Ion Cyclotron Resonance Heating in Toroidal Plasmas

Johan Hedin

DOCTORAL THESIS
ALFVÉN LABORATORY
DIVISION OF FUSION PLASMA PHYSICS
ROYAL INSTITUTE OF TECHNOLOGY

STOCKHOLM 2001
Cover
The cover picture shows the electric field $E_r$ at $t = 0.35 + s$ for $n_{\phi} = -25$ in the case of the Tore Supra similar deuterium discharge, presented in paper VII. Behind the torus are some of the formulas for pre-calculating the coefficients of the dielectric tensor and small fragments of the code calculating the dielectric tensor.

Production of this thesis
This thesis is typeset with \LaTeX\, using the font set Lucida Bright from Y & Y.

ISBN 91-7283-023-9

Stockholm December 2000 Universitetsservice US AB
Ion cyclotron resonance heating in toroidal plasmas

Johan Hedin

A thesis submitted in the partial fulfillment of the requirement for the degree of Doctor in Philosophy

AT THE

ALFVÉN LABORATORY
DIVISION OF FUSION PLASMA PHYSICS
ROYAL INSTITUTE OF TECHNOLOGY

STOCKHOLM 2001
Ion cyclotron resonance heating in toroidal plasmas

Johan Hedin

Abstract

Radio frequency heating by fast magnetosonic waves in the ion cyclotron range of frequencies, ICRF, has developed into one of the main auxiliary heating methods for thermonuclear fusion experiments. The time evolution of the distribution functions of the heated species is determined by the balance between the wave-particle interactions and the Coulomb collisions. The ion cyclotron heating can produce highly anisotropic velocity distributions with large energetic tails. The orbits of these high energy ions will experience large deviations from the magnetic flux surfaces. The transfer of toroidal momentum from the wave will give rise to a spatial inward or outward induced transport of the high energy ions depending of the toroidal direction of wave propagation. Further, the wave propagation and damping properties are determined by the details of the distribution functions. Hence, both the Fokker-Planck equation of the distribution function and the wave equation have to be solved self-consistently.

A method for self-consistent calculations of ion cyclotron resonance heating including finite orbit widths and RF induced transport has been developed, and implemented in the SELFO code. The code has successfully been used to explain experimental results. A shift of the resonance due to non-standard orbits on the LFS, consistent with experimental observations at JET, is examined in detail. In the analysis, it is found that the drift terms of a symmetric spectrum do not cancel, but both the inward and outward drift are present. For a directed spectrum, the inclusion of the subdominant peak with high $|n_\phi|$ of opposite sign, is found to be important. For fast wave electron current drive experiments at Tore Supra, the SELFO code recovers the current drive efficiencies of the experiments. The asymmetry due to the toroidal direction of propagation of the wave is explained with the influence of RF induced transport on the ion tail formation.
Descriptors

Fusion plasma, RF heating, self-consistent calculations, ion cyclotron resonance, fast wave current drive, finite orbits widths, RF induced transport, ICRF, ICRH, antenna phasing
Preface

My first contact with the research area of fusion was during a presentation by Prof. Elisabeth Rachlew. The field fascinated me, since it involved many branches of physics, and still has a clear ultimate goal of an environmentally friendly energy source. I later did my Master's thesis at the Alfvén laboratory and continued as a Ph.D. student. It is my firm belief, that fusion energy is one of the few options for the next century energy requirements.

A good supervisor is crucial for outcome of the Ph.D. studies. I have greatly enjoyed working with my supervisor, Prof. Torbjörn Hellsten. The discussions during the data analysis have been both pleasant and stimulating. During the work, Torbjörn Hellsten gave me just about right amount of freedom, while still stopping me from running into dead ends.

During my first time at the laboratory as a Ph.D. student, Dr. Johan Carlsson and Doc. André Jaun, did provide me with valuable help on getting started with the numerics and physics. After getting in to the field, we have had stimulating physics and numerics discussions. During the time at the lab, I have had the pleasure of sharing Johan Carlsson’s deep knowledge of operating systems and computer hardware. Developing FIDO-TNG and writing papers in collaboration with Thomas Johnson has been rewarding projects. During the summers an intensive numerics course have been developed and given in collaboration with André Jaun and Thomas Johnson. The project has been great fun, and resulted in a few conference contributions.

I want to thank for all the support I have got during the years from my beloved girl friend Maria Reyier, who has helped me with proof-reading and improved the English and clarity of the text, my parents, my brothers, my sister, my grand parents and all my relatives. It is a great sorrow that my grandfather, who was always very interested in and well-informed about fusion, is not able to read this thesis. Dr. Gunnar Hedin and Dr. David Larsson have been teaching me the wisdoms of life from a Ph.D. student’s point of view. Sam Hokin introduced me to the wonderful world of Java. In close collaboration with Martin Laxåback and Thomas Hurtig, a project for significantly increased computing power of the laboratory have been successfully initialized. I’m confident it will be very useful in the future. I wish Jenny-Ann Malmberg good luck as the
new Sun system manager. It has been a great pleasure to discuss with Prof. Nils Brenning, who share my passion for lousy horror movies. I also would like to thank the rest of the lunch party for stimulating discussions during lunch time. Finally, I'll give an acknowledgment to all my colleges, who have helped me in my work at the laboratory.
To Maria
List of papers

Reprinted in the thesis

This thesis is based on the work presented in the reprinted papers I–X. The development of the dielectric tensor calculations and the interfacing of SELFIO are carried out by the thesis author. The first author has the main responsibility of the included papers. The analysis and proposals were done in close collaboration with the supervisor. In paper II, the contribution by the thesis author is limited to the calculation of the $\alpha$-particle distribution function, power partitions and electron current drive. The last paper was already included in the M. Sc. degree and should therefore be regarded more as an orientation of the work performed. For this paper the supervision was shared between T. Hellsten and D. F. H. Start.

First steps towards a self-consistent treatment

Paper I


Paper II


Influence of the orbit topology

Paper III

The influence of non-standard orbits on the ICRH power deposition in tokamaks, J. Hedin, T. Hellsten and L.-G. Eriksson, Nuclear Fusion, 11 (2000) 1819–1824 (Accepted as a letter, but the size of the figures was increased, and hence the result was too long.)
Paper IV


Paper V


Paper VI

The influence of finite drift orbit width on the ICRF heating in toroidal plasmas, J. Hedin, T. Hellsten, L.-G. Eriksson and T. Johnson, Submitted to Nuclear Fusion, Also as internal report ALF-2000-107

Parasitic ion heating during fast wave electron current drive

Paper VII


Paper VIII


Paper IX

Miscellaneous

Paper X

A. Jaun, J. Hedin and T. Johnson, Teaching Numerical Method for Partial Differential Equations with the Internet, Submitted to Computing in Science & Engineering

Paper XI

J. Hedin, Studies of heating efficiencies and models of RF-sheaths for the JET antennae, Master project, Alfén laboratory, Stockholm, ISSN 1102-2051, 1996

Not reprinted in the thesis

The following papers include contributions of the thesis author during the Ph.D. studies, but are not included in the thesis, since they are intermediate steps, too long, lacks proceedings, or because the contribution of the thesis author is minor.

Paper XII


Paper XIII


Paper XIV

Paper XV


Paper XVI


Paper XVII


Paper XVIII


Paper XIX

# Contents

1 Introduction 1  
1.1 Basic equations ................................. 7  
1.2 The wave equation ............................... 8  
1.3 Distribution function evolution ................. 10  
1.4 Self-consistent calculations .................... 15  

2 Numerical treatment of self-consistent calculations 17  
2.1 The wave field .................................. 17  
2.2 The distribution function ......................... 19  
2.3 Dielectric tensor ................................ 20  

3 Development of FIDO-TNG 25  
3.1 Planned improvements ............................ 25  
3.2 Design of FIDO-TNG .............................. 25  
3.2.1 The point classes ............................... 27  
3.2.2 The Mesh class ................................ 27  
3.2.3 The Table3D class .............................. 30  
3.2.4 The Diagnostics class ......................... 32  
3.2.5 The OrbitSolver class ....................... 38  
3.2.6 The FlexibleVector class .................... 38  

4 Summation of the publications 39  
4.1 Paper I: Parasitic alpha particle absorption ........ 39  
4.2 Paper II: Minority ion cyclotron current drive ...... 40  
4.3 Paper III–VI: Influence of the orbit topology ....... 40  
4.4 Paper VII–IX: Parasitic ion absorption during FWED 43  
4.5 Paper X: Teaching numerical methods using Internet based tools .......................... 44  

5 Conclusions 45  

6 Resume and personal reflections 47
A Parallelization of FIDO 1.x 49
  A.1 Architectures of parallel machines 49
  A.2 Parallel algorithm in FIDO 1.x 50
    A.2.1 Parallel algorithm for the Monte Carlo calculations 52
    A.2.2 Parallel algorithm for the dielectric tensor integration 53

B Technical environment 57
  B.1 C++ and object oriented design 57
  B.2 Automake, Autoconf and Libtool 58
  B.3 CVS 58
  B.4 Input file parsing 59
  B.5 Documentation 59

C Paper I 61

D Paper II 75

E Paper III 85

F Paper IV 93

G Paper V 109

H Paper VI 117

I Paper VII 135

J Paper VIII 143

K Paper IX 149

L Paper X 155

M Paper XI 165
1 Introduction

It is most likely that the worlds energy consumption will increase in the future [1], but even if it stays at the present level, the risk of global warming using fossil fuels [2], will create a public demand for a large scale energy source without net CO₂ emission. The fusion research aims for such a nuclear energy source, with less drawback than fission energy. A fusion reactor will give only Helium as exhaust. Due to the low plasma density, the reactor needs a constant inflow of fuel to be running, therefore there is no possibility of a core melt down. However, the large neutron flux will create some radioactive waste from core reactor parts being replaced, but it will be active for hundreds of years and not hundreds of thousands of years. Furthermore the decay time can be reduced by choosing suitable materials.

The basic principle of fusion energy is to use the energy released when light nuclei fuse into heavier ones, which is the energy source of the sun. In a fusion power plant, the reaction

\[ ^2H + ^3H \rightarrow ^4He + n \quad (Q = 17.6 \text{ MeV}) \]  

(1.1)

is proposed to be used, where 14.1 MeV of the released energy is kinetic energy of the neutron. The hydrogen isotopes \(^2H\) and \(^3H\) are also known as deuterium D and tritium T.

Since the D and T nuclei are positively charged, they require sufficient kinetic energy to overcome the Coulomb barrier, in order for the short range nuclear forces to fuse the nuclei. In a thermonuclear plasma, the temperature is sufficiently high for fusion reactions in the tails of the distribution functions. In the current designs, the temperature is in the order of \(10^8 \degree C\). In this temperature regime, the nuclei and electrons are unbound, yielding a completely ionized plasma. The plasma has to be confined by magnetic fields, since contact with the machine wall would cool the plasma, making it impossible to reach sufficient temperature. The magnetic confinement is based on that charged particles spiral along a magnetic field lines as shown in figure 1.1. The frequency of the rotation around the magnetic field line is denoted the fundamental cyclotron frequency.

There exist a variety of different magnetic configurations, but in this thesis, only the most common magnetic configuration, the tokamaks,
will be treated. In a tokamak, a strong toroidal magnetic field is induced by external coils, as shown in figure 1.2 from Ref. [3]. In addition, a weaker poloidal magnetic field, induced by a current in the plasma, is required for confining the plasma. The particles spiral around the resulting twisted magnetic field lines as shown in figure 1.3 from Ref. [4]. The largest present experiment, JET\(^1\) outside of Oxford, is shown in figures 1.4–1.5 from Ref. [5].

