Lower Hybrid and Electron Cyclotron Current Drive With Bootstrap Current in Tokamaks

by

Steven Donald Schultz

B.S., Applied Mathematics, Engineering and Physics
University of Wisconsin (1990)

Submitted to the Department of Physics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1999

© Massachusetts Institute of Technology 1999. All rights reserved.

Author .....
Department of Physics
July 29, 1999

Certified by.....
Abraham Bers
Professor, Department of Electrical Engineering
Thesis Supervisor

Accepted by .....
Thomas J. Greytak
Professor, Associate Department Head for Education

Science

Lower Hybrid and Electron Cyclotron Current Drive With Bootstrap Current in Tokamaks

by

Steven Donald Schultz

Submitted to the Department of Physics
on July 29, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Steady-state operation of tokamak reactors requires noninductive current generation in the plasma. Radio frequency (RF) current drive in combination with the bootstrap current may achieve this goal. By calculating RF current drive and the bootstrap current in a self-consistent kinetic manner, we find synergistic effects in the total non-inductive current in tokamaks. We include quasilinear diffusion in the Drift Kinetic Equation (DKE) in order to generalize neoclassical theory to highly non-Maxwellian electron distributions that are associated with RF current drive. The calculations are performed with the FASTFP-NC code, which is able to find both RF current drive and the bootstrap current accurately. The code makes use of the Fokker-Planck solver of the FASTFP code [1]. Predictions of the current drive effectiveness are made for lower hybrid and electron cyclotron wave current drive scenarios on high-bootstrap current tokamak machines. A scaling factor is presented which relates the synergistic current density to the total RF-driven current density for lower-hybrid current drive scenarios.

Thesis Supervisor: Abraham Bers
Title: Professor, Department of Electrical Engineering

Acknowledgments

My thanks go out to all of the following:

To my research advisor, Prof. Abraham Bers, for his help and support throughout my time at MIT. He has always been available, enthusiastic and insightful.

To Dr. Abhay Ram, for his friendship and his time and patience in answering questions or listening as I worked out my own answers.

To Dr. Magdi Shoucri and Dr. Issie Shkarofsky, for offering the use of the FASTFP code which was absolutely essential to this work, and also for the time they took to provide me with assistance in understanding the code.

To Prof. Dieter Sigmar, for all our conversations, which contributed greatly to my understanding of the bootstrap current and other aspects of neoclassical and classical transport.

To Laura von Bosau, for help with all the little and not-so-little details.

To all the friends I have made at MIT, without whom I would have missed so much fun and so many fascinating conversations: Fari Firooznia (who has been like a brother to me), John, David, Matt, Mary, Brian and Jen, all the gang from the Thirsty Ear Pub, and so many good friends too numerous to list but never forgotten.

To my Mom and Dad, without whom I never would have made it to MIT in the first place, and who have always been supportive even when they didn't understand what I was doing. To Grandma Ann for her prayers and love, and to my sister Lori, and all my family.

Most especially to Hilary, my best friend, my partner and my angel: you have meant so much to me, and this accomplishment is as much yours as it is mine.

And to God, for everything, and especially for bringing all of the above people into my life.

Contents

1 Introduction **17**

1.1 Motivation for This Work 17

1.2 Relevant Literature 19

1.2.1 Neoclassical Theory of Transport 20

1.2.2 Radio Frequency Current Drive 21

1.2.3 Steady-State Tokamak Scenarios 22

1.3 Overview of This Work 23

2 Analytic Framework for Kinetic Calculations **25**

2.1 Introduction 25

2.2 Electron Motion in a Tokamak 25

2.2.1 Tokamak Magnetic Field Coordinates 26

2.2.2 Guiding Center Motion 28

2.2.3 Trapped and Untrapped Electrons 31

2.3 Kinetic Theory 33

2.3.1 Drift Kinetic Equation 34

2.3.2 The Fokker-Planck Collision Operator 37

2.3.3 The Quasilinear Diffusion Operator 39

2.4 Expansion of the Drift Kinetic Equation 40

2.4.1 Flux-Surface and Bounce Averaging 40

2.4.2 Time Scales and Ordering 42

2.4.3 Small Drift Expansion 46

2.5 Total Inductive Current Drive with RF and Bootstrap 48

3 Introduction to the Bootstrap Current and RF Current Drive 51

3.1 Introduction 51

3.2 The Bootstrap Current 52

3.3 RF Current Drive 58

3.3.1 Lower Hybrid Current Drive 58

3.3.2 Electron Cyclotron Current Drive 68

4 The FASTFP-NC Code 79

4.1 Introduction 79

4.2 The FASTFP Fokker-Planck Code 80

4.3 Numerical Calculation of the Bootstrap Current – The FASTFP-NC Code 84

4.3.1 Radial Derivative 84

4.3.2 Poloidal Dependence of $\tilde{f}$ 85

4.3.3 Particular Solution of g 87

4.4 Benchmarking Calculations of FASTFP and FASTFP-NC 90

4.4.1 Benchmarking RF Current Drive Calculations of FASTFP . . 90

4.4.2 Benchmark Calculations of Bootstrap Current with FASTFP-NC 90

5 Lower Hybrid Current Drive Cases with Neoclassical Transport Effects Including Bootstrap Current 103

5.1 Introduction 103

5.2 FASTFP-NC Calculation of LHCD and Bootstrap Current 104

5.3 Quanititative Predictions of Synergism Between Lower Hybrid and Bootstrap 109

5.4 Further Numerical Results with LHCD and Bootstrap Current 111

5.5 Analysis of LHCD-Bootstrap Results 113

6 Electron Cyclotron Current Drive with Neoclassical Transport Effects Including Bootstrap Current 115

6.1 Introduction 115

6.2 Numerical Results with ECCD and Bootstrap Current 116
6.3 Analysis and Discussion 119

7 Conclusions **123**
7.1 Summary 123
7.2 RF Current and Bootstrap Current – Conclusions 124
7.3 Directions for Future Study 126

A Collision Operator for FASTFP Code **127**

B Approximation of the Integral I for Neoclassical Theory **129**

C Plasma Parameters for FASTFP-NC Calculations **133**
C.1 Alcator C-Mod Type Parameters 133
C.2 ARIES-RS Type Parameters 134
C.3 DIII-D Type Parameters 135

D Codes Used in This Work **137**
D.1 The FASTFP-NC Code 137
D.2 The FASTFP Fokker-Planck Code 210

E Bounce Averaging of Collision Operators **257**
E.1 Distributions Without θ Dependence 259
E.2 Distributions of the Form $\tilde{f}$ 261

F Bounce Averaging of Quasilinear Operators **267**
F.1 General Bounce Average 267
F.2 Lower Hybrid Diffusion 269
F.3 Electron Cyclotron Diffusion 271

G Matrix Form of Collision and Quasilinear Operators for FASTFP and FASTFP-NC Codes **275**

List of Figures

2-1 Toroidal geometry showing the variation of the toroidal coordinates (r, θ, ζ) and the major radius R_0. 27

2-2 Trapped and untrapped guiding center orbits in a tokamak projected onto a plane of constant ζ, showing radial drifts off a flux surface. . . 33

3-1 Functions $\tilde{f}$ (dashed line) and g (solid line) in the nonrelativistic Lorentz gas limit as a function of ξ_0 for a fixed $v = 4v_{Te}$ at $\epsilon = 0.1$ and $\theta = 0$. The dash-dot line is the approximate value of g from (3.14). 56

3-2 (a) Parallel distribution function $F(w)$ from (3.32)for the one-dimensional model of LHCD as a function of $w = v_\parallel / v_{Te}$, for w_1=3.5, w_2=6.0, and D_0=10. (b) Same function showing plateau region only. 63

3-3 Contour plot in $(p_\parallel / p_{Te}, p_\perp / p_{Te})$ space of the LH-modifed electron distribution function. The dashed lines indicate the resonant region of phase space $v_\parallel \in (3.5 v_{Te}, 6.0 v_{Te})$. Parameters are Alcator C-Mod type (Table C.1)with r=0.15 m and D_0=4.0. 66

3-4 Parallel distribution function $F(p_\parallel) = \int 2\pi p_\perp dp_\perp f$ as a function of $p_\parallel / p_{Te}$ for the LH-modified electron distribution function. Parameters are Alcator C-Mod type (Table C.1)with r=0.15 m, D_0=4.0 for $v_\parallel \in (3.5 v_{Te}, 6.0 v_{Te})$. 67

3-5 Resonance condition $\omega - k_\parallel v_\parallel - n\Omega = 0$ in $(p_\parallel / m_e c, p_\perp / m_e c)$ phase space for $N_\parallel = 0.35$ and $n\Omega_0 / \omega = 0.97$. The marks indicate the direction of EC quasilinear diffusion in momentum space. 72

3-6 Contour plot in $(p_\parallel/p_{Te}, p_\perp/p_{Te})$ space of the EC-modifed electron distribution function, for DIII-D type parameters (Table C.3)with r=0.3 m, $2\Omega_0/\omega = 0.96$, $N_\parallel$=0.35, and D_0=0.4. 76

3-7 Same as Figure 3-6, with the dashed lines indicating contours of constant D_{ec}, with the contours representing D_{ec}=1.0,0.5, 0.25, 0.125, and 0.0625 moving outward from the center. 76

3-8 Antisymmetric part $[F(p_\parallel) - F(-p_\parallel)]/2$ of the parallel distribution function $F(p_\parallel) = \int 2\pi p_\perp dp_\perp f$ as a function of $p_\parallel/p_{Te}$ for the EC-modified electron distribution function. Parameters are DIII-D type (Table C.3)with r=0.3 m, $2\Omega_0/\omega = 0.96$, $N_\parallel$=0.35, and D_0=0.4. . . . 77

4-1 Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Z_i=12, T_i=0, and all other parameters of Alcator C-Mod type (Table C.1). Squares are calculations of FASTFP-NC; dashed line is actual value [91]. . . 93

4-2 Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Alcator C-Mod type parameters (Table C.1) with T_i and dT_e/dr set to zero. Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91]. 94

4-3 Transport coefficient L_{31} as a function of $\epsilon^{1/2}$ calculated by FASTFP-NC (squares) and from Reference [91] (dashed line). 95

4-4 Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Alcator C-Mod type parameters (Table C.1) and T_i=0. Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92]. 96

4-5 Transport coefficient L_{32} as a function of $\epsilon^{1/2}$ calculated by FASTFP-NC (squares) and from Reference [91] (dashed line) and Reference [92] (dash-dot line). 97

4-6 Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Alcator C-Mod type parameters (Table C.1). Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92]. 98

4-7 Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for AIRES-RS type parameters (Table C.2). Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92]. 99

4-8 Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for DIII-D type parameters (Table C.3). Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92]. 100

5-1 (a) Functions $\tilde{f}$ (dashed line) and g (solid line) as a function of ξ_0 for constant $p = 4p_{Te}$, for Alcator C-Mod type tokamak parameters (Table C.1) at r of 0.15m. (b) Perturbed parallel distribution function $F_1 \equiv \int 2\pi p_\perp dp_\perp \, (\tilde{f} + g)$ as a function of $p_\parallel$ for the same case. 105

5-2 Parallel distribution function $F = \int 2\pi p_\perp dp_\perp f$ for LHCD as a function of $p_\parallel / p_{Te}$, with RF only (dashed line) and with RF plus bootstrap current (solid line). Parameters are Alcator C-Mod type (Table C.1) with r=0.15 m, D_0=4.0 for $v_\parallel \in (3.5 v_{Te}, 6.0 v_{Te})$. 107

5-3 Perturbed parallel distribution function $F_1 = \int 2\pi p_\perp dp_\perp (\tilde{f} + g)$ for LHCD as a function of $p_\parallel / p_{Te}$, with bootstrap current only (dashed line) and bootstrap with LH (solid line). Parameters are the same as Figure 5-2. 108

6-1 Resonance condition $\omega - k_\parallel v_\parallel - n\Omega = 0$ in $(p_\parallel / p_{Te}, p_\perp / p_{Te})$ phase space for $n\Omega_0/\omega = 0.97$ and different values of $N_\parallel$. 116

6-2 Contour plot in $(p_{\|}/p_{Te}, p_{\perp}/p_{Te})$ space of the EC-modifed electron distribution function (solid contours) and the Maxwellian distribution unmodified by EC (dashed contours). Parameters are DIII-D type (see Table C.3) with r=0.3 m, $2\Omega_0/\omega = 0.96$, $N_{\|}$=0.35, and D_0=0.4. . . . 120

List of Tables

5.1 RUN-001: Output of FASTFP-NC calculations for LHCD with Alcator C-Mod-type parameters (see Table C.1). 106

5.2 Output of FASTFP-NC calculations for LHCD with variations of Alcator C-Mod-type parameters (see Table C.1). 112

5.3 Ratio of the synergistic current density to LH-only current density: Calculated and predicted for FASTFP-NC runs in Table 5.2. 113

6.1 Current density and power dissipated results of FASTFP-NC runs based on DIII-D tokamak parameters (See Table C.3), with $2\omega_{ce}/\omega$=0.96 and D_0=0.4, on the flux surface r=0.3 m. 117

6.2 Current density and power dissipated results of FASTFP-NC runs based on Alcator C-Mod tokamak parameters (See Table C.1), with $2\omega_{ce}/\omega$=0.97 and D_0=0.4, on the flux surface r=0.15 m. 118

6.3 Current density and power dissipated results of FASTFP-NC runs based on ARIES-RS tokamak parameters (See Table C.2), with $2\omega_{ce}/\omega$=0.96 and D_0=0.4, on the flux surface r=1.1 m. 118

6.4 Synergistic current density $J_{\parallel}^{(S)}$ from the results of the FASTFP-NC runs in Tables 6.1, 6.2, and 6.3, as calculated directly and from the electron heating effect expressed in (6.1). 121

C.1 Plasma parameters of FASTFP-NC runs based on Alcator C-Mod in high-bootstrap LHCD mode 134

C.2 Plasma parameters of FASTFP-NC runs based on ARIES-RS design plans. 134

C.3 Plasma parameters of FASTFP-NC runs based on DIII-D tokamak with high pressure and temperature. 135

Chapter 1

Introduction

1.1 Motivation for This Work

In considering the tokamak as a possible fusion reactor, one of its most significant limitations may be the large plasma current needed to generate the poloidal magnetic field. Experimental tokamaks have relied on magnetic induction to produce an electric field in the toroidal direction and drive this current. Such *inductive, ohmic current* is necessarily limited to being pulsed and would be impractical for a fusion reactor which will be relied on to provide continuous electrical power. *Non-inductive current drive* has been studied extensively for this reason. The primary means for driving the current which have been considered are the continuous injection of particle beams or of waves. The injection of particle beams is limited by their ability to penetrate into a high-density reactor plasma; MeV negative-ion beams are still under development to overcome this limitation. We will not consider beam driven currents further in this thesis.

The use of waves (excited from power sources external to the plasma) to drive plasma currents – so-called "RF current drive" or RFCD – has received much more attention and has been successful in driving megaamperes of current in tokamak-confined plasmas. However, the power requirements of RFCD are such that when scaled to provide the total current in a reactor, where they would be part of the recirculating power, they become prohibitively large.

The discovery of the bootstrap current created a much more optimistic view of the goal of a practical steady-state tokamak. The bootstrap current is generated by the gradients of density and temperature which exist when a plasma is confined by a magnetic field, and by its dynamics it is driven in the same direction as the pre-existing plasma current in a tokamak. Most steady-state tokamak proposals under consideration today rely on the bootstrap current to provide a large fraction of the plasma current.

However, the bootstrap current is insufficient to provide the total current of a tokamak. For this reason the need for a combination of bootstrap current and radio frequency current drive has been recognized for some time [2, 3]. Since gradients of density and temperature vanish at the plasma center, a seed current is necessary in the plasma center; in theory this can be a very small current. The more important limitation is that a stable MHD equilibrium cannot be assured in general that can be supported by its own bootstrap current. In addition some source of current is needed in the startup phase of a tokamak plasma to confine the plasma until a bootstrap current can be built up.

Since the bootstrap current cannot create the magnetic field necessary for MHD equilibrium and stability by itself, some other form of current drive is essential. Radio frequency (RF) waves are the most likely means to be used to create the remainder of the current and provide the control of the current profile needed to stabilize the confinement. There is a wide variety of waves which can exist in a plasma, and there are nearly as many different proposed mechanisms for generating a plasma current with RF waves. Since the goal of tokamak study is an economical fusion energy reactor current must be driven at as low a power cost as possible, a necessity which has the driven study of RF current drive effectiveness, current generated per unit RF power supplied externally.

Because both radio frequency waves and the bootstrap current have limits on their effectiveness, it is presently believed that future steady-state tokamaks will make use of a combination of these two current sources. It is only natural to consider the effect which RF waves will have on the neoclassical transport which includes

the bootstrap current. However, the theory of RF current drive and neoclassical theory have each generally neglected the effects of the other. Neoclassical theory assumes that owing to the effect of interparticle collisions, the distribution functions of electrons and ions will only be slightly perturbed from their thermal equilibrium state, a Maxwellian. However, the effect of RF waves for current drive is usually to create a strong distortion to the electron distribution function. In the same way, RF current drive theory has only included the effects of an inhomogeneous magnetic field – the primary requirement of the neoclassical theory – in a simple way. The trapping of electrons in a magnetic field well was long ago recognized as very important, as RF power dissipated on such trapped electrons was treated as wasted, since these electrons will not carry a circulating current. However, there has been little consideration of how the bootstrap current may impact the damping of RF waves and the ability to drive currents with these waves, or how intense RF waves may modify the bootstrap current.

Starting from the first principles which led to RF current drive and neclassical transport theories, we derive a kinetic theory which accounts for both wave-particle interactions and the radial drifts which produce the bootstrap current. We find that this theory produces a total current which can be greater than the sum of the currents predicted by RF current drive and bootstrap current theories separately. The origins of this *synergistic* current will be explored in detail.

1.2 Relevant Literature

A great deal of research has gone into studying the problem of steady-state current drive in plasmas. We discuss the origins of neoclassical transport theory and efforts to most accurately calculate the bootstrap current in tokamaks. We then mention a number of relevant works on the subject of current drive using radio frequency waves. Finally, we discuss the present state of studies on the subject of tokamaks at steady-state.

1.2.1 Neoclassical Theory of Transport

In extending tokamak transport theory to the limit of low collisionality, Galeev and Sagdeev [4] observed that the plasma current parallel to the magnetic field will contain a component proportional to and driven by the perpendicular gradients of density and temperature of electrons and ions. Since such gradients will always be present any time a plasma is confined, this current is intrinsic to all toroidal plasmas, although it is only substantial in high-β plasmas, where β is the ratio of the plasma pressure to the magnetic pressure.

Bickerton, Connor and Taylor [5] gave the bootstrap current its name when they considered the possibility of a "bootstrap" tokamak (as in the phrase "pulling oneself up by one's own bootstraps") relying only on the bootstrap current. Although the bootstrap current was predicted by Galeev and Sagdeev in 1967, the first experimental evidence of its existence in tokamaks would not be found until over a decade later [6].

Rosenbluth, Hazeltine and Hinton [7] were the first to systematically calculate the transport coefficients for neoclassical theory, to lowest order in the inverse aspect ratio of the torus. A more complete transport neoclassical transport theory was later provided by Hinton and Hazeltine [8]. The work of Hirshman and Sigmar, accounting for multispecies ions, culminated in the most thorough work on the subject of neoclassical transport to date [9]. A good recent review of several models for calculating the bootstrap current is given by Kessel [10].

The effects of RF waves on the bootstrap current have been considered by several authors. The idea that RF can change the density and temperature profiles of a plasma and therefore change neoclassical transport was considered by Antonsen and Yoshioka [11], and more recently by Becoulet, et.al. [12]. A number of authors [13, 14, 15] have studied the parallel flow of ions when strong ion heating results in distribution functions strongly distorted from a Maxwellian. A recent work by Helander, Hastie, and Connor [16] has considered the effect of strong electron heating by electron cyclotron waves on the bootstrap current.

1.2.2 Radio Frequency Current Drive

The general theory of waves in plasmas has been the object of extensive research. All of the basics of this subject have been compiled in Stix's *The Theory of Plasma Waves* [17] (and the more recent version *Waves in Plasmas* [18]) and *Waves in Anisotropic Plasmas* by Allis, Buchsbaum, and Bers [19]. These texts are the starting point for any RF theory of waves in plasmas.

The prediction of Landau [20] that there exists a means of kinetic collisionless wave damping showed that the energy of waves could be transferred to charged particles moving in phase with the waves. The effect on the particles is to produce diffusion in velocity space. Kennel and Engelmann [21] produced the quasilinear theory of wave-particle interaction which quantified this diffusion of electrons, as well as diffusion due to cyclotron damping. Further theories expanded upon quasilinear theory: Lerche [22] produced a calculation analagous to Kennel and Engelmann's in the relativistic limit, while Bernstein and Baxter [23], Berk [24], and Kerbel and McCoy [25] generalized this work to inhomogeneous plasmas. An excellent discussion of the bounce averaging of the quasilinear operator is given by Hammett in his thesis [26]. Computer codes used to solve the reduced Boltzmann equation with Fokker-Planck collision terms are an essential part of current drive calculations [27].

Wort [28] is generally credited with being the first to suggest the idea of using RF waves to drive a current in a tokamak, with the suggestion that it would be more efficient to push slower electrons using low phase-velocity waves. Fisch [29] noted that it may be more efficient for waves to transfer momentum to fast moving electrons, since they are less likely to lose their momentum through Coulomb collisions. Thus, he predicted that the Landau damping of lower-hybrid waves could be an efficient means of driving a current, a hypothesis which has been proven by many experiments. The principle behind lower hybrid current drive is probably the simplest mechanism for driving a current: direct transfer of unidirectional momentum from waves to electrons moving along the magnetic field. The launching and coupling of lower hybrid waves into a plasma is a complicated process which has been studied extensively

[30, 31, 32, 33, 34, 35].

The mechanisms for current drive using unidirectional electron cyclotron (EC) waves were proposed by Fisch and Boozer [36], who suggested a way for EC waves to reduce the collisionality of forward moving electrons to increase the plasma conductivity, and by Ohkawa [37], using selective trapping of electrons moving in one direction to drive a current in the other direction.

A number of works have considered "neoclassical" current drive theory, by accounting for trapped electrons [38, 39, 40, 41]. In each of these cases the radial drifts which give rise to the bootstrap current were not included, however.

Fisch [42] compiled an excellent review of the theory of RF current drive which discusses most of the above work in detail.

1.2.3 Steady-State Tokamak Scenarios

Igor E. Tamm and Andrei D. Sakharov first proposed the idea for a system of plasma confinement which would eventually become known as the tokamak [43]. The system uses a strong magnetic field to insulate a high-temperature plasma from the walls of a vacuum chamber. Since the magnetic force only prevents the plasma from moving perpendicular to the field, the field lines would be closed by creating the device in the shape of a torus. However, the magnetic field of this configuration is non-uniform, and if the field is entirely in the direction of the main axis of the torus (the *toroidal* direction), plasma drifts will result in loss of confinement. An additional magnetic field component going around the main axis (in the *poloidal* direction) would bring the plasma to equilibrium, and could be created by a current running through the plasma. The tokamak was shown to be able to attain better plasma confinement than other existing machines by the research of the Soviets in the 1960's, summarized by Lev A. Artsimovich in 1968 [44]. Experimental research with tokamaks has steadily increased both their confinement capabilities and stability. This progress is measured primarily in terms of two quantities which are of importance in sustaining fusion reactions: the product of the plasma density and the energy confinement time ($n\tau_E$) and the temperature [45].

Bickerton, Connor and Taylor [5] and Kadomtsev and Shafranov [46] independently pointed out that the self-generated "bootstrap current" could produce a large fraction of the necessary plasma current in a tokamak, and the idea of a steady-state tokamak with fully non-inductive current drive became much more practical. Bickerton, et.al. were the first to show that it was not possible to have 100% bootstrap current and that a "seed current" at the tokamak's axis was necessary.

The limitations to the bootstrap current fraction are primarily determined by MHD equilibrum and stability considerations. Because the bootstrap current is proportional to the poloidal beta β_p, it is limited by the Troyon limit for the maximum β which is possible in a stable tokamak [47]. The fraction of bootstrap current which can be achieved is believed to be at least 90% in reversed-shear scenarios which have been recently proposed and studied [48].

At the time of this publication there are major design initiatives for steady-state tokamaks under way in the United States and Japan. The ARIES studies [49] are a set of works sponsored by the United States Government to determine feasible designs for a fusion power plant. The tokamak designs which came out of the ARIES work have been used in this work as a good example of the type of tokamak required for steady-state operation. The most recent ARIES-RS [50] is a reversed-shear tokamak which seems to be the most practical design of the ARIES project to date. The Japanese SSTR reactor design [51] also relies on high bootstrap currents with RF current drive.

1.3 Overview of This Work

We calculate the transport parallel to the magnetic field on a single flux surface in a tokamak, including the effect of RF waves and the bootstrap current. This is done by calculating the distribution function of electrons numerically, using a Fokker-Planck and quasilinear code. The code which is used has the capability of treating two different means of RF current drive, by lower-hybrid waves and electron cyclotron waves.

To get a basic idea of the parallel transport, we are able to reduce the complexity of the problem by making some simplications and assumptions. The calculations are performed with a given set of RF fields which are not affected to be self-consistent with the plasma dynamics which are calculated. Effects perpendicular to the flux surface are assumed (density and temperature gradients) and not calculated self-consistently; the effects of RF waves on transport perpendicular to the magnetic field are not included. The calculations of the parallel transport performed here are a first important step toward a more complete theory and computation of RFCD and neoclassical transport outlined in the suggestions for future work in Chapter 7.

A brief outline of this thesis is given here, with more detailed outlines of each chapter given in its respective introduction. In Chapter 2 we introduce the equations which must be solved to analyze the current in a tokamak with both RF wave and neoclassical effects. The basic mechanics of RF current drive and the bootstrap current are only analytically solvable from these equations in the most simplified of cases. Chapter 3 describes these cases, providing the reader with a basic introduction of the three current drive mechanisms considered here: the bootstrap current, lower-hybrid current drive, and electron cyclotron current drive. To guide our analysis of the more complicated problem of the interaction of the effects of RF current drive and the bootstrap current, we realized that it would be helpful to have a numerical calculation of the total current density. To this end we designed the FASTFP-NC code, which solves the problem of finding the distribution function on a flux surface of a tokamak. Chapter 4 describes the development and the features of the FASTFP-NC code and its 2D Fokker-Planck solver FASTFP. We also show benchmarking calculations which show that FASTFP-NC can calculate RF current drive and the bootstrap current separately. In Chapter 5 the case of current drive by lower hybrid waves is analyzed in detail, and used as the basis for providing a simple model for understanding its interaction with the bootstrap current. In Chapter 6, numerical calculations and discussion of electron cyclotron current drive with the bootstrap current are given. Finally, in Chapter 7 we discuss the predictions of this work and suggest new avenues for study in the future.

Chapter 2

Analytic Framework for Kinetic Calculations

2.1 Introduction

The plasma current density can be derived using kinetic theory. In kinetic theory, the macroscopic fluid-dynamic fields of a plasma are derived by taking moments of the distribution functions of electrons and ions. In particular we can calculate the electrical current density in the plasma from these distributions.

In this chapter we derive the equations necessary to find the electron distribution function including the effects of RF wave-electron interactions and radial drifts. In Section 2.2 we describe the magnetic field geometry of a tokamak, and derive coordinates for the six-dimensional phase space of the electrons. Sections 2.3 and 2.4 introduce the equation used to find the electron distribution function, and show how the equation is simplified by averaging over fluctuations which occur on short time scales. The results are summarized in Section 2.5.

2.2 Electron Motion in a Tokamak

We describe the motion of individual electrons in a tokamak. In this calculation both collisions and the small-amplitude RF fields are ignored, and we find the unperturbed

orbits of electrons in a steady-state tokamak-type magnetic field which we denote as $\mathbf{B}_0(\mathbf{x})$

An electron with charge q_e in a homogeneous magnetic field $\mathbf{B}$ acted upon by the Lorentz force $q_e\mathbf{v} \times \mathbf{B}$ will have a periodic circular motion in the plane perpendicular to $\mathbf{B}$, while streaming without change in the velocity along the magnetic field. Since we are considering relativistic electrons the electron's velocity is expressed in terms of its momentum, $\mathbf{v} = \mathbf{p}/\gamma m_e$, where the relativistic factor γ is

$$\gamma = \left(1 + \frac{p^2}{m_e^2 c^2}\right)^{1/2}. \tag{2.1}$$

and the electron rest mass is m_e. The circular gyromotion has a frequency $\omega_{ce} = |q_e| B/\gamma m_e$ and a radius of gyration (gyroradius) $\rho_L = v_\perp/\omega_{ce}$, where $v_\perp = p_\perp/\gamma m_e$ is the velocity perpendicular to the magnetic field. The center of the circle is called the *guiding center*, and in a homogeneous field the guiding center motion is directly along the magnetic field lines.

In the tokamak, where the equilibrium magnetic field $\mathbf{B}_0(\mathbf{x})$ is not homogeneous, we can still approximate the electron motion by a circular orbit around a guiding center provided that the scale length of variations in $\mathbf{B}_0$ is much longer than the gyroradius. The inhomogeneous magnetic field causes the guiding center to drift perpendicular to the magnetic field. The drifts arise from two sources, the gradient of B_0 perpendicular to the field and the curvature of the field lines.

2.2.1 Tokamak Magnetic Field Coordinates

We use the steady-state magnetic field $\mathbf{B}_0$ as a basis for the coordinate system of a tokamak. The spatial geometry of a tokamak with a circular poloidal cross-section, having a large aspect ratio (R_0/a), is shown in Figure 2-1. The *toroidal coordinates* (r, θ, ζ) are used to label this geometry as shown. The distance from the central axis to the plasma can be expressed in terms of the major radius of the torus R_0 and the poloidal angle θ as $R = R_0(1 + \epsilon\cos\theta)$, where $\epsilon = r/R_0$.

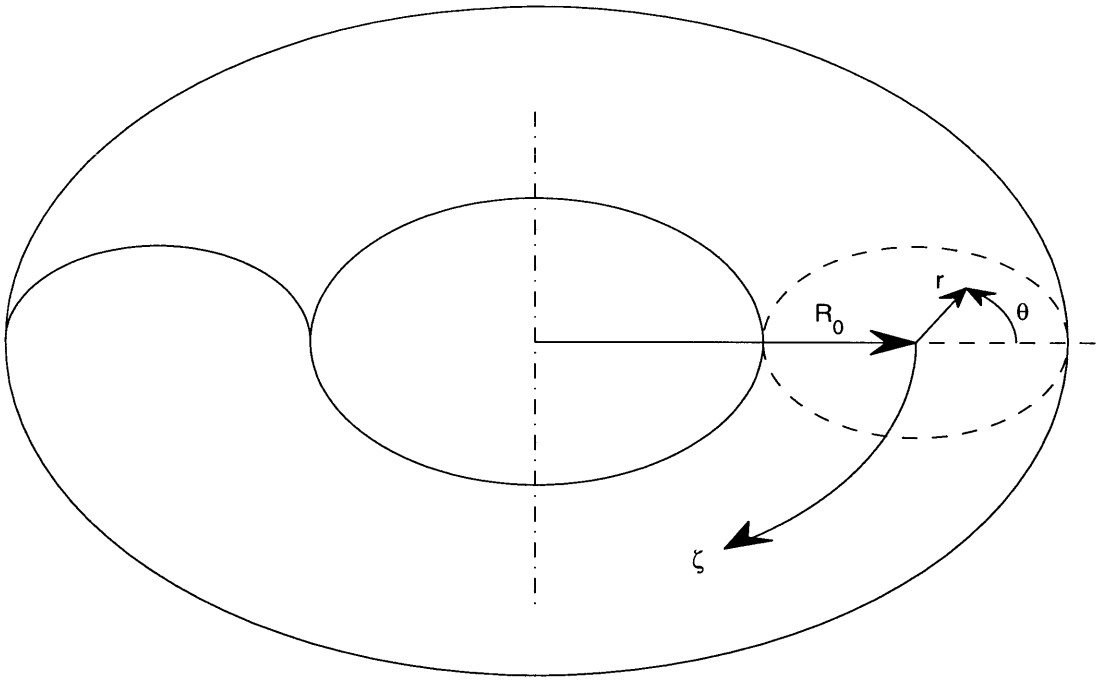

Figure 2-1: Toroidal geometry showing the variation of the toroidal coordinates (r, θ, ζ) and the major radius R_0.

The magnetic field of the tokamak is typically calculated from Ampere's Law,

$$\nabla \times \mathbf{B}_0 = \mu_0 \mathbf{J} \tag{2.2}$$

(where μ_0 is the free-space permeability) and the equations of plasma magnetohydrodynamics (MHD) [52, 53, 54]. For our purposes, however, we will assume a known $\mathbf{B}_0$ consistent with the properties of equilibrium MHD. The plasma is assumed to be symmetric about the main axis of the torus; i.e., all physical properties are independent of the toroidal angle ζ. In addition, we will assume there is no Shafranov shift so that r=0 is the center of the plasma [53]. In this case, $\mathbf{B}_0$ has components in the ζ and θ directions only, and the magnetic field lines wind around the torus at a constant value of r. Because the magnetic field lines are thus contained within surfaces of constant r, these nested tori are often referred to as magnetic surfaces, or alternatively as flux surfaces.

The magnetic field geometry of the system can expressed in terms of *flux functions*,

functions of r only [8]:

$$\mathbf{B}_0 = I\nabla\zeta + \nabla\zeta \times \nabla\psi \tag{2.3}$$

with flux functions $I(r)$ and $\psi(r)$. Because $|\nabla\zeta| = 1/R$, we can therefore express the components of the field as

$$B_{0,\zeta} = \frac{I}{R} \tag{2.4}$$

and

$$B_{0,\theta} = \frac{1}{R}\frac{d\psi}{dr}. \tag{2.5}$$

The function ψ is called the poloidal flux, since it is equal to (or within a factor of 2π of, depending on one's definition) the magnetic flux from $B_{0,\theta}$ enclosed in the surface r=const. It is common to use $\psi(r)$ as a spatial coordinate instead of r. We also note from (2.4) and (2.5) that both the toroidal field and the poloidal field are inversely proportional to R.

2.2.2 Guiding Center Motion

The guiding center motion can be described in terms of constants of the motion. In particular, the energy of the an electron is conserved, since a constant magnetic field cannot change a charged particle's energy. Since the relativistic energy is $E = \gamma m_e c^2$ and γ is a function of the momentum magnitude p only, p is a constant of the motion. The other important constant of the motion is the *magnetic moment*,

$$\mu = \frac{p_\perp^2}{2\gamma m_e B_0}, \tag{2.6}$$

which is an adiabatic invariant resulting from the periodicity of the gyromotion in a weakly varying magnetic field. Since p and γ are constant, the conservation of the magnetic moment can be expressed as

$$\frac{p_\perp^2}{B_0} = const. \tag{2.7}$$

It is advantageous to use a form of pitch-angle coordinates in the kinetic equation.

We define the coordinates (p, ξ, ϕ) as

$$\begin{aligned} p &= \sqrt{p_{\|}^2 + p_{\perp}^2} \\ \xi &= \frac{p_{\|}}{p} \\ \phi &= \cos^{-1} \frac{\mathbf{p}_{\perp} \cdot \hat{x}}{p_{\perp}} \end{aligned} \tag{2.8}$$

The pitch angle coordinate ξ is the cosine of the angle between the $\mathbf{p}$ vector and $\hat{b}$, the unit vector along $\mathbf{B}_0$. The gyroangle ϕ is defined relative to a local unit vector $\hat{x}$ perpendicular to $\hat{b}$. For cases where $B_0(\mathbf{x})$ does not vary on the scale of the gyroradius, the fast gyromotion is

$$\frac{d\phi}{dt} = \Omega = \frac{q_e B_0}{\gamma m_e} \tag{2.9}$$

In the pitch-angle coordinates the conservation of energy and magnetic moment become

$$\begin{aligned} p &= const. \\ \frac{1-\xi^2}{B_0} &= const. \end{aligned}$$

Thus ξ is a function of the electron's position $\mathbf{x}$ through B_0,

$$\xi = \sigma \sqrt{1 - \mu B_0(\mathbf{x}) \frac{2\gamma m_e}{p^2}} \tag{2.10}$$

where σ is the sign of $p_{\|}$.

