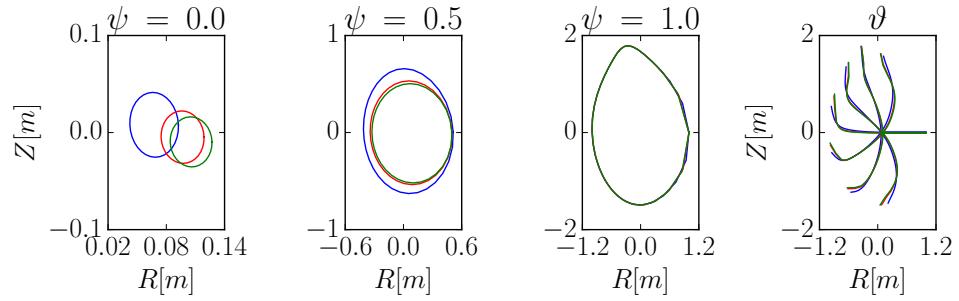


FIGURE 4.10: The effect of a change in input parameter $\Delta\tilde{F}_1^2$. The three leftmost figures show from left to right the shape of the inner-most, center flux-surface, and outer-most constant poloidal flux surface respectively, while the rightmost figure shows eight toroidal flux surfaces. For the blue lines $\Delta\tilde{F}_1^2 = -29$, the red lines $\Delta\tilde{F}_1^2 = 0$, and the green lines $\Delta\tilde{F}_1^2 = 29$. The quick q -profile estimation assumes that these lines have the same shape, but as can be seen this is not always the case. The change in shape is strongest for the inner poloidal flux-surfaces. The effect on the toroidal flux-surfaces is minor, but stronger on the outside compared to the inside.

The assumption holds for the outer poloidal flux-surfaces, but less for the inner poloidal flux-surfaces. For toroidal flux-surfaces the effect is minor, but stronger on the outside compared to the inside.

Chapter 5

Conclusion

In this report a solution has been proposed to solve the mismatch between ASDEX Upgrade data and FINESSE input. A software package with visual tool has been designed to provide the experimenter to better match the q -profiles from ASDEX and FINESSE. The relative error in the estimation of the $q(\rho)$ -profile from FINESSE input data is below 1% if 33 points or more are used for the FEM grid, well below the requirement of 2%. The quick estimation method can be used to predict the q -profile from FINESSE by varying the input parameters. The relative error of this method depends strongly on the parameter being varied, and by which amount it is varied. Nevertheless, the domain over which these parameters can be varied while staying inside the given requirements is large enough for an experimenter to correctly match the q -profiles. This gives an experimenter a quicker way to match ASDEX data with FINESSE input.

Chapter 6

Outlook

It should be possible to write an algorithm that automatically solve the mismatch between ASDEX Upgrade data and **FINESSE** input, as the quick estimation method takes only about 1 ms. However, designing such an algorithm is hard because the problem is highly multidimensional. An algorithm would need to take into account, for example, the possible non-convergence of **FINESSE** and needs to make sure the assumptions for the quick estimation method are met.

An easier alternative to an algorithm would be the creation of a database of q -profiles and their **FINESSE** input files. An experimenter could then find a q -profiles closely matching his, and the manually edit it until it matches accurately enough for his or her needs.

Some more improvements to **PF2q** could be made as well. Currently, the tool only has a visual interface. A command line interface would help speeding up profile matching for multiple shots. Currently **PF2q** does not take into account toroidal flow of the plasma. **FINESSE** uses this rotation to make a more realistic model of the plasma, which could increase the accuracy of the model. Also, multiple extensions could be added, for example tools to assess NTM stability for scenario building, and tools for current drive studies.

These extensions can be used to gain a deeper understanding of the tokamak plasma, which in the end result in a better fusion reactor, bringing us closer to a functioning, sustainable, and cost effective power plant.

The **PF2q** code will be included in the ITER software repository, and is publicly available on GitHub (<https://github.com/Karel-van-de-Plassche/PF2q>).