To achieve the required temperatures, the plasma is heated by Ohmic heating, neutral beams and \textit{rf}\(^2\) waves, and internally by the kinetic energy of the fusion born \(\alpha\)-particles. If the \(\alpha\)-particle heating is sufficient for sustaining the required temperature, the plasma is ignited. For a reactor, the produced energy from the fusion reactions must exceed the energy required to maintain the cooling of the supra-conducting magnets and for plasma control by current drive and heating. The thesis deals with how external heating in the form of \(\textit{rf}\) power in the ion cyclotron range of frequencies, ICRF, affects the plasma.

Ion cyclotron resonance heating is based on launching waves with multiples of the ion cyclotron frequency. Unfortunately, if a wave with the same frequency as the fundamental cyclotron resonance frequency as one of the main ion species is launched, the wave will polarize in such a way that the damping at the cyclotron resonance becomes negligible. In order to overcome this problem, minority ions with different charge to mass ratio are introduced, e.g. 1\% \(^3\)He in a DT plasma. The minority ions will then heat the background plasma through collisions. Heating

---

\(^1\)Joint European Torus

\(^2\)Radio Frequency
Figure 1.2 A schematic picture of a tokamak. The strong toroidal magnetic field is induced by external coils. The poloidal magnetic field is induced by the plasma current, obtained by using the plasma as secondary winding in a transformer.

Figure 1.3 The unperturbed motion of a low energy particle will be a spiral along the twisted magnetic field line.
**Figure 1.4** The largest magnetic fusion experiment in the world, JET outside of Oxford.
on a minority will create very high energy ions. Energies in the order of MeV are not uncommon. The RF waves are launched from one or more antennae on the outer side of the torus. The waves are in the frequency range of 20–100 MHz and are used for plasma heating and current drive/control. The RF power is transported in transmissions lines from the generators to the current straps in the antennae. The principal construction of the four A2 antennae at JET are shown in figure 1.6. The construction with four current straps will allow, through different phasing of the straps, a possibility to control the toroidal wave spectrum of the launched wave. The oscillating current in the strap will produce an electromagnetic wave, propagating from the current strap, through an evanescent vacuum region, into the plasma, where it will couple to a fast magnetosonic wave. However, the power launched by the antenna does not always couple to the plasma in the way expected. At JET, different coupling efficiencies are seen for different phasings of the antennae [6]. This has been studied experimentally and modeled numerically [6–9]. The locations of the JET antennae are shown in figure 1.7.
1. Introduction

**Figure 1.6** The principal construction of the JET RF-antennae. The Faraday screen is used to short circuit the component of the electrical field parallel to the magnetic field.

**Figure 1.7** The A2 antenna at JET. The figure shows a 3D sketch of the vacuum vessel to the left, and a cross section of the vacuum vessel to the right.
1.1 Basic equations

In the framework of classical electromagnetic theory, the electric and magnetic fields are described by Maxwell’s equations

\[ \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \]  
(1.2a)

\[ \nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \]  
(1.2b)

\[ \nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \]  
(1.2c)

\[ \nabla \cdot \mathbf{B} = 0 \]  
(1.2d)

where \( \mathbf{J} \) is the total current density and \( \rho \) is the total charge density.

High energy ions in fusion plasmas may have energies of a few tens MeV, while the rest mass of a proton is 938 MeV/c\(^2\). Therefore, non-relativistic mechanics is sufficient for describing the ion physics.

The motion of a particle is described by Newton’s second law

\[ m \frac{d\mathbf{v}}{dt} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \]  
(1.3)

Here, the gravitational force is neglected. The motion of the particle is coupled to Maxwell’s equations (1.2a–1.2d), since

\[ \mathbf{J}(\mathbf{x}) = q\mathbf{v}\delta(\mathbf{x}) \]  
(1.4)

\[ \rho = q\delta(\mathbf{x}) \]  
(1.5)

where \( \delta \) is the Dirac distribution. Real fusion plasmas are more complicated than a single charged particle in a vacuum field. The physics of a plasma is described by the many particle problem of interacting charged particles where

\[ \rho = \sum_i q_i \delta(x_i) \]  
(1.6)

\[ \mathbf{J} = \sum_i q_i \mathbf{v}_i \delta(x_i) \]  
(1.7)

\[ m_i \frac{d\mathbf{v}_i}{dt} = q_i (\mathbf{E} + \mathbf{v}_i \times \mathbf{B}) \]  
(1.8)

and \( i \) is summed over all particles in the plasma. Solving this set of equations, taking into account mirror currents in the shell, would demand huge, and at present time, non-existent computer resources. Therefore the many particle problem is approximated with a smooth distribution function \( \sum_\eta f_\eta(x,\mathbf{v}) \) in phase space, where \( \eta \) is summarized over the electrons and the ion species. The collective behavior of the plasma is described by the Vlasov equation

\[ \frac{\partial f_\eta(x,\mathbf{v})}{\partial t} + \mathbf{v} \cdot \nabla f_\eta(x,\mathbf{v}) - \frac{q_\eta}{m_\eta} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_\eta}{\partial \mathbf{v}} = 0 \]  
(1.9)
The charge density $\rho$ is hence the zeroth velocity moment of the distribution function

$$\rho(x) = \sum_{\eta} \int_{-\infty}^{\infty} d^3 v \ q_\eta f_\eta(x, v)$$  \hspace{1cm} (1.10)$$

and the current density $J$ is the first velocity moment of the distribution function

$$J(x) = \sum_{\eta} \int_{-\infty}^{\infty} d^3 v \ v q_\eta f_\eta(x, v)$$  \hspace{1cm} (1.11)$$

Thus the system of the five coupled partial differential equations (1.2a–1.2d) and (1.9) has to be solved.

The system has two natural very different time scales, which allows the separation of the problem into a short and long time scale. The short time scale is the wave period. During this time scale, the distribution function is regarded static, except for small oscillations coupled to a wave field. The longer time scale is used for studying the evolution of the distribution function. The effect of the wave field is then averaged over a wave period.

### 1.2 The wave equation

Solving the system of equations for the short time scale involves combining Maxwell’s equations (1.2a–1.2d) in the temporal gauge, i.e. no scalar potential,

$$\frac{1}{c^2} \frac{\partial^2 A}{\partial t^2} + \nabla \times \nabla \times A = \mu_0 J$$  \hspace{1cm} (1.12)$$

where

$$E = -\frac{\partial A}{\partial t}$$  \hspace{1cm} (1.13)$$
$$
$$B = \nabla \times A$$  \hspace{1cm} (1.14)$$

The total current density $J$ is divided into an external current $J_e$ and an internally induced current $J_i$

$$J = J_e + J_i$$  \hspace{1cm} (1.15)$$

By Fourier transforming equation (1.12) in time, and combining it with equation (1.15) the wave equation is obtained

$$\nabla \times \nabla \times E(x, \omega) - \frac{\omega^2}{c^2} \ e(x, \omega) E(x, \omega) = i \mu_0 \omega J_e(x, \omega)$$  \hspace{1cm} (1.16)$$
1.2. The wave equation

where the response of the plasma is expressed in the dielectric tensor $\epsilon$, implicitly given by

$$J_i = i\omega\epsilon_0(\epsilon - 1)E$$

(1.17)

By assuming the electric field to induce only a small perturbation $f_1^\eta$ of the distribution function $f_\eta$

$$f_\eta(x, v) = f_\eta^0(x, v) + f_1^\eta(x, v)$$

(1.18)

$$f_1^\eta(x, v) \ll f_\eta^0(x, v)$$

(1.19)

and a plane propagating wave in a locally homogeneous plasma, an explicit expression of the dielectric tensor can be calculated by linearization of equations of motions. Starting with a cold plasma, i.e. the ions and electrons are treated as fluids, the equations of motions for the ions are

$$mdv/dt = qE(t) + qv\times B_0$$

(1.20)

assuming one ion species only. The cold dielectric tensor is obtained by lineraising (1.20), and using equation (1.17)

$$\epsilon(k, \omega) = \begin{pmatrix}
1 - \omega_0^2/\omega^2 & -i\omega_0\omega/\omega(\omega^2 - \Omega^2) & 0 \\
-i\omega_0\omega/\omega(\omega^2 - \Omega^2) & 1 - \omega_0^2/\omega^2 & 0 \\
0 & 0 & 1 - \omega_0^2/\omega^2
\end{pmatrix}$$

(1.21)

The same procedure applies for solving the dielectric tensor for a general distribution function in $v$. $J_i = \sigma E$ is obtained by linearization of the Vlasov equation (1.9). The rotation around the magnetic field lines will give rise to Bessel functions in the dielectric tensor

$$\epsilon(k, \omega) = \left(1 - \frac{\omega_p^2}{\omega^2}\right) 1 - \sum_{n=-\infty}^{\infty} \int d\mathbf{v} \left( \frac{n\Omega_\eta}{\nu_\perp} \frac{\partial f_\eta}{\partial \nu_\perp} + k_n \frac{\partial f_\eta}{\partial \nu_\parallel} \right)$$

$$\times \frac{\Pi_\eta(\nu_\perp, \nu_\parallel; n)}{n\Omega_\eta + k_n\nu_\parallel - \omega - i0^+}$$

(1.22)

where

$$\Pi_\eta(\nu_\perp, \nu_\parallel; n) = \begin{pmatrix}
n\Omega_\eta^2/k_n J_n^2 & iv_\perp n\Omega_\eta/k_n J_n J'_n & v_\parallel n\Omega_\eta/k_n J_n^2 \\
-i v_\perp n\Omega_\eta^2/k_n J_n J'_n & v_\parallel^2 (J_n')^2 & -i v_\perp v_\parallel J_n J'_n \\
v_\parallel n\Omega_\eta/k_n J_n^2 & iv_\parallel v_\perp J_n J'_n & v_\parallel^2 J_n^2
\end{pmatrix}$$

(1.23)

and

$$J_n = J_n(\rho) \quad J'_n = \frac{d(f_n(\rho))}{d\rho} \quad \rho = \frac{k_n \nu_\perp}{\Omega_\eta}$$

(1.24)
$J_n$ is the $n$:th Bessel function. The derivation can be found in a standard textbook in plasma waves [10]. The finite parallel velocity enters the tensor as Doppler broadening of the resonance. $\epsilon_{xx}$, $\epsilon_{xy}$, $\epsilon_{yx}$, and $\epsilon_{yy}$ describe cyclotron damping and dispersion. $\epsilon_{yy}$, $\epsilon_{yz}$, $\epsilon_{zy}$, and $\epsilon_{zz}$, describe electron Landau and TMP damping.