As a result of the axisymmetry of a tokamak the canonical angular momentum of the electron in the toroidal direction is also a constant of the motion. The canonical toroidal momentum is

$$P_{\zeta} = \left(\gamma m_e \mathbf{v} + q_e \mathbf{A}_0\right)_{\zeta} R \tag{2.11}$$

where $\mathbf{A}_0$ is the magnetic vector potential ($\nabla \times \mathbf{A}_0 = \mathbf{B}_0$). We can express the ζ

component $A_{0,\zeta}$ in terms of the poloidal flux, by comparing two expressions for $B_{0,\theta}$,

$$B_{0,\theta} = (\nabla \times \mathbf{A}_0)_\theta = -\frac{1}{R}\frac{d\left(A_{0,\zeta}R\right)}{dr} \tag{2.12}$$

and

$$B_{0,\theta} = \frac{1}{R}\frac{d\psi}{dr} \tag{2.13}$$

Therefore the poloidal flux ψ can be expressed as $-A_{0,\zeta}R$, and

$$P_\zeta = \gamma m_e v_\zeta R - q_e\psi = const. \tag{2.14}$$

As stated earlier the guiding center motion of the electron is assumed to consist of motion along the magnetic field and drifts due to the gradient and curvature of $\mathbf{B}_0$ [55],

$$\mathbf{v} = v_\parallel \hat{b} + \mathbf{v}_D \tag{2.15}$$

Because P_ζ is a constant of the motion, clearly

$$\mathbf{v} \cdot \nabla P_\zeta = \mathbf{v} \cdot \nabla \left(\gamma m_e v_\zeta R - q_e\psi\right) = 0 \tag{2.16}$$

From this equation we can derive the guiding center drift which the electron must have to conserve toroidal momentum [8, 56]:

$$\mathbf{v}_D \cdot \nabla \left(\gamma m_e v_\zeta R - q_e\psi\right) = -v_\parallel \hat{b} \cdot \nabla \left(\gamma m_e v_\zeta R - q_e\psi\right) \tag{2.17}$$

If we make the *a priori* assumption that $\left|v_\parallel\right| \gg |\mathbf{v}_D|$ we get

$$v_\zeta \approx v_\parallel \hat{b} \cdot \frac{\nabla\zeta}{|\nabla\zeta|} = v_\parallel \frac{I}{RB_0} = v_\parallel \frac{B_{0,\zeta}}{B_0}, \tag{2.18}$$

where we used (2.3) for the magnetic field, and (2.17) with this assumption becomes

$$-q_e \mathbf{v}_D \cdot \nabla\psi \approx -v_\parallel \hat{b} \cdot \nabla \left(\gamma m_e v_\parallel \frac{I}{B_0}\right) \tag{2.19}$$

Axisymmetry shows that

$$\hat{b} \cdot \nabla = \frac{B_{0,\theta}}{B_0} \frac{1}{r} \frac{\partial}{\partial \theta} \tag{2.20}$$

and since

$$\mathbf{v}_D \cdot \nabla \psi = v_{Dr} \frac{d\psi}{dr} = v_{Dr} R B_{0,\theta} \tag{2.21}$$

we get a useful expression for the radial drift:

$$v_{Dr} = \frac{v_\parallel}{r} \frac{I}{R B_0} \frac{\partial}{\partial \theta} \left(\frac{v_\parallel}{\Omega} \right). \tag{2.22}$$

We note from (2.4) that $I/RB_0 = B_{0,\zeta}/B_0$, and in the large aspect ratio limit we take $B_0 \approx B_{0,\zeta}$, so that our expression for the drift is

$$v_{Dr} \approx \frac{v_\parallel}{r} \frac{\partial}{\partial \theta} \left(\frac{v_\parallel}{\Omega} \right). \tag{2.23}$$

We can now verify our assumption of the ordering of the magnitude of the drift. From (2.23),

$$\frac{v_{Dr}}{v_\parallel} = \frac{1}{r} \frac{\partial}{\partial \theta} \left(\frac{v_\parallel}{\Omega} \right) \sim \frac{v_{Te}}{R\Omega} \sim \frac{\rho}{R} \tag{2.24}$$

where we estimate taking electrons with temperature T_e, so that v_{Te} is the electron thermal velocity $\sqrt{T_e/m_e}$ and ρ is the thermal gyroradius v_{Te}/Ω. In all toroidal fusion plasmas of interest the ion gyroradius must be much smaller than the minor radius $a < R_0$. We can estimate this ratio in terms of the plasma properties:

$$\frac{v_{Dr}}{v_\parallel} \sim \frac{\rho}{R} \sim 7.5 \times 10^{-5} \frac{\sqrt{T_{keV}}}{R_m B_T} \tag{2.25}$$

with the electron temperature in keV, the major radius in meters, and the magnetic field in Tesla. Typical tokamaks have $T_{keV} \sim 1 - 10$, $R_m \sim 0.5 - 6$, and $B_T \sim 1 - 6$.

2.2.3 Trapped and Untrapped Electrons

Consider the variation of $\xi \equiv p_\parallel/p$ as a function of poloidal angle as given by (2.10). We can define ξ_0 to be the value of of $p_\parallel/p$ of the electron when it passes the midplane

$\theta = 0$ of the tokamak and also define

$$\Psi(r,\theta) \equiv \frac{B_0(r,\theta)}{B_0(r,0)} = \frac{1+\epsilon}{1+\epsilon\cos\theta} \tag{2.26}$$

(where axisymmetry gives $B_0(\mathbf{x}) = B_0(r,\theta)$) since the magnetic field in a tokamak is inversely proportional to the major radius as shown in Section 2.2.1. Then the magnetic moment (2.6), being a constant of the motion, can be calculated when the electron passes through θ=0 where its value is

$$\mu = \frac{p_\perp^2}{2\gamma m_e B_0(r,\theta)} = \frac{p^2}{2\gamma m_e}\frac{1-\xi_0^2}{B_0(r,0)} \tag{2.27}$$

and (2.10) becomes

$$\xi = \sigma\sqrt{1-\Psi(r,\theta)\,(1-\xi_0^2)}. \tag{2.28}$$

Because the radial drifts are small the guiding center orbit of an electron can be taken as primarily within a flux surface with only a small radial excursion. We can treat r as a fixed parameter in (2.28), so that ξ varies primarily due changes in θ resulting from the electron's motion parallel to the magnetic field.

If there exists some value of θ where $\xi = 0$, then the parallel motion stops there, and the electron will be reflected back in the opposite direction. Such an electron is *trapped* in regions of low magnetic field. From (2.28), we see that the condition for trapping is given by

$$\max\left[\Psi(r,\theta)\right](1-\xi_0^2) > 1. \tag{2.29}$$

Since the maximum value of Ψ is $(1+\epsilon)/(1-\epsilon)$ as seen in (2.26), we find that trapped electrons are those on orbits with

$$\xi_0^2 < \xi_{0T}^2 \equiv \frac{2\epsilon}{1+\epsilon}. \tag{2.30}$$

The trapping angle θ_T is the limiting value of θ, found to be

$$\cos\theta_T = \frac{1}{\epsilon}[(1+\epsilon)(1-\xi_0^2)-1] = 1 - 2\frac{\xi_0^2}{\xi_{0T}^2} \tag{2.31}$$

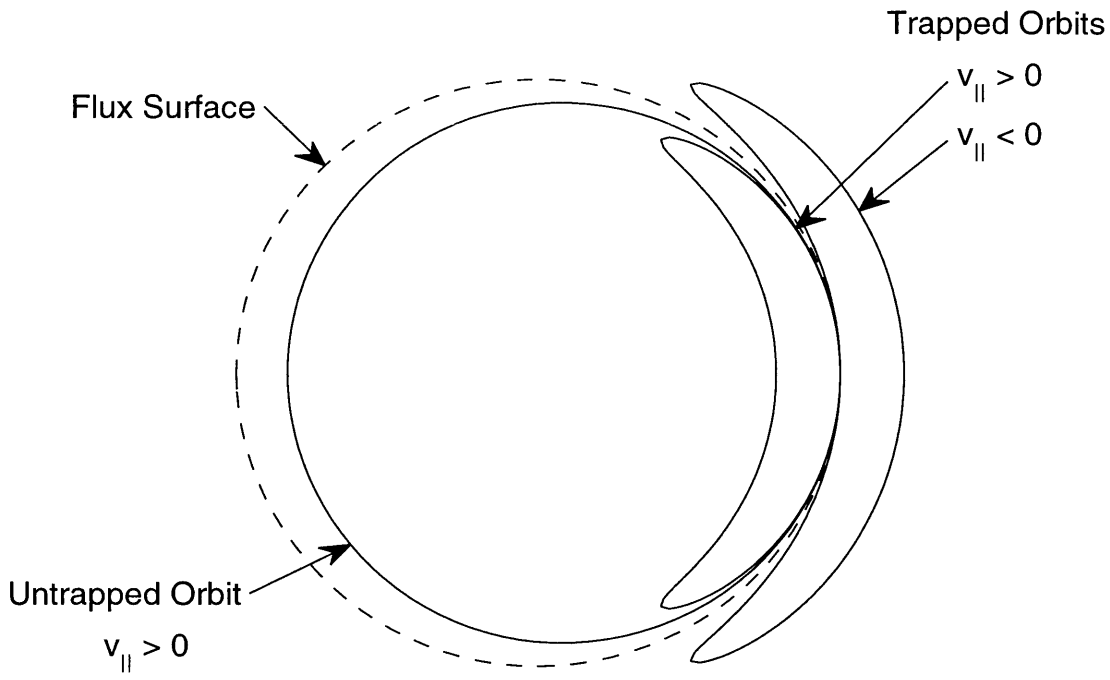

Figure 2-2: Trapped and untrapped guiding center orbits in a tokamak projected onto a plane of constant ζ, showing radial drifts off a flux surface.

Figure 2-2 shows examples of trapped and untrapped (or circulating) orbits projected into the poloidal cross section of the tokamak (a surface with constant ζ). The radial drifts are shown, so that the orbits do move slightly off the flux surface. The orbits of trapped electrons are referred to as *banana orbits* owing to their shape in this projection.

2.3 Kinetic Theory

The (one-particle) distribution function $f_a(\mathbf{x}, \mathbf{p}, t)$ is a probability density of the particles of species a in a six-dimensional continuum phase space described by the three coordinates of spatial position $\mathbf{x}$ and the three components of the particle's momentum vector $\mathbf{p}$. In particular, the current density in a single-ion species plasma is given by the charge-weighted particle flow densities of ions and electrons,

$$\mathbf{J}(\mathbf{x}, t) = q_i \mathbf{\Gamma}_i(\mathbf{x}, t) + q_e \mathbf{\Gamma}_e(\mathbf{x}, t) = q_i \int d^3p \; \mathbf{v} f_i(\mathbf{x}, \mathbf{p}, t) + q_e \int d^3p \; \mathbf{v} f_e(\mathbf{x}, \mathbf{p}, t) \quad (2.32)$$

where q_a is the charge of species a. When driving a current with RF waves it is far easier to induce a flow of electrons than of the heavier ions. For this reason ions are usually left unaffected by high-frequency RF current drive, and one generally needs only to find the function f_e. In the following we assume that the value of the function f_i is known and unperturbed by RF waves.

The electron distribution function $f_e(\mathbf{x}, \mathbf{p}, t)$ satisfies the relativistic *Boltzmann equation*

$$\frac{\partial f_e}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_e + q_e \left[\mathbf{E}(\mathbf{x}, t) + \mathbf{v} \times \mathbf{B}(\mathbf{x}, t)\right] \cdot \nabla_{\mathbf{p}} f_e = \left(\frac{df_e}{dt}\right)_C \tag{2.33}$$

where we now use subscripts to distinguish the gradient in configuation space $\nabla_{\mathbf{x}}$ from the gradient in momentum space $\nabla_{\mathbf{p}}$. The term $\left(\frac{df}{dt}\right)_C$ is the time rate of change of f_e due to interparticle collisions, which will be described in Section 2.3.2. As we will be interested in finding only the electron distribution function, we will omit the subscript e from f_e in the remainder of this work.

2.3.1 Drift Kinetic Equation

We observe that each term in (2.33) has the dimensions of f divided by time, and that the left hand side of the equation is the total convective derivative of f with respect to time. For this reason each term of (2.33) is represented by a characteristic time scale, which we call τ, and the smaller the value of τ, the larger that term is. In this section we average the kinetic equation to remove fluctuations in f which occur on short time scales, leaving only the long term behavior of f which is of interest in a steady-state tokamak. This process is made easier when the phase space of the electrons is described in terms of the constants of the motion and adiabatic invariants derived in Section 2.2.2.

The first such change to the kinetic equation involves averaging over the gyroperiod and the period of the RF waves. In general we will perform these averages together, which is useful for electron cyclotron waves since the scales can be comparable when the wave frequency is in the electron cyclotron range.

In the kinetic equation the fast gyromotion of electrons appears in the term

$$q\left[\mathbf{v}\times\bar{\mathbf{B}}(\mathbf{x})\right]\cdot\nabla_{\mathbf{p}}f=\Omega\frac{\partial f}{\partial\phi} \tag{2.34}$$

The characteristic time of this term is clearly the gyroperiod. Numerically, the electron gyrofrequency is found to be

$$\omega_{ce}\equiv|\Omega|=1.76\times10^{11}\frac{B_T}{\gamma}\ \mathrm{sec}^{-1} \tag{2.35}$$

where B_T is the magnetic field in Tesla and we keep the relativistic factor γ. As stated earlier, a typical tokamak has a magnetic field of 1 to 6 Tesla.

The process of gyroaveraging,

$$\langle\mathcal{A}\rangle_{\phi}=\oint\frac{d\phi}{2\pi}\mathcal{A}, \tag{2.36}$$

eliminates the term (2.34) from the kinetic equation and removes the variable ϕ from the equations. Simultaneously, we also want to average over the time and length scales defined by the period and wavelength of the RF waves which provide the fields $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{B}}$. The RF fields are assumed to fluctuate in a periodic manner in space and time, and so averaging in time over a wave period, denoted by $\langle...\rangle_{\omega}$, gives

$$\left\langle\tilde{\mathbf{E}}\right\rangle_{\omega}=\left\langle\tilde{\mathbf{B}}\right\rangle_{\omega}=0 \tag{2.37}$$

We also assume that the RF fields are weak,

$$\frac{1}{v_{ph}}\left|\tilde{\mathbf{E}}\right|\sim\left|\tilde{\mathbf{B}}\right|\ll\left|\mathbf{B_0}\right|, \tag{2.38}$$

where v_{ph} is the magnitude of the wave's phase velocity. Maxwell's equations tell us that just as charged particles are affected by electromagnetic fields, the fields are also affected by the particles. For this reason, a true and accurate picture of plasma physics would require us to find the plasma distribution function and the electromagnetic fields self-consistently. Such an analysis is beyond the scope of this

work, and as a starting point $\mathbf{B_0}(\mathbf{x})$, $\tilde{\mathbf{E}}(\mathbf{x},t)$ and $\tilde{\mathbf{B}}(\mathbf{x},t)$ are assumed to be given and unchanged by the plasma dynamics which are subsequently derived.

We use the notation $\bar{f} \equiv \langle f \rangle_{\phi,\omega}$ for the combined wave and gyroaverage, and $\tilde{f} \equiv f - \bar{f}$. Applying the averaging process to (2.33) gives two equations,

$$\frac{\partial \bar{f}}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} \bar{f} = \left(\frac{d\bar{f}}{dt} \right)_C - \nabla_{\mathbf{p}} \cdot \mathbf{\Gamma}_{ql} \tag{2.39}$$

and

$$\frac{\partial \tilde{f}}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} \tilde{f} + q \left(\frac{\mathbf{p}}{\gamma m_e} \times \mathbf{B_0} \right) \cdot \nabla_{\mathbf{p}} \tilde{f} = -q \left(\tilde{\mathbf{E}} + \frac{\mathbf{p}}{\gamma m_e} \times \tilde{\mathbf{B}} \right) \cdot \nabla_{\mathbf{p}} \bar{f} \tag{2.40}$$

The derivative with respect to time in (2.39) refers implicitly to variations on a time scale longer than the wave period. The quasilinear flux, the time-averaged current density in momentum space due to RF waves, is defined as the second-order term

$$\mathbf{\Gamma}_{ql} \equiv \left\langle q \left(\tilde{\mathbf{E}} + \frac{\mathbf{p}}{\gamma m_e} \times \tilde{\mathbf{B}} \right) \tilde{f} \right\rangle_{\phi,\omega}. \tag{2.41}$$

The calculation of $\mathbf{\Gamma}_{ql}$ from a prescribed set of electric and magnetic fields has been performed by Kennel and Engelmann [21] for an infinite, homogeneous plasma using a spatial integration over a wavelength as $\langle ... \rangle_{\omega}$.

The averaged equation (2.39) is known as the *drift kinetic equation* (DKE). Using the guiding center velocity from (2.15) in Section 2.2.2 we get

$$\frac{\partial \bar{f}}{\partial t} + v_{\parallel} \frac{B_{0,\theta}}{B_0} \frac{1}{r} \frac{\partial \bar{f}}{\partial \theta} + \mathbf{v}_D \cdot \nabla_{\mathbf{x}} \bar{f} = \left(\frac{d\bar{f}}{dt} \right)_C - \nabla_{\mathbf{p}} \cdot \mathbf{\Gamma}_{QL} \tag{2.42}$$

where the averaged distribution function $\bar{f}$ is a function of (r, θ, p, ξ, t). Since we are considering steady-state behavior of the plasma we can neglect the t dependence and the first term of (2.42).

To simplify our notation in the remainder of this work, we drop the bar from $\bar{f}$. We will also drop the subscript zero from $\mathbf{B}_0$ and its components, since the first order fields have been averaged out and we no longer need to distinguish them.

2.3.2 The Fokker-Planck Collision Operator

Because interparticle collisions are discrete, random events, the rate of change of f due to collisions is treated approximately, using a collision operator, which we refer to as C. The collision operator uses statistical averaging to describe $(df/dt)_c$.

This operator treats binary collisions (which dominate over three or more body collsions) of electrons with other electrons and with the plasma ions. The operator can then be taken to be *bilinear* in the distribution functions,

$$\left(\frac{df}{dt}\right)_C \equiv \sum_b C_{eb}(f, f_b) = C_{ee}(f, f) + C_{ei}(f, f_i) \tag{2.43}$$

As stated earlier, we are only interested in deriving the electron distribution function, so we take the ion distribution f_i to be a known Maxwellian. In addition, we linearize the electron-electron collision operator by assuming that the electrons are not strongly heated, so that the bulk electron distribution function remains nearly equal to its thermal, Maxwellian value f_M. The linear operator is therefore

$$C(f) \equiv C_{ee}(f, f_M) + C_{ee}(f_M, f) + C_{ei}(f, f_i) \tag{2.44}$$

The form of the *Fokker-Planck* collision operator is derived by considering the effect of a large number of small-angle deflections caused by Coulomb interactions of the charged electrons and ions. Large-angle collisions are not considered due to the small probability with which they occur. Assuming a Markov-type process, a statistical averaging over these small-angle collisions results in an operator of the form [57, 58]:

$$\begin{aligned} C_{ab}(f_a, f_b) &= -\nabla_{\mathbf{p}} \cdot \boldsymbol{\Gamma}_c \\ \boldsymbol{\Gamma}_c &= \left\langle \frac{\Delta \mathbf{p}}{\Delta t} \right\rangle_c f_a + \frac{1}{2} \nabla_{\mathbf{p}} \cdot \left(\left\langle \frac{\Delta \mathbf{p} \Delta \mathbf{p}}{\Delta t} \right\rangle_c f_a \right), \end{aligned} \tag{2.45}$$

where $\langle\ \rangle_c$ is used here as a shorthand for the statistical averaging using the Markov probability function for the scattering of a test particle by $\Delta \mathbf{p}$, due to a background

of particles of species b with distribution f_b. The first term in (2.45) is observed to have the form of a friction force, and the second term gives diffusion in momentum space.

The linearized operators for electron scattering off of electron and ion Maxwellian backgrounds can be expressed in (p, ξ) coordinates [59, 1]:

$$\begin{aligned} C_{ee}(f, f_M) + C_{ei}(f, f_i) &= -\nu_e \frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\left(A(p)\frac{\partial f}{\partial p} + F(p) f\right)\right] \\ &\quad -\nu_e \frac{B_t(p)}{p^2}\frac{\partial}{\partial \xi}\left[(1-\xi^2)\frac{\partial f}{\partial \xi}\right] \end{aligned} \tag{2.46}$$

Here, we explicitly normalize the collision operator to show its dependence on the thermal collision frequency [60]

$$\nu_e = \frac{e^4 n_e \ln \Lambda}{4\pi \epsilon_0^2 m_e^2 v_{Te}^3} \tag{2.47}$$

The potentials $A(p)$, $B_t(p)$, and $F(p)$ are calculated from the electron Maxwellian distribution f_M and the ion distribution f_i from the complete operator of Braams and Karney [61], who extended the Rosenbluth potential form [62] of the collision operator to apply to the relativistic collision operator of Beliaev and Budker [63].

Also necessary to the linearized Fokker-Planck operator is a term $C_{ee}(f_M, f)$ which gives the collisional scattering of the bulk Maxwellian electrons off of the distribution f. This integral operator on f is necessary to preserve the conservation properties of the full collision operator. For current drive calculations, where the first-order moment of the distribution function is to be found, it is sufficient for the operator to conserve momentum in the parallel direction. For this reason the integral is simplified by only taking the integral of the component of f proportional to the $l = 1$ Legendre polynomial of ξ,

$$C_{ee}(f_M, f) \approx \nu_e I_1(f_M, f_{l=1}\xi) = I_1(f_M, f_{l=1})\xi \tag{2.48}$$

$$f_{l=1} \equiv \frac{3}{2}\int_{-1}^{1} \xi f \, d\xi \tag{2.49}$$

Further details of the construction of this collision operator are given in Appendix A.

2.3.3 The Quasilinear Diffusion Operator

The quasilinear diffusion operator for the resonant interaction between electrons and RF waves was first derived for an infinite, homogeneous plasma by Kennel and Engelmann [21]. Lerche [22] performed the analagous calculation for a relativisitic plasma. From this calculation, the *quasilinear operator* $Q(f) = -\nabla_{\mathbf{p}} \cdot \mathbf{\Gamma}_{ql}$ is found to be in (p, ξ) coordinates:

$$Q(f) = \lim_{V\to\infty} \frac{1}{V} \int \frac{d^3k}{(2\pi)^3} \sum_{n=-\infty}^{\infty} \mathcal{L}\left[p^2(1-\xi^2) D_{\mathbf{k}} \delta(\omega - k_{\|} v_{\|} - n\Omega) \mathcal{L}(f)\right] \tag{2.50}$$

$$\begin{aligned} \mathcal{L}(\mathcal{A}) &= \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(\frac{k_{\|} p}{\gamma m_e \omega} - \xi\right) \frac{1}{p^2} \frac{\partial \mathcal{A}}{\partial \xi} \\ &= \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\mathcal{A}) + \frac{\partial}{\partial \xi}\left[\left(\frac{k_{\|} p}{\gamma m_e \omega} - \xi\right)\mathcal{A}\right]\right) \end{aligned} \tag{2.51}$$

$$D_{\mathbf{k}} = q_e^2 \left| E_{\mathbf{k},+} e^{-i\alpha} J_{n-1} + E_{\mathbf{k},-} e^{+i\alpha} J_{n+1} + \frac{p_{\|}}{p_{\perp}} E_{\mathbf{k},\|} J_n \right|^2 \tag{2.52}$$

$$E_{\mathbf{k},\pm} = \frac{1}{2}(E_x \pm i E_y)_{\mathbf{k}} \tag{2.53}$$

$$\begin{aligned} k_x &= k_{\perp} \cos\alpha \\ k_y &= k_{\perp} \sin\alpha \end{aligned}$$

where we have taken the Fourier decomposition of the RF electric field to be

$$\tilde{\mathbf{E}}(\mathbf{x}, t) = \mathrm{Re}\left[\int \frac{d^3k}{(2\pi)^3} \; e^{i\mathbf{k}\cdot\mathbf{x} - i\omega t} \tilde{\mathbf{E}}_{\mathbf{k}}\right] \tag{2.54}$$

and the argument of the Bessel functions J_n is $(k_{\perp} v_{\perp}/\Omega)$. We note that the quasilinear operator is a second-order differential operator in $\mathbf{p}$. Electrons which are able to interact most strongly with the RF waves move in resonance with the wave phase; this is expressed by the resonance condtion $\omega - k_{\|} v_{\|} - n\Omega = 0$.

2.4 Expansion of the Drift Kinetic Equation

Taking the drift kinetic equation (2.42) in steady state form, using v_{Dr} from (2.23) and neglecting $v_{D\theta} \ll v_{\|}(B_\theta/B)$, we get

$$\frac{v_{\|}}{r}\frac{B_\theta}{B}\frac{\partial f}{\partial \theta} + \frac{v_{\|}}{r}\frac{\partial}{\partial \theta}\left(\frac{v_{\|}}{\Omega}\right)\frac{\partial f}{\partial r} = C(f) + Q(f) \tag{2.55}$$

The solution f of the tokamak drift kinetic equation is a function of four variables, the spatial coordinates r and θ and the momentum-space coordinates p and ξ. Finding the solution of a partial differential equation in four variables is beyond our capabilities, and some way of simplifying the problem is necessary.

The expansion in small parameters described here is the basis for neoclassical transport theory [64, 8, 9]; however, here we generalize to cases where the distributions are non-Maxwellian as a result of RF waves, keeping the quasilinear term alongside the collision operator to all orders in the expansions which follow.

2.4.1 Flux-Surface and Bounce Averaging

One technique which allows us to simplify the DKE is averaging over the poloidal angle θ. In a later section we will discuss approximations to justify this averaging, but here we simply introduce the integration techniques to average over θ and their notation.

The *flux-surface average* is defined as the integral over the infinitesimal volume between the flux surface at radius r and a neighboring surface at $r + \Delta r$. The volume element in toroidal coordinates is $dV = dr\ rd\theta\ Rd\zeta$, so that the flux surface average becomes

$$\langle \mathcal{A} \rangle = \frac{\int dr\ rd\theta\ Rd\zeta \mathcal{A}}{\int dr\ rd\theta\ Rd\zeta} \approx \frac{r\ \Delta r \int_{-\pi}^{\pi} d\theta\ R\mathcal{A}}{r\ \Delta r \int_{-\pi}^{\pi} d\theta\ R} = \frac{\int_{-\pi}^{\pi} d\theta\ R\mathcal{A}}{\int_{-\pi}^{\pi} d\theta\ R} \tag{2.56}$$

Because the magnetic field is inversely proportional to the major radius, this can be expressed in the more general form

$$\langle \mathcal{A} \rangle = \frac{\int_{-\pi}^{\pi} d\theta\ \mathcal{A}/B}{\int_{-\pi}^{\pi} d\theta\ 1/B} \tag{2.57}$$

The *bounce average* is a variation of the flux-surface average in which the integration is performed over a periodic electron orbit. For circulating orbits, it is simply defined as

$$\{\mathcal{A}\} \equiv \frac{1}{\tau_t} \int_{-\pi}^{\pi} d\theta \frac{r}{|v_\parallel|} \frac{B}{B_\theta} \mathcal{A} \tag{2.58}$$

The average is normalized by the transit time,

$$\tau_t \equiv \int_{-\pi}^{\pi} d\theta \frac{r}{|v_\parallel|} \frac{B}{B_\theta} \tag{2.59}$$

We note immediately that the bounce average is the annihilator of the poloidal derivative in (2.55),

$$\left\{ \frac{v_\parallel}{r} \frac{B_\theta}{B} \frac{\partial f}{\partial \theta} \right\} = \frac{1}{\tau_t} \sigma \int_{-\pi}^{\pi} d\theta \frac{\partial f}{\partial \theta} = \frac{1}{\tau_t} \sigma \left[f \right]_{\theta=-\pi}^{\theta=\pi} = 0 \tag{2.60}$$

owing to the 2π periodicity in θ. Once again σ is the sign of $v_\parallel$.

For trapped orbits the bounce average is a bit more complicated,

$$\{\mathcal{A}\} \equiv \frac{1}{\tau_b} \frac{1}{2} \sum_{\sigma=\pm 1} \sigma \int_{-\theta_T}^{\theta_T} d\theta \frac{r}{v_\parallel} \frac{B}{B_\theta} \mathcal{A} = \frac{1}{\tau_b} \frac{1}{2} \sum_{\sigma=\pm 1} \int_{-\theta_T}^{\theta_T} d\theta \frac{r}{|v_\parallel|} \frac{B}{B_\theta} \mathcal{A} \tag{2.61}$$

where the bounce time is now defined as

$$\tau_b \equiv \int_{-\theta_T}^{\theta_T} d\theta \frac{r}{|v_\parallel|} \frac{B}{B_\theta} \tag{2.62}$$

Note that this is only half of the time required for the trapped electron to complete a full orbit, but we define τ_b this way to make it consistent with the definition of τ_t in (2.59). It can again be shown that this annihilates the poloidal derivative term of (2.55),

$$\begin{aligned} \left\{ \frac{v_\parallel}{r} \frac{B_\theta}{B} \frac{\partial f}{\partial \theta} \right\} &= \frac{1}{\tau_b} \sum_\sigma \sigma \int_{-\theta_T}^{\theta_T} d\theta \frac{\partial f}{\partial \theta} \\ &= \frac{1}{\tau_b} \sum_\sigma \sigma \left[f \right]_{\theta=-\theta_T}^{\theta=\theta_T} \\ &= \frac{1}{\tau_b} \left[f(\sigma=+1) - f(\sigma=-1) \right]_{\theta=-\theta_T}^{\theta=\theta_T} = 0 \end{aligned} \tag{2.63}$$

In the end, we make use of the fact that $v_{\|} = 0$ at the turning points, and so f is independent of the sign σ.

We can also rewrite the bounce averaging integrals (2.58) and (2.61) in terms of ξ and ξ_0,

$$\{\mathcal{A}\} = \frac{1}{\lambda}\left[\frac{1}{2}\sum_{\sigma}\right]_T \int_{-\theta_c}^{\theta_c} \frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\mathcal{A} \tag{2.64}$$

We generalize the bounce average to include trapped and circulating electrons by defining the critical angle

$$\begin{aligned} \theta_c &= \pi \text{ for circulating electrons} \\ &= \theta_T \text{ for trapped electrons,} \end{aligned}$$

and by including the sum in brackets $[\ ]_T$ only for trapped electrons. Here, λ is a form of the transit or bounce time (2.59) and (2.62) normalized to $2\pi r B/v\xi_0 B_\theta$,

$$\lambda \equiv \int_{-\theta_c}^{\theta_c} \frac{d\theta}{2\pi}\frac{\xi_0}{\xi} = \tau_{t,b}\cdot\left(\frac{v\xi_0}{2\pi r}\frac{B}{B_\theta}\right). \tag{2.65}$$

From the integral form in (2.65), we can see that λ is a function of ξ_0 as well as an implicit function of the flux surface r, but is not dependent on p.

2.4.2 Time Scales and Ordering

We discuss the characteristic time scales for each term in (2.55) and show how they are ordered in a typical tokamak.

Transit and Bounce Times

The characteristic time of the θ-derivative term of the DKE (2.55) is the electron transit or bounce time, defined in Section 2.4.1. We can find a typical transit frequency ω_t by taking $v_{\|} \sim v_{Te}$ and the distance travelled by the electron moving along

the magnetic field to be of the order of the major radius R:

$$\omega_t \sim \frac{v_{Te}}{R} = 1.3 \times 10^7 \frac{\sqrt{T_{keV}}}{R_m} \text{ sec}^{-1} \tag{2.66}$$

with the electron temperature in keV, and the major radius in meters. Since trapped electrons have

$$\frac{v_{\|,0}}{v} = \xi_0 < \xi_{0T} = \sqrt{\frac{2\epsilon}{1+\epsilon}} \sim \sqrt{\epsilon} \tag{2.67}$$

for $\epsilon \ll 1$, the bounce frequency of trapped electrons on average will be lower than the typical transit time,

$$\omega_b \sim \frac{v_{\|}}{R} \sim \frac{\sqrt{\epsilon} v_{Te}}{R} \sim \sqrt{\epsilon}\, \omega_t. \tag{2.68}$$

Collision and Quasilinear Times

The Fokker-Planck theory considers the cumulative effect of many small-angle collisions in calculating the rate of change of a particle distribution. From this theory, we estimate the electron-electron collision time [60] as a characteristic time for Coulomb collisions to deflect an electron's path by a significant angle (on the order of 90^o):

$$\tau_{ee} = \frac{4\pi\epsilon_0 m^2 v_{Te}^3}{q^4 n_b \ln\Lambda} \tag{2.69}$$

The Coulomb logarithm $\ln\Lambda$ is a slowly varying function of density and temperature; we take its value to be approximately 15. The Fokker-Planck collision term of the kinetic equation scales like the collision frequency times f and can be estimated:

$$\nu_e = \frac{1}{\tau_{ee}} \sim 5.3 \times 10^5 \frac{n_{20}}{{T_{keV}}^{3/2}} \text{ sec}^{-1} \tag{2.70}$$

Here, the density n_{20} is in units of 10^{20}m^{-3}, which is 0.1 to 1 for large-scale tokamak experiments, although it may be up to 3 or 4 in reactor scale plasmas.

We can then consider ν_e/ω_t to be a collisionality parameter, and from (2.66) and

(2.70) we get the approximation

$$\frac{\nu_e}{\omega_t} \sim 0.04 \frac{n_{20} R_m}{{T_{keV}}^2}. \tag{2.71}$$

We recall that T_{keV} and R_m are on the order of 1, so that $\nu_e/\omega_t \sim 0.01$.

For trapped electrons we can consider an alternate collisional timescale determined not by the time for deflection of the path by 90^o but by the time needed for the electron to be deflected so that it is no longer on a trapped orbit. We can approximate the change to the pitch angle ξ_0 necessary to make trapped particles become untrapped [4]:

$$\Delta\xi_0 \sim \xi_{0T} = \sqrt{\frac{2\epsilon}{1+\epsilon}} \sim \sqrt{\epsilon} \tag{2.72}$$

for $\epsilon \ll 1$. Because the small-angle collisions produce a random-walk change in the pitch angle ξ_0, the *effective* collision frequency for detrapping is approximately

$$\nu_{eff} \sim \nu_e(90^o) \frac{1}{(\Delta\xi_0)^2} \sim \frac{\nu_e}{\epsilon}. \tag{2.73}$$

The collisional time scale, which is the time required for a non-Maxwellian distribution of particles to relax to thermal equilibrium, is much shorter than the *transport* time scale, the time scale on which the significant macroscopic properties of the plasma change. For this reason, existing theories of plasma transport [60, 8, 9] are formulated upon the assumption that the distribution functions of electrons and ions are Maxwellian, or only weakly perturbed from a Maxwellian. However, RF waves used for current drive act on the collisional time scale of the plasma, and therefore are able to prevent the electron distribution function from relaxing to a Maxwellian, instead giving f a value which has a first-order moment, producing a current. If the RF waves are used continuously, this effect, the non-Maxwellian state of the electron distribution, will persist even on the time scale of transport. Therefore, the quasilinear time scale is assumed to be of the same order as the collision time scale, and the effect of the RF waves can be treated as instantaneous on the transport time scale.

Guiding Center Drifts

We define a drift time scale by estimating the time it would take for radial drifts to move an electron across the magnetic field by a distance on the order of the minor radius a, $\tau_D \sim a/v_{Dr}$.

We define δ to be a small paramater indicating the drift ordering. As shown in Section 2.2.2 the radial drift speed is smaller than the parallel velocity by a factor typically on the order of ρ/R. Since the transit or bounce time was shown to be on the order of $R/\left|v_{\|}\right|$ we get from (2.25)

$$\delta \equiv \frac{\tau_{t,b}}{\tau_D} \sim \frac{R/\left|v_{\|}\right|}{a/v_{Dr}} \sim \frac{\rho}{a} \sim 7.5 \times 10^{-5} \frac{\sqrt{T_{keV}}}{a_m B_T} \tag{2.74}$$

for electron temperature in keV, a in meters, and B in Tesla. Since these quanities are on the order of 1 in the chosen units, (2.74) shows that $\delta \sim 1 \times 10^{-4} \ll 1$ for a typical tokamak.

Banana Regime Ordering

From our estimation of the collisionality parameter ν_e/ω_t and the drift parameter δ we see that typically

$$\delta \ll \frac{\nu_e}{\omega_t} \ll 1. \tag{2.75}$$

The banana regime ordering is a stronger form of the low-collisionality limit $\nu_e/\omega_t \ll 1$. In this case all the electrons, circulating and trapped, are assumed to be able to complete their orbits (closed in the poloidal angle) in a time too short for collisions to deflect them from their orbit. (This neglects the small number of orbits which take a very long time to complete: those on the trapped-passing boundary $\xi_0 \approx \xi_{0T}$, and those for very slow-moving electrons with $v \ll v_{Te}$.)

So in addition to having $\nu_e/\omega_t \ll 1$ for circulating particles, we also require $\nu_{eff}/\omega_b \ll 1$ for trapped particles. We define a new collisionality parameter $\nu_* \equiv \nu_{eff}/\omega_b$, and (2.68) and (2.73) show that $\nu_e/\omega_t \sim \epsilon^{3/2}\nu_* \ll \nu_*$. Therefore, the banana

regime is defined by the inequality

$$\delta \ll \epsilon^{3/2}\nu_* \ll \nu_* \ll 1. \tag{2.76}$$

2.4.3 Small Drift Expansion

We expand the electron distribution function in δ,

$$f = f_0 + \delta f_1 + \cdots. \tag{2.77}$$

Since ν_* is much greater than δ, we treat the collision and quasilinear terms as being of order δ^0. From the expanded DKE (2.55) one obtains up to order δ

$$\frac{v_\parallel}{r}\frac{B_\theta}{B}\frac{\partial f_0}{\partial \theta} - C(f_0) - Q(f_0) = 0 \tag{2.78}$$

$$\frac{v_\parallel}{r}\frac{B_\theta}{B}\frac{\partial \delta f_1}{\partial \theta} - C(\delta f_1) - Q(\delta f_1) = -\frac{v_\parallel}{r}\frac{\partial}{\partial \theta}\left(\frac{v_\parallel}{\Omega}\right)\frac{\partial f_0}{\partial r} \tag{2.79}$$

Zero Order Solution

The lowest order form of the DKE, (2.78), can be taken at constant r, but is still a partial differential equation in three variables, θ, p, and ξ. This can be reduced using the banana regime ordering (2.76) and taking the quasilinear time scale,

$$\tau_{QL} \sim \frac{p_{Te}^2}{D} \tag{2.80}$$

for diffusion coefficient D, to be on the order of the collision time. Then since the collisional and quasilinear frequencies are much less than the the transit and bounce frequencies, (2.78) becomes

$$\frac{v_\parallel}{r}\frac{B_\theta}{B}\frac{\partial f_0}{\partial \theta} \approx 0 \tag{2.81}$$

Thus the lowest order distribution f_0 is independent of the poloidal angle θ, which allows us to remove θ completely from this equation. We perform the bounce average,

defined in Section 2.4.1, on (2.78), and find

$$\{C(f_0)\} + \{Q(f_0)\} = 0 \tag{2.82}$$

Therefore f_0 is derivable by solving (2.82), a partial differential equation in two dimensions, with the proper boundary conditions.