Appendix A

Description of PF2q code

All modules are meant to be used inside a python script file. An example `analyse.py` is supplied with PF2q.

A.1 `asdex.py`

This module contains the `AsdexDataSet` class to interact with ASDEX Upgrade data. The class contains the $p(\rho)$, $q(\rho)$, and ρ profiles, as well as the α , ϵ , and $I(\psi = 1)$ constants. There are also functions defined to manipulate and plot data, as can be found in table A.1

Class	Name	Description
<code>AsdexDataSet</code>	<code>convert_rho_to_psi</code>	Convert the $q(\rho)$ to a $q(\psi)$ profile
	<code>plot</code>	Plot $p(\rho)$ and $q(\rho)$
	<code>load_matlab</code>	Load all profiles and constants from a MATLAB file

TABLE A.1: Description of all `asdex` functions.

A.2 fem.py

This module contains multiple classes and functions needed to make and manipulate the FEM grid used in calculations, for example for all the integrals described in chapter 3.

Class	Name	Description
-	<code>surfaceTriangle</code>	Calculate the surface of a triangle
Map	<code>calculate_poloidal</code>	Transpose grid from $(x, y) \rightarrow (r, \chi)$ coordinates
	<code>calculate_dl</code>	Calculate dl along a flux surface
	<code>contour_integral</code>	Calculate the contour integral along a flux surface
	<code>plot</code>	Plot the grid with lines of constant ϑ and Ψ
	<code>triangulate</code>	Triangulate the map
QuadrilateralMap	<code>abcdize</code>	Find the coordinates of the four vertices of each element
	<code>plot</code>	Plot the grid with lines of constant ϑ and Ψ
TriangularMap	<code>volume_integral</code>	Calculate the volume integral $\iiint x \, dV$
	<code>ring_integral</code>	Calculate the ring integral $\iint_0^R x \, dA$
	<code>surface_integral</code>	Calculate the surface integral $\iint dA$
	<code>calculate_centroid</code>	Calculate the position of the triangle centroid
	<code>centroid_interpolation</code>	Estimate the value on the centroid
	<code>barycentric_interpolation</code>	Use barycentric interpolation to estimate the value of x on a triangle.

TABLE A.2: Description of all `fem` functions.

A.3 tools.py

This module contains some general purpose tools that are used throughout PF2q.

Class	Name	Description
-	<code>range_0_2pi</code>	Remap all values to $[0, 2\pi]$
	<code>smart_diff</code>	Calculate <code>numpy diff</code> using that the first and last poloidal points are the same point.
	<code>rescale</code>	Linearly rescale the given profile
	<code>plot_grid</code>	Plot the given grid
	<code>plotyy</code>	Plot using multi-y axes
	<code>create_windows_shortcut</code>	Create a shortcut on a Windows system
	<code>relative_error</code>	Calculate the relative error η
	<code>badness</code>	Calculate the <i>badness</i> of a certain profile
	<code>combine_badness</code>	Combine the <i>badness</i> values in a single value
	<code>small_is_zero</code>	Round small values to 0
	<code>fit_finesse_poly</code>	Fit a polynomial without too large coefficients
	<code>frange</code>	<code>range</code> for floats
	<code>plot_estimate_q</code>	Plot an estimation of the q -profile
	<code>extrap1d</code>	Extrapolate polynomial values

TABLE A.3: Description of all `tools` functions.

A.4 finesse.py

This module contains all function to manipulate FINESSE data. Many of these functions have a direct relation to equations in this report.