By assuming a Maxwellian distribution function

$$f(v_\perp, v_\parallel) = \left(\frac{m}{2\pi T}\right)^{3/2} \exp \left[-\frac{v_\perp^2 + v_\parallel^2}{2(T/m)}\right]$$  \hspace{1cm} (1.25)$$

here with the temperature $T$ given in Joule, it is possible to analytically integrate the dielectric tensor (1.22) in velocity space. The result is here presented for a single component plasma.

$$\epsilon(k, \omega) = 1 - \frac{\omega^2_p}{\omega^2} \left[ \sum_{n=-\infty}^{\infty} \frac{Z_0}{Z_n} \Pi_n^M(\beta, Z_n; n) [1 - W(Z_n)] - Z_0^2 \epsilon_{zz} \right]$$  \hspace{1cm} (1.26)$$

where

$$\Pi_n^M(\beta, Z_n; n) = \begin{bmatrix} \frac{n^2}{\beta} \Lambda_n(\beta) & i n \Lambda_\beta(\beta) & k \frac{n}{|k_\perp| \sqrt{\beta}} Z_n \Lambda_n(\beta) \\ -i n \Lambda_\beta(\beta) & \frac{n^2}{\beta} \Lambda_n(\beta) - 2 \beta \Lambda_\beta(\beta) & -i \frac{k}{|k_\perp| \sqrt{\beta}} Z_n \Lambda_n(\beta) \\ \frac{k}{|k_\perp| \sqrt{\beta}} Z_n \Lambda_n(\beta) & i \frac{k}{|k_\perp| \sqrt{\beta}} Z_n \Lambda_\beta(\beta) & Z_n^2 \Lambda_n(\beta) \end{bmatrix}$$  \hspace{1cm} (1.27)$$

and

$$Z_n = \frac{\omega - n \Omega}{|k_\parallel| (T/m)^{1/2}} \hspace{1cm} \beta = \frac{k^2 T}{m \Omega^2}$$

$$\Lambda_n = I_n(\beta) e^{-\beta} \hspace{1cm} \epsilon_{zz} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$W(Z) = 1 - Ze^{-Z^2/2} \int_0^Z dy e^{y^2/2} + i(\pi/2)^{1/2} Ze^{-Z^2/2}$$  \hspace{1cm} (1.28)$$

1.3 Distribution function evolution

During the calculation of the time evolution of the distribution function $f(x, v)$, the wave field is assumed to be stationary. The time evolution of the distribution function is described by the Boltzmann equation

$$\frac{\partial f_\eta(x, v)}{\partial t} + v \cdot \nabla f_\eta(x, v) - q_\eta \frac{a_\eta}{m_\eta} (E + v \times B) \cdot \frac{\partial f}{\partial v} \frac{\partial f_\eta}{\partial t} \bigg|_c$$  \hspace{1cm} (1.29)$$
where the right hand side is a non-trivial collision operator. Writing the force from the wave field as an operator on the distribution function, the Fokker-Planck equation is obtained

$$\frac{\partial f_\eta(x, v)}{\partial t} + v \cdot \nabla f_\eta(x, v) = C[f_\eta(x, v)] + Q[f_\eta(x, v)]$$

(1.30)

where $C$ is the collision operator and $Q$ is the quasi-linear RF operator. The collisions are dominated by the Coloumb collisions, described by the Coulomb operator by Spitzer

$$C[f] = -\frac{1}{v^2} \frac{\partial}{\partial v} \left( v^2 \alpha f \right) + \frac{1}{2v^2} \frac{\partial^2}{\partial v^2} \left( v^2 \beta f \right)$$

$$+ \frac{1}{4v^2} \frac{\partial}{\partial \xi} \left( 1 - \xi^2 \right) \frac{\partial}{\partial \xi} (\gamma f)$$

(1.31)

where $\xi = v_\perp / v$. $\alpha$, $\beta$ and $\gamma$ are the Spitzer’s Coulomb diffusion coefficients [11]. The steady state solution of equation (1.30) without RF heating, i.e. $C[f] = 0$ and $Q[f] = 0$, is a Maxwellian distribution.

By assuming that the distribution function could be approximated by a local homogenous isotropic distribution function, Stix obtained a steady state solution of the Fokker-Planck equation [12], for which the operators $C$ and $Q$ are

$$C[f(v)] = \frac{1}{v^2} \frac{\partial}{\partial v} \left[ -\alpha v^2 f + \frac{1}{2} \frac{\partial}{\partial v} \left( v^2 \beta f \right) \right]$$

(1.32)

$$Q[f(v)] = \frac{1}{v^2} \frac{\partial}{\partial v} \left( K v^2 \frac{\partial f}{\partial v} \right)$$

(1.33)

Where $K$ is a function of the RF power density. The steady state solution of the Fokker-Planck equation is then

$$f(v) = f(0) \exp \left[ -\int_0^v dv' \frac{-2\alpha v^2 + (\beta b^2)^\prime}{\beta v^2 + 2Kv^2} \right]$$

(1.34)

In a toroidal geometry, the magnetic field in the torus is inhomogeneous, introducing the curvature and $\nabla B$ drifts. As a consequence, the gyro center of the orbits will drift across the magnetic flux surfaces in the direction $\nabla B \times B$. The unperturbed motion of a particle in an axisymmetric tokamak, can be described by three integrable or adiabatic invariants and three angles [13]. By choosing the following invariants

$$\tilde{E} = \frac{v^2}{2}$$

(1.35a)

$$\Lambda = \frac{B_0 v^2}{B v^2}$$

(1.35b)

$$\tilde{P}_\phi = R v_\perp + \frac{q}{m} \psi$$

(1.35c)
some points in invariant space will correspond to more than one drift orbit [14]. An additional label is therefore introduced in order to separate the different orbits. The invariant space is divided in regions [15], where each region corresponds to different types of orbits, as shown in figure 1.8 from Ref. [16]. The coordinate system used is shown in figure 1.9 from ref [3].

By manipulations of the expression for the wave–particle interactions, it is found that [16, 17]

$$\Delta \tilde{E} = \frac{eZ v_{\text{res}}}{m} \left| E_r J_{r-1} (k_{\perp} \rho) + E_{-r} J_{r+1} (k_{\perp} \rho) \right| \left[ \frac{2 \pi m}{n \mid \Omega \mid} \right] \cos \phi \quad (1.36a)$$

$$\Delta \Lambda = \left( \frac{n \omega_{c0}}{\omega} - \Lambda \right) \frac{\Delta \tilde{E}}{\tilde{E}} \quad (1.36b)$$

$$\Delta \tilde{P}_\phi = \frac{n \omega_{c0}}{\omega} \Delta \tilde{E} \quad (1.36c)$$

By integrating equation (1.36) the characteristics along which the diffu-
An ion may change from one characteristics to another given by equation (1.37) due to altering of \( C_1 \) and \( C_2 \) by Coulomb collisions [16] or interactions with modes with other toroidal mode numbers. The characteristics for a single \( n_{\phi} \) in the invariant space are illustrated in figure 1.10 from ref. [18].

As the energy increases, \( \Lambda \) will be driven towards \( \Lambda_{\text{res}} = n\omega c_0 / \omega \), i.e. the turning points of the trapped ions are driven towards the cyclotron resonance. The flux surface of the turning points of a trapped orbit is given by

\[
\Psi_T = \frac{m}{q} \left( \frac{n_{\phi}}{\omega} \hat{E} + C_1 \right) \tag{1.38}
\]

It follows from equation (1.38) that waves having \( n_{\phi} > 0 \) will drive the trapped orbits outward, and waves with \( n_{\phi} < 0 \) inward, as shown in figure 1.11, resulting in an outward or inward particle pinch [19]. This effect, seen experimentally [20], is found to be important for the calculations [18,21–32].
Figure 1.10 The characteristics given by equation (1.37) in the invariant space \((\tilde{E}, \Lambda, \tilde{P}_\phi)\) projected on the \((\Lambda, \tilde{P}_\phi)\)-plane. The full lines illustrate paths along which ions with an initial energy of 10 and 100 keV move due to wave-particle interactions. Dotted and dashed lines are separating the invariant space into different regions according to the orbit topology. The numbers of the regions refer to figure 1.8. The dotted lines are drawn for 1 MeV ions. The dashed ones are independent of the energy.

Figure 1.11 The time evolution (dotted, dashed, solid) of an orbit for \(n_\phi > 0\) to the left, and \(n_\phi < 0\) to the right.
1.4 Self-consistent calculations

The ion cyclotron heating can produce distributions with highly anisotropic large energetic tails. By varying frequency, concentration of the resonating species, toroidal wave spectrum and coupled power, the wave absorption can provide ion and electron heating, non-inductive current drive, stabilization or destabilization of MHD-modes and give rise to plasma rotation. The versatility of the method makes predictions of the performance difficult for many scenarios, because the sensitivity on the details of the distribution function. The distortion of the distribution function affects the wave propagation through the dielectric tensor $\epsilon$ (1.22), and thus, the profile of deposited power, which will in its turn affect the distribution function. Self-consistent calculations of the wave field and distribution function is hence needed in order to model RF heating.
2 Numerical treatment of self-consistent calculations

Self-consistent calculations of RF heating solves the coupled problem of the Fokker-Planck equation and the wave equation. This is achieved by using the codes LION and FIDO, described in sections 2.1–2.2, as major components in the new code package SELFO [24, 28, 33]. The dielectric tensor (1.22) is calculated from the simulated ion distribution including the full Bessel functions. The calculated dielectric tensor is used as input to the LION code. A new wave field and new power partition between the ions and the electrons are calculated with LION. The new wave field and the power partition are then used in FIDO to evolve the distribution in time. This is repeated in an iterative manner. The scheme of the calculations is shown in figure 2.1. The most resent plug-in in the SELFO package is the THETA code [34]. In THETA, the direct RF power to the background and the collisional power transfer from the heated ions, calculated by FIDO, are used to calculate the time evolution of the background temperature profiles using a semi-empirical model [35].

Previously, self-consistent calculations have been carried out in 1D [36], in 2D for lower hybrid current drive [37], for electron cyclotron heating and current drive [38] and for ion cyclotron heating [39]. The calculations presented here are however the first calculations including finite orbit widths and RF induced transport.

2.1 The wave field

In the applications of RF heating, the antenna is usually treated as a sheath current \( J_i \). The wave equation is solved in a spatial mesh, including the vacuum region between the plasma and the wall. In the high frequency domain, the wall can be regarded as perfectly conducting. The dielectric tensor is treated as a constant of matter while solving the wave equation.

In toroidal geometry, the LION\(^1\) code [40, 41] solves the wave equa-

\(^1\)Lausanne ION-2-D-toroidal-global-wave-code
2. Numerical treatment of self-consistent calculations

The LION code

The FIDO code

\( \chi_D(r, k) \)

\( E(x) \)

MHD equilibria

The THETA code

\( T_{ie}(r) \)

\( p_{ie}^f(r) \)

\( p_{ie}^{coll}(r) \)

Figure 2.1 Algorithm used in the SELFO code. The susceptibility of the simulated ions, calculated with the FIDO code, is fed into the LION code. The structure and amplitude of the wave field obtained is used in the next step of the iteration, calculating the further time evolution of the distribution function. A semi-empirical transport model THETA is used for the time evolution of the background temperatures and densities.
2.2. The distribution function

The distribution function (1.16) using the warm dielectric tensor (1.26). If the wavelength of the fast wave and the ion Bernstein wave matches anywhere in the plasma, mode conversion might occur. For the fast wave, the LION code includes all relevant physics as long as the local homogeneous approximation is valid and no mode conversion occurs. To include mode conversion a second order in expansion in $\rho$ is needed. The dielectric tensor will then include spatial derivatives of the distribution function. The PENN code is designed for solving the wave field to second order in $\rho$ [42]. Attempts are made to develop a code valid to all orders of $\rho$ by using a full Fourier decomposition in space [43].

The equation of the perpendicular components of the fast wave equation in cold plasmas and the MHD equation have similar structure. Therefore the first version of the LION code was based on the MHD code ERATO [44]. By approximation

$$k_i \approx \frac{n_i \Phi}{R}$$

in the dielectric tensor, finite parallel temperature is introduced according to the luke warm model. Both harmonic ion absorption and electron damping, with the possibility of calculating the electron current drive [45, 46], has later been included in LION. LION uses FEM for solving the wave equation in the angle coordinate $\chi$ and the flux coordinate $s = \sqrt{\Psi}$ in toroidal geometry. Fourier decomposition is used in the toroidal direction.

By expanding to the second order effect of the Larmor radii, as done in the PENN\(^2\) code [42], mode conversion can be treated, but the calculations then consume more CPU time. However, for high energy ions, it is important to keep the full Bessel functions. This is done in the SELFIO calculations, making them valid for fast wave calculations given that the local homogeneous approximation is valid.