First Order Solution

The banana regime ordering (2.76) again helps to reduce the first order part of the DKE (2.79)

$$\frac{v_\parallel}{r}\frac{B_\theta}{B}\frac{\partial \delta f_1}{\partial \theta} - C(\delta f_1) - Q(\delta f_1) = -\frac{v_\parallel}{r}\frac{\partial}{\partial \theta}\left(\frac{v_\parallel}{\Omega}\right)\frac{\partial f_0}{\partial r} \tag{2.83}$$

which becomes

$$\frac{v_\parallel}{r}\frac{B_\theta}{B}\frac{\partial \delta f_1}{\partial \theta} \approx -\frac{v_\parallel}{r}\frac{\partial}{\partial \theta}\left(\frac{v_\parallel}{\Omega}\right)\frac{\partial f_0}{\partial r} \tag{2.84}$$

This equation is solvable by integrating over θ, keeping r fixed, since f_0 is independent of θ,

$$\delta f_1 = -\frac{B}{B_\theta}\left(\frac{v_\parallel}{\Omega}\right)\frac{\partial f_0}{\partial r} + g = -\frac{v_\parallel}{\Omega_\theta}\frac{\partial f_0}{\partial r} + g \tag{2.85}$$

where $\Omega_\theta = qB_\theta/m$ is the poloidal gyrofrequency. The function g is the "integration constant", and therefore is independent of θ. We use the notation

$$\tilde{f} \equiv -\frac{v_\parallel}{\Omega_\theta}\frac{\partial f_0}{\partial r}. \tag{2.86}$$

and thus $\delta f_1 = \tilde{f} + g$.

To find the θ-independent function g, we can again use the bounce average to reduce (2.83):

$$\{C(\delta f_1)\} + \{Q(\delta f_1)\} = \left\{C(\tilde{f} + g)\right\} + \left\{Q(\tilde{f} + g)\right\} = 0. \tag{2.87}$$

Because the collision and quasilinear operators are linear, we can rewrite this as

$$\{C(g)\} + \{Q(g)\} = -\left\{C(\tilde{f})\right\} - \left\{Q(\tilde{f})\right\} \tag{2.88}$$

2.5 Total Inductive Current Drive with RF and Bootstrap

The previous section shows that the problem of calculating the electron distribution in a tokamak, including the effects of collisions and RF quasilinear diffusion, and to first order in radial drifts, requires finding the solution to the system of equations

$$f \approx f_0 + \tilde{f} + g$$

$$\{C(f_0)\} + \{Q(f_0)\} = 0 \tag{2.89}$$

$$\tilde{f} = -\frac{v_\parallel}{\Omega_\theta}\frac{\partial f_0}{\partial r} \tag{2.90}$$

$$\{C(g)\} + \{Q(g)\} = -\left\{C(\tilde{f})\right\} - \left\{Q(\tilde{f})\right\}. \tag{2.91}$$

From this, we can calculate the current density as described in (2.32), with the ion flux taken as known,

$$\mathbf{J}(\mathbf{x},t) = q_i \mathbf{\Gamma}_i(\mathbf{x},t) + q_e \int d^3p\ \mathbf{v}\left(f_0 + \tilde{f} + g\right). \tag{2.92}$$

To calculate a flux-surface average of this current density, we consider the cross section $\zeta = const.$ of the tokamak. For simplicity, in this cross section the flux surface will be taken as an annular cross section between r and $r + \Delta r$. The current density is averaged over this area:

$$\text{Average of } J_\parallel = \frac{\int_{-\pi}^{\pi} \Delta r\ r d\theta J_\parallel}{\int_{-\pi}^{\pi} \Delta r\ r d\theta} = \int_{-\pi}^{\pi} \frac{d\theta}{2\pi}\ J_\parallel \tag{2.93}$$

Expressed in terms of the volume flux-surface average (2.57), this is

$$\text{Average of } J_\parallel = \frac{\left\langle J_\parallel B \right\rangle}{\langle B \rangle} \tag{2.94}$$

In Chapter 3, we show how these kinetic equations are used to solve the separate problems of determining RF current drive and the bootstrap current, and observe

that the combined problem of RFCD with bootstrap requires the solution of the full set of equations (2.89)-(2.91).

Chapter 3

Introduction to the Bootstrap Current and RF Current Drive

3.1 Introduction

Before we can consider the interactions between radio frequency current drive and the bootstrap current, a summary of the relevant features of each of these elements by itself is appropriate. We present simple models for the bootstrap current, lower-hybrid current drive, and electron cyclotron current drive. We also show how the kinetic equations introduced in Chapter 2 are used in separate calculations to find the bootstrap current and the current driven by RF waves.

In Section 3.2 we discuss the bootstrap current in a Maxwellian plasma, illustrating the discussion by considering the simple case where the electron collision operator is dominated by collisions with ions. In Section 3.3 we describe RF current drive theory, using both simple models and more detailed calculations; lower hybrid wave current drive (LHCD) is discussed in Section 3.3.1, electron cyclotron wave current drive (ECCD) in Section 3.3.2.

3.2 The Bootstrap Current

A Maxwellian plasma can be described in terms of the plasma parameters: density n, temperature T, and pressure $p = nT$, for electrons and ions. Neoclassical transport theory [8, 9] is primarily concerned with the calculation of transport coefficients, which relate the fluxes of particles and energy in the plasma to the gradients of density and temperature (as well as the equilibrium electric field, when one exists) in a linear manner. The coefficients are found by solving the Fokker-Planck equations for electrons and ions in the expanded form (2.89-2.91) for a plasma near thermal equilibrium, i.e. having Q=0 and f_0 equal to a Maxwellian.

We show how this sort of calculation is performed in a simple case, then discuss the more general expressions used for the bootstrap current.

Simple Physical Picture

It is possible to derive a simple picture of the bootstrap current in the absence of RF by looking at the exhange of momentum between trapped and circulating electrons [64, 65].

Trapped electrons are able to carry a small diamagnetic flux in the toroidal direction, owing to the radial drifts. To illustrate this we look at Figure 2-2 in Chapter 2. Two trapped electron orbits are shown with $v_\parallel$ in opposite directions. Because the pressure is higher inside of the flux surface than outside, we typically get more and faster electrons on orbits with $v_\parallel > 0$ than on orbits with $v_\parallel < 0$. We note that Figure 2-2 shows orbits projected onto the plane ζ =constant, and the electrons are moving primarily along the magnetic field. So there is a net flux of electrons along the magnetic field in the positive direction. For $\epsilon \ll 1$ this diamagnetic flux has the form

$$n_t V_t \sim \epsilon^{3/2} \frac{1}{|e|\, B_\theta} \frac{dp_e}{dr} \tag{3.1}$$

where

$$n_t V_t \equiv \int_{Trapped} d^3p v_\parallel f \tag{3.2}$$

is the parallel moment of the distribution function for trapped electrons.

The momentum of these trapped electrons is transferred to circulating electrons by the detrapping of the trapped electrons through a small-angle collision. The frequency of detrapping collisions is ν_e/ϵ, as shown in (2.73). So the rate of momentum transfer to circulating electrons is

$$(m_e n_t V_t) \cdot \left(\frac{\nu_e}{\epsilon}\right).$$

The primary loss mechanism for circulating electrons is collisions with ions. We can approximate this loss of momentum as $-m_e n_e \nu_{ei}(V_e - V_i)$, where V_e and V_i are the parallel flows of ions and electrons respectively. Balancing these two gives the steady-state momentum balance

$$(m_e n_t V_t) \cdot \left(\frac{\nu_e}{\epsilon}\right) - m_e n_e \nu_{ei}(V_e - V_i) = 0. \tag{3.3}$$

Using (3.1) and taking $\nu_{ei} \sim \nu_e$, we get the following form for the bootstrap current,

$$j_{BS} \equiv -|e|\, n_e(V_e - V_i) \sim -\epsilon^{1/2} \frac{1}{B_\theta} \frac{dp_e}{dr}. \tag{3.4}$$

Nonrelativistic Lorentz Gas Model

In this section we discuss the derivation of the bootstrap current without RF waves using a simple model. We start by considering a nonrelativistic plasma with a fixed, motionless background of ions with charge $Z_i \gg 1$ and ion temperature much lower than the electron temperature.

As shown in Chapter 2, the electron distribution is $f \approx f_0 + \tilde{f} + g$, and the zero order distribution f_0 is taken to be a local nonrelativistic Maxwellian on the flux surface r,

$$f_0 = f_M = \frac{n_e(r)}{(2\pi T_e(r)/m_e)^{3/2}} \exp\left[-\frac{m_e v^2}{2T_e(r)}\right] \tag{3.5}$$

The density and temperature are functions of r, and we take the radial derivative of f_0,

$$\frac{\partial f_M}{\partial r} = \left[\frac{d\ln n_e}{dr} + \left(\frac{m_e v^2}{2T_e} - \frac{3}{2}\right)\frac{d\ln T_e}{dr}\right] f_M \tag{3.6}$$

From this it is a quick step to find

$$\tilde{f} = -\frac{v_\parallel}{\Omega_\theta}\frac{\partial f_M}{\partial r} = -\frac{v\xi}{\Omega_\theta}\left[\frac{d\ln n_e}{dr} + \left(\frac{m_e v^2}{2T_e} - \frac{3}{2}\right)\frac{d\ln T_e}{dr}\right] f_M \tag{3.7}$$

where ξ is $v_\parallel/v$.

It is possible to calculate the function g analytically when the collision operator is taken in the Lorentz gas limit, which occurs when the ions have a charge $Z_i \gg 1$. Then the electron-ion collisions dominate over electron-electron collisions and the operator can be taken to have the Lorentz form

$$C(f) = \frac{\nu_{ei}}{2}\frac{\partial}{\partial\xi}\left[(1-\xi^2)\frac{\partial f}{\partial\xi}\right], \tag{3.8}$$

where the electron-ion thermal collision frequency is

$$\nu_{ei} = \frac{q_e^2 q_i^2 n_i \ln\Lambda}{4\pi\epsilon_0^2 m_e^2 v^3} = \frac{Z_i^2 q_e^4 n_i \ln\Lambda}{4\pi\epsilon_0^2 m_e^2 v^3}. \tag{3.9}$$

In (2.90) we noted that g is calculated from the Fokker-Planck equation

$$\{C(g)\} = -\left\{C(\tilde{f})\right\}.$$

The bounce average of the Lorentz operator is derived in Appendix E with $B_t(p)/p^2 = -\nu_{ei}/2$ in the Lorentz limit; (E.12) gives

$$\{C(g)\} = \frac{\nu_{ei}}{2}\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[(1-\xi_0^2)\lambda\left(1-\frac{\Delta}{\xi_0^2}\right)\frac{\partial g}{\partial\xi_0}\right] \tag{3.10}$$

and the source term in this equation found from (E.37),

$$S \equiv -\left\{C(\tilde{f})\right\} = -\nu_{ei}H(|\xi_0| - \xi_{0T})\frac{1}{\lambda(1+\epsilon)}\left(\frac{v\xi_0}{\Omega_{\theta,min}}\frac{\partial f_M}{\partial r}\right). \tag{3.11}$$

The symmetry of the collision operator is used to set boundary conditions for g in the trapped region. A quantity $\mathcal{A}$ is said to be *symmetric* in σ, the sign of ξ_0, if $\mathcal{A}(\sigma = +1) = \mathcal{A}(\sigma = -1)$ and is referred to as *antisymmetric* if $\mathcal{A}(\sigma = +1) =$

$-\mathcal{A}(\sigma = -1)$. The source term (3.11), being proportional to ξ_0, is antisymmetric in σ. This means that $\{C(g)\}$ is also antisymmetric, and since the collision operator conserves symmetry, g must be as well. However, g must also be symmetric in σ in the trapped orbit region of phase space $|\xi_0| < \xi_{0T}$. A proof of this is straightforward: since $v_\parallel$ is zero at the trapping angle $\theta = \theta_T$, g is not dependent on the sign of $v_\parallel$ there; but since g is independent of θ, g is symmetric for all θ. Thus, the boundary condition is that g is zero for all trapped orbits, since only this trivial solution can be both symmetric and antisymmetric.

The other boundary condition is that $\partial g/\partial \xi_0$ must be finite at the farthest end of the passing region of phase space, $\xi_0 = \pm 1$. The solution of $\{C(g)\} = S$ with these two boundary conditions is

$$\begin{aligned} g &= -H(\xi_0 - \xi_{0T}) \int_{\xi_{0T}}^{\xi_0} \frac{d\xi_0}{\lambda(1+\epsilon)(1-\Delta/\xi_0^2)} \left(\frac{mv}{qB_{\theta,min}} \frac{\partial f_M}{\partial r} \right) \\ &\quad + H(-\xi_0 - \xi_{0T}) \int_{\xi_{0T}}^{|\xi_0|} \frac{d\xi_0}{\lambda(1+\epsilon)(1-\Delta/\xi_0^2)} \left(\frac{mv}{qB_{\theta,min}} \frac{\partial f_M}{\partial r} \right) \\ &= -\sigma \; H(|\xi_0| - \xi_{0T}) I(|\xi_0|) \left(\frac{mv}{qB_{\theta,min}} \frac{\partial f_M}{\partial r} \right) \end{aligned} \tag{3.12}$$

where the integral I is[1]

$$I(\xi_0) = \int_{\xi_{0T}}^{\xi_0} \frac{d\xi_0}{\lambda(1+\epsilon)(1-\Delta/\xi_0^2)}. \tag{3.13}$$

As shown in Appendix B, it is possible to get an estimate of I in the large-aspect-ratio limit $\epsilon \ll 1$:

$$I(\xi_0) \approx (\xi_0 - \xi_{0T}) \left(1 + \frac{1}{4} \frac{\xi_{0T}}{\xi_0} \right) \tag{3.14}$$

[1]In many works on neoclassical theory (e.g. [9, 7]) this integral is seen in the form

$$v \int_\lambda^{\lambda_c} \frac{d\lambda}{\langle v_\parallel \rangle} \quad \text{or} \quad \int_\lambda^{\lambda_c} \frac{d\lambda}{\langle \sqrt{1-\lambda B} \rangle}$$

where λ is a pitch-angle coordinate proportional to $(1-\xi_0^2)$. Our choice of notation uses the pitch-angle coordinate ξ_0 and reflects our preference to use bounce averages rather than flux-surface averages when the average depends on the orbit.

where we recall that $\xi_{0T} = \sqrt{2\epsilon/(1+\epsilon)} \sim \sqrt{\epsilon}$.

In Figure 3-1 we show both $\tilde{f}$ and g as a function of ξ_0, at a fixed value of $v = 4v_{Te}$. This was calculated with ϵ=0.1, so that the trapped-passing boundary is at $\xi_{0T} = 0.43$. Two versions of g are shown, one found by using the full integral I and one from the approximation (3.14), which allows us to verify that the approximation is good.

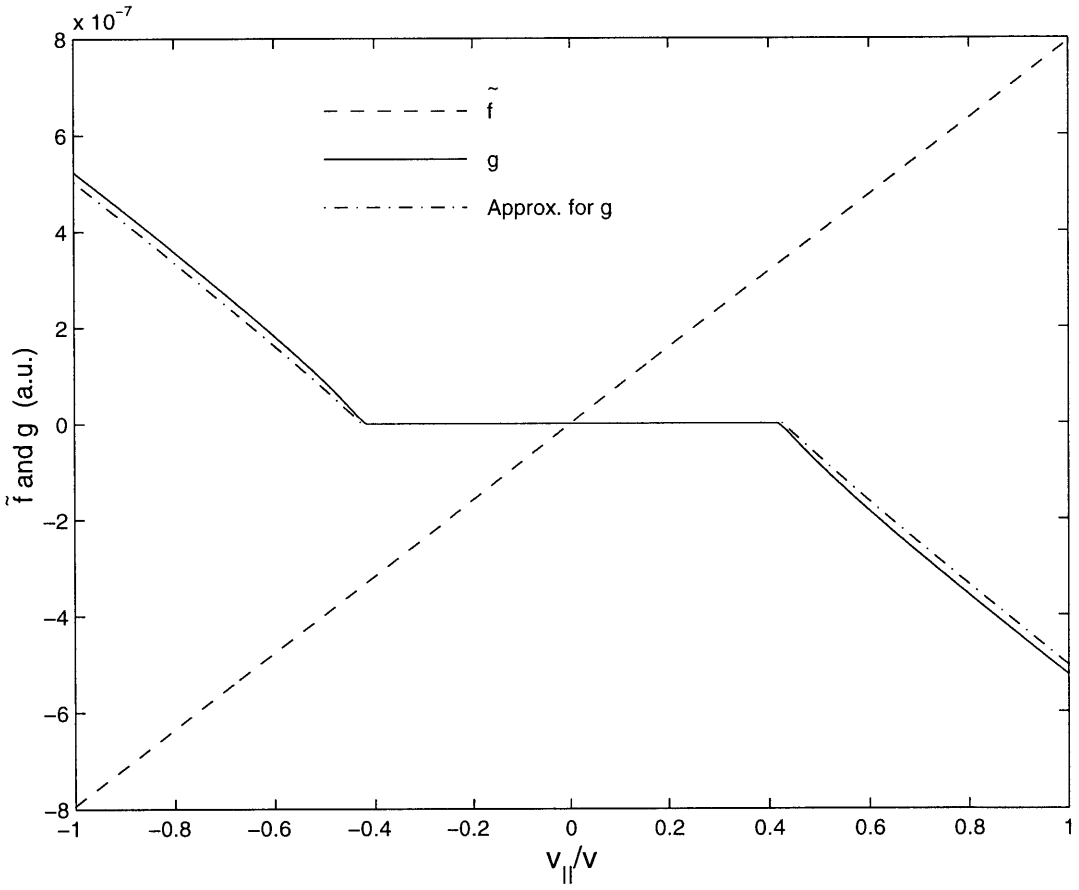

Figure 3-1: Functions $\tilde{f}$ (dashed line) and g (solid line) in the nonrelativistic Lorentz gas limit as a function of ξ_0 for a fixed $v = 4v_{Te}$ at $\epsilon = 0.1$ and $\theta = 0$. The dash-dot line is the approximate value of g from (3.14).

To get a physical picture of the distribution function, first consider the limit $\epsilon \to 0$. In this limit there are no trapped particles; ξ_{0T} goes to zero, and therefore we obtain a limiting case of g equal to $-\tilde{f}$, and the collisional response cancels out the radial drift effects in the distribution function. However, when there is a nonzero ϵ and some

electrons are trapped, the fact that g must be zero in the trapped region of phase space means that it cannot cancel out $\tilde{f}$ there, and a net current remains.

The bootstrap current is then found by taking the integral of the functions $\tilde{f}+g$, and averaging over a flux surface,

$$J_b = \frac{\langle J_\parallel B \rangle}{\langle B \rangle} = \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} q \int d^3p \; v_\parallel (\tilde{f}+g) \tag{3.15}$$

Using (3.7) and (3.12) this can be evaluated to be

$$J_b = -1.46\sqrt{\epsilon}\frac{n_e T_e}{B_\theta}\left(\frac{d\ln n_e}{dr} + \frac{d\ln T_e}{dr}\right) \tag{3.16}$$

which is the same result as obtained by Rosenbluth, et.al [7] for $Z_i \gg 1$ and $T_i = 0$.

General Expression for the Bootstrap Current

In general, the bootstrap current can be expressed in terms of the plasma parameters n_e, n_i, T_e, and T_i through transport coefficients, in the form [8, 9]

$$\langle J_{\parallel,BS} B \rangle = -p_e \frac{B_\zeta}{B_\theta}\left(L_{31}A_1 + L_{32}A_2 + L_{34}A_4\right) \tag{3.17}$$

where

$$\begin{aligned} A_1 &= \frac{d\ln p_e}{dr} + \frac{T_i}{Z_i T_e}\frac{d\ln p_i}{dr} \\ A_2 &= \frac{d\ln T_e}{dr} \\ A_4 &= \alpha_i \frac{T_i}{Z_i T_e}\frac{d\ln T_i}{dr} \end{aligned} \tag{3.18}$$

The dimensionless transport coefficients L_{31}, L_{32}, and L_{34}, and the factor α_i are to be calculated from the distribution functions. It is worth noting that ions also have a parallel flux due to neoclassical transport, and that if we keep a finite ion temperature the electron flux must be calculated relative to the ion frame of reference to get this expression.

3.3 RF Current Drive

The term RF current drive is a label that describes a wide variety of techniques for using RF waves to induce a plasma current. Such waves have a variety of frequencies, wavenumbers, and polarizations as shown in Section 1.2.2, and for each case the quasilinear operator $Q(f)$ will have a different form [42].

In general the problem of RF current drive is solved while neglecting the radial electron drift entirely. So the problem to be solved reduces to (2.89),

$$\{C(f_0)\} + \{Q(f_0)\} = 0$$

with

$$\mathbf{J}(\mathbf{x}, t) = q_e \int d^3p \; \mathbf{v} f_0 \tag{3.19}$$

as the total current. Physically, the quasilinear diffusion of electrons in momentum-space results in a non-Maxwellian distribution f_0 at steady-state. These electrons are constantly losing energy and momentum to ions and the electrons in the bulk of the distribution through collisions, and thus the distribution f_0 must be maintained by the continous transfer of RF power to the plasma. In steady state the power density P_d dissipated by collisions is equal to [42]

$$\int d^3p \; \left(\gamma m_e c^2\right) Q(f_0) = \int d^3p \; \frac{\mathbf{p}}{\gamma m_e} \cdot \mathbf{\Gamma}_{ql} \tag{3.20}$$

where we used $Q(f) = -\nabla_{\mathbf{p}} \cdot \mathbf{\Gamma}_{ql}$ and integration by parts with Gauss' divergence theorem. For practical applications the quantity of interest is the current driven per unit of RF power dissipated. This can be expressed in terms of the ratio J/P_d of current density to density of power dissipated.

3.3.1 Lower Hybrid Current Drive

We construct a simple picture of lower hybrid waves at a particular flux surface r. The lower-hybrid range of frequencies takes its name from the lower hybrid resonace

frequency for cold plasma waves propagating at right angles to the magnetic field,

$$\omega_{LH} \approx \frac{\omega_{pi}^2}{1 + \omega_{pe}^2/\omega_{ce}^2}, \tag{3.21}$$

where ω_{pe} are ω_{pi} are the electron and ion plasma frequencies respectively ($\omega_{ps} = \sqrt{q_s n_s/\epsilon_0 m_s}$) and the cyclotron frequency ω_{ce} has already been defined. Since $\omega_{pe} \sim \omega_{ce}$ in tokamak plasmas lower hybrid waves have frequencies near ω_{pi}, which falls between the cyclotron frequencies,

$$\omega_{ci} \ll \omega \ll \omega_{ce}. \tag{3.22}$$

Heating and current drive schemes using lower hybrid waves make use of electron Landau damping [20] rather than resonance damping to transfer power to the plasma, and in fact in most experiments the lower-hybrid waves are launched at frequencies slightly greater than the lower hybrid resonance frequency everywhere in the plasma.

Waves in the LH range are launched using a phased waveguide array with an unidirectional spectrum in $k_\parallel$. These waves have a parallel index of refraction $N_\parallel = ck_\parallel/\omega$ which is greater than one, and thus they do not propagate in a vacuum. Therefore the antenna must be close to the plasma edge, and the waves are able to tunnel to a density where a cutoff occurs beyond which LH waves can propagate [32].

Upon reaching a flux surface r inside the plasma where current drive occurs, the spectrum in $k_\parallel$ has a lower limit, given by conditions of accessibility to the plasma, and an upper limit, determined by the condition for wave damping. We discuss these here.

When the value of $k_\parallel$ is too low, lower hybrid waves will become evanescent (with $N_{\perp,i}$ nonzero) at densities lower than the local density $n_e(r)$, and therefore they cannot propagate to the flux surface of interest. This accessibility problem has been well studied [30, 33, 34], and using cold plasma wave theory, one finds two cases for the lower limit of $N_\parallel$: defining $\omega_h = \sqrt{\omega_{ce}\omega_{ci}}$,

1. If $\omega < \omega_h$ and $\left(\frac{\omega_{pi}(r)}{\omega}\right)^2 > \frac{(\omega/\omega_h)^2}{1-(\omega/\omega_h)^2}$, the LH waves are accessible for

$$N_\parallel^2 > \frac{1}{1-(\omega/\omega_h)^2} \tag{3.23}$$

2. If $\omega < \omega_h$ and $\left(\frac{\omega_{pi}(r)}{\omega}\right)^2 < \frac{(\omega/\omega_h)^2}{1-(\omega/\omega_h)^2}$, or if $\omega > \omega_h$, the LH waves are accessible for

$$N_\parallel^2 > 1 + 2\left(\frac{\omega_{pe}(r)}{\omega_{ce}}\right)^2 - \left(\frac{\omega_{pi}(r)}{\omega}\right)^2 + 2\left(\frac{\omega_{pe}(r)}{\omega_{ce}}\right)\sqrt{1+\left(\frac{\omega_{pe}(r)}{\omega_{ce}}\right)^2 - \left(\frac{\omega_{pi}(r)}{\omega}\right)^2} \tag{3.24}$$

In a tokamak, after the LH waves are launched from the edge they are observed to undergo a significant upshift in $k_\parallel$. There are several linear and nonlinear mechanisms that can give rise to such an upshift [66, 67, 68, 69]. For this reason the upper bound on the $k_\parallel$ spectrum is not determined by the antenna structure but by the propagation of the waves. The waves can increase in $k_\parallel$ until their energy is transferred to the plasma by electron Landau damping. This process is understood by the kinetic theory of waves, and an approximate condition can be derived for significant power to be deposited by electron Landau damping of LH waves [70, 30],

$$2\pi^{1/2}\left(\frac{m_i}{m_e}\right)^{1/2} x_{oe}^3 e^{-x_{oe}^2} k_\parallel \Delta x \sim 1 \tag{3.25}$$

where $x_{oe} = \omega/\sqrt{2}k_\parallel v_{Te}$ and Δx is the distance over which Landau damping will occur. By setting Δx to a distance small relative to the tokamak size this equation gives an upper bound to the values of $k_\parallel$ that will reach the flux surface.

The generalized quasilinear operator (2.50) for a relativistic plasma was given in Section 2.3.3. Since the lower hybrid frequency is much lower than the electron cyclotron frequency the condition for wave-particle resonance with electrons, given by $\omega - k_\parallel v_\parallel - n\Omega = 0$, can only be met for n=0. If we denote the lower and upper bounds on $k_\parallel$ as k_1 and k_2 respectively, the spectrum in $k_\parallel$ can be translated directly into a range in $v_\parallel$ from the resonance condition $v_\parallel = \omega/k_\parallel$: $v_1 < v_\parallel < v_2$, where $v_1 \equiv \omega/k_2$ and $v_2 \equiv \omega/k_1$.

Between the two limits $v_1 < v_\parallel < v_2$ the wave fields are sufficiently high so that $D_{\mathbf{k}}$ in (2.50) is much greater than the thermal collisional diffusion coefficient $\nu_e p_{Te}^2$. For this reason it is sufficient to take the diffusion coefficient to be a constant, denoted as D_0, which is greater than 1 when normalized to $\nu_e p_{Te}^2$.

The linear operator (2.51) in this case becomes

$$\mathcal{L}(\mathcal{A}) = \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \frac{1-\xi^2}{p^2}\frac{1}{\xi}\frac{\partial \mathcal{A}}{\partial \xi} = \frac{1}{p_\parallel}\frac{\partial \mathcal{A}}{\partial p_\parallel} \tag{3.26}$$

where the derivative with respect to $p_\parallel$ is taken at constant $p_\perp$. Thus lower hybrid diffusion occurs in the $p_\parallel$ direction while $p_\perp$ is unchanged by the waves.

So, in (p, ξ) coordinates, the LH quasilinear operator is

$$\begin{aligned} Q(f) &= \frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2 \xi D_{LH}(v_\parallel)\left(\xi\frac{\partial f}{\partial p} + \frac{1-\xi^2}{p}\frac{\partial f}{\partial \xi}\right)\right] \\ &\quad + \frac{1}{p}\frac{\partial}{\partial \xi}\left[(1-\xi^2) D_{LH}(v_\parallel)\left(\xi\frac{\partial f}{\partial p} + \frac{1-\xi^2}{p}\frac{\partial f}{\partial \xi}\right)\right] \end{aligned} \tag{3.27}$$

$$\begin{aligned} D_{LH}(v_\parallel) &= D_0, \quad v_1 < v_\parallel < v_2 \\ &= 0 \quad\quad v_\parallel < v_1, v_\parallel > v_2. \end{aligned}$$

One-dimensional Nonrelativistic Model

We can write (3.27) as

$$Q(f) = \frac{\partial}{\partial p_\parallel} D_{LH}(v_\parallel)\frac{\partial f}{\partial p_\parallel} \tag{3.28}$$

where the partial derivatives in $p_\parallel$ are at constant $p_\perp$. Because the diffusion is entirely in the direction of $p_\parallel$ we can get an idea of the physics of LHCD from a simple, one-dimensional model [71, 31, 29]:

To get a physical picture of lower hybrid current drive, we consider here current drive in a non-relativistic homogeneous plasma. In this limit we can observe the features of LHCD but we expect to get only approximate predictions of the current

drive in a relativistic toroidal plasma.

The *ad hoc* justification for the 1D model for lower hybrid current drive is that the parallel velocity-space dynamics are more important than the perpendicular dynamics. The electrons are still able to move perpendicular to the magnetic field, but this motion is assumed to be independent of the parallel motion. The distribution function is assumed to have a form

$$f = f_M(v_\perp)F(v_\parallel) \tag{3.29}$$

where f_M is a Maxwellian of fixed temperature T_e. The dependence of the problem on $v_\perp$ is removable by integration.

The resulting collision and quasilinear operators are [71]

$$\begin{aligned} C(F) &= (2+Z_i)\frac{\partial}{\partial w}\left(\frac{1}{w^3}\frac{\partial}{\partial w}+\frac{1}{w^2}\right)F(w) \\ Q(F) &= \frac{\partial}{\partial w}D_{LH}\frac{\partial}{\partial w}F(w) \end{aligned} \tag{3.30}$$

where the normalized parallel velocity is $w = v_\parallel/v$, and in the nonrelativistic case D_{LH} is normalized to the thermal units $\nu_e v_{Te}^2$. The collision operator is an approximation in the limit of $\left|v_\parallel\right| > v_{Te}$.

The steady-state limit (where D_{LH} is assumed to be constant in time) in this homogeneous plasma case is given by solving $C(F)+Q(F)=0$, and the solution F is

$$F(w) = C\exp\left[\int_0^w \frac{-w\ dw}{1+w^3 D_{LH}/(2+Z_i)}\right], \tag{3.31}$$

where C is a normalization constant.

We take D_{LH} to be a constant D_0, much greater than 1 in a range of velocities $w_1 < w < w_2$, and zero for all other values of w. Matching the solutions at w_1 and w_2, the result is

$$F(w) = C\exp\left(-\frac{w^2}{2}\right) \qquad w < w_1$$

$$
\begin{aligned}
&= C \exp\left(-\frac{w_1^2}{2} - \frac{2+Z_i}{D_0 w_1}\right) \exp\left(\frac{2+Z_i}{D_0 w}\right) && w_1 < w < w_2 \\
&= C \exp\left[-\frac{w_1^2}{2} + \frac{w_2^2}{2} - \frac{2+Z_i}{D_0}\left(\frac{1}{w_1} - \frac{1}{w_2}\right)\right] \exp\left(-\frac{w^2}{2}\right) && w > w_2
\end{aligned}
\tag{3.32}
$$

The constant C is chosen to make the density correct,

$$
n = \int_{-\infty}^{\infty} F(w) dw. \tag{3.33}
$$

The distribution function has a nearly-flat plateau in the range $w_1 < w < w_2$. A particular example of $F(w)$ is shown in Figure 3-2, with the plateau clearly visible.

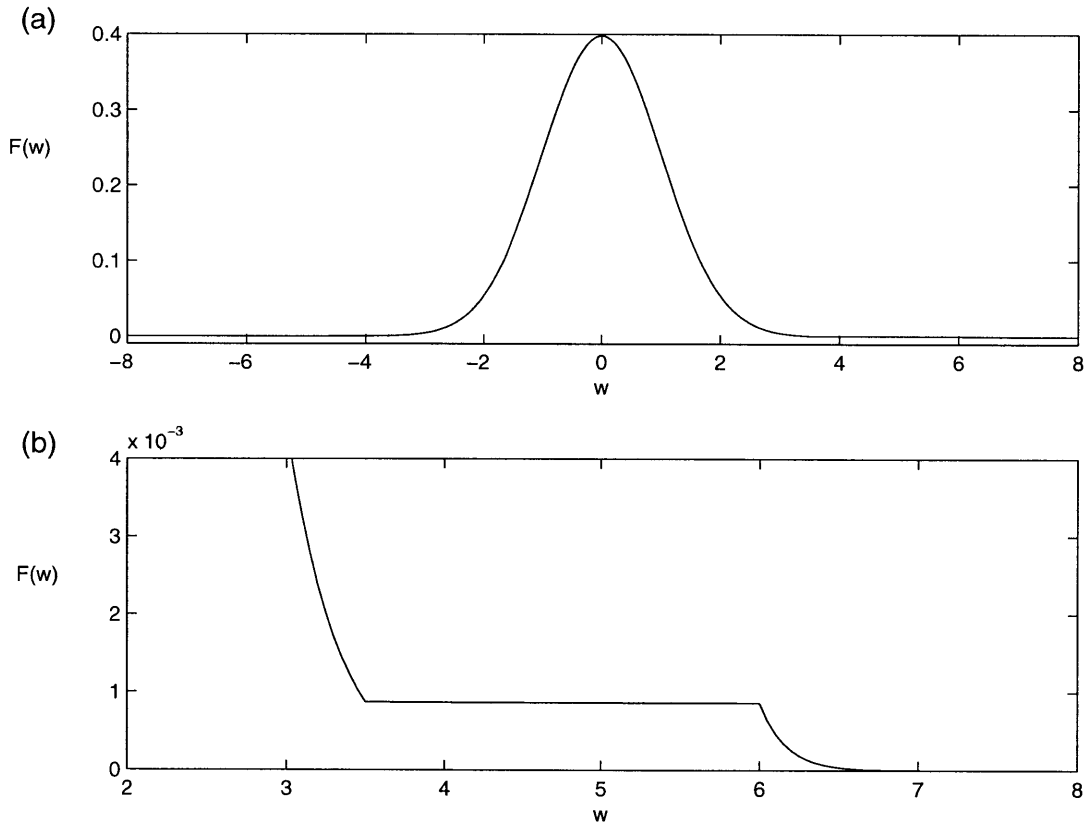

Figure 3-2: (a) Parallel distribution function $F(w)$ from (3.32)for the one-dimensional model of LHCD as a function of $w = v_{\|}/v_{Te}$, for w_1=3.5, w_2=6.0, and D_0=10. (b) Same function showing plateau region only.

The plasma current is carried by the electrons in this plateau. These electrons are

constantly losing energy and momentum to the electrons in the bulk of the distribution through collisions, and thus the RF must transfer power (through Landau damping) to replace this collisional dissipation at a constant rate to maintain the current. The current density is found to be

$$\begin{aligned} J_{\|} &= qnv_{Te}\int dw\ wF(w) \\ &\approx qnv_{Te}F_M(w_1)\frac{w_2^2 - w_1^2}{2} \end{aligned} \tag{3.34}$$

and the density of power dissipated by collisions is

$$\begin{aligned} P_d &= \nu_e nmv_{Te}^2 \int dw\ \frac{w^2}{2}\frac{\partial}{\partial w}D_{LH}\frac{\partial}{\partial w}F(w) \\ &\approx \nu_e nmv_{Te}^2(2+Z_i)F_M(w_1)\ln\frac{w_2}{w_1} \end{aligned} \tag{3.35}$$

The amount of current driven per unit of RF power absorbed is calculated by taking the ratio of the quantities found above:

$$\frac{J_{\|}}{P_d} \approx \frac{q}{\nu_e m v_{Te}}\frac{w_2^2 - w_1^2}{2(2+Z_i)\ln(w_2/w_1)} \tag{3.36}$$

Karney and Fisch [72] showed that, for $Z_i = 1$, the actual value of $J_{\|}/P_d$ calculated numerically in two dimensions is greater than this by a factor of about 2.5.

Other Work on Lower-Hybrid Current Drive

Various works have presented other models which more accurately approximate the fully two-dimensional current drive problem. Fuchs, et.al. [73] proposed to model the LH current drive in two dimensions by allowing the temperature of the Maxwellian part $f_M(v_\perp)$ to have a varying temperature $T_\perp(w)$ determined by the combined actions of parallel diffusion and pitch-angle collisions. This gave an evaluation of $J_{\|}/P_d$ which was more accurate than (3.36) and compared well with results from 2D (velocity space) Fokker-Planck codes. A different, general approach to calculating $J_{\|}/P_d$ was given by Antonsen and Chu [74] using the *adjoint technique* [75]. This makes use of

the self-adjointness property of the collision operator, and one finds that the solution to the Spitzer-Härm problem [76] is a Green's function for the current drive problem. A simple form of the adjoint technique [39] gives a result

$$\frac{J_{\|}}{P_d} \approx \frac{q_e}{\nu_e m_e v_{Te}} \frac{4}{5+Z_i} \frac{w_2^2 - w_1^2}{2\ln(w_2/w_1)} \tag{3.37}$$

which differs from (3.36) by a factor of $4(2+Z_i)/(5+Z_i)$. This factor is greater than one; for example, it is equal to 2 for $Z_i = 1$. Calculations of $J_{\|}/P_d$ have been performed by many authors for cases of lower hybrid current drive [74, 39, 11].