Class	Name	Description
	<code>calculate_beta</code>	Calculate β using equation (3.5)
	<code>calculate_betap</code>	Calculate β_p using equation (3.6)
<code>FinesseDataSet</code>	<code>calculate_rho</code>	Calculate ρ using equation (1.8)
	<code>calculate_p</code>	Calculate p using equations (2.21) and (3.5) to (3.7)
	<code>calculate_common_geometric</code>	Calculate x, y, R_0 and R using equations (2.16) and (2.17)
	<code>calculate_common_physical</code>	Rescale B_φ and B_p using equations (2.18) and (2.20)
	<code>estimate_from_output</code>	Internal check using FINESSE output instead of input to estimate the q -profile
	<code>assume_dp_dF_correct</code>	Calculate F_0, F_1, p_0 and p_1
<code>FinesseSession</code>	<code>run_finesse</code>	Run FINESSE
	<code>read_output_data</code>	Read FINESSE output from file
<code>FinesseInput</code>	<code>read_input_file</code>	Read FINESSE input from file
	<code>input_to_file</code>	Save FINESSE input to file
	<code>boundary_to_file</code>	Save Ψ_1 to file
	<code>read_boundary_file</code>	Read Ψ_1 from file
	<code>save_input_dialog</code>	GUI to save FINESSE input
	<code>load_input_dialog</code>	GUI to load FINESSE input
<code>EstimationCase</code>	<code>estimate_q</code>	Estimate q using the quick estimation

TABLE A.4: Description of all `finesse` functions.

A.5 pf2qvis.py

This module contains all the necessary function to draw the GUI of `pf2qvis`.

Class	Name	Description
FpTool	<code>_define_badness</code>	Define the <i>badness</i> boxes below the <i>q</i> -profile plot
	<code>_define_estimate_q</code>	Define the <i>q</i> -profile plot
	<code>_draw_finesse_q</code>	Draw the <i>q</i> from the output of FINESSE and the target <i>q</i>
	<code>_draw_estimate_q</code>	Draw the result of the quick <i>q</i> estimation
	<code>_estimate_q</code>	Run the quick <i>q</i> estimation
	<code>_define_below_plt_2</code>	Define the boxes below the 2nd plot
	<code>_define_buttons</code>	Define the buttons at the bottom of the GUI
	<code>_run_finesse</code>	Run FINESSE
	<code>_rerun_finesse</code>	Run FINESSE and update plots
	<code>_save_input</code>	Save FINESSEinput file
	<code>_connect_button_events</code>	Connect the events that happen when you press a button
	<code>_switch_tool</code>	Switch from <i>p</i> -profile to <i>F</i> -profile tool
	<code>_load_input</code>	Load FINESSEinput file
	<code>_read_sliders</code>	Read the values from the sliders
	<code>_reset_sliders</code>	Reset the sliders to their initial value

TABLE A.5: Description of all `pf2qvis` functions.

Class	Name	Description
PTool	<code>_update_estimates</code>	Re-estimate q and draw the q - and p -profiles
	<code>_update_finesse</code>	Re-run finesse and update GUI
	<code>_draw_finesse_p</code>	Draw the p -profile from FINESSE output
	<code>_draw_estimate_p</code>	Draw the estimated p -profile
	<code>_recalculate_p</code>	Calculate the estimated p -profile
	<code>_connect_slider_events</code>	Connect the event that happen when you move a slider
	<code>_define_sliders</code>	Define the sliders to change the p -profile
	<code>_define_p</code>	Define the p -profile plot
FTool	<code>_update_estimates</code>	Re-estimate q and draw the q - and F -profiles
	<code>_update_finesse</code>	Re-run finesse and update GUI
	<code>_connect_slider_events</code>	Connect the event that happen when you move a slider
	<code>_define_I_encl</code>	Define the I_{encl} -profile plot
	<code>_draw_I_encl_finesse</code>	Draw the I_{encl} -profile reconstructed from FINESSE output
	<code>_draw_I_encl_est</code>	Draw the estimated I_{encl} -profile
	<code>_define_sliders</code>	Define the sliders to change the F -profile
CombiTool	-	Combines the PTool and FTool

TABLE A.6: Description of all `pf2qvis` functions.