### 2.2 The distribution function

The full toroidal 6D distribution function can be reduced to three invariants in a space of constants of motion of the unperturbed drift orbit

$$\tilde{E} = \frac{v^2}{2}$$

$$\Lambda = \frac{B_0 v^2}{BV^2}$$

$$\tilde{P}_\phi = R v_s + \frac{q \Psi}{m}$$

\(^2\)The god for the mountain range the Pennine Alps in Switzerland, originating from the Roman god Penn.
where $\Psi$ denotes the flux function. The orbit averaged distribution function is solved with the Monte Carlo code FIDO\(^3\) [3, 16, 47], using the three orbit invariants $(\tilde{E}, \Lambda, \tilde{P}_\phi)$ as Monte Carlo test particles, assuming a stationary wave field and neglecting the self collisions of the Monte Carlo particles. To distinguish between the orbits with the same invariants, a label $\sigma$ is introduced. The parallelization of the code is described in appendix A. The Monte Carlo solution of the orbit averaged distribution function consists of a sum of Dirac distributions

$$f(\tilde{E}, \Lambda, \tilde{P}_\phi; \sigma) = \sum_{i=1, j=0, \pm 1}^N \frac{1}{J} \delta \left( \tilde{E}_i - \tilde{E}, \Lambda_i - \Lambda, (\tilde{P}_\phi)_i - \tilde{P}_\phi \right) \delta_{\sigma, j}$$  (2.3)

where $N$ denotes the number of test particles in the simulation and $J$ the Jacobian. The mapping of invariants to real phase space is described by the function $\tilde{r}$

$$\tilde{r}(\tilde{E}, \Lambda, \tilde{P}_\phi, \tau; \sigma) = (r, \theta, v_\parallel, v_\perp) = (r, v)$$  (2.4)

as a curve in the 4D phase space. The curve is parameterized by the time $0 \leq \tau \leq 1$, normalized with the poloidal drift orbit period.

### 2.3 Dielectric tensor

The dielectric tensor is calculated from the distribution function (2.3), given by the FIDO code. The distribution function in FIDO is composed by a set of the triples of orbit invariants in equations (2.2a–2.2c) and the label $\sigma$. One Monte Carlo test particle contributes to several volume elements, and the volume elements are traversed by the orbits of several Monte Carlo test particles, as shown in figure 2.2. For each test particle, the corresponding drift orbit is traced in 4D phase space using the function $\tilde{r}$ in equation (2.4), by running the parameter $\tau$ from 0 to 1. The contribution by the orbit is added to the distribution function expressed in a linear finite element base

$$f(r, v) = \sum_{i,j,k,l} f_{ijkl} \varphi_i(r) \varphi_j(\theta) \varphi_k(v_\parallel) \varphi_l(v_\perp)$$  (2.5)

where

$$\varphi_i(\xi) = \begin{cases} 0 & \text{if } \xi < \xi_{i-1} \\ \xi_i - \xi_{i-1} & \text{if } \xi_{i-1} \leq \xi < \xi_i \\ \xi - \xi_i & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 1 - \frac{\xi - \xi_i}{\xi_{i+1} - \xi_i} & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{if } \xi \geq \xi_{i+1} \end{cases}$$  (2.6)

\(^3\)Finite-width Ion Drift Orbits
2.3. Dielectric tensor

Derivatives of the distribution function reduces to derivatives of the basis functions, e.g.

\[
\frac{\partial f(r,v)}{\partial v_i} = \sum_{ijkl} f_{ijkl}(r) \varphi_i(r) \varphi_j(\theta) \frac{\partial \varphi_k(v_i)}{\partial v_i} \varphi_l(v_{i\perp})
\] (2.7)

Note that

\[
\Omega = \Omega[B(r)] \Rightarrow \Omega = \Omega_{ij}
\] (2.8)

\[
k_{ii} = \frac{n\phi}{R} \Rightarrow k_{ii} = k_{ii}^{ij}
\] (2.9)

\[
k_{i\perp} = k_{i\perp}(r) \Rightarrow k_{i\perp} = k_{i\perp}^{ij}
\] (2.10)

The sub and superscript indices are not co and contra variant components, but a way of avoiding stacking all indices as subscripts. The indexes of \(\Omega_{ij}, k^{ij}\) and \(k_{i\perp}^{ij}\) will, for simplicity, be dropped. The dielectric tensor is obtained by inserting the distribution function (2.5) in (1.22)

\[
\epsilon(k, \omega, r, \theta) = \left(1 - \frac{\omega^2}{\omega_p^2}\right) \left[1 - \sum_{n=0}^{\infty} \sum_{i,j,k,l} h_{ijkl}(r) \varphi_i(\theta) \varphi_j(\theta)ight]
\]

\[
\times \left[ \int \frac{d\varphi_k(v_{i\perp})}{v_{i\perp}} \sum_{n=0}^{\infty} \frac{\varphi_l(v_{i\perp})}{v_{i\perp}} \frac{n\Omega_i \Pi_i(v_i; n)}{\Omega_i} \frac{d\varphi_k(v_{i\perp})}{d\varphi_k(v_{i\perp})} + \int \frac{d\varphi_l(v_i; n)}{\Omega_i} \frac{d\varphi_k(v_{i\perp})}{d\varphi_k(v_{i\perp})} \right]
\]

\[
\times \left[ \int d\varphi_i \left( \frac{k_i \Pi_i(v_i; n)}{n\Omega_i + k_i v_i - \omega - i0^+} \varphi_i(v_i) + \int d\varphi_i \frac{k_i \Pi_i(v_i; n)}{n\Omega_i + k_i v_i - \omega - i0^+} \varphi_i(v_i) \right) \right]
\] (2.11)
2. Numerical treatment of self-consistent calculations

where $\Pi_{\bot}^q(v_{\bot};n) \otimes \Pi_{\bot}^p(v_{\bot};n) = \Pi_{\bot}^q(v_{\bot}, v_{\bot}; n)$ and $\otimes$ is defined as $(A \otimes B)_{ij} = (A)_{ij}(B)_{ij}$. The singularity at $n\Omega_0 + k_n v_n - \omega = 0$ is treated using the Plemelj formula

$$
\int_{-\infty}^{\infty} \frac{dv}{v-a-i0^+} f(v) = i\pi f(v) \bigg|_{v=a} + \wp \int_{-\infty}^{\infty} \frac{dv}{v-a} f(v) \quad (2.12)
$$

where $k > 0$ and $\wp$ is the Cauchy principal value, defined as

$$
\wp \int_{-\infty}^{\infty} f(v) \, dv = \lim_{\delta \to 0} \left( \int_{-\infty}^{x_s-\delta} f(v) \, dv + \int_{x_s+\delta}^{\infty} f(v) \, dv \right) \quad (2.13)
$$

where $x_s$ is the position of the singularity. The imaginary part of the dielectric tensor will give the wave absorption by the particles.

Only the components perpendicular to the magnetic field are included in the LION code. $k_{\bot}$ is calculated from the fast wave dispersion relation in the LION code. In order to speed up the calculations, all the integrals are pre-calculated according to

$$
c_{nijl}^{xxv_{\bot}} = 2\pi \int_{0}^{\infty} dv_{\bot} \frac{n^3 \Omega^3}{k_{\bot}^2} J_n(\rho) \frac{\partial \varphi_1(v_{\bot})}{\partial v_{\bot}} \quad (2.14a)
$$

$$
c_{nijk}^{xv_{\bot}} = \wp \int_{-\infty}^{\infty} dv_{\bot} \frac{1}{n\Omega + k_n v_n - \omega} \varphi_k(v_{\bot}) \quad (2.14b)
$$

$$
c_{nijk}^{v_{\bot}v_{\bot}} = i\pi \frac{\partial \varphi_k(v_{\bot})}{\partial v_{\bot}} \bigg|_{v_{\bot}=(\omega-n\Omega)/k_{\bot}} \quad (2.14c)
$$

$$
c_{nijl}^{xxv_{\bot}} = 2\pi \int_{0}^{\infty} dv_{\bot} \frac{n^2 \Omega^2}{k_{\bot}^2} J_n(\rho) J_n'(\rho) \varphi_1(v_{\bot}) \quad (2.14d)
$$

$$
c_{nijk}^{xv_{\bot}v_{\bot}} = \wp k_{\bot} \int_{-\infty}^{\infty} dv_{\bot} \frac{1}{n\Omega + k_n v_n - \omega} \frac{\partial \varphi_k(v_{\bot})}{\partial v_{\bot}} \quad (2.14e)
$$

$$
c_{nijk}^{v_{\bot}v_{\bot}v_{\bot}} = i\pi \frac{\partial \varphi_k(v_{\bot})}{\partial v_{\bot}} \bigg|_{v_{\bot}=(\omega-n\Omega)/k_{\bot}} \quad (2.14f)
$$

$$
c_{nijl}^{xxv_{\bot}} = 2\pi \int_{0}^{\infty} dv_{\bot} i v_{\bot} \frac{n^2 \Omega^2}{k_{\bot}^2} J_n(\rho) J_n'(\rho) \frac{\partial \varphi_1(v_{\bot})}{\partial v_{\bot}} \quad (2.14g)
$$

$$
c_{nijl}^{xxv_{\bot}} = 2\pi \int_{0}^{\infty} dv_{\bot} i v_{\bot} \frac{n\Omega}{k_{\bot}^2} J_n(\rho) J_n'(\rho) \varphi_1(v_{\bot}) \quad (2.14h)
$$

(2.14i)
2.3. Dielectric tensor

\[
\varepsilon_{\gamma \gamma v'_\perp} = 2\pi \int_0^\infty dv'_\perp v'_\perp^2 n\Omega \left[f'_n(\rho)\right]^2 \frac{\partial \varphi_l(v'_\perp)}{\partial v'_\perp} \tag{2.15a}
\]

\[
\varepsilon_{\gamma \gamma v'_\perp} = 2\pi \int_0^\infty dv'_\perp v'_\perp^3 \left[f'_n(\rho)\right]^2 \varphi_l(v'_\perp) \tag{2.15b}
\]

The dielectric tensor is reduced to a sum of coefficients

\[
\varepsilon(\mathbf{k}, \omega, r, \theta) = \left(1 - \frac{\omega^2_p}{\omega^2}\right) I + \sum_{\eta} \sum_{n=-\infty}^{\infty} \sum_{i,j,k,l} h_{ijkl} \varphi_i(r) \varphi_j(\theta) I_{ijkl} \tag{2.16}
\]

where

\[
\begin{align*}
(I_{ijkl})_{xx} &= c_{xxv'_\perp}^{nijl} \left(c_{\Re v'_\perp}^{njk} + c_{\Im v'_\perp}^{njk}\right) + c_{xxv'_\perp}^{nijl} \left(c_{\Re v'_\perp}^{njk} + c_{\Im v'_\perp}^{njk}\right) \tag{2.17a} \\
(I_{ijkl})_{xy} &= c_{xyv'_\perp}^{nijl} \left(c_{\Re v'_\perp}^{njk} + c_{\Im v'_\perp}^{njk}\right) + c_{xyv'_\perp}^{nijl} \left(c_{\Re v'_\perp}^{njk} + c_{\Im v'_\perp}^{njk}\right) \tag{2.17b} \\
(I_{ijkl})_{yx} &= - (I_{ijkl})_{xy} \tag{2.17c} \\
(I_{ijkl})_{yy} &= c_{yyv'_\perp}^{nijl} \left(c_{\Re v'_\perp}^{njk} + c_{\Im v'_\perp}^{njk}\right) + c_{yyv'_\perp}^{nijl} \left(c_{\Re v'_\perp}^{njk} + c_{\Im v'_\perp}^{njk}\right) \tag{2.17d}
\end{align*}
\]

It is possible to add the contribution of each particle directly to the dielectric tensor without having to store the full 4D distribution (2.5). This gives a significant reduction of the memory consumptions, since the full distribution function requires

\[
N_f = n_r n_\theta n_s n_\perp = \mathcal{O}\left(n^4\right) \tag{2.18}
\]

points, whereas using equation (2.16) requires

\[
N_I = 4n_r n_\theta n_s + 6n_r n_\theta n_\perp + n_r n_\theta n_\perp = \mathcal{O}\left(n^3\right) \tag{2.19}
\]

points. Calculating the dielectric tensor without storing the full distribution function and without pre-calculating the integrals is extremely CPU expensive, since the integrals have to be calculated for each particle in every spatial mesh point. The pre-calculations and the dielectric tensor summation are carried out in parallel according to appendix A.
3 Development of FIDO-TNG

The 1.x, currently 1.2pl6, version of the FIDO code [3,16,47] has successfully been used to explore new areas of ICRF physics [3,17–34,47–58], as stand alone or as a major component of the SELFO code [24,50,59]. At 1999-02-12, version 1.2 was branched as the last circular version. The 1.2 branch was from that date only patched, although some of the patches are rather substantial. A new version of the FIDO code, currently 2.0prealpha5, is in development under the name FIDO-TNG. The suffix TNG\(^1\) is originating from the second version of the famous TV series Star Trek [60] called “Star Trek: The Next Generation” [61]. The new version is aiming for more realistic cross sections and more modularity.