Numerical Calculation of LHCD

By understanding the one-dimensional model, we are able to see the most important features of lower hybrid current drive. The steady-state distribution function contains a plateau or "tail" in the region of momentum space where electrons are in resonance with the LH wave spectrum. For a quasilinear operator in the (now relativistic) form (3.27) with D_{LH} much larger than the thermal collisional diffusion coefficient, the slope $\partial f/\partial p_{\|}$ becomes very small.

We use the Fokker-Planck code FASTFP, which will be described in Chapter 4, to solve the problem of lower hybrid current drive without the bootstrap current for a typical and illustrative case.

The plasma parameters used in this solution are chosen to match a hypothetical high bootstrap current scenario in the Alcator C-Mod tokamak [77]. The remaining plasma parameters are detailed in Table C.1 of Appendix C. In this scenario, the lower hybrid current will be located off of the axis, at a minor radius of about 0.15m, which gives a value of $\epsilon = 0.23$. This calculation is performed with $D_0 = 4.0$ in the range $v_1 < v_{\|} < v_2$ with $v_1 = 3.5 v_{Te}$, as determined by (3.25), and $v_2 = 6.0 v_{Te}$, as determined by (3.23).

Figure 3-3 shows a contour plot of the electron distribution f as a function of $(p_{\|,0}, p_{\perp,0})$. The primary change which occurs when we consider this relativistic LHCD problem is that the resonance region is a function of $p_{\perp}$ as well as $p_{\|}$. Figure 3-3 shows

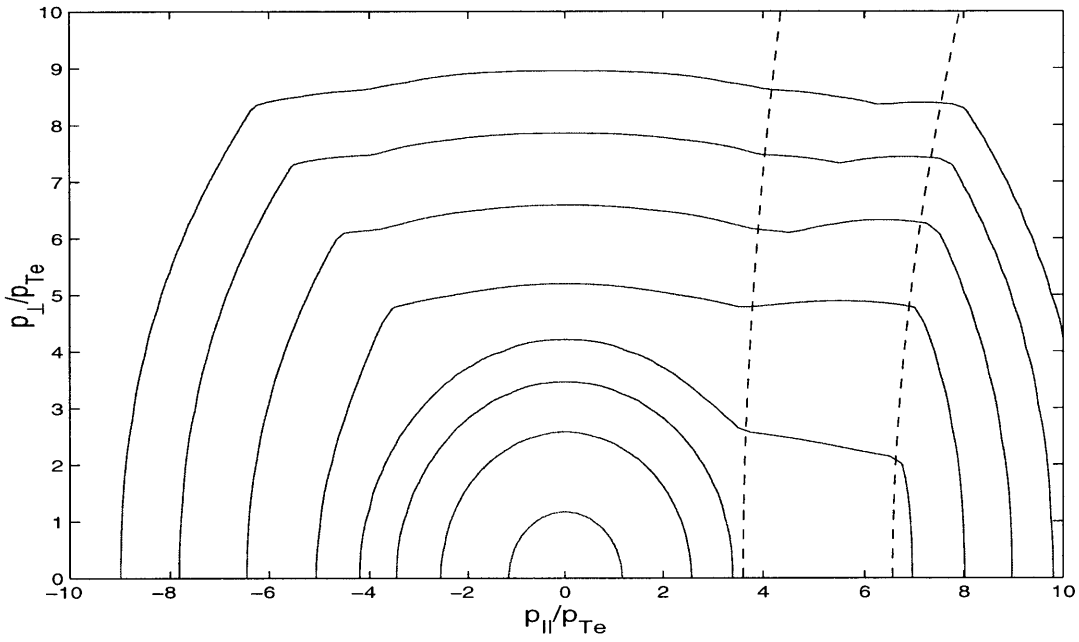

Figure 3-3: Contour plot in $(p_{\|}/p_{Te}, p_{\perp}/p_{Te})$ space of the LH-modifed electron distribution function. The dashed lines indicate the resonant region of phase space $v_{\|} \in (3.5v_{Te}, 6.0v_{Te})$. Parameters are Alcator C-Mod type (Table C.1)with r=0.15 m and D_0=4.0.

with dashed lines the resonance region $v_1 < v_{\|} < v_2$ in the phase space $(p_{\|}, p_{\perp})$. The local current density parallel to the magnetic field is found by taking a moment of this distribution function, $\left\langle J_{\|} B \right\rangle / \langle B \rangle = 10.6$ MA/m^2.

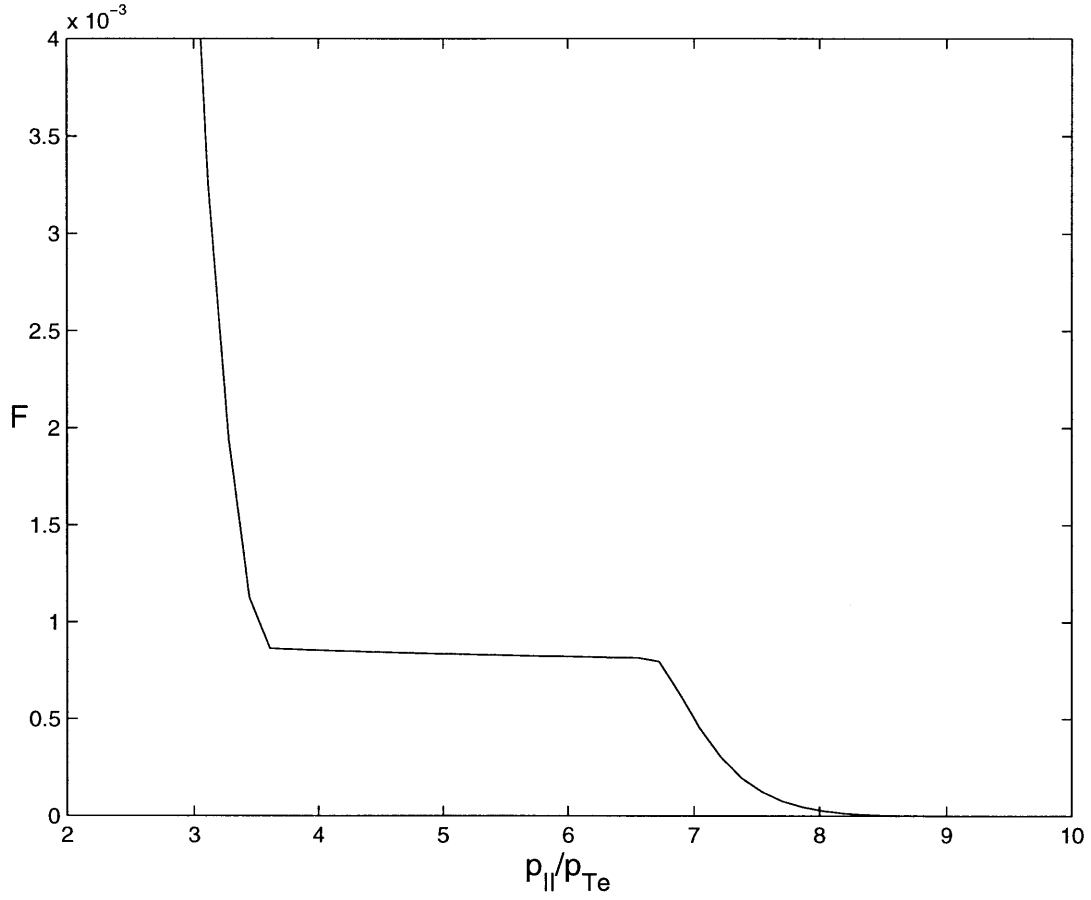

Figure 3-4: Parallel distribution function $F(p_{\|}) = \int 2\pi p_{\perp} dp_{\perp} f$ as a function of $p_{\|}/p_{Te}$ for the LH-modified electron distribution function. Parameters are Alcator C-Mod type (Table C.1)with r=0.15 m, D_0=4.0 for $v_{\|} \in (3.5 v_{Te}, 6.0 v_{Te})$.

We note the appearance of the plateau in phase space. Figure 3-4 shows this more clearly by integrating out the $p_{\perp}$ dependence, leaving

$$F(p_{\|}) \equiv \int 2\pi p_{\perp,0}\, dp_{\perp,0}\, f. \tag{3.38}$$

Assuming this plasma is only weakly relativistic, we find that the minimum value of $p_{\|}$ in this plateau is approximately 3.5 p_{Te}, which we refer to as p_1. A reasonable estimate

of the height of the plateau is the value of the parallel Maxwellian distribution at $p_\parallel = p_1$, according to the estimate given by the one-dimensional picture of LHCD (3.32).

FASTFP also allows us to calculate $J_\parallel/P_d$ described in the previous section. For this case the code gives $J_\parallel/P_d$=15.1 in the thermal units $q/\nu_e m v_{Te}$. The approximate value (3.37) for $w_1 = 3.5$ and $w_2 = 6$ and $Z_i = 1$, gives

$$\frac{J_\parallel}{P_d} \approx \frac{q_e}{\nu_e m_e v_{Te}} \frac{4}{5+Z_i} \frac{w_2^2 - w_1^2}{2\ln(w_2/w_1)} = 14.7 \frac{q}{nmv_{Te}}. \tag{3.39}$$

This comparision is a good benchmark for the FASTFP code in calculations of LHCD without the bootstrap current.

3.3.2 Electron Cyclotron Current Drive

RF waves near the electron cyclotron frequency and its harmonics are not ideally suited to drive a current along the magnetic field, since there is no significant parallel momentum transfer from waves to electrons in cyclotron damping. However EC current drive is possible by two different methods. The simpler and more general method was suggested by Fisch and Boozer [36]. A spectrum of waves which are unidirectionally launched will interact preferrentially with electrons moving in one direction along the magnetic field (which we refer to as the forward direction) according to the wave-particle resonance condition $\omega - k_\parallel v_\parallel - n\Omega = 0$. These forward-moving electrons will be heated by the waves, increasing their perpendicular momentum. Because the collisional cross-section of charged particles decreases as they become more energetic, the forward-moving electrons collide with ions less frequently than those moving in the opposite direction and feel less collisional drag. The ions, on average, are more likely to collide with the electrons moving in the opposite direction, and the resulting momentum transfer gives the ion distribution a net negative flux which is small compared to the electron thermal velocity. Since the overall momentum must be conserved, the electrons will have acquired a positive flux and there is a net current.

An earlier work by Ohkawa [37] gave a mechanism which relies on unidirection-

ally trapping electrons for current drive. The increase in perpendicular momentum due to EC waves means that those electrons which are nearly trapped can become trapped. As noted earlier, for a unidirectional spectrum parallel to the magnetic field this interaction is preferentially with forward-moving electrons. In steady-state this trapping will be balanced by untrapping due to collisions, which occurs with equal probability in both directions along the magnetic field. This creates a surplus of electrons circulating in the backward direction, and there is a net electron flux opposite to the direction of wave-particle interaction.

It has been noted [78] that the total ECCD in a tokamak will have contributions from both of these current-drive mechanisms. Since the currents driven by these two means are in opposite directions, there is an intermediate regime where they can cancel out, producing poor current drive. Optimal ECCD must act on electrons far from the trapped-passing boundary (Fisch-Boozer) or very close to it (Ohkawa).

Electron cyclotron waves are launched with a frequency near a cyclotron harmonic, $\omega \approx n\Omega$. Taking ω to be positive, n will be a negative integer since $\Omega = q_e B_0/\gamma m_e$ and the electron charge is negative, and we will often talk about $|n|$ instead.

We take the general relativistic quasilinear operator (2.50) for the specific case of electron cyclotron waves. For a given mode $\mathbf{k}$ we find it illustrative to consider three separate elements of $Q(f)$. First, there is the wave-particle resonance condition, signified by the delta function of $\omega - k_\parallel v_\parallel - n\Omega$. Second, we find the direction of electron diffusion in momentum phase space, by looking at the linear operator $\mathcal{L}(\mathcal{A})$ in (2.51). Finally, the strength of the wave-particle interaction is given by the diffusion coefficient $\mathbf{D_k}$ (2.51).

An important point will be clearly observed by taking these three parts separately. We will find that the resonance condition and the diffusion direction are entirely determined by setting only two parameters, the parallel index of refraction $N_\parallel$ and the ratio $n\Omega_0/\omega$ (where $\Omega_0 \equiv \gamma\Omega$ is independent of p). Note that this ratio will always be approximately 1 and is not dependent on the choice of n but only on how close one is to the harmonic $\omega = n\Omega_0$. The diffusion coefficient is a more complicated quantity which depends on the value of n and the power and polarization of the waves.

Wave-Particle Resonance

The particular electrons which are affected by the waves are those moving in resonance with the waves, $\omega - k_{\|}v_{\|} - n\Omega = 0$. In the case of resonance with $n \neq 0$ we find the effect of the relativistic factor γ to be highly important. Both $k_{\|}v_{\|} = k_{\|}p_{\|}/\gamma m_e$ and $\Omega = q_e B_0/\gamma m_e$ are inversely proportional to γ. Using $\gamma = \sqrt{1 + p^2/m_e^2 c^2}$ the relativisitic resonance condition can be written

$$\sqrt{1 + \frac{p^2}{m_e^2 c^2}} - N_{\|}\frac{p_{\|}}{m_e c} - \frac{n\Omega_0}{\omega} = 0 \tag{3.40}$$

where the nonrelativistic cyclotron frequency $\Omega_0 \equiv \gamma\Omega$ is independent of $\mathbf{p}$.

EC waves are generally launched with $N_{\|} < 1$ so that they can propagate in the vacuum from the antenna to the plasma. The resonance curve for the case $N_{\|} < 1$ can be shown to be an ellipse in $(p_{\|}, p_{\perp})$ space:

$$\frac{(p_{\|} - a)^2}{(\Delta p_{\|})^2} + \frac{p_{\perp}^2}{(\Delta p_{\perp})^2} = 1 \tag{3.41}$$

where

$$a = m_e c \frac{(n\Omega_0/\omega)N_{\|}}{1 - N_{\|}^2} \tag{3.42}$$

$$\Delta p_{\|} = m_e c \frac{1}{(1 - N_{\|}^2)^{1/2}} \left[\frac{(n\Omega_0/\omega)^2}{1 - N_{\|}^2} - 1\right]^{1/2} \tag{3.43}$$

$$\Delta p_{\perp} = m_e c \left[\frac{(n\Omega_0/\omega)^2}{1 - N_{\|}^2} - 1\right]^{1/2} \tag{3.44}$$

This ellipse depends only upon $N_{\|}$ and $n\Omega_0/\omega$ as predicted.

Direction of Electron Diffusion

Given the resonance condition $\omega - k_{\|}v_{\|} - n\Omega = 0$, the linear differential operators in (2.50) are now

$$\mathcal{L}(\mathcal{A}) = \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(\frac{k_{\|}p}{\gamma m_e \omega} - \xi\right)\frac{1}{p^2}\frac{\partial \mathcal{A}}{\partial \xi}$$

$$= \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(1 - \xi^2 - \frac{n\Omega_0}{\gamma\omega}\right)\frac{1}{p^2\xi}\frac{\partial \mathcal{A}}{\partial \xi} \tag{3.45}$$

We note that the direction of diffusion is

$$\Delta p = \frac{1}{p} \tag{3.46}$$

$$\Delta\xi = \left(\frac{k_\parallel p}{\gamma m_e \omega} - \xi\right)\frac{1}{p^2} \tag{3.47}$$

Recalculating this in terms of $\Delta p_\parallel$ and $\Delta p_\perp$,

$$\Delta p_\parallel = \frac{k_\parallel}{\gamma m_e \omega} \tag{3.48}$$

$$\Delta p_\perp = \left(1 - \frac{k_\parallel p_\parallel}{\gamma m_e \omega}\right)\frac{1}{p_\perp} = \frac{n\Omega_0}{\gamma\omega}\frac{1}{p_\perp} \tag{3.49}$$

for electrons in resonance. That the direction of diffusion is primarily in $p_\perp$ can be seen by taking the ratio,

$$\frac{\Delta p_\parallel}{\Delta p_\perp} = \frac{k_\parallel p_\perp / m_e}{n\Omega_0} = \frac{\omega}{n\Omega}\frac{p_\perp}{m_e c}N_\parallel \tag{3.50}$$

which is small for nearly perpendicular propagation when $N_\parallel < 1$ and $p_\perp \sim m_e v_{Te} \ll m_e c$.

In Figure 3-5 we show a plot of the resonance condition in momentum phase space with axes $(p_\parallel, p_\perp)$, for $N_\parallel = 0.35$ and $n\Omega_0/\omega = 0.97$. The electron orbits satisifying the resonance condition lie on the solid line, and the short marks on this ellipse show the direction of electron quasilinear diffusion.

The Diffusion Coefficient

To simplify the calculation of the diffusion coefficient $D_{\mathbf{k}}$ in (2.52), we assume that

$$\frac{k_\perp v_\perp}{\Omega} = N_\perp \frac{p_\perp}{m_e c}\frac{\omega}{\Omega_0} \ll 1 \tag{3.51}$$

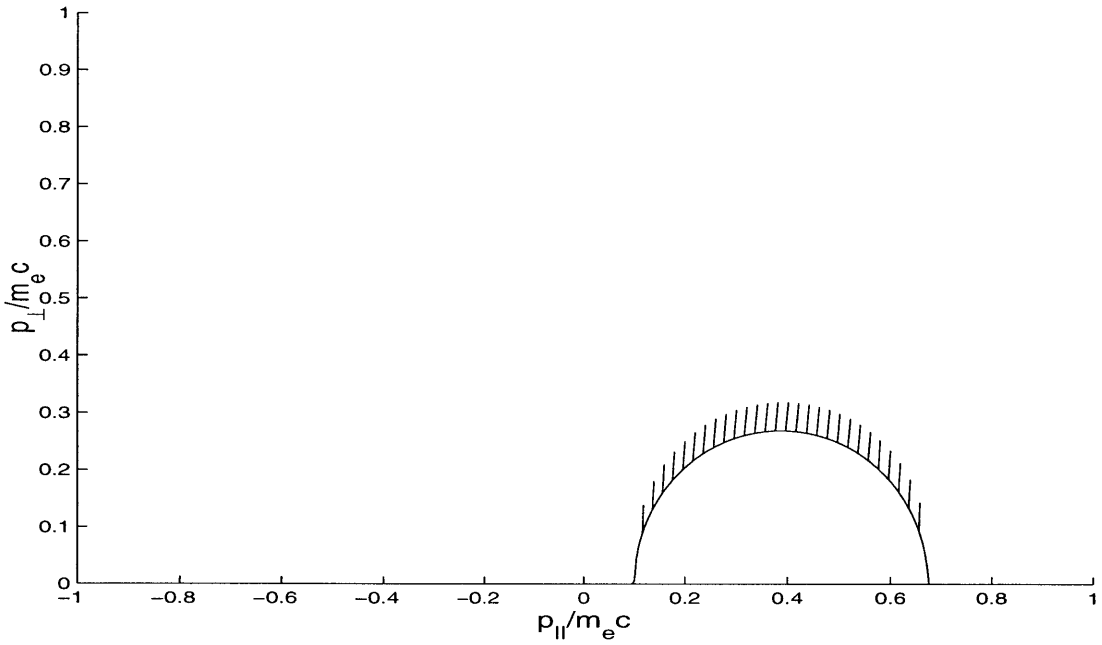

Figure 3-5: Resonance condition $\omega - k_{\|}v_{\|} - n\Omega = 0$ in $(p_{\|}/m_e c, p_{\perp}/m_e c)$ phase space for $N_{\|} = 0.35$ and $n\Omega_0/\omega = 0.97$. The marks indicate the direction of EC quasilinear diffusion in momentum space.

for resonance with thermal electrons having $p_e \sim m_e v_{Te} \ll m_e c$. In this limit the Bessel function $J_n(z)$ can be approximated by $(z/2)^n/n!$. As noted earlier, since electrons have a negative charge the cyclotron resonance gives $n < 0$. So we use the Bessel function identity $J_{-|n|} = (-1)^{|n|} J_{|n|}$, and the Bessel function approximation shows that $J_{|n|-1} \gg J_{|n|} \gg J_{|n|+1}$ for $k_{\perp}v_{\perp}/\Omega \ll 1$.

At this point it is important to discuss the polarization of the waves. EC waves in the limit $N_{\|} = 0$ are separated into two kinds: the O-mode, having $\tilde{\mathbf{E}}$ parallel to the equilibrium magnetic field, and the X-mode, with perpendicular elliptical polarization. We use the same labels for current drive in which $N_{\|}$ is finite but $N_{\|} \ll N_{\perp}$. The X-mode diffusion coefficient with $J_{|n|-1} \gg J_{|n|} \gg J_{|n|+1}$ is approximately

$$D_{\mathbf{k}} \approx q_e^2 \left| E_{\mathbf{k},-} e^{-i\varphi} J_{|n|-1} \right|^2 \approx \left| E_{\mathbf{k},-} \right|^2 \left[\frac{1}{(|n|-1)!} \right]^2 \left(\frac{k_{\perp} v_{\perp}}{2\Omega} \right)^{2|n|-2} \tag{3.52}$$

while the O-mode, being dominated by $E_{\mathbf{k},\parallel}$, has diffusion coefficient

$$D_{\mathbf{k}} \approx q_e^2 \left| \frac{p_\parallel}{p_\perp} E_{\mathbf{k},\parallel} J_{|n|} \right|^2 \approx \left| E_{\mathbf{k},\parallel} \right|^2 \left[\frac{1}{|n|!} \right]^2 \left(\frac{k_\perp v_\perp}{2\Omega} \right)^{2|n|} \frac{p_\parallel^2}{p_\perp^2}. \tag{3.53}$$

For modes with comparable fields O-mode diffusion is lower than X-mode diffusion by a factor of $k_\perp v_\parallel / \Omega$, and since these expressions get smaller as $|n|$ increases, the modes which are most promising for current drive and heating are those with the lowest $|n|$.

The mode with the greatest interaction with electrons is the X-mode with $|n|$=1. However, this mode is not easily launched in a tokamak plasma: if launched with $\omega > \omega_{ce}$, it encounters the right-hand cutoff

$$\omega = \omega_R = \omega_{ce} \left(\frac{1}{2} \sqrt{1 + 4 \frac{\omega_{pe}^2}{\omega_{ce}^2}} + \frac{1}{2} \right) \tag{3.54}$$

as the density increases [17]. So the $|n|$=1 X-mode can only be launched on the high-magnetic-field side of the tokamak on the inside of the torus. This can be difficult since the space near the center of the tokamak is usually taken up by other equipment and because smaller aspect ratios (R_0/a) are becoming more common in tokamaks.

Thus to get a strong wave-electron interaction most EC heating and current drive experiments launch the O-mode with $\omega \approx \omega_{ce}$ or the X-mode with $\omega \approx 2\omega_{ce}$. The former case also has limitations: the O mode propagates only when the density is low enough that $\omega_{pe} < \omega$ [17]. The plasma frequency is on the order of the cyclotron frequency in typical tokamak,

$$\frac{\omega_{pe}}{\omega_{ce}} = 3.21 \frac{\sqrt{n_{e,20}}}{B_T} \tag{3.55}$$

where $n_{e,20}$ is the electron density in units of $10^{20}/\mathrm{m}^3$, and B_T is the magnetic field in Tesla.

In this work, we consider ECCD with the $|n|$=2 X-mode, since we are interested in interactions with plasmas having high bootstrap currents, and the condition of low

density necessary for the O-mode will limit the bootstrap current.

Now we continue with the diffusion coefficient (3.52),

$$D_{\mathbf{k}} \approx q_e^2 \left|E_{\mathbf{k},-}\right|^2 \left(\frac{k_\perp v_\perp}{2\Omega}\right)^2 = 2\sqrt{\frac{\mu_0}{\epsilon_0}} p_{\mathbf{k}} \left(N_\perp \frac{p_\perp}{m_e c} \frac{\omega}{2\Omega_0}\right)^2. \tag{3.56}$$

the power flow density in this mode is $p_{\mathbf{k}} = \sqrt{\epsilon_0/\mu_0}\left|E_{\mathbf{k},-}\right|^2/2$.

We consider a RF beam which crosses the flux surface r. The power in the RF beam is assumed to have a Gaussian dependence upon the distance from the center of the beam. Thus the Fourier spectrum is also Gaussian in $k_\parallel$ with a standard deviation $\Delta k_\parallel$. So the diffusion coefficient has the Gaussian form

$$D_{\mathbf{k}} = q_e^2 2\sqrt{\frac{\mu_0}{\epsilon_0}} p_{\mathbf{k},0} \exp\left[-\frac{(k_\parallel - k_{\parallel,0})^2}{(\Delta k_\parallel)^2}\right] \left(N_\perp \frac{p_\perp}{m_e c} \frac{\omega}{2\Omega_0}\right)^2 \tag{3.57}$$

for peak power flow density $p_{\mathbf{k},0}$.

Quasilinear Operator for EC Waves

Taking the integral over $\mathbf{k}$ in (2.50), the operator is

$$Q(f) = \mathcal{L}'\left[p^2(1-\xi^2) D_{ec} \exp\left[-\frac{(k_{\parallel,res} - k_{\parallel,0})^2}{(\Delta k_\parallel)^2}\right] \mathcal{L}(f)\right] \tag{3.58}$$

where

$$k_{\parallel,res} \equiv \frac{\omega - n\Omega}{v_\parallel} = \frac{\gamma - n\Omega_0/\omega}{p_\parallel/m_e c} \frac{\omega}{c} \tag{3.59}$$

and

$$D_{ec} = q_e^2 2\sqrt{\frac{\mu_0}{\epsilon_0}} p_{\mathbf{k},0} \left(N_\perp \frac{p_\perp}{m_e c} \frac{\omega}{2\Omega_0}\right)^2 \frac{1}{\Delta k_\parallel \left|v_\parallel\right|}. \tag{3.60}$$

The linear differential operators are now

$$\begin{aligned} \mathcal{L}(\mathcal{A}) &= \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(1 - \xi^2 - \frac{2\omega_{ce,0}}{\gamma\omega}\right) \frac{1}{p^2 \xi} \frac{\partial \mathcal{A}}{\partial \xi} \\ \mathcal{L}'(\mathcal{A}) &= \frac{1}{p^2}\left\{\frac{\partial}{\partial p}(p\mathcal{A}) + \frac{\partial}{\partial \xi}\left[\left(1 - \xi^2 - \frac{2\omega_{ce,0}}{\gamma\omega}\right)\frac{\mathcal{A}}{\xi}\right]\right\} \end{aligned} \tag{3.61}$$

The local diffusion coefficient D_{ec} is proportional to the power flow density of the RF waves, which is the total power in the beam P divided by the area of the beam which we can call A_b. But since an orbiting electron is only passing through the beam for a small portion of its orbit, when we bounce average the quasilinear operator the effect is to reduce the diffusion coefficient from its local value by a factor of $A_b/(4\pi^2 r R_0)$, where $4\pi^2 r R_0$ is the area of the entire flux surface [79]. Thus the bounce-averaged diffusion coefficient can be calculated by taking the RF power divided by the flux surface area. The result is [80, 79]

$$\begin{aligned} D_{ec} &= D_0 \frac{p_\perp^2}{p_{Te}^2} \\ D_0 &= q_e^2 \frac{2\sqrt{\mu_0/\epsilon_0}\, P}{32\pi^2 r R_0} \left(N_\perp \frac{p_{Te}}{m_e c} \frac{\omega}{2\Omega_0} \right)^2 \frac{1}{\Delta k_\parallel \left| v_\parallel \right|}. \end{aligned} \tag{3.62}$$

Numerical Calculation of ECCD

A numerical calculation of the electron distribution function from the FASTFP code, without bootstrap current, can be used to study the features of electron cyclotron current drive.

This calculation uses parameters consistent with experiments on the DIII-D tokamak, shown in Table C.3 in Appendix C. The current drive is taken at r of about 0.3m, or ϵ of about 0.18, with $2\omega_{ce,0}/\omega = 0.96$, $N_{\parallel,0} = 0.35$, and $\Delta N_\parallel = 0.02$. The diffusion coefficent is taken to have a peak value of $D_0 = 0.4$, which we find to be consistent with a wave power of approximately 5.3 MW from (3.62).

For this case, the current density calculated by the FASTFP code is 1.11 MA/m^2, and there is a wave power density dissipated of 4.75 MW/m^3. In Figure 3-6 we see a contour plot of the distribution function f_0 in $(p_\parallel, p_\perp)$ space. We show the same contour plot in Figure 3-7, this time with another set of dashed contours superimposed to show the value of the diffusion coefficient. This comparision make it easier to see how the EC waves result in a deformation of the distribution function.

In ECCD the current is carried well in the bulk of the electron distribution, in the form of a slight shift from the resting Maxwellian. To observe this, in Figure

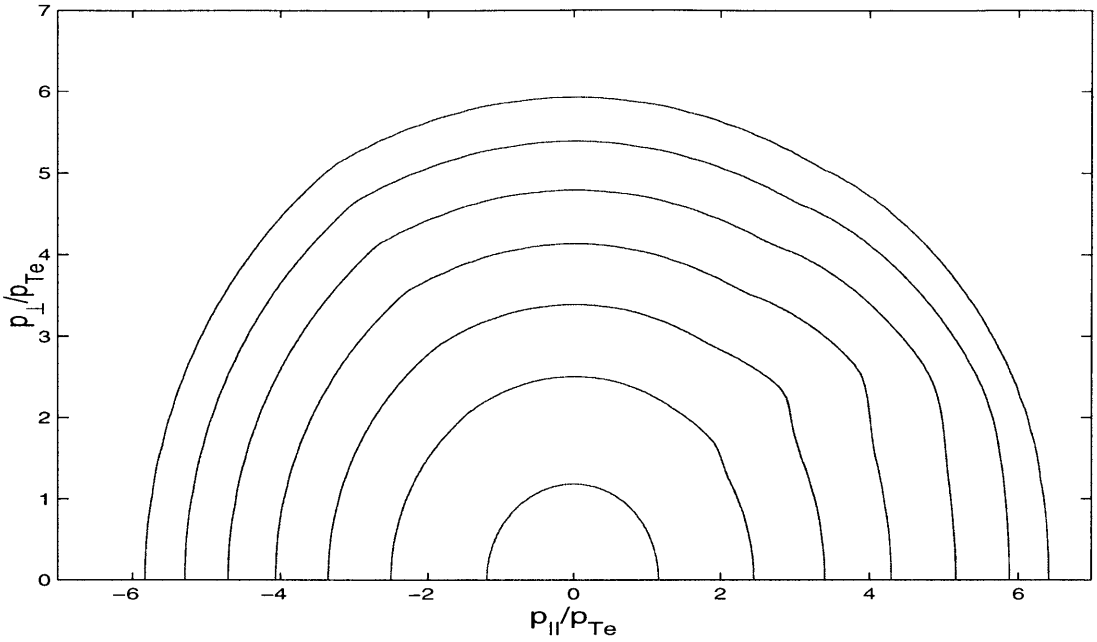

Figure 3-6: Contour plot in $(p_\parallel/p_{Te}, p_\perp/p_{Te})$ space of the EC-modifed electron distribution function, for DIII-D type parameters (Table C.3)with r=0.3 m, $2\Omega_0/\omega = 0.96$, $N_\parallel$=0.35, and D_0=0.4.

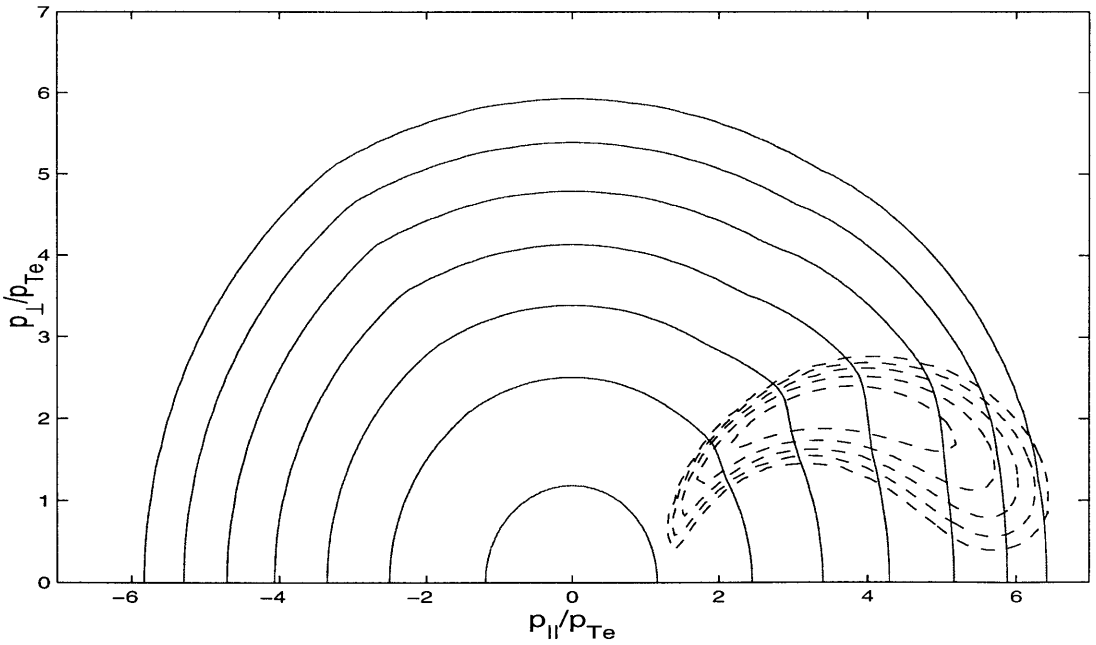

Figure 3-7: Same as Figure 3-6, with the dashed lines indicating contours of constant D_{ec}, with the contours representing D_{ec}=1.0,0.5, 0.25, 0.125, and 0.0625 moving outward from the center.

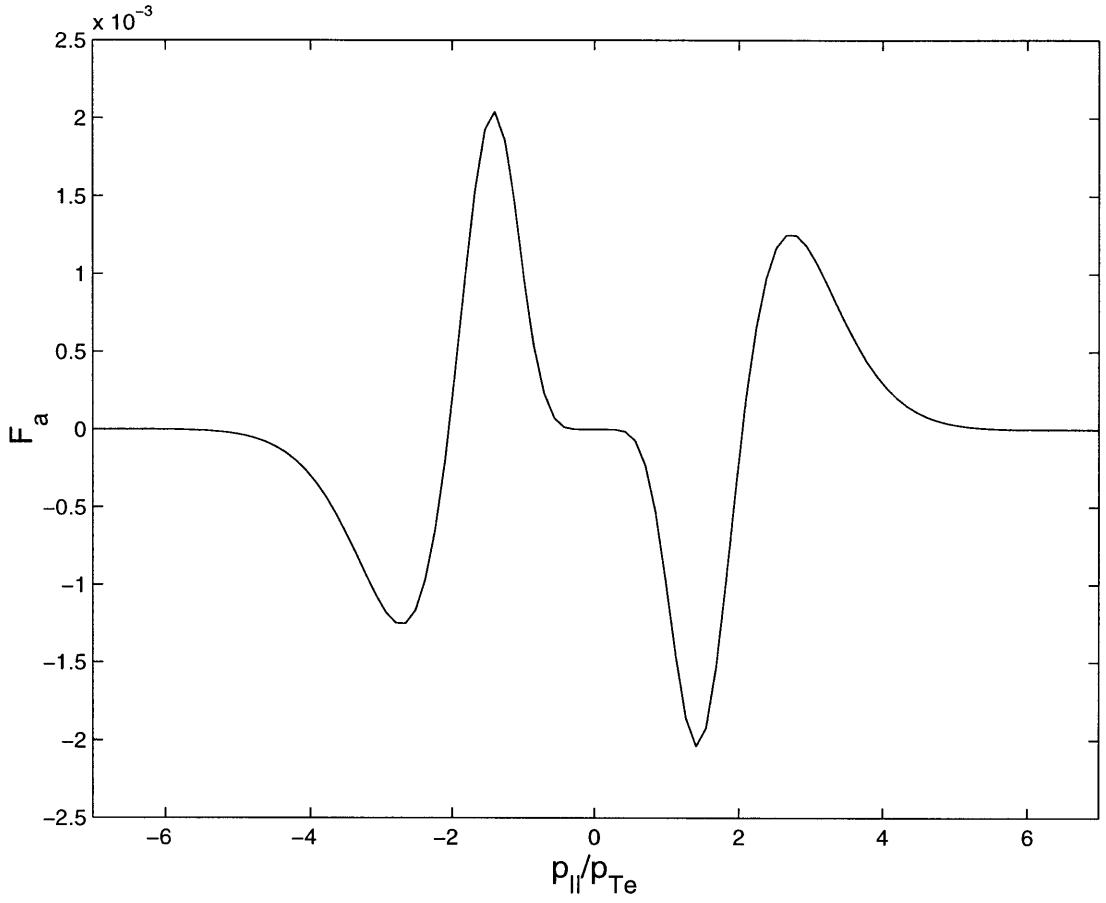

Figure 3-8: Antisymmetric part $[F(p_\parallel) - F(-p_\parallel)]/2$ of the parallel distribution function $F(p_\parallel) = \int 2\pi p_\perp dp_\perp f$ as a function of $p_\parallel/p_{Te}$ for the EC-modified electron distribution function. Parameters are DIII-D type (Table C.3)with r=0.3 m, $2\Omega_0/\omega = 0.96$, $N_\parallel$=0.35, and D_0=0.4.

3-8 we show the antisymmetric (in $p_\parallel$) part of the integrated distribution function $F = \int 2\pi p_\perp^2 \; f$. This antisymmetric part is defined by

$$F_a \equiv \frac{F(p_\parallel) - F(-p_\parallel)}{2}. \tag{3.63}$$

If we take the plasma to be only weakly relativistic, the electron flux density can be found to be

$$\Gamma_{e,\parallel} = \int dp_\parallel \; v_\parallel F_a \tag{3.64}$$

Unlike the visible plateau which was seen in the integrated form of the LH-modified distribution function, there is no clear region which would dominate the integral (3.64).