Bibliography

- [1] Institute of Atomic Energy. Key world energy statistics 2014, 2014. URL <http://www.iea.org/publications/freepublications/publication/KeyWorld2014.pdf>.
- [2] Institute of Atomic Energy. World energy outlook 2014, 2014. URL <http://www.iea.org/textbase/nppsm/weo2014sum.pdf>.
- [3] World Nuclear Association. Current and future generation lithium.
- [4] Max-Planck-Gesellschaft. Plasma boundary - the divertor. URL <http://www.ipp.mpg.de/11900/plasmabegrenzung>.
- [5] J.P. Goedbloed, R. Keppens, and S. Poedts. *Advanced magnetohydrodynamics*. University Press, Cambridge, 2010.
- [6] M. Shimada, D.J. Campbell, V. Mukhovatov, et al. Chapter 1: Overview and summary. *Nuclear Fusion*, 47(6):S1, 2007. URL <http://stacks.iop.org/0029-5515/47/i=6/a=S01>.
- [7] F Jaulmes. *Kinetic behaviour of ions in tokamak inductive scenarios*. PhD thesis, University of Technology Eindhoven, 2015.
- [8] I T Chapman. Controlling sawtooth oscillations in tokamak plasmas. *Plasma Physics and Controlled Fusion*, 53(1):013001, 2011. URL <http://stacks.iop.org/0741-3335/53/i=1/a=013001>.
- [9] O. Sauter, E. Westerhof, M. L. Mayoral, et al. Control of neoclassical tearing modes by sawtooth control. *Phys. Rev. Lett.*, 88:105001, Feb 2002. doi: 10.1103/PhysRevLett.88.105001. URL <http://link.aps.org/doi/10.1103/PhysRevLett.88.105001>.
- [10] D. J. Campbell, D. F. H. Start, J. A. Wesson, et al. Stabilization of sawteeth with additional heating in the jet tokamak. *Phys. Rev. Lett.*, 60:2148–2151, May 1988. doi: 10.1103/PhysRevLett.60.2148. URL <http://link.aps.org/doi/10.1103/PhysRevLett.60.2148>.

- [11] F. Jaulmes, E. Westerhof, and H.J. de Blank. Redistribution of fast ions during sawtooth reconnection. *Nuclear Fusion*, 54(10):104013, 2014. URL <http://stacks.iop.org/0029-5515/54/i=10/a=104013>.
- [12] Ya.I. Kolesnichenko and Yu.V. Yakovenko. Theory of fast ion transport during sawtooth crashes in tokamaks. *Nuclear Fusion*, 36(2):159, 1996. URL <http://stacks.iop.org/0029-5515/36/i=2/a=I04>.
- [13] A. J. C. Beliën, M. A. Botchev, J. P. Goedbloed, et al. Finesse: Axisymmetric mhd equilibria with flow. *Journal of Computational Physics*, 182, July 2002. URL http://ac.els-cdn.com/S0021999102971536/1-s2.0-S0021999102971536-main.pdf?_tid=509d3e6c-19b7-11e5-84ec-00000aab0f6b&acdnat=1435071262_8b84f62e5be17abd5cb0d5998d4b111c.
- [14] R.J. Hawryluk. An empirical approach to tokamak transport. *Physics of Plasmas Close to Thermonuclear Conditions*, 1, 1980. URL <http://www.xsciforge.com/about/articles/nubeam.pdf>.
- [15] Transp website. URL <http://www.xsciforge.com/about/articles/nubeam.pdf>.
- [16] Wikipedia community. Grad-shafranov equation, June 2015. URL https://en.wikipedia.org/wiki/Grad%20Shafranov_equation.
- [17] Hugo J de Blank. Plasma equilibrium in tokamaks. *Fusion Science and Technology*, 61(2T):89–95, 2012. URL <http://www.differ.nl/sites/default/files/z-ppc-notesequi.pdf>.
- [18] G Hommen, M de Baar, J Citrin, et al. A fast, magnetics-free flux surface estimation and q -profile reconstruction algorithm for feedback control of plasma profiles. *Plasma Physics and Controlled Fusion*, 55(2):025007, 2013. URL <http://stacks.iop.org/0741-3335/55/i=2/a=025007>.