3.1 Planned improvements

The main purpose of the upgrade is to support non-circular equilibria. The TNG version will internally generate a Stringer equilibrium [62,63], or read one from the CHEASE code [64]. All diagnostics will be generated in 2D.

During the development of the TNG version of FIDO, great emphasis has been put on modularity and documentation of the classes and algorithms. The design goal is a code that is easy to maintain and improve in the future. Details about the technical environment is presented in appendix B. The development is eased by the fact that a lot of the numerical problems have been addressed and proven to work in the 1.x version of FIDO.

3.2 Design of FIDO-TNG

The classes in FIDO-TNG are shown in figure 3.1. The most evolved of those will be described in the following sections.

\(^1\)The suffix TNG is not official determinated, but used as an internal project name.
Figure 3.1 The classes in the FIDO-TNG code.
3.2. Design of FIDO-TNG

3.2.1 The point classes

The points classes are available in the four coordinate systems \((R,Z)\), \((\Psi,\chi)\) and \((s,\chi)\) with child class in \((s,\chi,v_\parallel,v_\perp)\). Conversions between the coordinates are done through methods in the \texttt{Mesh} class.

3.2.2 The \texttt{Mesh} class

The representation of the mesh is irrelevant for the rest of the code. The orbit solver requires the information of the mesh properties and the derivatives of those in a given point. Thus the mesh data structure is hidden from the outside in the \texttt{Mesh} object, allowing a complete rewrite of the representation without disturbing the rest of the code. The \texttt{Mesh} object is reusable in other fusion plasma codes requiring the profiles in a given point, e.g. ray tracing codes.

The main \texttt{Mesh} object has two children classes used only for the initialization of the object. The \texttt{StringerEquilibria} class generates a Stringer equilibrium [62, 63]. The \texttt{CircularEquilibria} class generates an equilibrium similar to FIDO 1.x used for comparing the two versions. A \texttt{CheaseEquilibria} class is planned reading an equilibrium from the CHEASE code [64].

Internally in the \texttt{Mesh} class, the coordinates \((R,Z,\chi)\) are stored in a 2D mesh equidistant in \(s\). For each mesh point the \(B_T, B_p\) etc. are stored. Flux surface properties, e.g. background densities, are stored as well in the \texttt{Mesh} class. The \(s\) and the \(\chi\) values are interpolated according to

\[
f = \sum f_{ij} \phi_{ij}(i + 1 - a, i + 1 - b) \tag{3.1}
\]

where \(i\) and \(j\) are the indices of the mesh cell, the distance between the mesh cells is 1, \(\phi\) is the 2D hat function and \(a\) and \(b\) are interpolation coefficients satisfying

\[
R = (1 - a)(1 - b)R_{ij} + a(1 - b)R_{i+1,j} + (1 - a)bR_{i+1,j+1} \tag{3.2}
\]

and

\[
Z = (1 - a)(1 - b)Z_{ij} + a(1 - b)Z_{i+1,j} + (1 - a)bZ_{i+1,j+1} \tag{3.3}
\]

The base function \(\phi\) is a class with inlined methods, enabling easy switch to another base function without compromising the efficiency. Most of the public methods in the \texttt{Mesh} class give a property of the mesh at a given point. The methods are generally available in three version for the three coordinate systems described in section 3.2.1. In order to give e.g. \(B_T(R,Z)\) or \(R(\Psi,\chi)\), the local mesh coordinates \(i, j, a\) and \(b\) have to be
found. First the mesh cell \((i,j)\) is located through an alternating secant method in \(i\) and \(j\), and then, if this method does not converge, searching the mesh cells outwards in a spiral from the last guess. Once the mesh cell is found, \(a\) and \(b\) are solved. The latter step might require a non-linear \(2 \times 2\) Newton Raphson solver. The \(\text{FindPoint}\) methods have been tested on various meshes, e.g. a Stringer mesh, shown in figure 3.2, a circular mesh, shown in figure 3.3 and a highly irregular mesh used for testing purpose, shown in figure 3.4. The last lookup of the \(\text{FindPoint}\) method is internally cached. This behaviour makes most of the methods in the \textit{Mesh} class not thread safe. However, the constructor and the destructor methods are thread safe, and allow every thread to create its own copy of the object, sharing the memory of the large equilibrium mesh and mesh properties. One way of obtaining a good initial guess, and thereby speed up the \(\text{FindPoint}\) method further, is to make course grids of the mesh indices \(i\) and \(j\) as functions of \((R,Z)\) and \((s,\chi)\) \cite{65}

The various derivatives are calculated consistently with equation (3.1), e.g. the derivative

\[
\frac{\partial f}{\partial a} = \sum f_{ij} \frac{\partial}{\partial a} [q_{ij}(i+1-a,j+1-b)] \tag{3.4}
\]

In order to calculate e.g. the derivative \(\frac{\partial R}{\partial \chi}\), the derivatives \(\frac{\partial \chi}{\partial R}\) etc. are needed

\[
\begin{pmatrix}
\frac{\partial R}{\partial s} & \frac{\partial R}{\partial Z} \\
\frac{\partial Z}{\partial s} & \frac{\partial Z}{\partial \chi}
\end{pmatrix} = \begin{pmatrix}
\frac{\partial s}{\partial R} & \frac{\partial s}{\partial Z} \\
\frac{\partial \chi}{\partial R} & \frac{\partial \chi}{\partial Z}
\end{pmatrix}^{-1} \tag{3.5}
\]

The right hand side can be expressed in the mesh coordinates \(a\) and \(b\).

\[
\frac{\partial s}{\partial R} = \frac{\partial s}{\partial a} \frac{\partial a}{\partial R} \tag{3.6}
\]

\[
\frac{\partial s}{\partial Z} = \frac{\partial s}{\partial a} \frac{\partial a}{\partial Z} \tag{3.7}
\]

\[
\frac{\partial \chi}{\partial R} = \frac{\partial \chi}{\partial a} \frac{\partial a}{\partial R} + \frac{\partial \chi}{\partial b} \frac{\partial b}{\partial R} \tag{3.8}
\]

\[
\frac{\partial \chi}{\partial Z} = \frac{\partial \chi}{\partial a} \frac{\partial a}{\partial Z} + \frac{\partial \chi}{\partial b} \frac{\partial b}{\partial Z} \tag{3.9}
\]

\(\frac{\partial s}{\partial R} = 0\) is a property of the mesh. \(\frac{\partial s}{\partial a}, \frac{\partial \chi}{\partial a}\) and \(\frac{\partial \chi}{\partial b}\) are given by equation (3.4). The remaining derivatives are given by another \(2 \times 2\) matrix inversion.

\[
\begin{pmatrix}
\frac{\partial a}{\partial R} & \frac{\partial a}{\partial Z} \\
\frac{\partial b}{\partial R} & \frac{\partial b}{\partial Z}
\end{pmatrix} = \begin{pmatrix}
\frac{\partial R}{\partial a} & \frac{\partial R}{\partial b} \\
\frac{\partial Z}{\partial a} & \frac{\partial Z}{\partial b}
\end{pmatrix}^{-1} \tag{3.10}
\]
3.2. Design of FIDO-TNG

Figure 3.2 A Stringer equilibrium.

Figure 3.3 A FIDO 1.x equilibrium.

Figure 3.4 A highly irregular mesh used for stressing the FindPoint methods.
3.2.3 The Table\textsuperscript{3D} class

The Monte Carlo increments are expensive to calculate, therefore tabulation and interpolation are necessary. A high resolution near the boundaries in the invariant space where the orbit changes topology is required, whereas in other parts, the increments are slowly varying, and hence a lower resolution is sufficient. An equidistant interpolation mesh is expensive in the slowly varying parts. A mesh with regular stacking with independent weighting of the x, y and z-axes is better but still has drawbacks. A very dense mesh in areas where it is not needed is generated when resolving e.g. the surface of a sphere in a box. Triangles are commonly used for arbitrary stacking in 3d, but finding the right triangles fast might be expensive. An adaptive interpolating mesh of independently subdivided boxes is therefore developed. An example of the flexibility of this method is show in figure 3.5, where a circle is resolved in a square using five levels of refinement.

The interpolation is done using the regular 3d hat basis $\phi$

$$f = \sum_{i,j,k} h_{ijk} \phi_{ijk}(x,y,z)$$

(3.11)

The mesh is first divided into a coarse equidistant mesh. A weight function $W(D) \in (0, 1)^3$, where $D$ is the domain of the cell, giving the refinement level three directions $x, y$ and $z$, is introduced as

$$n_{\text{recursive subdivisions in } \xi} = W_\xi N_{\text{max depth}}$$

(3.12)

where $\xi$ is the $x$, $y$ and $z$-component. For simplicity, a function $w$ is introduced

$$W_\xi = \max_{(x,y,z) \in D} w_\xi(x,y,z) \quad \xi \in \{x,y,z\}$$

(3.13)

Once a cell is populated with values, it will be recursively subdivided independent in all three directions until the depth given by the weight function is reached. The interpolation values will not be recalculated nor stored in duplicates at points in the finer grid matching points in a coarser grid. While refining a mesh cell, a list of the next level of subdivided cells created is stored, i.e. a mesh cell can find its subdivided cells in time $O(1)$. Finding a point in an equidistant mesh is of order $O(1)$, and hence the lookup speed of this tables will be of order $O(N_{\text{max depth}})$.

For optimal performance, the table must be refined in regions where the derivatives are changing rapidly. One way of expressing this condition is to use a weight function

$$w = \left(\frac{|\partial^2 f|}{\partial x^2}, \frac{|\partial^2 f|}{\partial y^2}, \frac{|\partial^2 f|}{\partial z^2}\right)$$

(3.14)
Figure 3.5 A interpolating table resolving a circle.
3. Development of FIDO-TNG

Evaluation equation (3.14) is in some cases numerically expensive. An approximation is sufficient given that

\[ w_{\xi}^{\text{approx}}(x) \geq w_{\xi}^{\text{optimal}}(x) \quad \forall \xi \in \{x,y,z\}, x \in \text{[table domain]} \] (3.15)

The implementation of this method of interpolating tables in FIDO-TNG is the template class Table3D. Using a template class enables interpolating tables to be built on all sorts of objects implementing a set of interpolation operators such as \(*=\langle\text{double}\rangle\) and \(+=\). The Table3D class has the option of transparently switch from reading interpolated values to do computations of the values in areas where interpolation is not sufficient.

An example of the independent refining along the three axes is shown in figure 3.6. The table was created using the weight function

\[ w(x,y,z) = \exp \left( - \left( x^2 + y^2 + z^2 \right) \right) \left( 1, \frac{1}{2}, \frac{1}{4} \right) \] (3.16)

The interpolation error of the table is demonstrated using the function

\[ f = \exp \left( - \left( x - \frac{\pi}{2} \cos(z) \right)^2 + \left( y - \frac{\pi}{2} \sin(z) \right)^2 \right) \] (3.17)

where \( x, y, z \in (-\pi, \pi) \). The, in this case, isotropic weight function is given by

\[ w = \exp \left( - \left[ x - \frac{\pi}{2} \cos(z) \right]^2 - \left[ y - \frac{\pi}{2} \sin(z) \right]^2 \right) \] (3.18)

The course table size is \( 8 \times 8 \times 8 \) with the max depth of 5. The absolute error of the interpolation is given in figure 3.7. The resulting 34015 table cells are shown in figure 3.8. As a comparison, the error and the points of an equidistant table with \( 34^3 = 39304 \) points, i.e. 35937 cells, is given in figures 3.9 and 3.10. Clearly, the adaptive table is performing much better, while still comparable in speed with the equidistant one.