The value of $(J_\parallel/P_d)$ for this case is 0.23 A·m/W, or 0.61 in the thermal units $q/\nu_e m v_{Te}$. Using the adjoint technique of Antonsen and Chu [74], we can find an approximate value of $(J_\parallel/P_d)$ for ECCD which interacts with electrons having a given $v_\parallel$ which is assumed to be much greater than v_{Te} [39]:

$$\frac{J_\parallel}{P_d} \approx \frac{q_e}{\nu_e m_e v_{Te}} \frac{3}{5 + Z_i} \left(\frac{v_\parallel}{v_{Te}}\right)^2. \tag{3.65}$$

Looking at Figure 3-7, we look for the point at which the EC diffusion affects the maximum number of electrons. The region of diffusion, roughly indicated by the dashed contours, reaches into the distribution function to a value of $p_\parallel$ somewhere between 1.0 p_{Te} and 1.5 p_{Te}. Assuming these electrons are weakly relativistic and with Z_i=1 we use (3.65) to predict $(J_\parallel/P_d)$ between 0.5 and 1.1. The calculated value of 0.61 falls well between these limits.

Chapter 4

The FASTFP-NC Code

4.1 Introduction

To accurately solve the set of equations outlined in Chapter 2 (2.89)-(2.91) which will give us the total non-inductive (RF-driven and bootstrap) current density, numerical methods for solving partial differential equations on computers provide the best approach. In this work, we will use the term Fokker-Planck code to refer to those codes used to solve the kinetic equation for a distribution f with both Fokker-Planck collision and quasilinear diffusion operators. The FASTFP-NC code, written specifically for this work, contains all of the details necessary to solve (2.89)-(2.91) and find the electron distribution function with RF and radial drifts. Appendix D gives a listing of this code.

In this chapter we discuss the features of the FASTFP-NC code and discuss its benchmarking. In Section 4.2 we introduce the FASTFP code, which solves the Fokker-Planck equation (2.89) for current drive only. In Section 4.3 we discuss the modifications which we have implemented so that FASTFP-NC can be used to solve the complete set of equations (2.89)-(2.91). Finally in Section 4.4 benchmark calculations using FASTFP-NC to find the bootstrap current in the absence of RF waves are shown and compared to predicted results.

4.2 The FASTFP Fokker-Planck Code

FASTFP is a Fokker-Planck code designed to solve the Fokker-Planck equation in two-dimensional momentum space [1, 81, 80, 82, 83, 84]. This code was written by Drs. M. Shoucri and I. Shkarofsky of the Centre Canadien de Fusion Magnétique, in Varennes, Quebec, Canada. FASTFP is self-contained and makes use of a simple model of the quasilinear diffusion, which as its name suggests allows for fast computation, in contrast to Fokker-Planck codes which are coupled to transport, MHD, and ray-tracing codes. This simple model is also useful when we take the analytical approach to our calculation of RF and bootstrap currents, since the numerical results can be more easily understood.

FASTFP is a relativistic code, using the linearized relativistic collision operator of Braams and Karney [61], who formulated the collision operator of Beliaev and Budker [63] in terms of scalar potentials. This collision operator is discussed in detail in Appendix A. The code includes the complete first Legendre harmonic term in the electron-electron collision operator, which is required for current drive problems where the conservation of momentum parallel to the magnetic field is essential. All variables in FASTFP are normalized to thermal units:

$$t \to t\nu_e, \qquad v \to v/v_{Te} \qquad p \to p/p_{Te} \tag{4.1}$$

with the thermal speed defined as $v_{Te} = \sqrt{T_e/m_e}$, the thermal momentum as $p_{Te} = m_e v_{Te}$, and the thermal collision frequency defined in (2.47).

Description of Quasilinear Operators

In its original form for a homogeneous plasma FASTFP contains two quasilinear diffusion operators: a lower-hybrid operator of the form (3.27) and an electron cyclotron operator of the form (3.58).

In the banana regime only the bounce-averaged collision and quasilinear operators are needed for solving the kinetic equations (2.89) and (2.91). The bounce-averaging of $C(f_0)$ is shown in detail in Appendix E. The result from (E.10), (E.12), and (E.19)

is

$$\{C(f_0)\} = -\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\left(A(p)\frac{\partial f}{\partial p}+F(p)f\right)\right]$$

$$-\frac{B_t(p)}{p^2}\frac{1}{\lambda}\frac{\partial}{\partial \xi_0}\left[(1-\xi_0^2)\lambda\left(1-\frac{\Delta}{\xi_0^2}\right)\frac{\partial f}{\partial \xi_0}\right]$$

$$+I_1\left(f_M, f_{l=1}\right)\sqrt{\frac{1+\epsilon}{1-\epsilon}}\frac{1}{\lambda}\xi_0\ H(|\xi_0|-\xi_{0T}) \tag{4.2}$$

where

$$\Delta = \left\{1-\frac{1}{\Psi}\right\},$$

for $\Psi = (1+\epsilon)/(1+\epsilon\cos\theta)$, $\xi_{0T} = \sqrt{2\epsilon/(1+\epsilon)}$, and the function H is the Heaviside unit step function

$$H(x) = 1 \text{ for } x > 0, \quad 0 \text{ for } x < 0.$$

The Legendre integral (2.49) is

$$f_{l=1,0} \equiv \frac{3}{2}\int_{-1}^{1} \xi_0 f_0\ d\xi_0.$$

The bounce-averaged quasilinear operators for lower-hybrid current drive and electron cyclotron current drive are calculated in Appendix F. For LH waves, because the resonant electrons are far in the tail of the distribution function $v_{\|} > 3v_{Te}$ the waves interact primarily with those electrons on orbits which are well circulating. So we take $\xi = \xi_0$ in the quasilinear operator (3.27), and the result is given in Appendix F, (F.15) and reproduced here in a slightly different form:

$$\{Q(f)\} = \frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\xi_0 D_{LH}\left(\xi_0\frac{\partial f}{\partial p}+\frac{1-\xi_0^2}{p}\frac{\partial f}{\partial \xi_0}\right)\right]$$
$$+\frac{1}{p}\frac{1}{\lambda}\frac{\partial}{\partial \xi_0}\left[(1-\xi_0^2)\lambda D_{LH}\left(\xi_0\frac{\partial f}{\partial p}+\frac{1-\xi_0^2}{p}\frac{\partial f}{\partial \xi_0}\right)\right] \tag{4.3}$$

$$D_{LH}(v_{\|}) = D_0, \quad v_1 < v_{\|,0} < v_2$$

$$= \quad 0 \qquad v_\parallel < v_1, v_{\parallel,0} > v_2.$$

EC waves are launched in the form of a beam which remains well-collimated as it passes through the plasma, until it damps near the cyclotron resonance. The location where the EC beam crosses the flux surface r is to be taken at a poloidal angle θ_b. Therefore the electrons which are resonant with the waves must have $v_{\parallel,b} = (\omega - n\Omega)/k_\parallel$, where

$$v_{\parallel,b} = v\sigma\sqrt{1 - \Psi_b(1 - \xi_0^2)} \tag{4.4}$$

and

$$\Psi_b \equiv \Psi(\theta_b) = (1 + \epsilon)/(1 + \epsilon\cos\theta_b). \tag{4.5}$$

The result is from Appendix F, (F.27)

$$\begin{aligned}
\{Q(f)\} &= \frac{1}{\lambda}\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^4\Psi_b^2(1-\xi_0^2)^2\frac{\lambda\bar{D}_0}{|v_{\parallel,b}|}\frac{\xi_0}{\xi_b}\left(\frac{\partial f}{\partial p} + \frac{1-\xi_0^2-(n\Omega/\omega\Psi_b)}{p\xi_0}\frac{\partial f}{\partial \xi_0}\right)\right] \\
&\quad + \frac{1}{\lambda}\frac{1}{p^2}\frac{\partial}{\partial \xi_0}\left[p^4\Psi_b^2(1-\xi_0^2)^2\frac{\lambda\bar{D}_0}{|v_{\parallel,b}|}\frac{\xi_0}{\xi_b}\frac{1-\xi_0^2-(n\Omega/\omega\Psi_b)}{p\xi_0}\right. \\
&\qquad \left. \cdot\left(\frac{\partial f}{\partial p} + \frac{1-\xi_0^2-(n\Omega/\omega\Psi_b)}{p\xi_0}\frac{\partial f}{\partial \xi_0}\right)\right] \qquad (4.6) \\
\bar{D}_0 &= \frac{D_0}{\lambda}\exp\left[-\left(\frac{\gamma\omega - n\Omega_0}{p_\parallel/m} - k_{\parallel,0}\right)_b^2/(\Delta k_\parallel)^2\right]
\end{aligned}$$

where the subscript b means that quantities are calculated at the point θ_b where the electron orbit crosses the RF beam.

Solution Method

FASTFP solves the Fokker-Planck equation on a two-dimensional $N_1 \times N_2$ grid in (p, ξ) momentum space. We take

$$p_i = (i-1)\Delta p, \qquad i = 1, ..., N_1, \tag{4.7}$$

$$\xi_j = -1 + (j-2)\Delta\xi, \qquad j = 2, ..., N_2 - 1, \tag{4.8}$$

$$\Delta p = \frac{p_{max}}{(N_1 - 1)}, \qquad \Delta\xi = \frac{2}{(N_2 - 2 - 1)}$$

To enforce the zero-flux boundary condition $\partial f/\partial\xi = 0$ at $\xi = \pm 1$, extra grid points $j = 1$ and $j = N_2$ are added, with $f_{j=1} = f_{j=3}$ and $f_{j=N_2} = f_{j=N_2-2}$. The solution $f(p, \xi)$ is thus represented by the set of values

$$f_{i,j} = f(p_i, \xi_j) \tag{4.9}$$

In discretized form the two-dimensional partial differential equation

$$\{C(f)\} + \{Q(f)\} = S \tag{4.10}$$

becomes a matrix equation on a 9-point stencil, meaning that the matrix form of the linearized, discrete collision and quasilinear operators consists of nine nonzero diagonals, including the main diagonal. To calculate the various partial derivatives in the operators, we need $f_{i,j}$ and its eight "neighbors" $f_{i+1,j}$, $f_{i-1,j}$, $f_{i,j+1}$, $f_{i,j-1}$, $f_{i+1,j+1}$, $f_{i+1,j-1}$, $f_{i-1,j+1}$ and $f_{i-1,j-1}$. Thus the collisional and quasilinear operators can be written in discrete form as

$$\begin{aligned} &a_{i,j}f_{i,j-1} + b_{i,j}f_{i-1,j} + c_{i,j}f_{i,j} + d_{i,j}f_{i+1,j} + e_{i,j}f_{i,j+1} + \\ &am_{i,j}f_{i-1,j-1} + ap_{i,j}f_{i+1,j-1} + em_{i,j}f_{i-1,j+1} + ep_{i,j}f_{i+1,j+1} \end{aligned} \tag{4.11}$$

The matrix elements $a_{i,j}, ..., ep_{i,j}$ are given in Appendix G.

The Fokker-Planck equation is solved using the relaxation method, introducing a time-like parameter and letting the distribution function evolve to a steady-state. The relaxation problem is solved in the form

$$\frac{\mathbf{f}(\tau + \Delta\tau) - \mathbf{f}(\tau)}{\Delta\tau} - \bar{M} \cdot \mathbf{f}(\tau + \Delta\tau) = \mathbf{S} \tag{4.12}$$

where $\mathbf{f}$ is the discretized distribution function is the form of a vector, $\bar{M}$ is the matrix form of the collision and quasilinear operator, $\mathbf{S}$ is the source as a vector, and τ is

the time-like variable, which evolves in steps of $\Delta\tau$. This equation is in fully implicit form [85], meaning that the diffusion operator $\bar{M}\cdot\mathbf{f}$ is evaluated at time $\tau+\Delta\tau$ rather than at τ, which would give a numerically-unstable solution [86].

Rewriting the matrix equation as

$$\left(\frac{1}{\Delta\tau}\hat{I}-\bar{M}\right)\cdot\mathbf{f}(\tau+\Delta\tau)=\frac{\mathbf{f}(\tau)}{\Delta\tau}, \tag{4.13}$$

the calculation of $\mathbf{f}(\tau+\Delta\tau)$ requires the inversion of a matrix with nine diagonals, which FASTFP performs using the strongly implicit method [87], using the subroutine SIP9D [83] written by Y. Peysson specifically for use in the FASTFP code.

4.3 Numerical Calculation of the Bootstrap Current – The FASTFP-NC Code

To perform current drive calculations in which the bootstrap current is also calculated, we have written the FASTFP-NC code to solve the set of equations (2.89)-(2.91).

The Fokker-Planck solver from FASTFP is incorporated in this work, to solve the equations (2.89) and (2.91). There are three major tasks which FASTFP-NC has to perform in addition to the FASTFP solutions. These are described in this section.

First, the function $\tilde{f}$ must be calculated from (2.90), using some model of the radial profiles of density and temperature. Secondly, the implicit poloidal dependence of the first-order component $\tilde{f}$ must be treated accurately when FASTFP-NC calculates $C(\tilde{f})$ and $Q(\tilde{f})$ for inclusion in the source term of (2.91). Thirdly, the solution g of (2.91) must be found.

4.3.1 Radial Derivative

The radial derivative $\partial f_0/\partial r$ is calculated numerically using a finite differencing in r,

$$\frac{\partial(f_0)_{i,j}}{\partial r}=\frac{(f_0)_{i,j}(r+\Delta r/2)-(f_0)_{i,j}(r-\Delta r/2)}{\Delta r} \tag{4.14}$$

This entails that the single-flux-surface FASTFP solver must be run three times – not only to find $(f_0)_{i,j}$ on our chosen flux surface r, but also on two nearby flux surfaces $r + \Delta r/2$ and $r - \Delta r/2$. Each surface has a different, specified density and temperature.

As shown in Section 2.2.2, the extent of the electron orbit in the r direction is small (since $|v_{Dr}| \ll v_{Te}$), which means that we can treat the radial dimension by including a simple model for the local radial derivative of the plasma parameters n and T. Calculations of MHD equilibrium and experimental observations give models for the density and temperature profiles of the plasma [88], and perpendicular transport effects which can modify the profiles are assumed to occur on time scales much longer than the scales of interest.

The diffusion coefficients for both lower hybrid and electron cyclotron current drive are assumed to be the same on nearby flux surfaces; $\Delta r \ll a$ will be taken as sufficently small.

4.3.2 Poloidal Dependence of $\tilde{f}$

We treat the poloidal (θ) dependence of $\tilde{f}$ analytically. Recalling that

$$\tilde{f} = -\frac{v_\parallel}{\Omega_\theta}\frac{\partial f_0}{\partial r} \tag{4.15}$$

this poloidal dependence is implicit in the poloidal magnetic field contained in $\Omega_\theta = q_e B_\theta/\gamma m_e$ and the variation of $v_\parallel$ along the electron orbit. Specifically, we use

$$B_\theta = \frac{1}{R}\frac{d\psi}{dr} \equiv \frac{B_{\theta,min}(r)}{1+\epsilon\cos\theta} \tag{4.16}$$

where $B_{\theta,min}$ is the minimum value of the poloidal magnetic field on a flux surface, and since $v_\parallel = (p/\gamma)\sqrt{1+\Psi(1-\xi_0^2)} = (p\xi_0/\gamma)(\xi/\xi_0)$, we can rewrite (4.15) as

$$\tilde{f} = -\left(\frac{\xi}{\Psi\xi_0}\right)\frac{p\xi_0}{qB_{\theta,min}}\frac{\partial f_0}{\partial r} \equiv \left(\frac{\xi}{\Psi\xi_0}\right)\tilde{f}' \tag{4.17}$$

The θ dependence is now confined to the factor in parentheses in (4.17), and we define

$$\tilde{f}' \equiv \tilde{f}(\theta = 0) = -\frac{p\xi_0}{qB_{\theta,min}} \frac{\partial f_0}{\partial r}. \tag{4.18}$$

The bounce average of $\tilde{f}$ is (E.25) in Appendix E,

$$\begin{aligned} \left\{\tilde{f}\right\} &= \frac{1}{\lambda}\left[\frac{1}{2}\sum_{\sigma}\right]_T \int_{-\theta_c}^{\theta_c} \frac{d\theta}{2\pi} \frac{\xi_0}{\xi} \left(\frac{\xi}{\Psi\xi_0}\right) \tilde{f}' \\ &= H(|\xi_0| - \xi_{0T}) \frac{1}{\lambda(1+\epsilon)} \tilde{f}' \end{aligned} \tag{4.19}$$

which is zero for trapped orbits, since $\tilde{f}'$ is antisymmetric in σ on trapped electrons.

The source term $\left\{C(\tilde{f})\right\}$ is calculated from the collision operator (2.46). The details are given in Appendix E and the result from (E.26), (E.33-E.35), and (E.43) is

$$\begin{aligned} \left\{C(\tilde{f})\right\} &= -\frac{1}{\lambda(1+\epsilon)} H(|\xi_0| - \xi_{0T}) \frac{1}{p^2} \frac{\partial}{\partial p} \left[p^2 \left(A(p) \frac{\partial \tilde{f}'}{\partial p} + F(p)\tilde{f}'\right)\right] \\ &\quad - \frac{1}{\lambda(1+\epsilon)} H(|\xi_0| - \xi_{0T}) \frac{B_t(p)}{p^2} \frac{\partial}{\partial \xi_0} \left[(1-\xi_0^2)\left(\frac{\partial \tilde{f}'}{\partial \xi_0} - \frac{\epsilon - \epsilon^2/2}{(1+\epsilon)} \frac{1}{\xi_0} \frac{\partial(\tilde{f}'/\xi_0)}{\partial \xi_0}\right)\right] \\ &\quad + I_1(f_M, f_{l=1}) \frac{1 - \{\Psi\}(1-\xi_0^2)}{\xi_0} H(|\xi_0| - \xi_{0T}) \end{aligned} \tag{4.20}$$

where again the Legendre integral is taken at $\theta = 0$,

$$f_{l=1} \equiv \frac{3}{2} \int_{-1}^{1} \xi_0 \tilde{f}' \, d\xi_0.$$

The assumption of localized RF deposition allows us find $\{Q(\tilde{f})\}$ by taking the value of $\tilde{f}$ at the angle of RF deposition θ_b. This means that it is sufficient to use the quasilinear operator with the distribution

$$\tilde{f}_b = \frac{\xi_b}{\Psi(r, \theta_b)\xi_0} \tilde{f}'. \tag{4.21}$$

The electron cyclotron operator from (4.6) for this case is

$$\begin{aligned}\{Q(\tilde{f})\} &= \frac{1}{\lambda}\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^4\Psi_b^2(1-\xi_0^2)^2\frac{\lambda\bar{D}_0}{|v_{\|,b}|}\left(\frac{\partial\tilde{f}_b}{\partial p}+\frac{1-\xi_0^2-(n\Omega/\omega\Psi_b)}{p\xi_0}\frac{\partial\tilde{f}_b}{\partial\xi_0}\right)\right] \\ &+\frac{1}{\lambda}\frac{1}{p^2}\frac{\partial}{\partial\xi_0}\left[p^4\Psi_b^2(1-\xi_0^2)^2\frac{\lambda\bar{D}_0}{|v_{\|,b}|}\frac{1-\xi_0^2-(n\Omega/\omega\Psi_b)}{p\xi_0}\right. \\ &\left.\cdot\left(\frac{\partial\tilde{f}_b}{\partial p}+\frac{1-\xi_0^2-(n\Omega/\omega\Psi_b)}{p\xi_0}\frac{\partial\tilde{f}_b}{\partial\xi_0}\right)\right]. \qquad (4.22)\end{aligned}$$

Lower hybrid waves are also taken to be in resonance primarily circulating electrons, and therefore we can approximate $\xi \approx \xi_0$ in the bounce averaging process as explained in Appendix F, Section F.2. This allows us to take $\tilde{f} \approx \tilde{f}'$ and from (4.3)

$$\begin{aligned}\{Q(\tilde{f})\} &= \frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\xi_0 D_{LH}\left(\xi_0\frac{\partial\tilde{f}'}{\partial p}+\frac{1-\xi_0^2}{p}\frac{\partial\tilde{f}'}{\partial\xi_0}\right)\right] \\ &+\frac{1}{p}\frac{\partial}{\partial\xi_0}\left[(1-\xi_0^2)D_{LH}\left(\xi_0\frac{\partial\tilde{f}'}{\partial p}+\frac{1-\xi_0^2}{p}\frac{\partial\tilde{f}'}{\partial\xi_0}\right)\right] \qquad (4.23)\end{aligned}$$

4.3.3 Particular Solution of g

In Section 3.2 we showed that considerations of the symmetry of g show that it must satisfy the boundary condition g=0 for trapped orbits. However, it is important to consider the possibility that this solution is not necessarily the solution when quasilinear diffusion is included.

A very important point about $\{Q(f)\}$ is brought to light by the test particle approach to calculating the quasilinear operator [79, 89, 90]. In this analysis, the quasilinear diffusion coefficient is found by taking the linear wave-particle interaction during a single pass of the electron through the RF beam, and averaging over multiple passes. The phase of the wave relative to the electron motion is taken to be uncorrelated and the result is a random walk in momentum space. Thus quasilinear diffusion is a process which takes place only after a large number of electron orbits in θ. Since each trapped electron orbit contains motion with positive and negative $v_\|$, and $\tau_b \ll \tau_{QL}$, the quasilinear operator is essentially symmetric in σ for trapped

particles.

The zero-order distribution f_0 must be symmetric in σ for trapped electrons since it is independent of θ The distribution $\tilde{f}$ will therefore be antisymmetric in σ for trapped electrons, since it contains the symmetric $\partial f_0/\partial r$ and an additional factor of $v_\parallel$.

The symmetry of Q combined with the antisymmetry of $\tilde{f}$ in the trapped region means that the bounce-averaged quasilinear operator $\{Q(\tilde{f})\}$ must be zero for trapped orbits. Since the collision operator is also symmetric for all orbits, the same argument applies to $\{C(\tilde{f})\}$. Therefore we have a very important result: **the source term**

$$S = \left\{C(\tilde{f})\right\} + \left\{Q(\tilde{f})\right\}$$

will be zero for trapped particles. For this reason, there exists a solution to the Fokker-Planck equation for g having g=0 in the trapped region of momentum space:

$$\{C(g)\} + \{Q(g)\} = S. \tag{4.24}$$

However, the question we must consider is whether this particular g, which we can call g_p, is the correct solution to the current drive problem.

Now, consider the case where f_0 is a Maxwellian and therefore independent of $\xi = p_\parallel/p$. In this case, we showed in Section 3.2 that both $\tilde{f}$ and g are antisymmetric in ξ for all orbits. This means that while $\tilde{f}$ and g contribute in a critical way to calculations of parallel particle flow density

$$\mathbf{\Gamma}_\parallel = \int d^3p\, v\xi\, (f_0 + \tilde{f} + g),$$

they do not contribute at all to the calculations of particle density, in other words,

$$\int d^3p\, (\tilde{f} + g) = 0.$$

However, when RF waves create a distorted f_0, the distribution functions f_0 and

$\tilde{f}$ do not have any clear symmetry properties outside of the trapped region. The zero-order component f_0 will contain components both symmetric and antisymmetric in ξ, with the latter particularly imporant in RF current drive. After solving the Fokker-Planck equation

$$\{C(f_0)\} + \{Q(f_0)\} = 0 \tag{4.25}$$

the distribution f_0 is initially normalized to have density $n(r)$ for electrons on the flux surface r,

$$n(r) = \int d^3p f_0. \tag{4.26}$$

As stated, $\tilde{f}$ is calculated directly from the radial derivative of f_0. The FASTFP-NC code can then be used to find the particular solution g_p of the equation (4.24) having $g_p = 0$ in the trapped region. However, we note that f_0 is the homogeneous solution to this partial differential equation for g, i.e. if

$$\{C(g_p)\} + \{Q(g_p)\} = S \tag{4.27}$$

then

$$\{C(g_p + cf_0)\} + \{Q(g_p + cf_0)\} = S \tag{4.28}$$

for any constant c.

Since $\tilde{f}$ and g_p have no definite symmetry properities, they can contribute to the density. So the constant c is chosen specifically to keep the density accurate (equal to $n(r)$) when $g = g_p + cf_0$,

$$n(r) = \int d^3p(f_0 + \tilde{f} + g_p + cf_0) \tag{4.29}$$

$$c = -\frac{1}{n(r)} \int d^3p(\tilde{f} + g_p). \tag{4.30}$$

From this condition the correct solution g is found to be g_p, calculated using the Fokker-Planck solver, plus cf_0, with c found as above.

4.4 Benchmarking Calculations of FASTFP and FASTFP-NC

4.4.1 Benchmarking RF Current Drive Calculations of FASTFP

The FASTFP code has been used benchmarked extensively by its authors, for current drive problems in tokamaks including current drive by lower hybrid and electron cyclotron waves [1, 81, 80].

Although a full benchmarking of the RFCD-only calculations of FASTFP will not be shown here, we refer to the examples of LHCD in Section 3.3.1 and of ECCD in Section 3.3.2 which show that the code gives good agreement with the predictions of $(J_{\|}/P_d)$ from the adjoint technique.

4.4.2 Benchmark Calculations of Bootstrap Current with FASTFP-NC

As a check of the capabilities of FASTFP-NC, it was essential to show that it is capable of calculating the bootstrap current in cases where there are no RF waves. We consider cases where f_0 is a Maxwellian, which reproduces the neoclassical theory already described [7, 9], and the results of the code are compared to predictions for the bootstrap current.

The numerical calculation of LHCD in Section 3.3.1 and the calculation of ECCD in Section 3.3.2 are provided as benchmark calculations which show good comparision to theory.

Predicted Results for Bootstrap Current

The most complete form of neoclassical transport calculation is the calculation of Hirshman and Sigmar [9] in which the plasma fluid and heat flows are found in terms of the thermodynamic driving terms from a set of viscosity and friction coefficients. Simpler expressions for the bootstrap current can be found in the large aspect ratio limit ($\epsilon \ll 1$) and in the banana regime ordering of low collisionality $\nu_* \ll 1$.

As stated in Section 3.2 the general form of the expression for the bootstrap current is

$$\left\langle J_{\parallel,BS} B \right\rangle = -p_e \frac{B_\zeta}{B_\theta} \left(L_{31} A_1 + L_{32} A_2 + L_{34} A_4 \right) \tag{4.31}$$

where

$$\begin{aligned} A_1 &= \frac{d \ln p_e}{dr} + \frac{T_i}{Z_i T_e} \frac{d \ln p_i}{dr} \\ A_2 &= \frac{d \ln T_e}{dr} \\ A_4 &= \alpha_i \frac{T_i}{Z_i T_e} \frac{d \ln T_i}{dr} \end{aligned} \tag{4.32}$$

where the transport coefficients L_{31}, L_{32}, and L_{34}, and the factor α_i must be calculated.

A good overview of the merits of the various models used to derive the bootstrap current is from Kessel [10]. Hirshman [91] performed a calculation in the limit of low collisionality ($\nu_* \to 0$) which gives

$$\begin{aligned} L_{31} &= x[0.754 + 2.21 Z_i + Z_i^2 \\ &\quad + x(0.348 + 1.243 Z_i + Z_i^2)]/D(x) && (4.33) \\ L_{32} &= -x[(0.884 + 2.074 Z_i)/D(x)] && (4.34) \\ L_{34} &= L_{31} && (4.35) \\ \alpha_i &= -1.172/(1 + 0.462 Z_i), && (4.36) \end{aligned}$$

where

$$\begin{aligned} D(x) &= 1.414 Z_i + Z_i^2 + x(0.754 + 2.657 Z_i + 2 Z_i^2) \\ &\quad + x^2(0.348 + 1.243 Z_i + Z_i^2) \end{aligned} \tag{4.37}$$

and

$$x \approx (1.46 \epsilon^{1/2} + 2.40 \epsilon)/(1 - \epsilon)^{1.5} \tag{4.38}$$

in the limit $\epsilon \ll 1$. Sauter, Angioni, and Lin-Liu [92] have performed another more recent calculation which suggests a more accurate formula for the coefficient L_{32}; in the low collisionality limit they find

$$L_{32} = -j_0 \left[\frac{0.51 + 1.31 Z_i}{Z_i(1 + 0.44 Z_i)} (X - X^4) + \frac{5.95 + 3.57 Z_i}{1 + 2.70 Z_i + 0.546 Z_i^2} (X^2 - X^4) - \frac{3.92 + 3.57 Z_i}{1 + 2.70 Z_i + 0.546 Z_i^2} (X^3 - X^4) \right] \tag{4.39}$$

where

$$X = 1 - \frac{(1-\epsilon)^2}{\sqrt{1-\epsilon^2}(1 + 1.46\sqrt{\epsilon})} \tag{4.40}$$

is the trapped particle fraction in the limit $\epsilon \ll 1$.

Comparisions of Bootstrap Current Calculations and Predictions

As a first test of the FASTFP-NC code, we perform calculations of the bootstrap current in the Lorentz gas limit, by giving the code an input of Z_i=12 for the ion charge. The ions are assumed to be cold (T_i=0). All the other parameters used by the code are the Alcator C-Mod type parameters in Table C.1 in Appendix C. The bootstrap current can be calculated analytically in the Lorentz limit. Figure 4-1 shows the bootstrap current as a function of minor radius normalized to a. The comparision of the FASTFP-NC results (indicated by the squares) to the known results (the dashed line) is very good except at particularly low values of r, which we will discuss later in this section.

The next comparision is for a plasma with a density gradient only with a uniform electron temperature. The parameters are again taken from Table C.1 but the temperature profile is replaced by a constant, and the density profile is changed to keep the pressure profile the same. (This is of course an unphysical plasma profile, but we use it for this illustrative case.) The ion temperature is again set to zero. In this limit, we get from 4.31

$$\left\langle J_{\parallel,BS} B \right\rangle = -p_e \frac{B_\zeta}{B_\theta} L_{31} \frac{d \ln p_e}{dr} \tag{4.41}$$

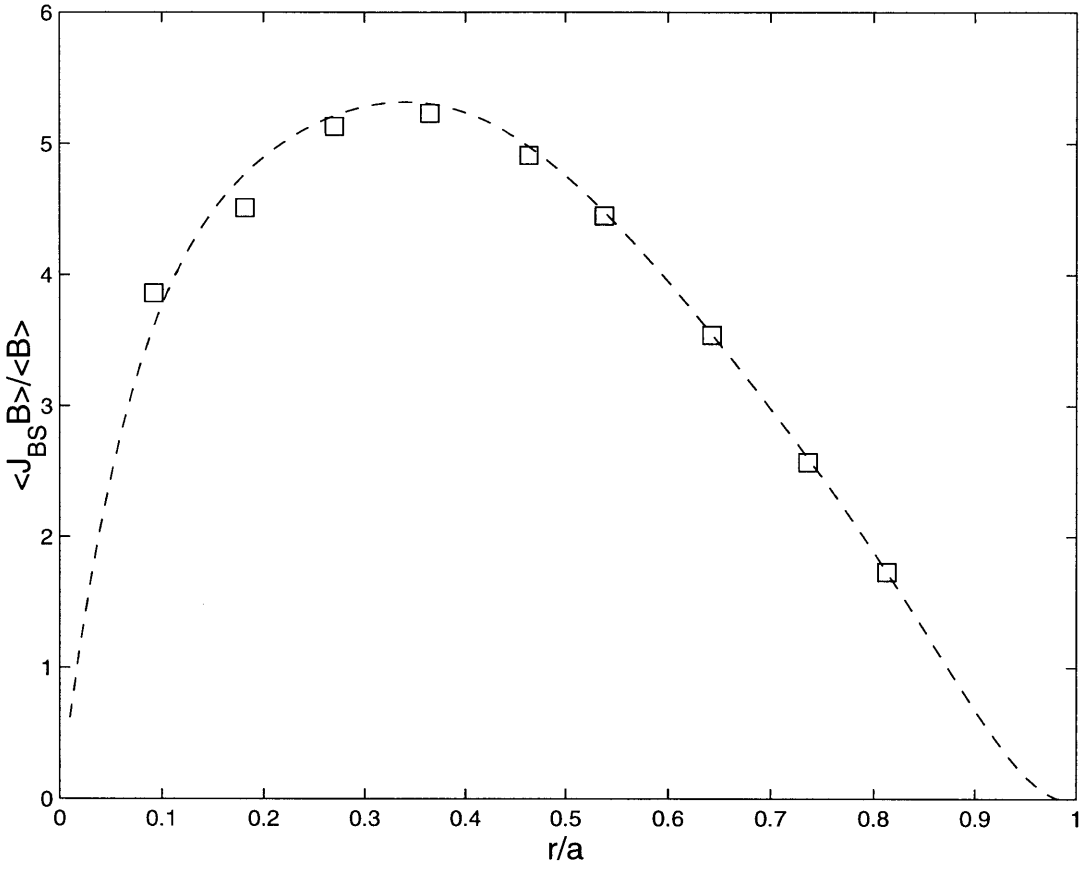

Figure 4-1: Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Z_i=12, T_i=0, and all other parameters of Alcator C-Mod type (Table C.1). Squares are calculations of FASTFP-NC; dashed line is actual value [91].

where L_{31} is predicted by Hirshman [91] in (4.33). Figure 4-2 shows good agreement

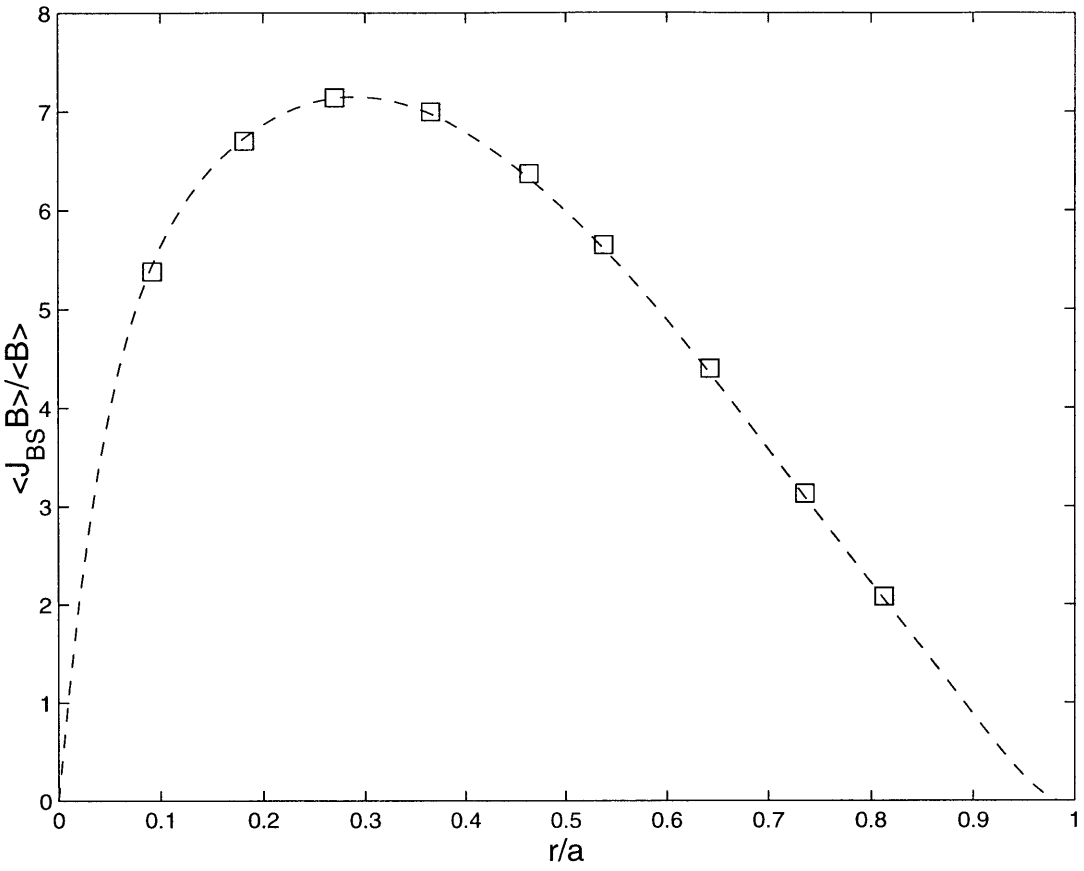

Figure 4-2: Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Alcator C-Mod type parameters (Table C.1) with T_i and dT_e/dr set to zero. Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91].

the of FASTFP-NC calculations (squares) of the bootstrap current to the predictions (dashed line). In Figure 4-3, the transport coefficient L_{31} (divided by $\sqrt{\epsilon}$ to remove a predicted $\sqrt{\epsilon}$ dependence) is displayed, as calculated by FASTFP-NC (squares) and from (4.33).

The next test case uses the full density and temperature profiles of Table C.1, still keeping T_i zero. In this case the bootstrap current (4.31) is

$$\left\langle J_{\parallel,BS} B \right\rangle = -p_e \frac{B_\zeta}{B_\theta} \left(L_{31} \frac{d \ln p_e}{dr} + L_{32} \frac{d \ln T_e}{dr} \right). \tag{4.42}$$

FASTFP-NC calculations of the bootstrap current in Figure 4-4 are again marked

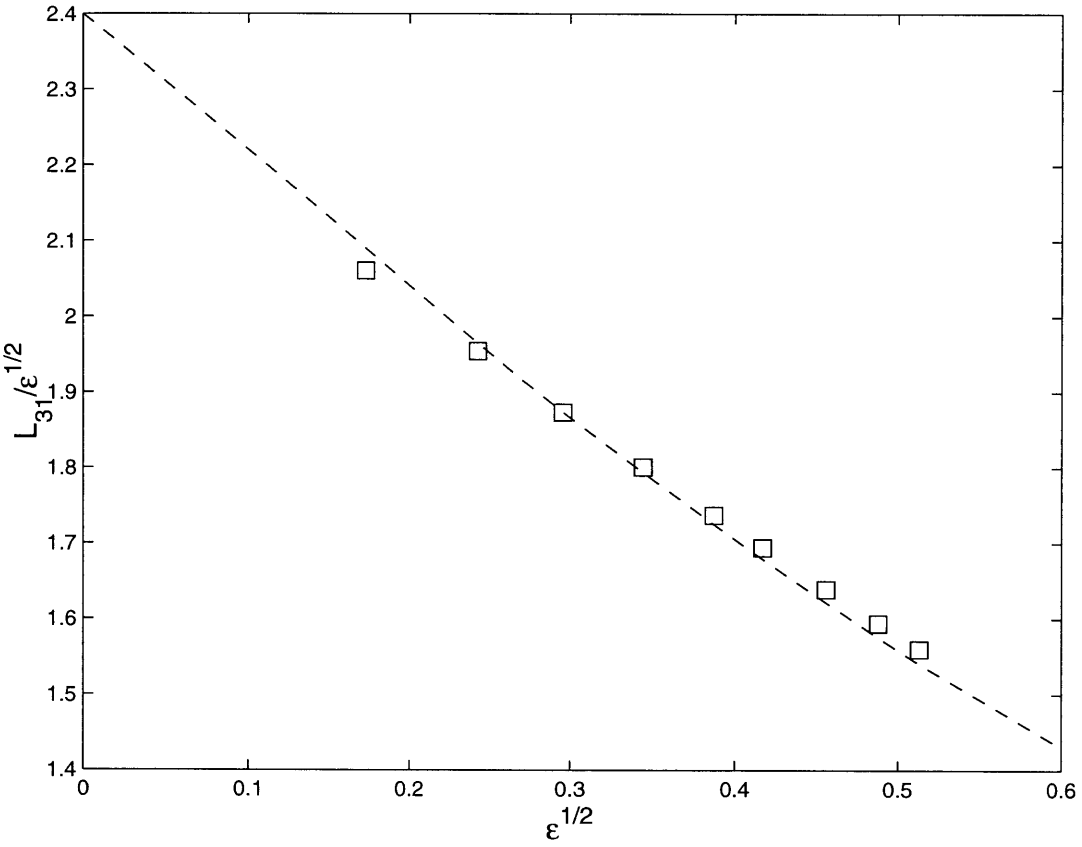

Figure 4-3: Transport coefficient L_{31} as a function of $\epsilon^{1/2}$ calculated by FASTFP-NC (squares) and from Reference [91] (dashed line).

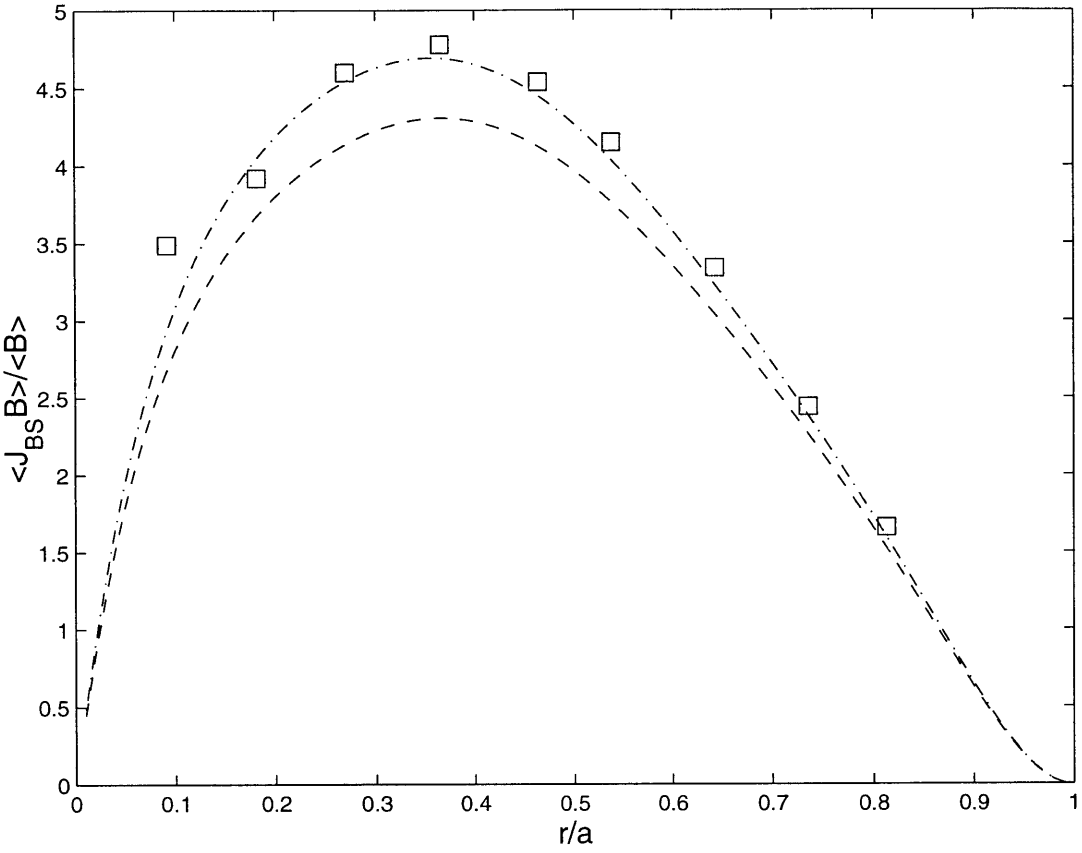

Figure 4-4: Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Alcator C-Mod type parameters (Table C.1) and T_i=0. Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92].

with a square. The dashed line is the prediction of Hirshman [91], while the dash-dot line uses the value of L_{32} given by Sauter, et. al. [92] in (4.39). We note that the FASTFP-NC results agree well with the dash-dot curve which indicates that it predicts the same value of L_{32} as Sauter, et.al. This is demonstrated in Figure 4-5, which shows $L_{32}/\sqrt{\epsilon}$ found in the same three ways. We note that the point at the lowest value of ϵ does not agree with the prediction; we will discuss inaccuracies at low ϵ below.

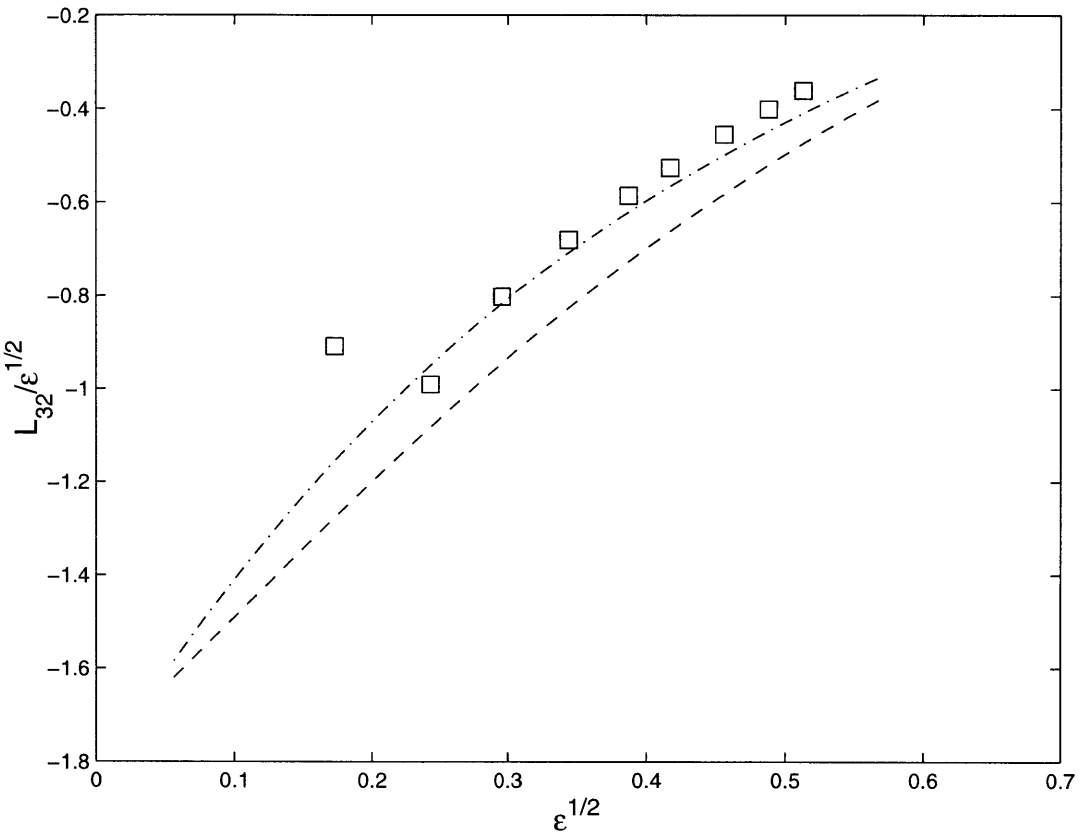

Figure 4-5: Transport coefficient L_{32} as a function of $\epsilon^{1/2}$ calculated by FASTFP-NC (squares) and from Reference [91] (dashed line) and Reference [92] (dash-dot line).

Finally, Figure 4-6 shows the full bootstrap current calculation of FASTFP-NC for the Alcator C-Mod type parameters with a finite ion temperature included. The FASTFP-NC code includes the neoclassical parallel flux of ions in the full calculation of the bootstrap current. The code uses the known value of the ion flux from Hirshman and Sigmar's work [9] and corrects the electron-ion collision operator to include this

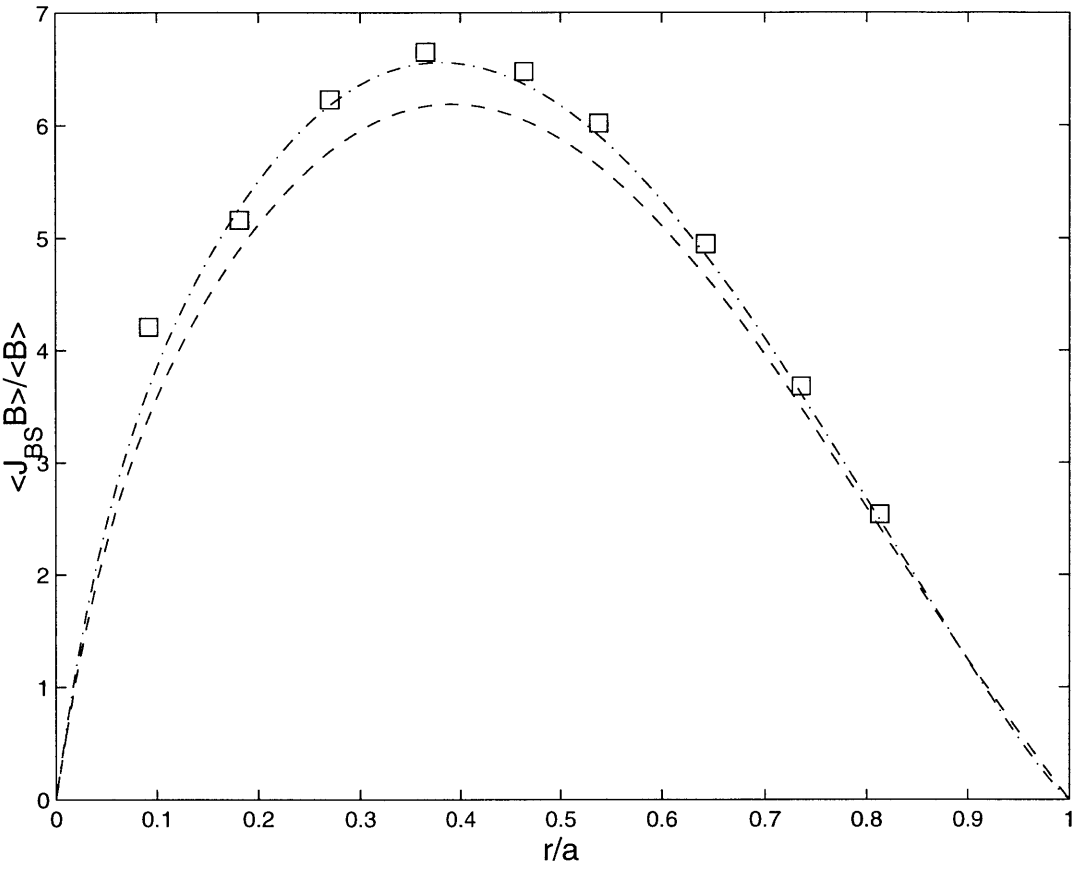

Figure 4-6: Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for Alcator C-Mod type parameters (Table C.1). Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92].

effect. As before, the squares are the FASTFP-NC calculation, the dashed line is the predictions of [91], and the dash-dot line is the prediction of [92]. To show the generality of the FASTFP-NC calculations, the same values of the bootstrap current are shown in Figures 4-7 and 4-8 for DIII-D and ARIES-RS type parameters respectively (see Tables C.3 and C.2).

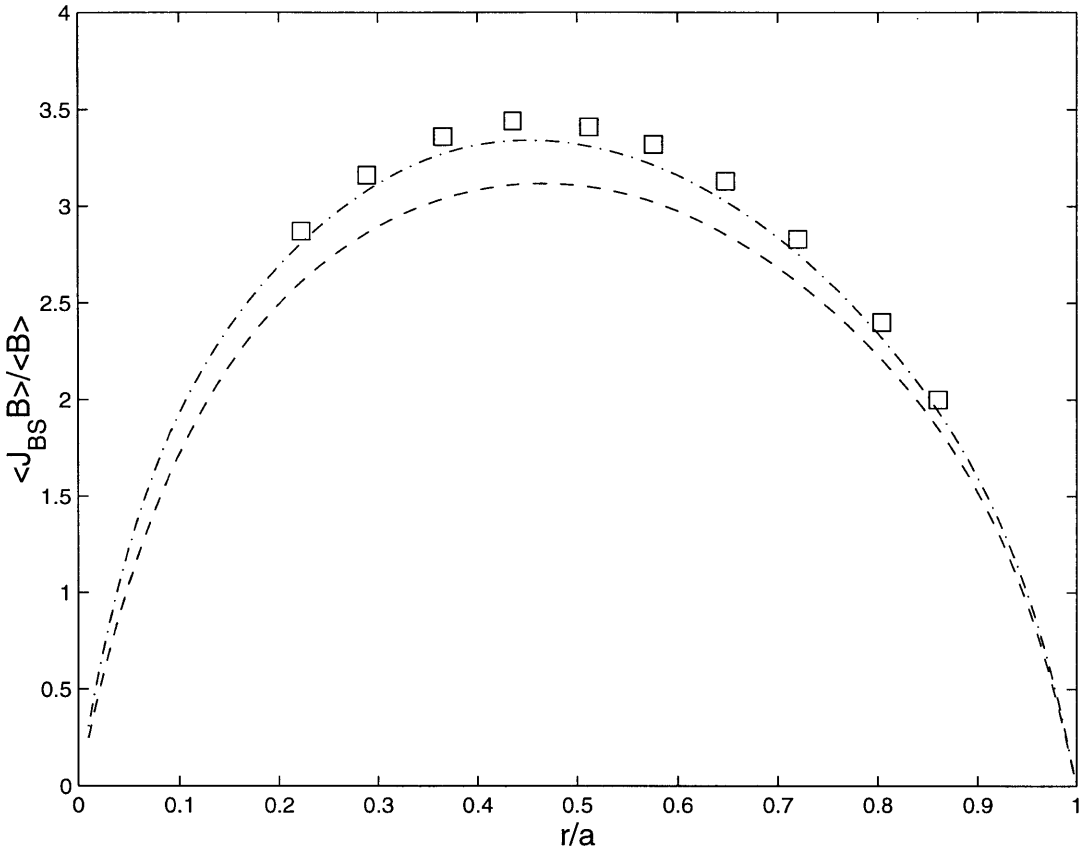

Figure 4-7: Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for AIRES-RS type parameters (Table C.2). Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92].

We have noted that the FASTFP-NC calculations do become less accurate at low values of r/a in many of the cases shown. This limitation appears to result from difficulties in accurately calculating the radial derivative $\partial f_0/\partial r$ over a discrete Δr when the gradients of density and temperature become small, as they do near the center of a tokamak. However, since the bootstrap current goes to zero near the

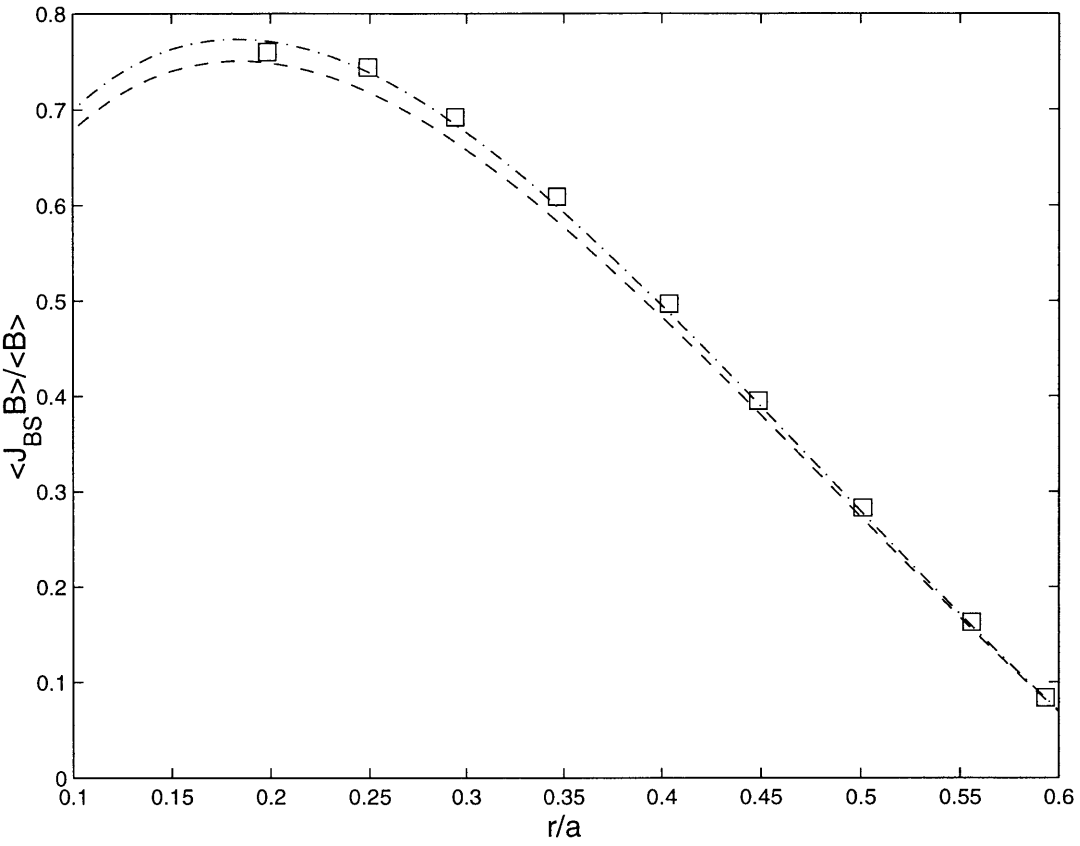

Figure 4-8: Bootstrap current $\left\langle J_{\parallel,B} B \right\rangle / \langle B \rangle$ as a function of r/a for DIII-D type parameters (Table C.3). Squares are calculations of FASTFP-NC; dashed line is predicted value from Reference [91] and dash-dot line is predicted value from Reference [92].

center of the tokamak, on-axis calculations of the current are not important for this work.

In the following chapters, FASTFP-NC will be used to calculate RF driven current and the bootstrap current together, using the complete set of equations (2.89)-(2.91).

Chapter 5

Lower Hybrid Current Drive Cases with Neoclassical Transport Effects Including Bootstrap Current

5.1 Introduction

We study the problem of lower hybrid current drive with the bootstrap current, using numerical results from the FASTFP-NC code. These current density calculations indicate the presence of lower hybrid-bootstrap synergism, which may have a simple analytic explanation.

In Section 5.2 one case of LHCD is calculated by FASTFP-NC and compared to the two baseline cases of LHCD and bootstrap current separately, and we find that the result leads to a synergism. This result leads us to an analytic picture of the effect of the bootstrap current on LHCD, which is presented in Section 5.3. Further calculations are then presented in Section 5.4 which show LH-bootstrap synergism, and these are compared to the predictions of the analytic model. We discuss the meaning of these results in Section 5.5.

5.2 FASTFP-NC Calculation of LHCD and Bootstrap Current

We now calculate the full distribution function $f_0 + \tilde{f} + g$ for electrons using the FASTFP-NC code described in Chapter 4. The case considered is the same LHCD case considered in Section 3.3.1: we take Alcator C-Mod type parameters from Table C.1 in Appendix C, with r equal to 0.15m, so that $\epsilon = 0.23$. We take the diffusion coefficient to be D_0=4.0 in the range $v_\parallel \in (v_1, v_2)$, where $v_1 = 3.5 v_{Te}$ and $v_2 = 6.0 v_{Te}$.

The first task the FASTFP-NC code performs is to calculate numerically the bootstrap current density on this flux surface, for the case where RF waves are not present and the local distribution function is an unperturbed Maxwellian. This is essential as a baseline for the calculations which follow. We denote the bootstrap current density calculated without RF waves as $J_\parallel^{(B)}$, and its flux-surface averaged value is found to be $\left\langle J_\parallel^{(B)} B \right\rangle / \langle B \rangle = 2.9$ MA/m^2. In Figure 5-1 we show the first order (in δ) distribution function parts $\tilde{f}$ and g for this case. We give these functions the name $\tilde{f}^{(B)}$ and $g^{(B)}$, where the superscript (B) denotes the distribution function for the case of no RF waves. These functions will be compared to the functions $\tilde{f}$ and g calculated with RF wave quasilinear diffusion included.

The next step of the FASTFP-NC calculation is to use the FASTFP portion of the code to find f_0 from the zero-order part of the DKE, $\{C(f_0)\} + \{Q(f_0)\} = 0$. This is the same as the LHCD-only case shown in Section 3.3.1, and f_0 is shown in Figures 3-3 and 3-4. The current density found for RF waves without the bootstrap current provides another baseline, and we denote it as $J_\parallel^{(R)}$:

$$J_\parallel^{(R)} = q_e \int d^3p \; v_\parallel f_0. \tag{5.1}$$

In Section 3.3.1 we gave the result for this calculation as 10.6 MA/m^2.

Finally, by calculating

$$\tilde{f} = -\frac{v_\parallel}{\Omega_\theta} \frac{\partial f_0}{\partial r} \tag{5.2}$$

and using the Fokker-Planck solver to find g we can find the total non-inductive

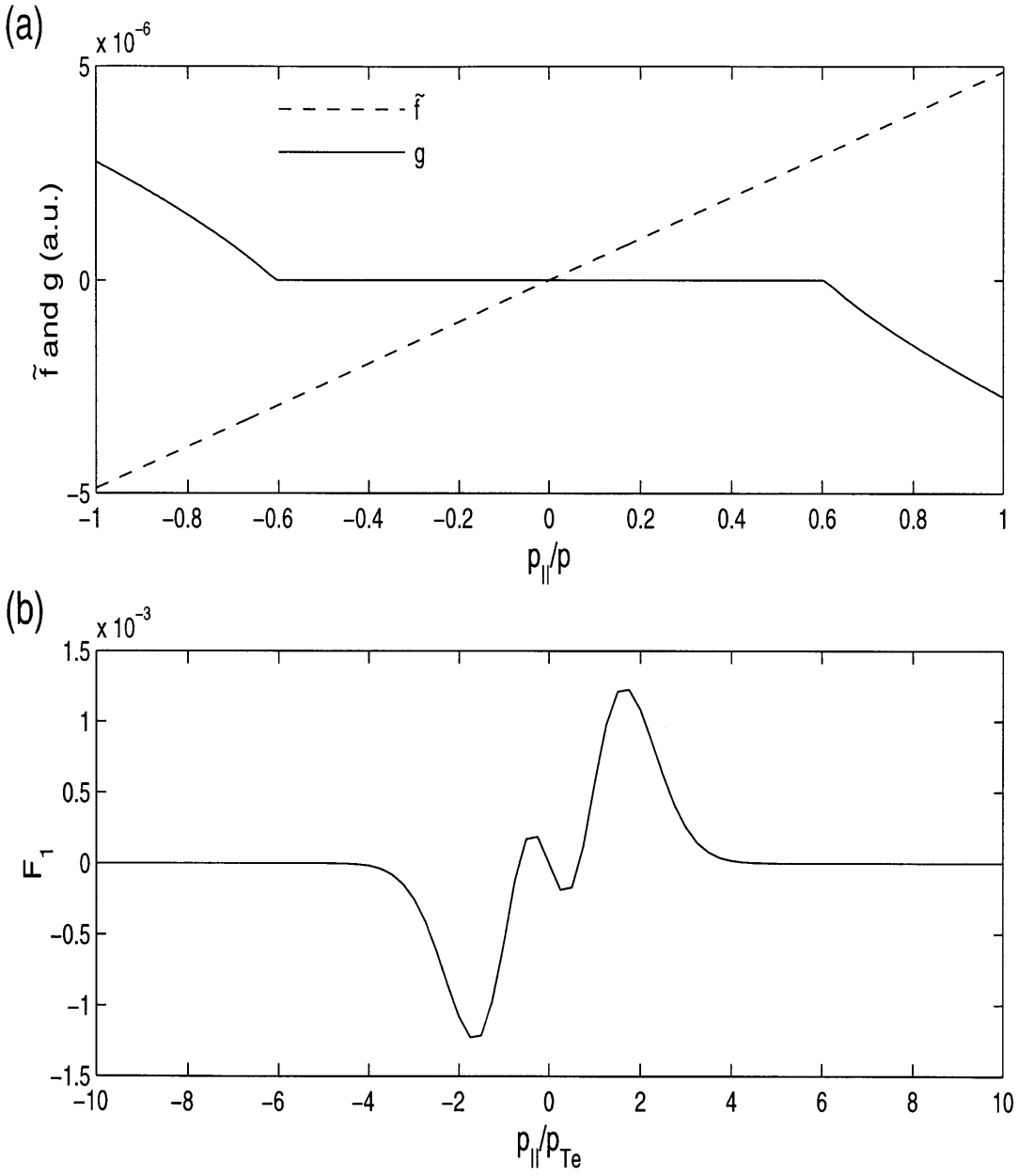

Figure 5-1: (a) Functions $\tilde{f}$ (dashed line) and g (solid line) as a function of ξ_0 for constant $p = 4p_{Te}$, for Alcator C-Mod type tokamak parameters (Table C.1) at r of 0.15m. (b) Perturbed parallel distribution function $F_1 \equiv \int 2\pi p_\perp dp_\perp \, (\tilde{f} + g)$ as a function of $p_{||}$ for the same case.

Run	Current Density in MA/m^2				Power Dens. in MW/m^3	
	BS Only $J_{\|\|}^{(B)}$	RF Only $J_{\|\|}^{(R)}$	Total $J_{\|\|}$	Synergism $J_{\|\|}^{(S)}$	RF Only $P_d^{(R)}$	Total P_d
001	2.9	10.6	14.0	0.4	23.7	24.7

Table 5.1: RUN-001: Output of FASTFP-NC calculations for LHCD with Alcator C-Mod-type parameters (see Table C.1).

current drive, which will include the bootstrap current, which we simply call $J_{\|}$. The flux-surface-averaged current density in this full calculation with lower hybrid current drive and neoclassical transport effects is $\left\langle J_{\|} B \right\rangle / \langle B \rangle = 14.0$ MA/m^2.

The simple approach of adding the RF driven current alone $J_{\|}^{(R)}$ to the bootstrap current alone $J_{\|}^{(B)}$ gives an average current density of only 13.5 MA/m^2. We define a *synergistic* current density as the difference in the current from these two approaches,

$$J_{\|}^{(S)} \equiv J_{\|} - (J_{\|}^{(B)} + J_{\|}^{(R)}) \tag{5.3}$$

The synergism between the LHCD and the bootstrap current for this case is found to be 440 kA/m^2. The results of all of these calculations are summarized in Table 5.1, under the heading RUN-001.

The extra current can be understood by looking at Figure 5-2 which shows the parallel distribution function

$$F \equiv \int 2\pi p_{\perp,0}\, dp_{\perp,0}\, \left[f_0 + \tilde{f} + g \right] \tag{5.4}$$

in comparision to

$$F_0 \equiv \int 2\pi p_{\perp,0}\, dp_{\perp,0}\, f_0. \tag{5.5}$$

The height of the LH-driven plateau is greater when the bootstrap current is included, and the current density is therefore greater. To understand the physical significance of this increase in height, we want to compare the first-order distribution function $\tilde{f} + g$ to its value in the absence of RF waves, $\tilde{f}^{(B)} + g^{(B)}$. This is shown at $\theta = 0$ (which is

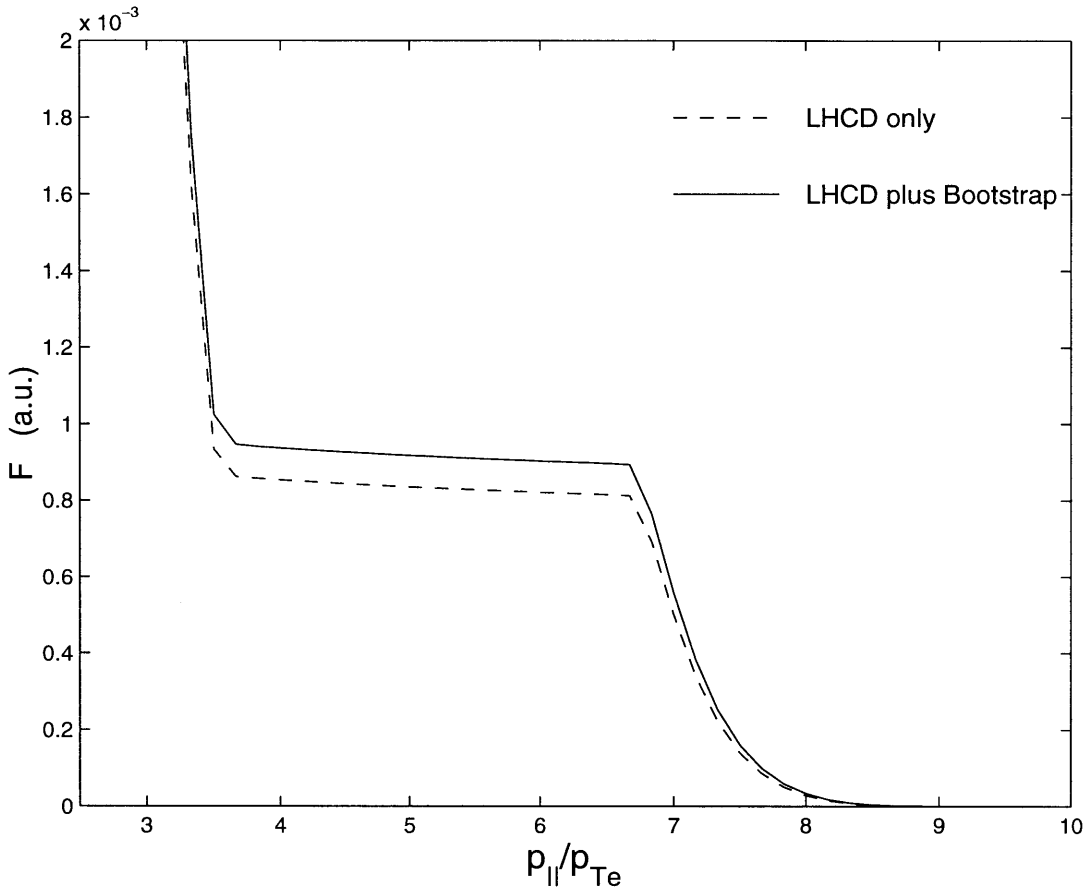

Figure 5-2: Parallel distribution function $F = \int 2\pi p_\perp dp_\perp f$ for LHCD as a function of $p_\parallel/p_{Te}$, with RF only (dashed line) and with RF plus bootstrap current (solid line). Parameters are Alcator C-Mod type (Table C.1) with r=0.15 m, D_0=4.0 for $v_\parallel \in (3.5v_{Te}, 6.0v_{Te})$.

where we assumed localized damping of the lower hybrid waves) in Figure 5-3. Once

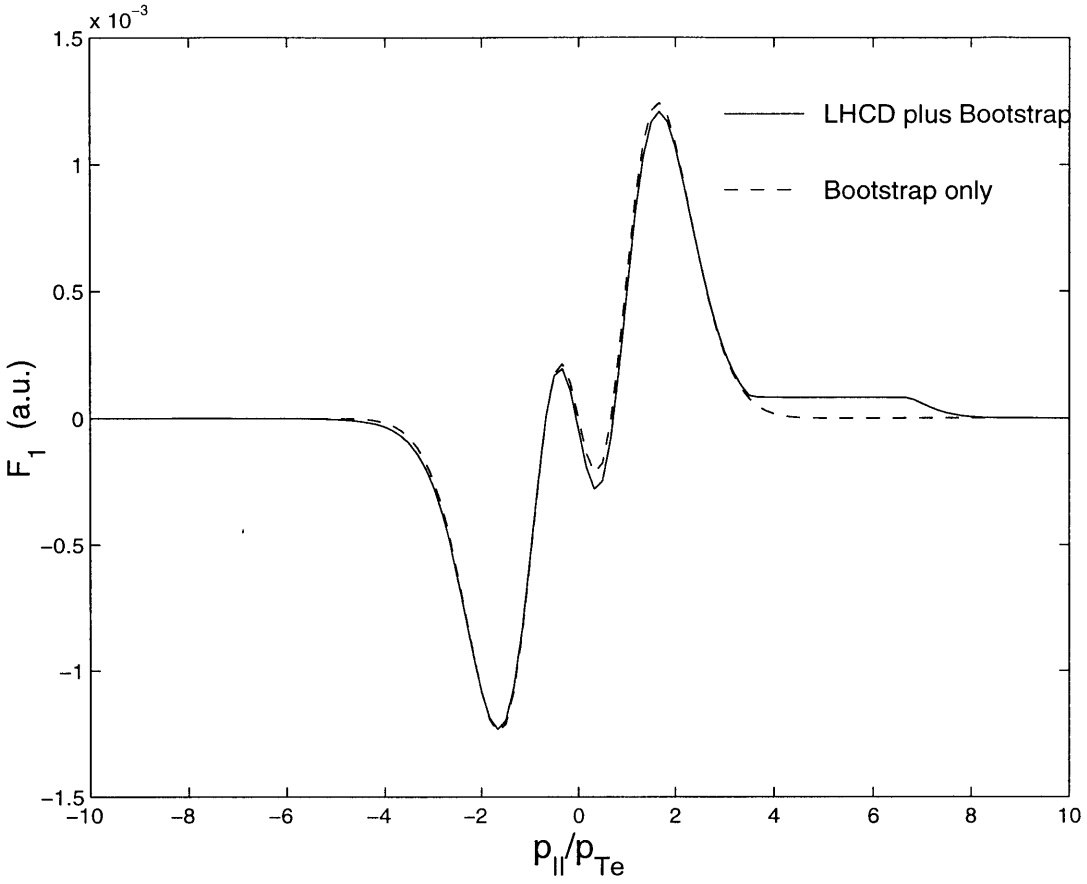

Figure 5-3: Perturbed parallel distribution function $F_1 = \int 2\pi p_\perp dp_\perp (\tilde{f}+g)$ for LHCD as a function of $p_\parallel/p_{Te}$, with bootstrap current only (dashed line) and bootstrap with LH (solid line). Parameters are the same as Figure 5-2.

again this is the parallel distribution function, with $p_\perp$ removed by integration,

$$F_1 \equiv \int 2\pi p_{\perp,0}\ dp_{\perp,0}\ \left[\tilde{f}+g\right] \tag{5.6}$$

We see that the extra current produced by the synergism appears as an additional height in the plateau in F_1.

The effective $(J_\parallel/P_d)$ of the current drive is calculated in a slightly different way by FASTFP-NC. Since the bootstrap current exists independent of RF current drive, and assuming that it is not affected by the RF fields, the limiting case as the RF

power goes to zero is $J_\parallel = J_\parallel^{(B)}$. So we define a new effective $(J_\parallel/P_d)$ as

$$\frac{J_\parallel - J_\parallel^{(B)}}{P_d}. \tag{5.7}$$

Referring to Table 5.1 for this particular case, the FASTFP-NC calculation of this effective $(J_\parallel/P_d)$ is 0.45 A·m/W, the same as the value we get when the bootstrap current is neglected entirely, $J_\parallel^{(R)}/P_D^{(R)}$=0.45 A·m/W. This shows that although there is an increase in the overall current density due to the synerigism, the power dissipated increases as well, so that there is no significant increase in the current drive efficiency.

5.3 Quanititative Predictions of Synergism Between Lower Hybrid and Bootstrap

Figures 5-2 and 5-3 hint at a simple picture of the interaction between lower hybrid current drive and the bootstrap current. From this picture, we can make predictions of the general effect of the bootstrap current on LHCD.