3.2.4 The Diagnostics class

The diagnostics is generated by tracing the orbits in an equidistant mesh in \((s,\chi)\). The handling of the mesh and tracing of the orbits is performed in the Diagnostics class. The output is saved in a well defined binary format in order to increase the speed and reduce the size of the output files. The child classes performing the actual diagnostics implements the public virtual functions

`CleanDiagnostics()`
`AddToDiagnostics(<particle deltaIncrement>)`
`Save(osstream)`
Figure 3.6 A interpolating table created using the weight function (3.16).
Figure 3.7 Absolute error of the interpolation for table size $8 \times 8 \times 8$ with the max depth of 5.
Figure 3.8 Table points of the interpolation for table size $8 \times 8 \times 8$ with the max depth of 5.
Figure 3.9 Absolute error for an equidistant table with $34^3 = 39304$ points.
Figure 3.10 Table points of an equidistant table with $34^3 = 39304$ points.
and the protected virtual function

\texttt{AddContributionToMesh(s\_i, \chi\_j, v\_par, v\_perp, weight)}

The constructor and destructor of the \texttt{Diagnostics} class are able to create copies sharing the memory of the diagnostics coordinate mesh and volume elements. Hence the volume elements are only calculated once for each type of diagnostics mesh.

\subsection*{3.2.5 The \texttt{OrbitSolver} class}

The calculation of the drift orbits for a given set of invariants is done in the \texttt{OrbitSolver} class. The result is cached in an instance of the \texttt{Orbit} class. Since several instances of the \texttt{OrbitSolver} class share a common cache, the orbit does not have to be recalculated every time a set of Monte Carlo operators or \texttt{Diagnostics} classes are given the same particle.

\subsection*{3.2.6 The \texttt{FlexibleVector} class}

The template class \texttt{FlexibleVector} class provides a vector with the capability of growing. Removing of objects and reuse of their positions are supported with methods for removing and finding the first free position. The implementation is a linked list of ordinary C++ vectors with an additional list for fast search of free positions.
4 Summation of the publications

4.1 Paper I: Parasitic alpha particle absorption

In paper I [50], the parasitic absorption by fusion born $\alpha$-particles in JET like DT plasmas is studied for on axis second harmonic tritium heating, a proposed scenario for ITER. Concerns have been raised regarding that the second harmonic $\alpha$-particle resonance might enter the plasma on the low field side as previously addressed in Ref. [66–68]. This paper is an extension of the contribution to EPS conference in Berchtesgaden [33]. In the EPS paper a simplified distribution function was used during the dielectric tensor calculations.

As a first step towards self-consistent calculations the $\alpha$-particle absorption is studied for one iteration with the SELFO code. First, a distribution function of the $\alpha$-particles is calculated with the FIDO code. The dependency of the orbit widths is studied by applying a low and a high toroidal current, resulting in thinner and wider orbits. A steady state distribution, without the RF turned on, is reached by using the fusion reaction rate for the background plasma profiles as a source and removing the $\alpha$-particles after one central slowing down time. The dielectric tensor is then calculated from that distribution function and fed into the LION code. The frequency is increased in three steps from second harmonic tritium heating to electron current drive scenarios, where the second harmonic deuterium and $\alpha$-particle absorption barely enters the plasma on the low field side. By including the finite orbit widths in the dielectric tensor, the effect on the absorption by the high energy $\alpha$-particles born in the center, making large excursions from the flux surfaces, can be addressed.

The second harmonic resonance is found to be insignificantly small. The density of $\alpha$-particles at the edge is still small. Further, using the unexpanded Bessel functions, the high energy ion gyro orbits effect will almost cancel the second harmonic absorption, since the $E_+$ and $E_-$ absorption will be comparable [54]. The $\alpha$-particle absorption is instead found to be completely dominated by the fundamental resonance, which will absorb 20–30% of the power. The wider orbits will broaden the absorption profiles. By inserting a 1% minority of $^3$He, the $\alpha$-particle
absorption drops to 10% and the overall central heating efficiency is significantly improved.

Regarding self-consistency, this paper is to be seen as a first step and a proof of principle. The dielectric tensor is calculated from a FIDO distribution function and fed into the LION code. In this paper, the coupling between the codes is found to be a correction to the previous results.

4.2 Paper II: Minority ion cyclotron current drive

In paper II [23], minority current drive is studied using the FIDO code. The paper is an extension of the letter on the same subject by Hellsten et al. [19]. The Fisch model [69] of ion current drive is found, even in the case of very high plasma current, i.e. thin orbits, to be valid only for low values of coupled RF power, mainly due to the effects of trapped orbits. By solving the Fokker-Planck equation with the FIDO code, the effects of the wide orbits, trapping and detrapping and RF induced transport will be included. The new mechanics of current drive by orbit effects are studied for the resonance on the high field side, on axis and on the low field side for four levels of RF power and a positive, negative and symmetric antenna spectrum. In particular, on axis minority ion cyclotron current drive is found, in contradiction to the Fisch model.

By using one iteration of the self-consistent scheme, the absorption by a fusion born \( \alpha \)-particle distribution function in a 25 keV ITER plasma with 50 MW of coupled RF power is calculated. The \( \alpha \)-particle distribution is generated in the same way as in paper I. A scenario promising for obtaining reversed shear, on axis fundamental heating in a DT plasma, was studied. The RF power was absorbed to 98% by the electrons, and the fundamental \( \alpha \)-particle resonance. By calculating the ion current drive with the FIDO code, including an expression for the back current by the majority ions and electrons, and the electron current drive with the LION code, reversed shear was found.

4.3 Paper III–VI: Influence of the orbit topology

The papers [18,28,29,31], are an investigation of the fundamentals of the influence of the orbit topology and RF induced transport during ICRH. First the effects of single mode number spectra are examined. It is found that for hydrogen heating in deuterium plasmas using the parameters of Tore Supra, the resonance is shifted all the way from the HFS to the LFS when heating with \( n_\phi < 0 \).
4.3. Paper III–VI: Influence of the orbit topology

Figure 4.1 A localized flat distribution of co-passing orbits in $v_\perp$ on the LFS will, as a result of the finite orbit width, result in peaks at different $v_\perp$ on different radii on the HFS.

For the case of $n_{\phi} > 0$, noise and possible emission is seen in the calculations in paper III. At the resonance location, where $v_\parallel = (\omega - \omega_c)/k_\parallel$, interactions result in a distribution function which is monotonically decreasing with the extreme of a flat function in $v_\perp$. Because of the orbit topology this is not necessarily the case in the whole plasma or even on a flux surface. Since orbits with higher energy deviate more from the flux surfaces, a spatial localized flat distribution function in the mid plane on the LFS, will create peaks in the mid plane on the HFS. The phenomena is schematically shown in figure 4.1. Experimental indications of positive slope of the distribution function have been seen by local measurements of the distribution functions of fast ions through collective Thomson scattering [70]. The positive slope of the distribution function, shown in figure 4.2, can result in velocity driven instabilities close to harmonics of the ion cyclotron frequency. The distribution function decreases in $v_\perp$ at the resonance $v_\parallel = (\omega - \omega_c)/k_\parallel$, but has an extra peak of co-passing orbits resulting in a positive slope in $v_\perp$. Calculations of the distribution function with the SELFO code show that a positive slope can occur at various positions in the plasma. The flattening of the distribution function as well as the reduction in wave-particle interaction due to the finite Larmor radius correction at the fundamental frequency, results, in this
case, in a weakening of the H absorption from 3.7 MW to around 2.2 MW at t=0.2 s.

Locally, increasing distribution functions in velocity space, leading to instabilities, can also be obtained in the absence of Doppler shifted absorption at the LFS due to the finite width of the banana orbits. Ions transversing the outer leg of trapped orbits resonating at $R_c$, will create peaks in the velocity plane of the distribution function at the LFS near the antenna.

The flux surface integrated absorbed power for waves with the wave numbers $n_{\phi} = -30$ and $n_{\phi} = 30$ as a function of $r$ is shown in figure 4.3.

A symmetric spectrum is in a first approximation composed of two superimposed waves with opposite toroidal directions of propagation. It is found that the RF induced drift terms do not cancel. A symmetric spectrum is rather the combination of the two drifts. The asymmetry is further enhanced by the fact that the wave mode with $n_{\phi} < 0$ is contracted towards the center of the plasma. The ion confinement is thereby enhanced, which will further increase the ion damping. The result is that the mode with $n_{\phi} < 0$ is likely to have higher ion single pass damping than the mode with $n_{\phi} > 0$, enhancing the inwards drift compare to the outwards one. Unfortunately, a small bug in the output routine described counter passing particles in region vi in figure 1.8 as potato orbits in in papers iii–v. This does however not alter the main physics or conclusions of the papers, which was to emphasis the importance of the non-standard counter passing orbits.

In the last paper, a realistic directed spectrum with one strong mode with low $|n_{\phi}|$ and one opposite directed weak mode with high $|n_{\phi}|$ is studied. The simulations shows that the weak part does significant con-
4.4. Paper VII–IX: Parasitic ion absorption during FWED

In papers [24, 26], the first self-consistent calculations of RF heating, including the finite orbit widths and RF induced transport, in toroidal geometry are presented. Details of the calculations are presented in Table 4.1.

A strong asymmetry in the fast wave electron current drive, is observed as a function of the toroidal direction of the launched wave for on axis third harmonic D plasmas [71–73]. Experimental indications of

**Table 4.1** Details of the calculations presented in papers VII and VIII.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monte Carlo Test particles</td>
<td>100000</td>
</tr>
<tr>
<td>Monte Carlo Time Steps</td>
<td>0.5 ms</td>
</tr>
<tr>
<td>Monte Carlo Time Steps Between LION runs</td>
<td>100</td>
</tr>
<tr>
<td>Dielectric Tensor Resolution</td>
<td>( n_r = 10, ; n_\theta = 20 ) ( n_{\nu_\parallel} = n_{\nu_\perp} = 64 )</td>
</tr>
<tr>
<td>Resolution in LION</td>
<td>( n_x = n_X = 120 )</td>
</tr>
<tr>
<td>FIDO--LION iterations</td>
<td>40</td>
</tr>
<tr>
<td>Run time (3 Sun Ultra I + 3 Sun Ultra 5)</td>
<td>( \approx 14 \text{ h} )</td>
</tr>
</tbody>
</table>

Figure 4.3 Integrated absorption for \( n_\phi = -30 \) and \( n_\phi = 30 \) (solid) at \( t = 0 \) s and \( n_\phi = -30 \) (dashed) and \( n_\phi = 30 \) (dotted) at \( t = 0.2 \) s.

Contribution to the orbit topology since the RF induced transport is higher for high \( |n_\phi| \) [21].

4.4 Paper VII–IX: Parasitic ion absorption during FWED

In papers [24, 26], the first self-consistent calculations of RF heating, including the finite orbit widths and RF induced transport, in toroidal geometry are presented. Details of the calculations are presented in Table 4.1.

A strong asymmetry in the fast wave electron current drive, is observed as a function of the toroidal direction of the launched wave for on axis third harmonic D plasmas [71–73]. Experimental indications of
fast particles has been observed in $\omega = 3\Omega_{c0}$ heating scenarios in ASDEX [74]. By using parameters similar to Tore Supra, the electron current drive efficiency is found to be depending on the amount of parasitic absorption by the ions. Ion absorption will inhibit current drive, since the driven ion current will be much smaller than the driven electron current. Due to the different phasing of the antennae, the ions are radially diffused outwards or inwards, and thereby forming different high energy tails. No such tails are observed in Tore Supra [75]. However, the tails are relative small and could be difficult to detect.