The dashed line in Figure 5-3 is $\tilde{f}^{(B)} + g^{(B)}$, the first order component of the distribution function in δ, for the bootstrap-current-only case (no RF waves). In a manner analagous to that used in Section 3.3.1 we note that the RF waves do not affect the distribution function at values of $p_\parallel$ less than $p_1 = 3.5 p_{Te}$. We can therefore estimate the height of the plateau as the value of $\tilde{f}^{(B)} + g^{(B)}$ at p_1:

$$\text{Extra Plateau Height } \approx F_1^{(B)}(p_\parallel = p_1) \equiv \int 2\pi p_{\perp,0}\, dp_{\perp,0}\, \left[\tilde{f}^{(B)} + g^{(B)}\right]_{p_\parallel = p_1}. \tag{5.8}$$

In order to find this height in a somewhat quantitative manner, we use the approximate form of g for the Lorentz gas limit, given in (3.12) with the integral (3.14). With this approximation

$$\tilde{f}^{(B)} + g^{(B)} \approx \left(\frac{p}{q_e B_{\theta,min}} \frac{\partial f_M}{\partial r}\right) \left[\xi_0 - \sigma\, H(|\xi_0| - \xi_{0T})(\xi_0 - \xi_{0T}) \left(1 + \frac{1}{4}\frac{\xi_{0T}}{\xi_0}\right)\right] \tag{5.9}$$

In the far circulating region of phase space, where the lower hybrid waves are most likely to interact with the electrons, we can approximate $\xi_0 \approx 1$, and thus, in the limit $\xi_{0T} \ll 1$,

$$\begin{aligned} F_1^{(B)}(p_\parallel = p_1) &\approx \left(\frac{p_1}{q_e B_{\theta,min}} \frac{\partial F_M}{\partial r}(p_1)\right)\left[1 - (1-\xi_{0T})\left(1 + \frac{1}{4}\xi_{0T}\right)\right] \\ \frac{F_1^{(B)}(p_1)}{F_M(p_1)} &\approx \frac{p_1}{q_e B_{\theta,min}}\left[\frac{d\ln n_e}{dr} + \left(\frac{p_1^2}{2m_e T_e} - \frac{3}{2}\right)\frac{d\ln T_e}{dr}\right]\left(\frac{3}{4}\xi_{0T}\right) \end{aligned} \quad (5.10)$$

In the limit $p_1^2 \gg p_{Te}^2$, the radial derivative $d\ln T_e/dr$ dominates,

$$\frac{F_1^{(B)}(p_1)}{F_M(p_1)} \approx \frac{p_1}{q_e B_{\theta,min}}\left(\frac{p_1^2}{2p_{Te}^2}\right)\frac{d\ln T_e}{dr}\left(\frac{3}{4}\xi_{0T}\right) \quad (5.11)$$

We express this in terms of the poloidal gyroradius, ($\rho_\theta = v_{Te}/\Omega_\theta = p_{Te}/q_e B_\theta$) at the midplane and also use $\xi_{0T} \approx \sqrt{2\epsilon}$, and we get

$$\begin{aligned} \frac{F_1^{(B)}(p_1)}{F_M(p_1)} &\approx \rho_\theta \left(\frac{p_1}{p_{Te}}\right)^3 \frac{d\ln T_e}{dr}\left(\frac{3\sqrt{2}}{8}\sqrt{\epsilon}\right) \\ &\approx 0.5\sqrt{\epsilon}\,\rho_\theta \frac{d\ln T_e}{dr}\left(\frac{p_1}{p_{Te}}\right)^3 \end{aligned} \quad (5.12)$$

This factor is the expected synergistic increase in the height of the lower hybrid plateau.

The above, approximate analysis shows that we can expect $J_\parallel^{(S)}$ to be proportional to the LH-driven current calculated alone $J_\parallel^{(R)}$

$$J_\parallel^{(S)} \approx \kappa J_\parallel^{(R)}, \quad (5.13)$$

where the approximate proportionality constant κ is

$$\kappa = \frac{\sqrt{\epsilon}}{2}\,\rho_\theta \frac{d\ln T_e}{dr}\left(\frac{v_1}{v_{Te}}\right)^3. \quad (5.14)$$

This factor will generally be small, since the poloidal gyroradius is typically less than a millimeter in most tokamaks.

Using this approximation, we return to the case of LHCD and the bootstrap current on Alcator C-Mod. For the parameters used, we get $\rho_\theta = 3.7 \times 10^{-4}$m, and $d \ln T_e/dr = 20\text{m}^{-1}$. With $\epsilon = 0.23$ and $v_1/v_{Te} = 3.5$ as previously given, this gives $\kappa \approx 0.08$. This is somewhat larger than the value found numerically by FASTFP-NC (see Table 5.1),

$$\kappa = \frac{\left\langle J_\parallel^{(S)} B \right\rangle}{\left\langle J_\parallel^{(R)} B \right\rangle} = \frac{0.5 \text{ MA/m}^2}{10.6 \text{ MA/m}^2} = 0.04 \tag{5.15}$$

Given the number of approximations which go into the derivation of (5.14) the fact that our calculation agrees with the prediction to within a factor of 2 is quite good.

5.4 Further Numerical Results with LHCD and Bootstrap Current

The analytic prediction of the amount of current drive synergism,

$$\frac{J_\parallel^{(S)}}{J_\parallel^{(R)}} \approx \frac{\sqrt{\epsilon}}{2} \rho_\theta \frac{d \ln T_e}{dr} \left(\frac{v_1}{v_{Te}} \right)^3 \tag{5.16}$$

is useful to predict how the synergistic current density scales with the parameters of the plasma. In this section we present further calculations with FASTFP-NC to check this scaling.

In each of the cases RUN-002 to RUN-009 analyzed with FASTFP-NC the parameters are the same as taken in RUN-001, only with one change to test our prediction of the scaling factor (5.16). We also present two cases called RUN-0A1 and RUN-0A2, in which we use parameters typical of the ARIES-RS tokamak design [50]. These parameters are described in Appendix C in Table C.2.

- **RUN-001:** Baseline LHCD case with Alcator C-Mod type parameters in Table C.1 and current drive at $\epsilon = 0.23$ with D_0=4.0, $v_1 = 3.5 v_{Te}$ and $v_2 = 6.0 v_{Te}$.
- **RUN-002:** Same as RUN-001 parameters but with $v_1 = 4.0 v_{Te}$ and $v_2 = 6.5 v_{Te}$.

Run	Current Density in MA/m^2				Power Dens. in MW/m^3	
	BS Only $J_\parallel^{(B)}$	RF Only $J_\parallel^{(R)}$	Total $J_\parallel$	Synergism $J_\parallel^{(S)}$	RF Only $P_d^{(R)}$	Total P_d
001	2.9	10.6	14.0	0.5	23.7	24.7
002	2.9	1.7	4.8	0.1	3.1	3.3
003	5.9	10.6	17.4	0.9	23.7	25.7
004	2.1	15.2	17.9	0.6	26.5	27.2
005	5.9	14.2	20.9	0.9	17.4	18.5
006	4.3	10.6	15.6	0.7	23.7	25.2
007	5.9	21.2	28.0	0.9	93.2	97.0
008	3.5	10.6	14.6	0.5	23.7	24.7
0A1	2.4	3.0	5.5	0.06	3.81	3.86
0A2	2.5	0.4	2.8	0.02	0.44	0.45

Table 5.2: Output of FASTFP-NC calculations for LHCD with variations of Alcator C-Mod-type parameters (see Table C.1).

- **RUN-003:** Same as RUN-001 parameters but with B_θ reduced by a factor of 2, so that ρ_θ is increased by a factor of 2.

- **RUN-004:** Same as RUN-001 parameters but with major radius R_0 increased to 1.36 m, so that ϵ is reduced by a factor of 2 to a value of 0.12.

- **RUN-005:** Same as RUN-001 parameters but with the local electron temperature T_e doubled (which increases ρ_θ by a factor of $\sqrt{2}$) while keeping the temperature gradient dT_e/dr fixed.

- **RUN-006:** Same as RUN-001 parameters but with the electron temperture gradient dT_e/dr doubled.

- **RUN-007:** Same as RUN-001 parameters but with the local electron density n_e doubled, while keeping the density gradient fixed. Density does not appear in (5.16), so the ratio should be the same as in RUN-001.

- **RUN-008:** Same as RUN-001 parameters but with the electron density gradient doubled. Density does not appear in (5.16), so the ratio should be the same as in RUN-001.

Run	Current Density in MA/m^2		Ratio of $J^{(S)}/J^{(R)}$	
	$J_{\|\|}^{(R)}$	$J_{\|\|}^{(S)}$	Numerical	Predicted
001	10.6	0.5	0.04	0.08
002	1.7	0.1	0.06	0.12
003	10.6	0.9	0.09	0.16
004	15.2	0.6	0.04	0.05
005	14.2	0.9	0.06	0.11
006	10.6	0.7	0.06	0.13
007	21.2	0.9	0.04	0.08
008	10.6	0.5	0.04	0.08
0A1	3.0	0.06	0.02	0.02
0A2	0.4	0.02	0.04	0.04

Table 5.3: Ratio of the synergistic current density to LH-only current density: Calculated and predicted for FASTFP-NC runs in Table 5.2.

- **RUN-0A1:** LHCD case with ARIES-RS type parameters in Table C.2 and current drive at $\epsilon = 0.20$ with D_0=4.0, $v_1 = 3.5v_{Te}$ and $v_2 = 6.0v_{Te}$.
- **RUN-0A2:** Same as RUN-0A1 parameters but with $v_1 = 4.0v_{Te}$ and $v_2 = 6.5v_{Te}$.

In Table 5.3 we list the results of these LHCD cases. We calculate the ratio of the synergistic current density to the LH-only current density and compare this to the prediction we made in (5.16). In these results we notice that in general our prediction within a factor of 2 in each case. This is particularly interesting in that it shows that while the prediction does not give an accurate value for the synergistic current it can show that the $J_{\|}^{(S)}/J_{\|}^{(R)}$ in (5.16) does have the correct variation with plasma parameters.

5.5 Analysis of LHCD-Bootstrap Results

The FASTFP-NC code shows that there will be a synergistic increase in the current driven by lower hybrid waves when a significant bootstrap current is present. This current increase is due to an increase in the height of the LH-driven plateau.

We consider the expression (5.16) for the relative increase in the RF-driven current density. In most laboratory plasmas the poloidal gyroradius is much smaller than the scale length of temperature variations ($\rho_\theta \; d\ln T_e/dr \ll 1$). We can observe that because (5.16) increases with $\sqrt{\epsilon}$, temperature, and the temperature gradient, and decreases with increasing poloidal field, then the synergistic current will become more significant when the bootstrap current is greatest, since

$$J_\parallel^{(B)} \sim \sqrt{\epsilon}\frac{n_e T_e}{B_\theta}\left(\frac{d\ln T_e}{dr}\right).$$

This is clearly observed by comparing RUN-003 to RUN-001 in Table 5.2. When the bootstrap current density is increased from 2.9 MA/m^2 to 5.9 MA/m^2, the synergistic current density increases from 0.5 MA/m^2 to 0.9 MA/m^2.

The other important factor in (5.16) is $(v_1/v_{Te})^3$. Because lower hybrid current drive takes place in tail of the distribution function, we expect v_1/v_{Te} to be greater than one, so that the power of 3 can make $(v_1/v_{Te})^3$ much larger than 1. Unfortunately, as noted in Section 3.3.1, the value of v_1 is determined by Landau damping of the lower hybrid waves, which cannot be controlled by external means; it is generally restricted to the case $v_1/v_{Te} \approx 3-4$.

The numerical results also indicate that the power dissipated increases along with the current density, and thus the effective $(J_\parallel/P_d)$ does not change as a result of the synergistic current. Because the power dissipated increases for a fixed value of v_1, lower hybrid waves with $k_\parallel = \omega/v_1$ will be more strongly Landau damped in this case.

Chapter 6

Electron Cyclotron Current Drive with Neoclassical Transport Effects Including Bootstrap Current

6.1 Introduction

The interaction of electron cyclotron waves and electrons in a plasma is significantly more complicated than the lower hybrid current drive interaction discussed in Chapter 5. The effect of radial drifts on parallel transport with electron cyclotron current drive (ECCD) is more difficult to predict, and the numerical calculation which can be preformed with FASTFP-NC is a valuable tool.

In Section 6.2 FASTFP-NC calculations show how ECCD and the bootstrap current interact. We discuss the results and attempt to understand them in Section 6.3.

6.2 Numerical Results with ECCD and Bootstrap Current

Using the FASTFP-NC code, we can find results of the total non-inductive current drive with electron cyclotron current drive, and try to find a synergistic effect similar to that produced by lower hybrid waves and bootstrap current. The results of some representative cases are shown in this section. In each case, we show a series of runs with different values of $N_{||}$. We note that as $N_{||}$ increases, the resonance curve moves further into the bulk of the plasma, an effect which is shown in Figure 6-1. Because

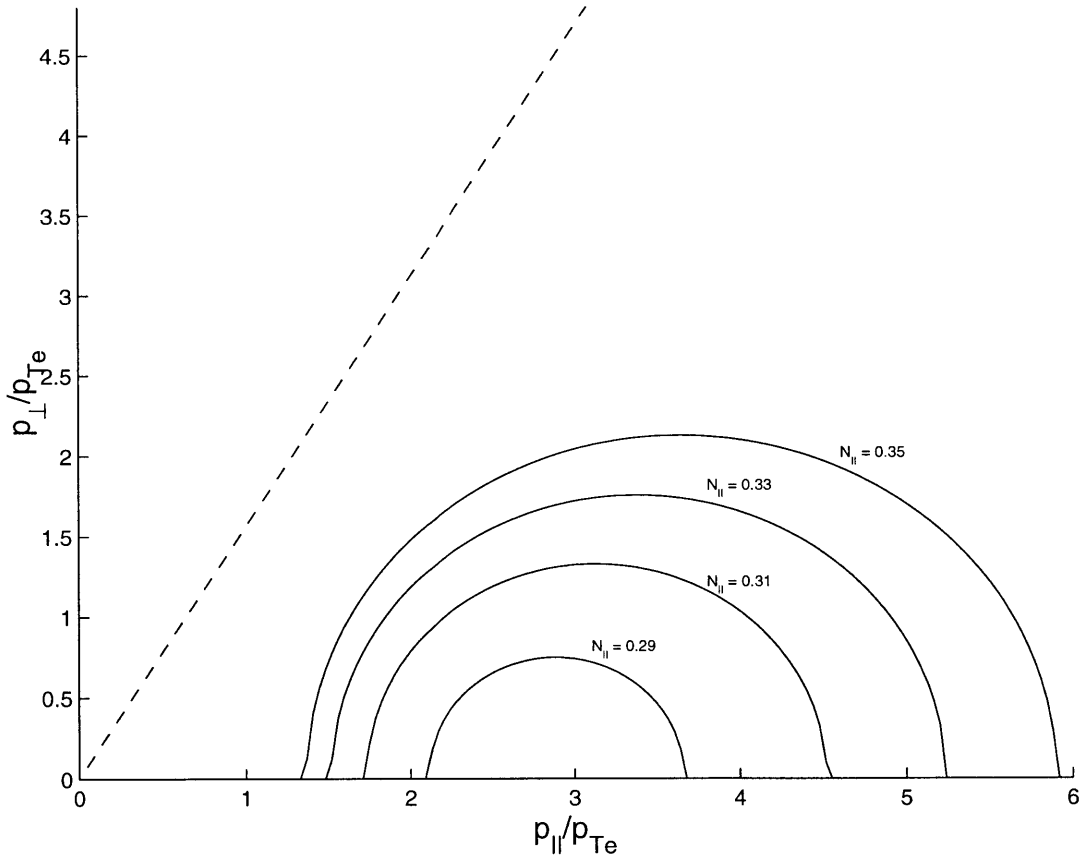

Figure 6-1: Resonance condition $\omega - k_{||}v_{||} - n\Omega = 0$ in $(p_{||}/p_{Te}, p_{\perp}/p_{Te})$ phase space for $n\Omega_0/\omega = 0.97$ and different values of $N_{||}$.

the bootstrap current is related to trapped particles, we will perform calculations in which the waves damp on the low-field side of the tokamak, at a poloidal angle θ=0.

Run	$N_\parallel$	Current Density in MA/m^2				Power Dens. in MW/m^3	
		BS Only $J_\parallel^{(B)}$	RF Only $J_\parallel^{(R)}$	Total $J_\parallel$	Synergism $J_\parallel^{(S)}$	RF Only $P_d^{(R)}$	Total P_d
RUN-D1	0.29	0.395	0.213	0.610	0.001	0.38	0.38
RUN-D2	0.31	0.395	0.643	1.042	0.004	1.38	1.38
RUN-D3	0.33	0.395	1.043	1.446	0.008	2.94	2.95
RUN-D4	0.35	0.395	1.109	1.515	0.011	4.75	4.76
RUN-D5	0.37	0.395	0.724	1.133	0.014	6.56	6.57

Table 6.1: Current density and power dissipated results of FASTFP-NC runs based on DIII-D tokamak parameters (See Table C.3), with $2\omega_{ce}/\omega$=0.96 and D_0=0.4, on the flux surface r=0.3 m.

However, in this case, the Ohkawa effect [37] will become detrimental to current drive when the resonance curve approaches the trapped electrons.

First, we use parameters from DIII-D, as shown in Table C.3 in Appendix C. This choice of parameters was motivated by recent experiments in ECCD on this tokamak [93]. For these calculations, however, the density and temperature are taken to be slightly higher than was observed in those experiments, in order to increase the bootstrap current in the hopes of increasing the synergism observed. Each case is for current drive at the flux surface r=0.3 m (ϵ=0.18). The results are tabulated in Table 6.1.

We observe that the total current increases with $N_\parallel$, owing to the greater penetration of the waves into the bulk of the electron distribution, until the last case, where the Ohkawa effect [37] has begun to reduce the current density. However, the synergistic current produced continues to increase as $N_\parallel$ increases, even when the Ohkawa effect appears. Unlike the significant synergism observed in the LHCD case, in these results the synergism is consistently less than 2% of the total current.

To find a more substantial synergistic effect, we expect that a tokamak with a higher bootstrap current might present an interesting case. Calculations similar to the previous case were therefore performed using plasma parameters for the tokamaks considered in Chapter 5, Alcator C-Mod and ARIES-RS.

In Table 6.2 we show results for a hypothetical ECCD scenario on Alcator C-Mod

Run	$N_\parallel$	Current Density in MA/m^2				Power Dens. in MW/m^3	
		BS Only $J_\parallel^{(B)}$	RF Only $J_\parallel^{(R)}$	Total $J_\parallel$	Synergism $J_\parallel^{(S)}$	RF Only $P_d^{(R)}$	Total P_d
RUN-C1	0.26	2.94	0.30	3.27	0.03	6.77	6.98
RUN-C2	0.28	2.94	0.74	3.77	0.09	31.0	31.7
RUN-C3	0.29	2.94	0.71	3.77	0.13	49.4	50.4
RUN-C4	0.293	2.94	0.64	3.72	0.14	55.6	56.6

Table 6.2: Current density and power dissipated results of FASTFP-NC runs based on Alcator C-Mod tokamak parameters (See Table C.1), with $2\omega_{ce}/\omega$=0.97 and D_0=0.4, on the flux surface r=0.15 m.

Run	$N_\parallel$	Current Density in MA/m^2				Power Dens. in MW/m^3	
		BS Only $J_\parallel^{(B)}$	RF Only $J_\parallel^{(R)}$	Total $J_\parallel$	Synergism $J_\parallel^{(S)}$	RF Only $P_d^{(R)}$	Total P_d
RUN-A1	0.29	2.40	0.24	2.64	0.00	4.33	4.34
RUN-A2	0.31	2.40	1.68	4.08	0.004	35.7	35.8
RUN-A3	0.33	2.40	2.73	5.14	0.01	83.7	83.8
RUN-A4	0.34	2.40	2.71	5.12	0.02	109	110
RUN-A5	0.35	2.40	2.26	4.67	0.02	135	135

Table 6.3: Current density and power dissipated results of FASTFP-NC runs based on ARIES-RS tokamak parameters (See Table C.2), with $2\omega_{ce}/\omega$=0.96 and D_0=0.4, on the flux surface r=1.1 m.

(see Table C.1 for details). This hypothetical current drive scenario, for the n=2 harmonic of the electron cyclotron frequency, would require an RF source at a very high frequency of nearly 220 GHz, which would be difficult with existing technology. As with the DIII-D results shown above, we take a diffusion coefficient, normalized to collisional diffusion, of $D_0 = 0.4$, which in this case is consistent with an injected power of 20 MW. In these cases the synergistic current is more substantial, often a large fraction of the RF-driven current. However, as with lower-hybrid waves, we notice that the power dissipated also increases in these cases.

We also perform calculations with similar numbers using ARIES-RS parameters, and the results are shown in Table 6.3. We observe that the synergistic currents are considerably lower for ARIES-RS than they were for the Alcator C-Mod runs.

6.3 Analysis and Discussion

It has long been understood [11, 12] that RF waves can have a profound effect on the bootstrap current by directly affecting the plasma parameters. In particular, RF can produce strong heating to increase the plasma pressure, and can also be used to alter the radial transport of the tokamak, which can result in steep pressure gradients [94].

We consider the possibility that the synergistic current which appears in EC-bootstrap calculations may be the result of a simple effect: the increase in electron temperature due to EC waves. As with LHCD in Chapter 5, we consider a single case of ECCD, listed in Table 6.1 as RUN-D4: parameters are taken for a DIII-D type tokamak (see Table C.3) with $\epsilon = 0.18$, for $2\omega_{ce}/\omega$=0.96 and $N_{\|}$=0.35 and D_0=0.4. The ECCD-modified electron distribution function f_0 for this case was shown as an example in Section 3.3.2, in Figures 3-6 and 3-7. Because of the pitch-angle scattering due to collisions, the energy transferred from the EC waves to the resonant electrons is redistributed to electrons at all pitch angles. This is shown in Figure 6-2 where we compare the EC-modified distribution (from Figure 3-6) to the unheated Maxwellian distribution of the electrons (dashed contours). We note that although the strongest energization of electrons by waves occurs in the resonant region, pitch angle scattering by collision means that there is energization at all pitch angles, even far from the resonant region and in the trapped region of phase space.

We take a simple model in which we calculate the difference in the bootstrap current due to a small change in electron temperature ΔT_e, while the density and the ion temperature are unchanged. We use the general expression for the bootstrap current in a Maxwellian plasma (4.31), and assume that the transport coefficients L_{31} and L_{32} are unchanged by the RF waves, so that (4.33) and (4.39) are used for these quantities. From (4.31), the change to the bootstrap current which results from ΔT_e is

$$\left\langle \Delta J_{\|,BS} B \right\rangle = -p_e \frac{B_\zeta}{B_\theta} \left[L_{31} \frac{d \ln n_e}{dr} \frac{\Delta T_e}{T_e} + (L_{31} + L_{32}) \frac{1}{T_e} \frac{d(\Delta T_e)}{dr} \right]. \qquad (6.1)$$

We calculate ΔT_e and $d(\Delta T_e)/dr$ from the distributions calculated by FASTFP-NC in the runs tabulated in Section 6.2. In Table 6.4 we show the flux-surface average

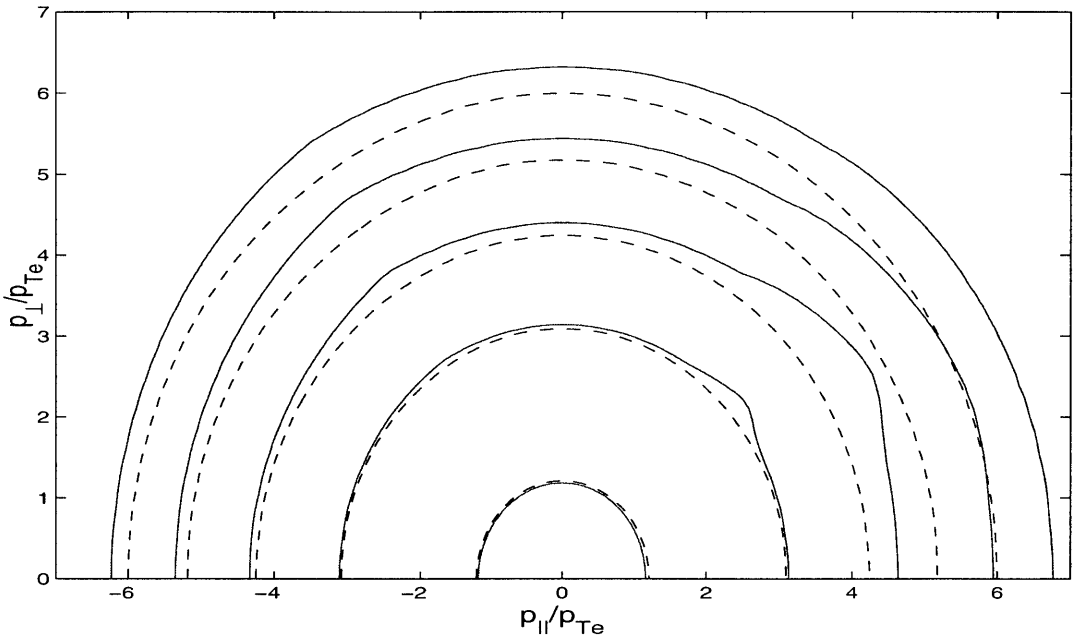

Figure 6-2: Contour plot in $(p_{\|}/p_{Te}, p_{\perp}/p_{Te})$ space of the EC-modifed electron distribution function (solid contours) and the Maxwellian distribution unmodified by EC (dashed contours). Parameters are DIII-D type (see Table C.3) with r=0.3 m, $2\Omega_0/\omega = 0.96$, $N_{\|}$=0.35, and D_0=0.4.

Run	Current Density in MA/m^2	
	$J_{\|\|}^{(S)}$	$\Delta J_{\|\|,BS}(\Delta T_e)$
RUN-D1	0.001	0.001
RUN-D2	0.004	0.003
RUN-D3	0.008	0.007
RUN-D4	0.011	0.012
RUN-D5	0.014	0.017
RUN-C1	0.03	0.02
RUN-C2	0.09	0.07
RUN-C3	0.13	0.10
RUN-C4	0.14	0.11
RUN-A1	0.00	0.00
RUN-A2	0.004	0.003
RUN-A3	0.01	0.01
RUN-A4	0.02	0.02
RUN-A5	0.02	0.02

Table 6.4: Synergistic current density $J_{\|\|}^{(S)}$ from the results of the FASTFP-NC runs in Tables 6.1, 6.2, and 6.3, as calculated directly and from the electron heating effect expressed in (6.1).

of the synergistic current density directly calculated from FASTFP-NC and the value of $\Delta J_{\|,BS}$ calculated using (6.1) for the same runs. From these results, it appears that there is some correlation between the amount of electron heating and the synergistic current, although the temperature increase does not explain the full synergistic current, especially in the calculations using Alcator C-Mod type parameters.

The FASTFP-NC results shown in this chapter seem to indicate that significant synergistic effects between EC waves and the bootstrap current will not appear unless the properties (especially temperature) of the bulk electrons are significantly changed, which requires an amount of power which is impractical in current drive scenarios. It would be of interest in the future to consider the synergistic currents produced in scenarios with bulk RF heating as well as current drive. Such a study would require a more sophisticated Fokker-Planck code; unfortunately a current drive code such as FASTFP has limitations which make such calculations unreliable due to its use of a linearized collision operator.

Chapter 7

Conclusions

In the preceding chapters we presented calculations of the total current density in cases of lower hybrid and electron cyclotron current drive with the bootstrap current. Here we summarize these results, discuss the conclusions drawn from these calculations and discuss interesting possibilities for future research on this subject.

7.1 Summary

The FASTFP-NC code has proved to be an essential tool to our understanding of RF-bootstrap synerigism. While Fokker-Planck codes have often been used to solve the problem of RF current drive, it is extremely useful to have a code which can find the total non-inductive current density, including the bootstrap current, numerically. A significant amount of modification was necessary to include bootstrap currents in the FASTFP code: trapped electrons and the trapped-passing boundary are treated accurately, collisions with ions include the neoclassical ion parallel flow which contributes to the bootstrap current, and improved accuracy allows the detection of small synergistic current densities in the combined RF-bootstrap current.

As an essential first step to verification of the usefulness of FASTFP-NC, accurate calculations of the bootstrap current without RF waves were performed. The calculations shown in Section 4.4.1 showed excellent agreement with the predictions of Hirshman [91] in all cases except those with temperature gradients. Although this

was of some concern during the benchmarking of the code, the work of Sauter, Angioni, and Lin-Liu [92] supports the validity of the calculations of FASTFP-NC in these cases.

Calculations of lower hybrid current drive with the bootstrap current showed the presense of a synergistic current density. The discretized electron distribution function which was output by FASTFP-NC indicated an approximate analytic picture for this phenomenon: an increase in the height of the plateau of the distribution function in the resonance region due to the bootstrap current. The ratio between the synergistic current and the predicted RF-driven current (without bootstrap) is given by (5.16).

Calculations of the synergism between electron cyclotron current drive and bootstrap current also showed an increased current, albeit a smaller relative increase than in the lower hybrid cases. In each case, the synergistic increase was accompanied by a measured increase in the electron temperature, due to a tail of energized electrons. We considered the possibility that the synergism is simply the increase in the bootstrap current due to electron heating, which was shown to be approximately consistent with the numerical data; however this model is most likely too simple to accurately treat a non-Maxwellian distribution due to RF waves.

7.2 RF Current and Bootstrap Current – Conclusions

To some degree the results of Chapters 5 and 6 indicate that the assumption that the bootstrap current is unchanged by RF current drive is reasonable when applied to existing experiments. RF-bootstrap synergism does give a numerically calculable increase in the current density, but one that is discovered to be rather small when the bootstrap current is low, as it is in presently existing tokamaks. This is itself an important result of the FASTFP-NC calculations, since much of the data analysis of these experiments depends on this assumption.

The importance of this synergistic effect will be more significant in cases where

the bootstrap current is enhanced, and we expect that the effect may play a role in future scenarios with higher bootstrap currents.

The interaction of lower hybrid waves with the bootstrap current shows a significant change in the current density. The expression (5.16) shows that we expect the synergistic current density to be proportional to the square root of the inverse aspect ratio ϵ and the square root of the electron temperature, and inversely proportional to the poloidal magnetic field and the scale length of radial variation of the temperature. We have noted that the bootstrap current itself has the same dependencies on these physical parameters, and the progression to tokamaks with higher bootstrap fractions leading to steady-state scenarios will have the added benefit of increasing the synergistic current density.

The cases of electron cyclotron current drive with the bootstrap current show a weaker synergism than lower hybrid waves when the waves are resonant with electrons in the tail of the electron distribution. The numerical results do indicate however that the synergism has the potential to become significant in cases where the spectrum of waves penetrates into the bulk of the electron distribution function and heats the electrons. The advantage of increasing the synergistic current density must be weighed against the decrease in the overall current drive efficiency as the waves penetrate into the bulk of the electrons. For this reason it appears that EC-bootstrap synergism is not as advantageous as LH-bootstrap synergism for current drive, but the increased bootstrap current will be a desirable side effect of EC electron heating.

Finally, we note an important point, that in all of the cases considered the synergistic current density generated is positive, i.e. in the same direction as the bootstrap and RF-driven currents. Therefore the interaction of RF current drive and the bootstrap current (when they are directed in the same direction along the magnetic field) will not have a *negative* effect on the total current, which would have been a problem for the future of steady-state tokamak research.

7.3 Directions for Future Study

The calculations performed in this work are focused entirely on transport in the direction of the equilibrium magnetic field. Clearly the effect of RF waves on transport across the magnetic field is an important consideration which could be studied in future works. Such effects could change the density and temperature profiles which would change the assumed profiles used in this work.

An assumption which was implicit in this work and many of the references on neoclassical theory which preceded it is that the banana orbits of the electrons are narrow, so that the excursion from the flux surface, Δr, is small. Recent considerations of the effect of wide ion banana orbits on ion neoclassical transport [95] show their importance, especially near the magnetic axis, and these particular orbits could be included in the RF-bootstrap problem in future works.

There is another mechanism for RF current drive which was not included in this work, using fast Alfvén waves (FAW). Because these waves damp on electrons closer to the bulk of the distribution function, the FASTFP code does not treat FAW damping accurately. Attempts to include a FAW quasilinear operator in the FASTFP-NC have not resulted in accurate calculations, and thus these attempts were not included in this thesis. However, a FAW-bootstrap calculation would be an interesting case to add to the lower hybrid and electron cyclotron current drive results presented here.

We have also considered only RF waves for current drive in this work without considering RF heating cases. This was largely necessitated by our use of a Fokker-Planck code with an electron-electron collision operator which does not conserve energy. As indicated by our numerical results for electron cyclotron current drive, the most interesting synergistic currents will be likely to occur when the electrons are significantly heated. A future work using a Fokker-Planck code designed to calculate EC heating of the bulk electrons could be used to calculate the bootstrap current according to the same methods used by FASTFP-NC.

Appendix A

Collision Operator for FASTFP Code

This is the Braams-Karney collision operator [61] as calculated in the homogeneous by Shoucri and Shkarovsky [1] for use in their FASTFP Fokker-Planck code. The collision operator is normalized in units of the collision frequency $\nu_e = e^4 n_e \ln\Lambda / 4\pi\epsilon_0^2 m_e^2 v_{Te}^3$ and the momentum p is normalized to the thermal momentum $m_e v_{Te}$.

The notation used here is as follows:

$$z = \frac{p}{m_e c}, \quad \gamma = \sqrt{1+z^2}, \quad v = \frac{p}{\gamma}, \quad \varsigma = \sinh^{-1} z$$

As defined in Ref. [61], we use the functions

$$J_{0[1]1} = 1, \quad J_{0[1]2}(z) = \gamma, \quad J_{1[1]2}(z) = \frac{1}{3}z,$$
$$J_{0[2]02}(z) = \frac{1}{2}zJ_{1[1]1}(z) = \frac{1}{4}\left(\gamma - \frac{\varsigma}{z}\right), \quad J_{0[2]11}(z) = \frac{1}{2}zJ_{1[1]0}(z) = \frac{1}{2}\left(\frac{\gamma\varsigma}{z} - 1\right),$$
$$J_{0[3]022}(z) = \frac{1}{4}zJ_{1[2]11}(z) = \frac{1}{32}\left[-3\gamma + \varsigma\left(\frac{3}{z} + 2z\right)\right], \quad J_{1[2]02}(z) = \frac{1}{12}\left(-\frac{3\gamma\varsigma}{z^2} + \frac{3}{z} + z\right),$$
$$J_{1[3]022}(z) = \frac{1}{288}\left[\gamma\varsigma\left(\frac{15}{z^2} + 6\right) - \frac{15}{z} - 11z\right].$$

For a plasma with a bulk electron Maxwellian distribution $f_M(p)$ with temperature T_e, and ions with charge Z_i and temperature T_i, we can derive $A(p)$, $B(p)$, $F(p)$, and

$I_1(f_M, f_{l=1}\xi)$ described in Section 2.3.2. The results are shown in Reference [1]:

$$F_1 = 4\pi \int_0^p dp'\ f_M(p')p'v' \left[\frac{1}{v^2} + \frac{1}{p^2}\left(1 - \frac{\gamma'\varsigma'}{z'}\right)\right], \tag{A.1}$$

$$F_2 = \frac{4\pi}{v}\left(1 - \frac{\varsigma}{z\gamma}\right)\int_p^\infty p' f_M(p')dp', \tag{A.2}$$

$$F(p) = F_1 + F_2 + \frac{1}{v^2} Z_i \frac{m_e}{m_i}, \tag{A.3}$$

$$A(p) = \frac{1}{v}(F_1 + F_2) + \frac{1}{v^3} Z_i \frac{m_e}{m_i}\frac{T_i}{T_e}, \tag{A.4}$$

$$\begin{aligned} B_1 = 4\pi \int_0^p dp'\ f_M(p')\frac{p'^2}{v}\Bigg\{ & \frac{3p^2 - p'^2}{6p^2} + \frac{1}{8\gamma'\gamma^2 z^2}\left[-3\gamma' + \varsigma'\left(\frac{3}{z'} + 2z'\right)\right] \\ & - \frac{1}{4z^2\gamma'}\left(\gamma' - \frac{\varsigma'}{z'} - \frac{2}{3}\gamma' z'^2\right) - \frac{1}{4\gamma'\gamma^2}\left(\gamma' - \frac{\varsigma'}{z'}\right)\Bigg\}, \end{aligned} \tag{A.5}$$

$$\begin{aligned} B_2 = 4\pi \int_p^\infty dp'\ f_M(p')\frac{p'^2}{v'}\Bigg\{ & \frac{1}{2} - \frac{\gamma^2}{6\gamma'^2} + \frac{1}{8\gamma\gamma'^2 z^2}\left[-3\gamma + \varsigma\left(\frac{3}{z} + 2z\right)\right] \\ & - \frac{\gamma}{4z^2\gamma'^2}\left(\gamma - \frac{\varsigma}{z} - \frac{2}{3}\gamma z^2\right) - \frac{v'^2}{4\gamma p^2}\left(\gamma - \frac{\varsigma}{z}\right)\Bigg\}, \end{aligned} \tag{A.6}$$

$$B(p) = B_1 + B_2 + \frac{Z_i}{2v}\left(1 - \frac{1}{v^2}\frac{m_e}{m_i}\frac{T_i}{T_e}\right), \tag{A.7}$$

$$\begin{aligned} I_1(f_M, f_{l=1}\xi) = {} & 4\pi f_M(p) f_{l=1}(p)\frac{\xi}{\gamma} + 4\pi\xi f_M(p)\int_0^p dp' f_{l=1}(p')\frac{p'^3}{\gamma\gamma' p^2}\Bigg\{\frac{1}{3} - \frac{2}{3}\gamma'\gamma + \frac{1}{5}\gamma p'^2 \\ & + \frac{1}{p'^2}\left(\gamma' - \frac{\varsigma'}{z'}\right) - \frac{\gamma}{z'^2}\left(\gamma' - \frac{\varsigma'}{z'} - \frac{2}{3}\gamma' z'^2\right) + \left(\frac{\gamma p^2 - 5}{6z'}\right)\left(-\frac{3\gamma'\varsigma'}{z'^2} + \frac{3}{z'} + z'\right) \\ & + \frac{\gamma}{2z'\beta_{th}^2}\left(-\frac{3\gamma'\varsigma'}{z'^2} + \frac{3}{z'} + z' - \frac{2}{5}z'^3\right) + \frac{\gamma}{2z'^2}\left[-3\gamma' + \varsigma'\left(\frac{3}{z'} + 2z'\right)\right] \\ & + \frac{p^2}{p'^2}\left(\frac{\gamma'\varsigma'}{z'} - 1\right) - \frac{\gamma}{12z'\beta_{th}^2}\left[\gamma'\varsigma'\left(\frac{15}{z'^2} + 6\right) - \frac{15}{z'} - 11z'\right]\Bigg\} \\ & + 4\pi\xi f_M(p)\int_p^\infty dp' f_{l=1}(p')\frac{p}{\gamma\gamma'}\Bigg\{\frac{1}{3} - \frac{2}{3}\gamma\gamma' + \frac{1}{5}\gamma' p^2 \\ & + \frac{1}{p^2}\left(\gamma - \frac{\varsigma}{z}\right) - \frac{\gamma'}{z^2}\left(\gamma - \frac{\varsigma}{z} - \frac{2}{3}\gamma z^2\right) + \left(\frac{\gamma' p'^2 - 5}{6z}\right)\left(-\frac{3\gamma\varsigma}{z^2} + \frac{3}{z} + z\right) \\ & + \frac{\gamma'}{2z\beta_{th}^2}\left(-\frac{3\gamma\varsigma}{z^2} + \frac{3}{z} + z - \frac{2}{5}z^3\right) + \frac{\gamma'}{2z^2}\left[-3\gamma + \varsigma\left(\frac{3}{z} + 2z\right)\right] \\ & + \frac{p'^2}{p}^2\left(\frac{\gamma\varsigma}{z} - 1\right) - \frac{\gamma'}{12z\beta_{th}^2}\left[\gamma\varsigma\left(\frac{15}{z^2} + 6\right) - \frac{15}{z} - 11z\right]\Bigg\} \end{aligned} \tag{A.8}$$

Appendix B

Approximation of the Integral I for Neoclassical Theory

In Section 3.2, the function g is found in the Lorentz gas limit, and expressed in terms of the integral $I(\xi_0)$ in (3.13),

$$I(\xi_0) = \int_{\xi_{0T}}^{\xi_0} \frac{d\xi_0}{\lambda(1+\epsilon)(1-\Delta/\xi_0^2)}. \tag{B.1}$$

where $\Delta = \{1 - 1/\Psi\}$ and $\Psi = (1+\epsilon)/(1+\epsilon\cos\theta)$. Here we find an approximation for this integral in the limit $\epsilon \ll 1$.