In the case of inward diffusion, the ion confinement is enhanced and a high energy tail is formed, whereas in the case of outward radial diffusion, confinement is degraded, resulting in the formation of a smaller tail. By self-consistent calculations, the former tail is found to give 80% of ion absorption, whereas the latter gives 50%, recovering the same asymmetry in the simulations as seen in the experiments. In these calculations both the feedback of the absorbed power and the finite orbit effects are crucial. Without the feedback of the absorbed power, the tail absorption would not be correctly described. The parasitic ion absorption would remain at the starting 10%. Without the finite orbit widths and rf induced transport, the difference of the tails depending of the direction of the toroidal wave, and hence, the asymmetry would not be resolved.

In paper IX [27], the behavior of the parasitic ion absorption as a function of $|n_\phi|$ is studied. From the figures in the paper, it is clear that the evolution of an ion tail and the time evolution of the wave fields are crucial components for describing the resulting fast wave electron current drive. The change in the ion damping and in the electron current drive efficiency due to that the wave field becomes more of single pass damping type is roughly of equal size.

4.5 Paper X: Teaching numerical methods using Internet based tools

Paper X is off topic from the rest of the thesis. During the summer 1999 and 2000, an intensive graduate course in numerical methods was taught using Internet based tools. The lecture notes are available both as printable PDF files and on the web [76]. The web version includes and interactive Java applet, illustrating the different schemes. The main focus of the course is the exercises, where the student modifies the Java applet and present their solutions as web pages.
5 Conclusions

The physics of RF heating involves a competition between roughly equal terms, e.g. the anti-hermitian parts of the dielectric tensor in the calculation of the power partitions between the ion species and the electrons. To predict the outcome of a given heating scenario, all the dominant effects have to be included. The wave propagation is dependent of the details of the distribution function, and the distribution function is altered by the wave-particle interaction. Self-consistent calculations are therefore necessary to describe the physics of the wave field and distribution function simultaneously. By iterating the distribution function from the FIDO code with the electric field from the LION code, the first self-consistent calculations of ICRH, including finite orbit widths and RF induced transport, have been obtained.

In the case of fusion born \( \alpha \)-particle absorption, finite orbits effects and non-Maxwellian distribution functions are found to be a correction to the power absorption profiles [33, 50]. However, in other cases, the time evolution of the wave field gives rise to new effects, not seen in previous calculations [18, 24–29, 31]. For minority heating at the HFS, it is found that the resonance will shift completely to the LFS when heating with \( n_\phi < 0 \) [28]. The shift is an effect of the RF induced transport affecting the orbit topology and the attenuation of the wave field at the unshifted cyclotron resonance. Resolving the latter effect requires time resolved wave field patterns. The shift of the resonance towards the low field side is present for a symmetric spectrum, consistent with experimental observations at JET [77]. The drift terms of the symmetric spectrum are not canceling, but both the inward and outward pinches are present.

In paper vi, realistic directed spectra are examined. The calculations show that in order to simulate such a spectrum, both the strong peak with low \( |n_\phi| \) and the opposite directed subdominant peak with high \( |n_\phi| \) have to be taken into account. Using only a single toroidal mode number can be misleading, and exaggerate the effects of RF induced diffusion.

In the case of fast wave electron current drive, the use of self-consistent calculations including finite orbit widths and RF induced diffu-
sion is crucial for recovering the physics. The same asymmetry in the
electron current drive efficiency as seen in the experiments [71–73] is
recovered in the simulations as an effect of the parasitic ion absorption,
affected by the mutual interaction of the finite orbits and the wave field.
Without the use of self-consistent models, the ion absorption would stay
at the initial 10%, instead of the 50% and 80% found in the simulation
for the $n_\phi > 0$ and $n_\phi < 0$ wave modes, respectively. Without the RF in-
duced diffusion and finite orbit widths, the asymmetry would not be
resolved.
6 Resume and personal reflections

The self-consistent calculations are shown to be crucial in many common RF heating scenarios. The physics of RF heating is found to be more sensitive to the non-linear coupling of the wave field and the distribution function, than first believed. As the ion tail develops, the damping can increase from 10% to 80%, and completely dominate the physics [24–27]. In the latest paper [31], basic RF physics is studied in detail. As for the future, more physics is to be found with the present version of SELFO. In the longer term future, FIDO-TNG will be a significant upgrade.

The construction of the SELFO code with modular design and well defined file formats for the interface between the two codes, enables parts of the code package to be replaced. If mode conversion is essential, a more evolved wave code could be used, e.g. the PENN code. The $\alpha k$ model for the wave field allows inclusion of the up-shift of the parallel wave number by the finite poloidal magnetic field [78], most important for low $|n_\phi|$. The use of MPI for the parallel version of the FIDO code, makes the code runnable on large super computers. As for now, the FIDO code is reported to be running on e.g. Solaris, AIX on an IBM SP/2 supercomputer and on Linux. The porting effort is reduced by using GNU autoconf system. In the long term future, the most beneficial way of doing numerical science is with open source [79–81]. In order for this to happen, the scientific effort of contributing to an open source project has to be officially recognized in the whole scientific community.

The computing power has grown tremendously since the first parallel production version of FIDO in 1996. In the early days, the computer farm at the department consisted of a Sun ELC, a Sun SS10/40 and a Sun SS4 on a shared BNC Ethernet. Today there is a twin CPU Sun Ultra 60, a Sun Ultra 10, three Sun Ultra 5 and older Ultras with four twin CPU Sun Blade 1000 and another Ultra 5 ordered. The net is switched 100 Mb TP Ethernet. Using an additional parallel private 100 Mb switch network for MPI traffic only is planned. The new computer power has and will be used to go to higher degree of complexity and accuracy, e.g. several mode numbers and coupling to the THETA code. The new machines are
most welcome, since the most complex calculations performed at the moment requires wall clock times in the order of 100 hours running in parallel on the whole system [82]. FIDO-TNG will probably require more computational power compared to FIDO v1.x, due to increased complexity in the geometry. The computer resources of the laboratory are good with the new machines, but still by far not unreasonable for a small laboratory performing simulations. In order to speed up the calculations, and hence further increase the resolution for a given amount of computer time, it is necessary to reduce the time spent on calculating the thermal bulk ions, which have a high collision frequency and are dominating in number. This can be achieved either using a weighting or using a hybrid Monte Carlo method, recently developed for 1D [83]. It is however not trivial to extend this model to three dimensions.

Monte Carlo methods involves interesting numerics and computer science. Many common numerical problems is treated from a slightly different angle, e.g. the demand for advanced and fast interpolation tables in 3D. With the development of first FIDO and then SELFO, a powerful tool is available, which can be used to further increase the understanding of the complicated ion behavior during ICRF heating.
A Parallelization of FIDO 1.x

Monte Carlo methods are intrinsic parallel, and therefore well suited for running on parallel computers. The most calculation intense parts of the FIDO code and the dielectric tensor integration are parallelizable with time $O(1)$ and work $O(n)$ on an EREW PRAM, i.e. the best possible scenario. More information about the work–time framework can be found in a textbook in parallel algorithms [84].

A.1 Architectures of parallel machines

There are two fundamental different types of parallel machines. In the shared memory machine, shown in figure A.1, the processors share the same main memory. Examples of such super computers are CRAY J90 and Sun Enterprise T-10000. When writing code for shared memory machines, the basic considerations are locking of memory and system resources by a specific processor. The threads library, normally used in such machines, includes all sorts of locking mechanisms [85, 86]. The second type of parallel machines have separate memory for each processor, as shown in figure A.2. Examples of such super computers are IBM SP/2 and CRAY T3E. These super computers are basically powerful workstations connected with a high speed network. A unit of processor and memory is called a node. Since each node has separate memory, messages and data need to be sent between the nodes. The two most
frequently used libraries for node communications are MPI\textsuperscript{1} \cite{87,88} and PVM\textsuperscript{2} \cite{89}. Both standards are available as free software for virtually all Unix systems, enabling development of distributed memory machine parallel code on a network of Unix workstations. The former type of super computers are easier to program, and it is easier to achieve good parallelization. The latter type is the most powerful, and is also available in a much smaller version as networks of workstations. The future top super computers will be a combination of the two, where each node is a shared memory machine as shown in figure A.3. An extended introduction to the different architectures and programming models of parallel machines is found in Ref. \cite{90}.

A.2 Parallel algorithm in FIDO 1.x

The work of parallelize the FORTRAN part of the FIDO code has been done in collaboration with J. Carlsson. The calculation of the dielectric tensor, is developed in C as parallel code. The parallelization is carried out using the MPI library on a network of Sun workstations. The FIDO-TNG code is planned to include a shared memory version, to be used as stand alone, or in combination with MPI. Both a sequential and a parallel version of the FIDO code is included in the 1.2-patches branch of the cvs\textsuperscript{3} tree \cite{91}, separated by a compiler option

```c
#include MPI
```

\textsuperscript{1}Message Passing Interface  
\textsuperscript{2}Parallel Virtual Machine  
\textsuperscript{3}Concurrent Versions System
A.2. Parallel algorithm in FIDO 1.x

Figure A.3 The future super computers are distributed memory machines with small shared memory machines as nodes.

```
parallel_loop_over_particles(arg1, ...);
```  

else
```
sequential_loop_over_particles(arg1, ...);
```  

endif /* MPI */

Both loops over the particles call the same function

```
take_timestep(particle, ...)
```  

Since the code runs on a set of desktop Unix workstations with different CPU power, the problem has to be distributed accordingly for maximum performance. Since the computers are used by others, the code must be able to redistribute the problem if the load of the nodes changes during the run. This is known as dynamic load balancing. In the Monte Carlo part of the FIDO code, the redistribution is done for each diagnostic time step. In the dielectric tensor calculation part, every small package is load balanced, yielding almost real time dynamic load balancing of the calculation. Using more frequent redistributions of the problem for the Monte Carlo part would not increase the efficiency, since the network versus CPU load is higher than for the dielectric tensor calculations.

In order to achieve the dynamic load balancing, a master–slave technique is used. One node is the master and the rest are slaves, carrying out the work. The structure of the `main()` function is the following

```c
int myid;
MPI_Comm_rank(MPI_COMM_WORLD, myid);
if (myid == 0){
    DoMasterStuff();
} else {
    WaitForMastersCommand();
```
A. Parallelization of FIDO 1.x

Figure A.4 Parallel MPI algorithm used in the Monte Carlo part of FIDO.

Both the master and a slave with lower priority runs often on the same node. This is a more optimal use of the computer resources, since the master is mostly waiting, and thereby not consuming cpu power. The algorithms used is schematically shown in figures A.4 and A.5. In the figures, the run state of, and the communication between, one master m and two slaves s are shown as a function of time. The run time of the slaves is shortened to increase clarity. This results in an exaggerated fraction wait versus run time for the slaves in the figures.

The communication between two nodes is known as point to point operation. Any node may communicate with any other node, but a node can only communicate with one other node at the same time. The time for one communication between two nodes will be denoted point to point operational time step.

A.2.1 Parallel algorithm for the Monte Carlo calculations

The algorithm used for the Monte Carlo calculations is shown in figure A.4. In the figure, the bottom slave node is faster than the top slave node. The iterations of step 2 and 3 below, will assign the particles to the slaves in such a way that the time to process the subset of particles
on each slave is equal.

1. The master starts by broadcasting the global parameters to the slaves. This is done internally in MPI by a tree algorithm, yielding $O(\log n)$ point to point operational time steps.

2. The master sends a small subset of the invariants for processing to the slaves. The master then waits for the slaves to report when done. If there are packages left, a new package will be sent to the reporting slave for further processing. A typical package in the FIDO code is 500 particles.

3. The slaves report to the master after processing the package of particles.

4. When all the particles are assigned to and processed by the slaves, the master tells the slaves to sum the power absorbed by their particles. The global sum of absorbed power is needed in order to normalize the electrical field, and thereby keep the global absorbed power constant.

5. The global sum of absorbed power is computed on to the master. This is also done by a tree algorithm, needing $O(\log n)$ point to point operational time steps.

6. The master broadcasts the normalization constant of the electric field to the slaves and tells them to process their assigned set of particles.

7. When the slaves are finished with the processing of their particles, the new global absorbed power is summarized to the master.

8. In order to be able to redistribute the problem, and to save the particles to a file, the master tells the slaves to send their particles to the master.

9. The slaves send their invariants to the master.

10. After the last time step, the master tells the slaves to quit.

A.2.2 Parallel algorithm for the dielectric tensor integration

In the dielectric integration there is no correspondence to the iteration of steps 6–7 in the algorithm presented above. An almost real time dynamic load balancing is possible in this case, since the nature of the problem implies more CPU load versus net load than a Monte Carlo problem. The algorithm is shown in figure A.5. In this example, the top slave node is faster than the bottom slave node.
1. The master starts by broadcasting the global parameters, including spatial resolved $k_\perp$, and all the particles to the slaves.

2. In order to build the mesh for the distribution function (2.5), the global min ($v_n$), max ($v_n$) and max ($v_\perp$) have to be found. The master assign small disjunct sets of invariants to the slaves, in which the slaves search for the local min ($v_n$), max ($v_n$) and max ($v_\perp$).

3. The slave reports to the master, when ready for a new task.

4. To find the global min ($v_n$), max ($v_n$) and max ($v_\perp$), the master tells the slaves to reduce their local min ($v_n$), max ($v_n$) and max ($v_\perp$).

5. The local min ($v_n$), max ($v_n$) and max ($v_\perp$) on the slaves are reduced to the global ones on the master.

6. The master broadcasts the mesh size in $v_n$ and $v_\perp$.

7. The master tells the slaves to pre calculate the integrals (2.14)-(2.15) in small disjunct hyper volumes in phase space.

8. When the integral for the whole mesh in phase space is calculated, the master tells the slaves to reduce the integrals onto the master.

9. The slaves reduces the integrals onto the master.

10. The master then broadcasts the full set of pre calculated integrals to all the slaves.
11. The master assigns the slaves small disjunct subsets of the particles for tracing the corresponding orbits in phase space and add the contribution to the dielectric tensor.

12. When all particles are processed by the slaves, the master tells the slaves to summarize the dielectric tensor.

13. The dielectric tensor is summarized to the master. This is done by a tree algorithm, needing $O(\log n)$ point to point operational time steps.

14. The master tells the slaves to quit.
B Technical environment

As for maximum modularity and flexibility without compromising the parallelization and performance properties, C++ [92–94] is chosen as the main language. The code is rebuilt from the start, making a complete object oriented design. In general modern tools commonly available with a predicted stable future are chosen. Table B.1 shows the tools used.

B.1 C++ and object oriented design

The basic idea of object oriented design is to gather both data structures and the operation on those structures into objects [95]. The object oriented way of designing is well suited for scientific codes. There are a lot of object oriented languages e.g. C++, Java [96] and SMALLTALK [97]. C++ was chosen since it is fast and general enough to run efficiently with both threads [85,86] and MPI [87,88], used on the present and future high end super computers and affordable workstations clusters (cf. appendix A). For the low end graphical parts, Java is chosen due to its pureness and huge standard class library. Even high end 3D visualization is possible through Java 3D [98,99] working natively with OpenGL [100,101]. The standard C++ class library [102], is chosen not to be used, since many of the compilers on the target platforms do not have a sufficiently good implementations of them yet. The naming convention of Java is used [103].

<table>
<thead>
<tr>
<th>C++</th>
<th>Main language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>Low performance graphics and GUI</td>
</tr>
<tr>
<td>CVS</td>
<td>Code management and version control</td>
</tr>
<tr>
<td>GNU Automake</td>
<td>Porting and compiling</td>
</tr>
<tr>
<td>GNU Autoconf</td>
<td>Porting, compiler optimization and compiling</td>
</tr>
<tr>
<td>GNU Libtool</td>
<td>Porting and compiling</td>
</tr>
<tr>
<td>Lex &amp; Yacc</td>
<td>Input file parsing</td>
</tr>
<tr>
<td>Doc++*</td>
<td>Code documentation</td>
</tr>
<tr>
<td>\LaTeX2ε &amp; latex2html</td>
<td>Documentation</td>
</tr>
</tbody>
</table>

*The stability of Doc++ is not known, but the underlay format Javadoc is stable.

Table B.1 The tools used for developing FIDO-TNG.
B.2 Automake, Autoconf and Libtool

FIDO 1.x uses GNU Autoconf for the task of compiler tuning, compiler and os dependent parts of the code and selections of compiler time options of the code. The result of the Autoconf program is the well known configure script. Although Autoconf is mainly intended for C/C++, the negative impact of the small tweaks needed for FORTRAN codes are still outweighed by order of magnitude by the functionality of Autoconf. Before the Autoconf versions there were a lot of combinations of pre processor flags to be set for different code behavior and platforms in the makefile. The sequential version of FIDO 1.x compiles on AIX, IRIX, HPUX, Linux, OSF, Solaris and UNICOS. The parallel version is tested on AIX using IBM MPI, Linux using Beowulf MPI, Solaris using MPICH and LAM and UNICOS using CRAY MPI. Autoconf is a framework for writing small tests, e.g. code compilation and running. The results of these tests are often set as a pre processor definition. A port to a specific platform and set of compilers is a combination of answers of the tests. This makes ports to new machines less expensive since it is very likely that the new machine will run with just a new combination of the existing tests. This is particularly true after a few ports when the set of tests has expanded. For this reason FIDO-TNG, almost exclusively developed on Solaris, is tested on Linux regularly to ensure no hard coded platform dependency is made in the code.

Automake and Libtool is further automating the process of building the code. Automake will build a makefile from a simple definition file called Makefile.am. Libtool will in a transparent manner take care of linking, building libraries and interfacing C/C++ and FORTRAN.

B.3 CVS

A multi developer versioning tool is a crucial part of code development involving several developers. The free system Concurrent Versions System CVS, based on RCS, is chosen for the task of FIDO development. The system provides each developer with a own local source tree. The changes are checked in to the main repository. Changes by other developers are merged into the local source tree. One of the advantages with a project based version tool like CVS compare to a file based tool like RCS, is the ability to make tags spanning all files in the project.
B.4 Input file parsing

There is no counterpart to the FORTRAN namelists in C++. However namelists should be treated with caution for a number of reasons. The behaviors of namelists are not defined until in FORTRAN-90. FORTRAN-77 namelists are not uniquely defined. With namelists, the user have to learn cryptic variable names in order to run a code. Further the error checking and flexibility is reduced to a minimum. A more natural way is to edit the input file in plain English, or \LaTeX\ notations, with the possibility to make comments. To achieve this Lex and Yacc\textsuperscript{1} \cite{lex, yacc} or their GNU counterparts Flex \cite{flex} and Bison \cite{bison} are used. Lex provides a token parser based on regular expressions \cite{lex-yacc}. Yacc builds a grammar and actions on those tokens. The result of the programs is C source files performing the input file processing. The combination of Lex and Yacc allows great flexibility of the input file, while still providing strong syntax and contents checking. An example of a FIDO-TNG input file is shown below

\begin{verbatim}
# Example input file
# This is a comment
Equilibrium = stringer
B_0 = 4
# We could also use
# Toroidal Magnetic Field = 4
I_p = 4e+06
# More comments
# Still converging on 1-2 iterations
# mesh accuracy = 0.000001
diagnostics size s = 10
diagnostics size chi = 30
size chi = 51
\end{verbatim}

B.5 Documentation

The documentation is written directly in the code using the Javadoc \cite{javadoc} standard. For every class a brief explanation is written at the top of the class definition. All methods and functions have their purpose and parameters documented within special comments sections at the definition in the source file. All the code is then processed with the utility Doc++ \cite{docplusplus}. Doc++ extends the Javadoc standard with the possibility of writing \LaTeX\ 2\epsilon \cite{latex2html} extensions. The output is either a \LaTeX\ document or HTML. In the HTML version, the formulas are converted to bit maps similar to latex2html \cite{latex2html} output. A compatible alternative to

\textsuperscript{1}Yet Another Compiler Compiler
Doc++ is the Doxygen package [135]. Additional documentation is written in \LaTeX\ and stored in the same source tree.

The following comment

\verbatim
/** \$g_{11}\$. Metric in $$(s, \chi)$$
    * @param point The point in $$(\psi, \chi)$$
    * @return \$g_{11}\$
  */
inline double g_11(const class Pointpsi\chi point){

will produce a method entry in the HTML and \LaTeX\ documentation as shown below.

<table>
<thead>
<tr>
<th>inline double g_11(const class Pointpsi\chi point)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{g_{11}. Metric in (s,\chi)}</td>
</tr>
<tr>
<td><strong>Return Value:</strong> \textit{g_{11}}</td>
</tr>
<tr>
<td><strong>Parameters:</strong> point — The point in (\psi,\chi)</td>
</tr>
</tbody>
</table>
Bibliography


1085–1095 (1998)

Melville, New York, AIP Conference proceedings 485

[52] M. Mantsinen et. al., In J. Jacquinot, G. Van Oost and R.R.
Frequency Heating and Current Drive of Fusion Devices*, pages

[53] M.J. Mantsinen et. al., In S. Bernabei and F. Paoletti, editors, *Radio
Frequency Power In Plasmas 13th Topical Conference*, page 120,
Annapolis, Maryland, (1999). AIP, Melville, New York, AIP
Conference proceedings 485


[56] M. Mantsinen, *Development and experimental evaluation of
theoretical models for ion cyclotron resonance frequency heating
of tokamak plasmas*, Ph.D. thesis, Helsinki University of
Technology, Espoo, (1999)

[57] M.J. Mantsinen et. al., *Plasma Physics and Controlled Fusion*, 42
1–18 (2000)

(2000)

[59] J. Hedin, Technical report, Alfvén Laboratory, Royal Institute of


[61] Paramount pictures, Star Trek; The Next Generation,


Communications*, 97 219 (1996)

[65] M. Wendt, Private communication

[75] B. Saoutic, Private communication
[81] GNU’s Not Unix!, http://www.gnu.org, Free Software Foundation (FSF)
[82] M. Laxåback, Private communication
[85] POSIX.1c standard


[106] ISO/IEC JTC1/SC22/WG5 (Fortran), http://www.nag.co.uk/sc22wg5/
[108] Silicon Graphics Inc.,
    http://www.sgi.com/developers/technology/irix.html
[109] Hewlett-Packard Company,
    http://unix.hp.com/operating/index.html
[110] L. Torvalds et. al., http://www.linux.org/
    as OSF in older releases
[114] D. Becker, P. Merkey et. al., The Beowulf Project,
    http://www.beowulf.org/, hosted by Scyld Computing Corporation
[115] MPICH MPI implementation,
    http://www-unix.mcs.anl.gov/mpi/mpich/
[118] University of Notre Dame, LAM MPI implementation,
    http://www.mpi.nd.edu/lam/
[119] D. MacKenzie et. al., Automake,
    http://www.gnu.org/software/automake/automake.html
[120] G. Matzigkeit et. al., GNU Libtool,
    http://www.gnu.org/software/libtool/libtool.html
[125] R. Corbett et. al., GNU bison,
    http://www.gnu.org/software/bison/
[127] Sun Microsystems Inc., Javadoc tool home page,
[128] R. Wunderling and M. Zöckler, DOC++ — A Documentation