First we recall that, as shown in (E.12),

$$1 - \frac{\Delta}{\xi_0^2} = \left\{\frac{\xi^2}{\Psi\xi_0^2}\right\}. \tag{B.2}$$

We denote the denominator of the integrand of I as $\mathcal{D}$,

$$\begin{aligned} \mathcal{D} &\equiv (1+\epsilon)\lambda\left(1 - \frac{\Delta}{\xi_0^2}\right) \\ &= (1+\epsilon)\lambda\left\{\frac{\xi^2}{\Psi\xi_0^2}\right\} \\ &= (1+\epsilon)\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\frac{\xi}{\Psi\xi_0} \end{aligned} \tag{B.3}$$

using the definition of the bounce average (2.64). Then since $\xi = \sigma\sqrt{1-\Psi(1-\xi_0^2)}$,

$$\mathcal{D} = (1+\epsilon)\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\sqrt{\frac{1-\Psi(1-\xi_0^2)}{\Psi^2\xi_0^2}}. \tag{B.4}$$

We can rewrite Ψ in a convenient form using the trigonometric identity $\cos\theta = 1-2\sin^2(\theta/2)$:

$$\begin{aligned}\Psi &= \frac{1+\epsilon}{1+\epsilon\cos\theta}\\ &= \frac{1+\epsilon}{1+\epsilon-2\epsilon\sin^2(\theta/2)}\\ &= \frac{1}{1-[2\epsilon/(1+\epsilon)]\sin^2(\theta/2)}\\ &= \frac{1}{1-\xi_{0T}^2\sin^2(\theta/2)}\end{aligned} \tag{B.5}$$

where the trapped-passing boundary pitch-angle ξ_{0T} is $\sqrt{2\epsilon/(1+\epsilon)}$. Substituting into (B.4) gives

$$\mathcal{D} = (1+\epsilon)\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\sqrt{\frac{1}{\xi_0^2}[1-\xi_{0T}^2\sin^2(\theta/2)]^2 - \frac{1-\xi_0^2}{\xi_0^2}[1-\xi_{0T}^2\sin^2(\theta/2)]}. \tag{B.6}$$

Now we expand the integrand of $\mathcal{D}$ to first order in ξ_{0T}^2 only, using the fact that $\epsilon = \xi_{0T}^2/(2-\xi_{0T}^2) \approx \xi_{0T}^2/2$:

$$\begin{aligned}\mathcal{D} &\approx (1+\epsilon)\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\sqrt{\frac{1}{\xi_0^2}[1-2\xi_{0T}^2\sin^2(\theta/2)] - \frac{1-\xi_0^2}{\xi_0^2}[1-\xi_{0T}^2\sin^2(\theta/2)]}\\ &\approx \left(1+\frac{\xi_{0T}^2}{2}\right)\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\sqrt{\frac{1}{\xi_0^2} - \frac{1-\xi_0^2}{\xi_0^2} + \xi_{0T}^2\sin^2(\theta/2)\left(-2\frac{1}{\xi_0^2} + \frac{1-\xi_0^2}{\xi_0^2}\right)}\\ &= \left(1+\frac{\xi_{0T}^2}{2}\right)\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\sqrt{1+\xi_{0T}^2\sin^2(\theta/2)\left(-1-\frac{1}{\xi_0^2}\right)}\\ &\approx \left(1+\frac{\xi_{0T}^2}{2}\right)\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\left[1+\frac{1}{2}\xi_{0T}^2\sin^2(\theta/2)\left(-1-\frac{1}{\xi_0^2}\right)\right]\\ &= \left(1+\frac{\xi_{0T}^2}{2}\right)\left[1+\frac{1}{4}\xi_{0T}^2\left(-1-\frac{1}{\xi_0^2}\right)\right]\end{aligned}$$

$$\approx \quad 1 + \frac{1}{4}\xi_{0T}^2 - \frac{1}{4}\frac{\xi_{0T}^2}{\xi_0^2}. \tag{B.7}$$

Then the integral I is

$$\begin{aligned} I(\xi_0) &= \int_{\xi_{0T}}^{\xi_0} d\xi_0 \frac{1}{\mathcal{D}} \\ &\approx \int_{\xi_{0T}}^{\xi_0} \frac{d\xi_0}{1 + \frac{1}{4}\xi_{0T}^2 - \frac{1}{4}\frac{\xi_{0T}^2}{\xi_0^2}} \\ &\approx \int_{\xi_{0T}}^{\xi_0} d\xi_0 \left(1 - \frac{1}{4}\xi_{0T}^2 + \frac{1}{4}\frac{\xi_{0T}^2}{\xi_0^2}\right) \\ &= \left(1 - \frac{1}{4}\xi_{0T}^2\right)(\xi_0 - \xi_{0T}) + \frac{1}{4}\xi_{0T}^2\left(-\frac{1}{\xi_0} + \frac{1}{\xi_{0T}}\right) \\ &= \left(1 - \frac{1}{4}\xi_{0T}^2 + \frac{1}{4}\frac{\xi_{0T}}{\xi_0}\right)(\xi_0 - \xi_{0T}). \end{aligned} \tag{B.8}$$

Finally we note that $\xi_{0T}^2 \ll 1$ allows us the neglect the second term, and we get the final approximation for I:

$$I(\xi_0) \quad \approx \quad \left(1 + \frac{1}{4}\frac{\xi_{0T}}{\xi_0}\right)(\xi_0 - \xi_{0T}). \tag{B.9}$$

Appendix C

Plasma Parameters for FASTFP-NC Calculations

The plasma parameters used in this work are taken to approximate existing tokamaks or, in the case of ARIES-RS, a steady-state tokamak design. However, the parameters are taken to have higher density and temperature than present day experiments, so that the bootstrap current is higher and the potential for synergism with RF current drive is greater. We neglect the Shafronov shift in all of these cases.

C.1 Alcator C-Mod Type Parameters

This set of plasma parameters is chosen to match a hypothetical high bootstrap current scenario in the Alcator C-Mod tokamak [77]. This scenario would increase the bootstrap current by having a low magnetic field, with the field on axis B_a of about 4.0 Tesla. The plasma current will be provided primarily by LHCD and the bootstrap current, with the fraction due to bootstrap being about 70%. The peak density of the plasma is taken to be 2.5×10^{20} m^{-3}, and the central electron temperature is 7.5 keV.

The remaining plasma parameters are detailed in Table C.1. In this scenario, the lower hybrid current will be located off of the axis, at a minor radius of about 0.15m, which gives a value of $\epsilon = 0.23$.

Parameter	Notation	Value
Major Radius	R_0	0.68 m
Minor Radius	a	0.22 m
Max. Electron Density	n_{e0}	2.5×10^{20} m^{-3}
Electron Density Profile	$n_e(r)$	$n_{e0}(0.8[1-(r/a)^2]^{0.5}+0.2)$
Max. Electron Temperature	T_{e0}	7.5 keV
Electron Temp. Profile	$T_e(r)$	$T_{e0}[1-(r/a)^2]^2$
Max. Ion Temperature	T_{i0}	4.0 keV
Ion Temp. Profile	$T_i(r)$	$T_{i0}[1-(r/a)^2]^2$
Magnetic Field on Axis	B_a	4.0 T
Ion Charge Number	Z_i	1

Table C.1: Plasma parameters of FASTFP-NC runs based on Alcator C-Mod in high-bootstrap LHCD mode

Parameter	Notation	Value
Major Radius	R_0	5.52 m
Minor Radius	a	1.38 m
Max. Electron Density	n_{e0}	6.5×10^{20} m^{-3}
Electron Density Profile	$n_e(r)$	$n_{e0}(0.8[1-(r/a)^2]^{0.6}+0.2)$
Max. Electron Temperature	T_{e0}	27.0 keV
Electron Temp. Profile	$T_e(r)$	$T_{e0}[1-(r/a)^2]^{1.2}$
Max. Ion Temperature	T_{i0}	20.0 keV
Ion Temp. Profile	$T_i(r)$	$T_{i0}[1-(r/a)^2]^{1.2}$
Magnetic Field on Axis	B_a	8.0 T
Ion Charge Number	Z_i	1

Table C.2: Plasma parameters of FASTFP-NC runs based on ARIES-RS design plans.

C.2 ARIES-RS Type Parameters

ARIES-RS [50] is the most recent result of the ARIES [49] design study to determine the feasibility and practicality of magnetic confinement concepts as fusion power plants. This tokamak design contains a reversed shear configuration and is able to achieve both high β and a bootstrap current fraction of 88%.

Parameter	Notation	Value
Major Radius	R_0	1.67 m
Minor Radius	a	0.67 m
Max. Electron Density	n_{e0}	4.3×10^{19} m^{-3}
Electron Density Profile	$n_e(r)$	$n_{e0}(0.76[1-(r/a)^2]^{1.3}+0.24)$
Max. Electron Temperature	T_{e0}	7.8 keV
Electron Temp. Profile	$T_e(r)$	$T_{e0}[1-(r/a)^2]^{1.1}$
Max. Ion Temperature	T_{i0}	6.2 keV
Ion Temp. Profile	$T_i(r)$	$T_{e0}[1-(r/a)^2]^{1.1}$
Magnetic Field on Axis	B_a	2.0 T
Ion Charge Number	Z_i	1

Table C.3: Plasma parameters of FASTFP-NC runs based on DIII-D tokamak with high pressure and temperature.

C.3 DIII-D Type Parameters

This calculation uses the parameters shown in Table C.3 which are consistent with plasmas in the tokamak DIII-D. This choice was motivated by ongoing experiments in ECCD [93]. Reference [93] gives the MHD equilibrium for a high-bootstrap current scenario that would be possible with ECCD.

Appendix D

Codes Used in This Work

D.1 The FASTFP-NC Code

```
      program fastfpnc
cccc Calculation of the total non-inductive current density
cccc due to RFCD and bootstrap current, from an electron distribution
cccc function expanded in small parameters according to
cccc neoclassical theory.
cccc Written by Steven D. Schultz as part of a Ph.D. thesis
cccc at the Massachusetts Institute of Technology.
cccc Incorporates the FASTFP program
cccc written by M. Shoucri and I. Shkarofsky.
cccc Completed July 1999.                                          10
      implicit double precision (a-h,o-z)
      parameter(c=2.9979d8,pi=3.14159265359)
      parameter(elec=1.602d-19,emass=9.1094d-31)
      parameter(nmax=20)
      parameter(n2=240,n1=480)
      double precision p(n1,n2),f0(n1,n2),g(n1,n2)
      double precision nul(n1,n2),dout(n1,n2)
      integer matOpen,matPutFull,matClose
      common /inp/ eps,r0,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass                   20
      common /times/ pmax,atmax,gamc,ntmax
```

```
cccc Note: Following the notation of original FASTFP code
cccc YMU refers to pitch angle coordinate p_{parallel}/p
cccc In thesis, this is referred to as XI
      double precision ymu(n2),tlam(n2),ba1(n2),xp(n1)
      double precision tlamc(n2),ba1c(n2)
      common /grid/ ymu,xp,tlam,ba1,tlamc,ba1c
      double precision epsn,ccol,rho,dens,te,ti,q,wpm
      common /surf/ epsn,ccol,rho,dens,te,ti,q,wpm,lp
      common /qllh/ v1,v2,dpara,ifw                                    30
      common /qlec/ omc,theta,dperp,dnres,dnpar

      itype=0
c itype=0 - Full Calculation of RF+bootstrap
c itype=1 - TEST CASE - ELIMINATED FROM FINAL VERSION OF CODE
c itype=2 - Use only a density gradient
c itype=3 - Find the lowest order (F0) distribution only
c           No neoclassical effects
c itype=4 - Bootstrap Current Only
c itype=5 - Calculation of L31 (transport coeff.)                      40
c           Bootstrap current w/ density gradient only
c itype=6 - Spitzer-H"arm problem
c itype=7 - Rosenbluth, Hazeltine, & Hinton distribution functions

cccc Read initial parameters
      call fct_init(itype,alimit)
           if (itype.eq.1) itype=0
cccc Output files
      open (unit=8,status='unknown',file='jbs.out')
      if (itype.eq.4) open(unit=3,status='unknown',file='l32.out')     50
      if (itype.eq.5) open(unit=3,status='unknown',file='l31.out')
      if (itype.eq.6) open(unit=3,status='unknown',file='sig.out')

      ifile=0
cccc Comment these lines for output redirection to screen
      open (unit=6,status='unknown',file='ffp.log')
      ifile=1
```

```
cccc Matlab file with info on convergence of code
      open (unit=13,status='unknown',file='converge.m')
      write (13,*) 'a=['                                                    60

cccc Read input value of  minor radius r from file 'rinput'
      open (unit=2,status='unknown',file='rinput')
      read (2,*) npmax
      if (npmax.gt.nmax) npmax=nmax
      if (itype.lt.4) npmax=0
      nprof=0
cccc Start loop here for multiple values of r when npmax != 0
cccc for bootstrap current profiles in r
 2    continue                                                              70
 1    continue
      read (2,*,end=99) rsurf
      epsc=rsurf/r0
      if ((epsc.gt.eps).or.(epsc.lt.0.))  goto 1

cccc GRIDSET defines the grid in P,MU space
cccc and defines EMI,EPL the values of  epsilon  for
cccc neighboring flux surfaces used to perform radial derivative
      call gridset(epsc,emi,epl)
                                                                            80
      do 10 i=1,n1
         do 8 j=1,n2
            p(i,j)=0.d0
            f0(i,j)=0.d0
            g(i,j)=0.d0
            nul(i,j)=0.d0
 8       continue
 10   continue

      if (itype.eq.7) then                                                  90
cccc An analytic expression for the Z>>1 distributions is
cccc given by Rosenbluth, Hazeltine, and Hinton.    We can
cccc output this distribution for comparison.
```

```
            call rhh(p,g,n1,n2,bth,dpr,tpr)
            call moments(p,n1,n2,1.d0,bth,denf,ajf,-1)
            call moments(g,n1,n2,1.d0,bth,deng,ajg,1)
            sifac=elec*dens*DSQRT(te/511.d0)*c
            goto 30
         elseif (itype.eq.6) then
cccc On output, f0 is the Spitzer-H"arm function with                    100
cccc trapped electrons explicitly excluded.  (Calc. of neoclassical
cccc conductivity).
         ipert=40
         if (npmax.ne.0) ipert=41
         call srface(epsc,epsc,adum1,adum2)
         call fastfp(1.d0,ntmax,pmax,te,ti,atmax,zb,mass,epsn*r0,r0,
     &        v1,v2,0.d0,ifw,omc,theta,0.d0,dnres,dnpar,
     &        f0,n1,n2,ipert,bth,dn0,aj0,pabs0,nul,0.d0,alimit,dout)
         if (npmax.ne.0) then
            call moments(f0,n1,n2,1.d0,bth,adn0,aaj0,1)                  110
            sigr=7.37d0*(1.d0-2.09d0*DSQRT(epsc))
            write (6,*) aaj0,sigr
            write (3,*) aaj0,sigr
         endif
         goto 30
      endif

      if (itype.eq.3) goto 19

cccc Bootstrap Current Only calculation                                  120
      if ((itype.eq.2).or.(itype.eq.5)) then
         call srface(emi,epsc,sden1,svm1)
         call srface(epl,epsc,sden2,svm2)
         goto 9
      endif

cccc Calculation on neighboring flux surfaces
      call srface(emi,epsc,sden1,svm1)
      call fastfp(1.d0,1,pmax/svm1,te,ti,atmax,zb,mass,epsn*r0,r0,
```

```
     &     v1/svm1,v2/svm1,0.d0,ifw,omc,theta,0.d0,dnres,dnpar,                130
     &     g,n1,n2,0,bth,den1,aj1,dum1,nul,0.d0,alimit,dout)

      call srface(epl,epsc,sden2,svm2)
      call fastfp(1.d0,1,pmax/svm2,te,ti,atmax,zb,mass,epsn*r0,r0,
     &     v1/svm2,v2/svm2,0.d0,ifw,
     &     omc,theta,0.d0,dnres,dnpar,
     &     p,n1,n2,0,bth,den2,aj2,dum1,nul,0.d0,alimit,dout)

 9    continue
      call srface(epsc,epsc,adum1,adum2)                                       140
      call fastfp(1.d0,1,pmax,te,ti,atmax,zb,mass,epsn*r0,r0,
     &     v1,v2,dpara,ifw,omc,theta,dperp,dnres,dnpar,
     &     f0,n1,n2,0,bth,dn0,aj0,dum1,nul,0.d0,alimit,dout)
      call moments(f0,n1,n2,1.d0,bth,adn0,aaj0,1)

cccc SIFAC = unit current density in SI units
      sifac=elec*dens*bth*c

cccc Numerical calculation of density and temperature gradients
cccc from profiles given                                                       150
      rdp=epl-epsc
      rdm=emi-epsc
      rd=rdp-rdm
      dpr=-rdm/rdp/rd*sden2-(rdp+rdm)/rdp/rdm
     &          +rdp/rdm/rd*sden1
      tpr=-rdm/rdp/rd*svm2*svm2-(rdp+rdm)/rdp/rdm
     &          +rdp/rdm/rd*svm1*svm1
      if ((itype.eq.2).or.(itype.eq.5)) then
c        DENSITY GRADIENT ONLY
        dpr=dpr+tpr                                                            160
        tpr=0.d0
      endif
      write(6,*) 'd(ln n)/dr,d(ln T)/dr=',dpr,tpr

cccc CALCULATION FROM HIRSHMAN, PHYS. FLUIDS 1988
```

```
cccc To be used in including ion neoclassical flow (upi)
      bj0=-elec*dens*c*bth*c*bth/wpm/r0/(1.d0+epsc)
      xx=(1.46d0*DSQRT(epsc)+2.40d0*epsc)/DSQRT((1.d0-epsc)**3)
      dx=1.414d0*zb + zb*zb + xx*(0.754 + 2.657*zb + 2.*zb*zb)
     &     +xx*xx*(0.348 + 1.243*zb + zb*zb)                          170
      al31=xx*(0.754 + 2.21*zb + zb*zb
     &     + xx*(0.348 + 1.243*zb + zb*zb))/dx
      al32=-xx*(0.884 + 2.074*zb)/dx
      ali=-1.172d0/(1.d0 + 0.462*xx)

cccc CALCULATION FROM SAUTER, ET.AL. PHYS. PLASMAS 1999
cccc To be used in including ion neoclassical flow (upi)
      xft=1-(1-epsc)**2/DSQRT(1-epsc**2)/(1+1.46*DSQRT(epsc))
      all32=-(.51+1.31*zb)/zb/(1+.44*zb)*(xft-xft**4)
     &     +(5.95+3.57*zb)/(1+.22*zb)/(1+2.48*zb)*(xft**2-xft**4)     180
     &     -(3.92+3.57*zb)/(1+.22*zb)/(1+2.48*zb)*(xft**3-xft**4)

cccc Ion neoclassical flow (from Hirshman, P.F. 1988)
      upi=bj0*(ti/te/zb)*(dpr+tpr+ali*tpr)/(-elec*dens*c*bth)
      write (6,*) 'UPI=',upi

cccc Renormalization of distributions on neighboring flux surf.
cccc to units of central flux surf.
      do 13 i=1,n1
        do 11 j=1,n2                                                  190
          g(i,j)=g(i,j)*sden1/(svm1**3)
          p(i,j)=p(i,j)*sden2/(svm2**3)
 11     continue
 13   continue

cccc Calculation of df0/dr  and f~
      pfac=c*bth/wpm/r0
      do 16 i=1,n1
        do 15 j=2,n2-1
          ppr=-rdm/rdp/rd*p(i,j)-(rdp+rdm)/rdp/rdm*f0(i,j)             200
     &          +rdp/rdm/rd*g(i,j)
```

```
          if ((itype.eq.2).or.(itype.eq.5)) then
            ppr=f0(i,j)*dpr
          endif
          p(i,j)=-xp(i)*ymu(j)*pfac*ppr
15       continue
        p(i,1)=p(i,3)
        p(i,n2)=p(i,n2-2)
16     continue
      call moments(p,n1,n2,1.d0,bth,adenf,aajf,-1)                    210

cccc Calculation of g for bootstrap current only case
      call fastfp(1.d0,ntmax,pmax,te,ti,atmax,zb,mass,epsn*r0,r0,
     &     v1,v2,0.d0,ifw,omc,theta,0.d0,dnres,dnpar,
     &     g,n1,n2,1,bth,deng,ajg,dum1,p,upi,alimit,dout)
      call moments(g,n1,n2,1.d0,bth,adeng,aajg,1)

cccc NOTE: Densities are labeled as "aden", current dens. as "aaj"
      aajp=aajf+aajg-upi
      aajpx=aajp                                                      220

cccc The following expression for the bootstrap current (w/o RF)
cccc is used for comparision of the numerical results
      ajbs=bj0*(al31*(dpr+tpr+(ti/te/zb)*(dpr+tpr+ali*tpr))
     &     +al32*tpr)
      ajbs1=bj0*(dpr+tpr)
      ajbs2=ajbs-ajbs1
      ajll=bj0*(al31*(dpr+tpr+(ti/te/zb)*(dpr+tpr+ali*tpr))
     &     +all32*tpr)
      write (6,*) 'BOOTSTRAP CURRENT:'                                230
      write (6,*) 'NUMER.',aajf*sifac,(aajg-upi)*sifac,aajp*sifac
      write (6,*) 'HIRSH.',ajbs1,ajbs2,ajbs
      write (6,*) 'SAUTER',ajbs1,ajll-ajbs1,ajll
cccc Calculation of the transport coefficient L31
cccc CALCULATION FROM HIRSHMAN, PHYS. FLUIDS 1988
      if (itype.eq.5) then
        bl31=-(1.d0+epsc)/pfac*aajp/dpr/(1.d0+ti/te/zb)
```

```
      write (6,*) 'L31/sqrt(epsilon)=',bl31/DSQRT(epsc)
      write (6,*) 'L31/sq(eps)-hirsh=',al31/DSQRT(epsc)
      write (3,*) epsc,bl31/DSQRT(epsc),al31/DSQRT(epsc)                    240
    endif
cccc Calculation of the transport coefficient L31
cccc CALCULATION FROM HIRSHMAN, PHYS. FLUIDS 1988
cccc  AND SAUTER, ET.AL. 1999
    if ((itype.eq.4).and.(ti/te.lt.0.01)) then
      bl32=(-(1.d0+epsc)/pfac*aajp-al31*(dpr+tpr))/tpr
      write (6,*) 'L32/sqrt(epsilon)=',bl32/DSQRT(epsc)
      write (6,*) 'L32/sq(eps)-hirsh=',al32/DSQRT(epsc)
      write (6,*) 'L32/sq(eps)sauter=',all32/DSQRT(epsc)
      write (3,*) epsc,bl32/DSQRT(epsc),al32/DSQRT(epsc)                    250
     &           ,all32/DSQRT(epsc)
    endif

    if (itype.ge.4) goto 30

cccc Calculation with RF waves:

    if (itype.eq.2) goto 19
cccc Calculation on neighboring flux surfaces
    call srface(emi,epsc,sden1,svm1)                                        260
    call fastfp(1.d0,ntmax,pmax/svm1,te,ti,atmax,zb,mass,epsn*r0,r0,
     &     v1/svm1,v2/svm1,dpara*svm1/sden1,ifw,
     &     omc,theta,dperp*svm1/sden1,dnres,dnpar,
     &     g,n1,n2,0,bth,den1,aj1,dum1,nul,0.d0,alimit,dout)
    call moments(g,n1,n2,svm1,bth,dum1,dum2,1)

    call srface(epl,epsc,sden2,svm2)
    call fastfp(1.d0,ntmax,pmax/svm2,te,ti,atmax,zb,mass,epsn*r0,r0,
     &     v1/svm2,v2/svm2,dpara*svm2/sden2,ifw,
     &     omc,theta,dperp*svm2/sden2,dnres,dnpar,                        270
     &     p,n1,n2,0,bth,den2,aj2,dum1,nul,0.d0,alimit,dout)
    call moments(p,n1,n2,svm2,bth,dum1,dum2,1)
```

```
19  continue
    call srface(epsc,epsc,adum1,adum2)
    call fastfp(1.d0,ntmax,pmax,te,ti,atmax,zb,mass,epsn*r0,r0,
   &     v1,v2,dpara,ifw,omc,theta,dperp,dnres,dnpar,
   &     f0,n1,n2,0,bth,dn0,aj0,pabs0,nul,0.d0,alimit,dout)
    call moments(f0,n1,n2,1.d0,bth,adn0,aaj0,1)
    sifac=elec*dens*bth*c                                          280
    write (6,*) 'RF curr=',aaj0*sifac
    write (6,*) 'Normalized power=',pabs0
    write (6,*) 'J/P=',aaj0/pabs0
    if (itype.eq.3) goto 30

cccc Renormalization of distributions on neighboring flux surf.
cccc to units of central flux surf.
    do 23 i=1,n1
      do 21 j=1,n2
        g(i,j)=g(i,j)*sden1/(svm1**3)                              290
        p(i,j)=p(i,j)*sden2/(svm2**3)
21     continue
23  continue

cccc Calculation of expected J^S/J^R for LHCD cases
cccc (see Chapter 5 of thesis)
    if (dpara.gt.1.d-12)
   &    write(6,*) 'Kappa=',sqrt(epsc)/2*pfac*tpr*(v1**3)

cccc Calculation of df0/dr  and f~                                 300
    do 26 i=1,n1
      do 25 j=2,n2-1
        ppr=-rdm/rdp/rd*p(i,j)-(rdp+rdm)/rdp/rdm*f0(i,j)
   &          +rdp/rdm/rd*g(i,j)
        if (itype.eq.2) ppr=f0(i,j)*dpr
        p(i,j)=-xp(i)*ymu(j)*pfac*ppr
25     continue
      p(i,1)=p(i,3)
      p(i,n2)=p(i,n2-2)
```

```
 26   continue                                                          310
      call moments(p,n1,n2,1.d0,bth,adenf,aajf,-1)

cccc Calculation of g for bootstrap current only case
      call fastfp(1.d0,ntmax,pmax,te,ti,atmax,zb,mass,epsn*r0,r0,
     &     v1,v2,dpara,ifw,omc,theta,dperp,dnres,dnpar,
     &     g,n1,n2,1,bth,deng,ajg,pabs1,p,upi,alimit,dout)
      call moments(g,n1,n2,1.d0,bth,adeng,aajg,1)
cccc Correct boundary conditions for g -- force the overall density
cccc to stay constant (see Chapter 4 of thesis)
      cnorm=-(adenf+adeng)/adn0                                         320
      write (6,*) 'CNORM=',cnorm
      aajg=aajg+cnorm*aaj0
      pabs1=pabs1+cnorm*pabs0
      do 28 i=1,n1
        do 27 j=1,n2
          g(i,j)=g(i,j)+cnorm*f0(i,j)
 27     continue
 28   continue
      aajp=aajf+aajg-upi
                                                                        330
cccc Screen output of current densities, power densities, J/P
      write (6,*) 'F1 curr=',aajp*sifac
      write (6,*) 'Total curr=',(aaj0+aajp)*sifac
      write (6,*) 'Synergistic curr=',(aajp-aajpx)*sifac
      write (6,*) 'Normalized power=',pabs1
      write (6,*) '(J-Jb)/P=',(aaj0+aajp-aajpx)/(pabs0+pabs1)

 30   continue
cccc Output to JBS.OUT
      write (8,199) epsc,aaj0*sifac,aajp*sifac,                         340
     &     aajpx*sifac,ajbs,ajll
 199  format(6(1x,e9.3))

cccc Plasma parameter outputs
      write (6,*) 'Density=',dens
```

```
      write (6,*) 'E. temp=',te,'  I. temp=',ti
      write (6,*) 'Epsilon=',epsc
      write (6,*) 'Current units (A/m^2):  ',sifac
      write (6,*) '----------------------------BREAK'
                                                                      350
cccc Iterate the loop when multiple values of r (in file RINPUT)
cccc are to be studied
      if (npmax.ne.0) then
        nprof=nprof+1
        if (nprof.ge.npmax) goto 99
        goto 2
      endif

cccc OUTPUT SECTION: x=grid in p,   y=grid in mu
cccc All matrices are written in the form of Matlab                   360
cccc formatted file in the form of complex numbers
      mfp=matOpen('kout.mat','w')
      mstat=matPutFull(mfp,'x',n1,1,xp,nul)
      mstat=matPutFull(mfp,'y',n2,1,ymu,nul)
      mstat=matPutFull(mfp,'f0',n1,n2,f0,nul)
      mstat=matPutFull(mfp,'ft',n1,n2,p,nul)
      mstat=matPutFull(mfp,'g',n1,n2,g,nul)
      if (itype.lt.4) mstat=matPutFull(mfp,'d',n1,n2,dout,nul)
      mstat=matClose(mfp)
                                                                      370
 99   continue
      write (13,*) '];'
      close (unit=13)
      close (unit=2)
      close (unit=8)
      close (unit=3)

      if (ifile.eq.0) read(5,*) i
      stop
      end                                                             380
```

```
      subroutine fct_init(itype,alimit)
cccc Reads the input deck file 'f3din' which contains
cccc a namelist of parameters -- see below
      implicit double precision (a-h,o-z)
      parameter(c=2.9979d8,pi=3.14159265359)
      parameter(n2=240,n1=480)
      common /inp/ eps,r0,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass
      common /qllh/ v1,v2,dpara,ifw                                    390
      common /qlec/ omc,theta,dperp,dnres,dnpar
      common /times/ pmax,atmax,gamc,ntmax
      namelist /fcjin/ a,r0,mass,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,b0,qmin,qmax,eq,
     &     v1,v2,dpara,ifw,pmax,ntmax,atmax,
     &     omc,theta,dperp,dnres,dnpar,itype,alimit
      open(unit=2,status='old',file='f3din')
      read (2,fcjin)
cccc  a    = Minor Radius
cccc  r0   = Major Radius                                              400
cccc  mass = Mass of background ions (amu)
cccc  zb   = Charge of background ions
cccc  den0 = Peak electron density (n_0)
cccc  dene = Fraction of density at plasma edge (n_edge/n_0)
cccc  gamn = Exponent of density profile (1-(r/a)^2)^gamn
cccc  te0  = Peak electron temperature (T_0)
cccc  ti0  = Peak ion temperature
cccc  tie  = Fraction of temp. at plasma edge (T_edge/T_0)
cccc  gamt = Exponent of temp. profile (1-(r/a)^2)^gamt
cccc  b0   = Magnetic Field on Axis                                    410
cccc  qmin = Safety Factor at Edge
cccc  qmax = Safety Factor on Axis
cccc  eq   = Exponent for q(r) profile (r/a)^eq
cccc  v1, v2 = Limits of LH spectrum
cccc  dpara = LH Diffusion coefficient (thermal units)
cccc  ifw  = Flag for Fast Alfven Wave diffusion (not used)
cccc  pmax = Maximum momentum (p) for grid
```

```
cccc  ntmax = Maximum number of time steps
cccc  atmax = Maximum "time" for run    (Time Step = atmax)
cccc  omc = Ratio n*wc/w at EC wave damping                                  420
cccc  theta = Poloidal angle at which ECCD occurs
cccc  dperp = EC diffusion coefficient (maximum value)
cccc  dnres = N_{parallel} of EC waves
cccc  dnpar = Delta N_{parallel} of EC wave spectrum
cccc  itype = Type of FASTFP-NC run (see start of code)
cccc  alimit = Convergence criterion << 1
c     a=0.5d0
c     r0=2.5d0
c     mass=12
c     zb=6.d0                                                                430
c     den0=1.d20
c     dene=5.d-2
c     gamn=0.5d0
c     te0=10.0
c     ti0=10.0
c     tie=1.d-3
c     gamt=2.d0
c     wc0=7.d11
c     qmin=1.01d0
c     qmax=3.01d0                                                            440
c     eq=3.d0
      eps=a/r0
      amu=5.446d-4/mass
      gamc=8.06d5*zb*zb*15.
      ifw=0

      wc0=b0*1.7588d11
      wp0=DSQRT(den0)*56.415

      close (unit=2)                                                         450

      return
      end
```

```
      subroutine gridset(epsc,emi,epl)
cccc Produces the momentum-space grid for use by the program.
      implicit double precision (a-h,o-z)
      parameter(pi=3.14159265359)
      parameter(n2=240,n1=480)
      common /inp/ eps,r0,zb,den0,dene,gamn,                              460
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass
      common /times/ pmax,atmax,gamc,ntmax
      double precision ymu(n2),tlam(n2),ba1(n2),xp(n1)
      double precision tlamc(n2),ba1c(n2)
      common /grid/ ymu,xp,tlam,ba1,tlamc,ba1c

      dmu=2.d0/DBLE(n2-2-1)
      do 2 j=2,n2-1
        ymu(j)=-1.d0 + dmu*DBLE(j-2)
2     continue                                                            470
      ymu(1)=ymu(3)
      ymu(n2)=ymu(n2-2)

cccc Values of EPSILON are shifted slightly to optimally fit
cccc the spacing on the grid in ymu
      epsco=epsc
      ymut=SQRT(2.d0*epsc/(1.d0+epsc))
      lp=NINT((1.d0+ymut)/dmu) + 2
      ymut=ymu(lp)
      epsc=(1.d0/(2.d0/ymut/ymut-1.d0))                                   480
      ymut=ymu(lp-1)
      epl=(1.d0/(2.d0/ymut/ymut-1.d0))
      ymut=ymu(lp+1)
      emi=(1.d0/(2.d0/ymut/ymut-1.d0))
      depse=(epl-epsc)/1024.d0
      epsc=epsc+1.*depse
      depse=0.008*eps
      epl=epsc+depse
      emi=epsc-depse
```

```
                                                                      490
      vk=0.d0
      delv=pmax/DBLE(n1-1)
      do 30 i=1,n1
        xp(i)=vk
        vk=vk+delv
 30   continue

      return
      end
                                                                      500
      subroutine srface(epst,epsc,sden,svm)
cccc Sets values of all flux surface quantites, and also gives
cccc quantities SDEN and SVM used to renormalize velocity and
cccc density for different flux surfaces
      implicit double precision (a-h,o-z)
      parameter(pi=3.14159265359)
      parameter(n2=240,n1=480)
      parameter(mmax=99,ees=1.d-07)
      common /inp/ eps,r0,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass                   510
      common /times/ pmax,atmax,gamc,ntmax
      double precision ymu(n2),tlam(n2),ba1(n2),xp(n1)
      double precision tlamc(n2),ba1c(n2)
      common /grid/ ymu,xp,tlam,ba1,tlamc,ba1c
      double precision epsn,ccol,rho,dens,te,ti,q,wpm
      common /surf/ epsn,ccol,rho,dens,te,ti,q,wpm,lp

      epsn=epst
      rho=epsn/eps
      dens=den0*((1.d0-dene)*(1.d0-rho**2)**gamn + dene)            520
      ti=ti0*((1.d0-tie)*(1.d0-rho**2)**gamt + tie)
      rhoc=epsc/eps
      denc=den0*((1.d0-dene)*(1.d0-rhoc**2)**gamn + dene)
      tic=ti0*((1.d0-tie)*(1.d0-rhoc**2)**gamt + tie)
      te=te0/ti0*ti
```

```
      sden=dens/denc
      svm=DSQRT(ti/tic)
      ccol=-gamc*(dens/zb)
      q=(qmax-qmin)*(rho**eq)+qmin
      wpm=(wc0/(1.d0+epsn))*epsn/q                                    530

ccc COMMENT  tlam is Tb*v||0
ccc          ba1 is <<1/psi - 1>> = <<-mu0t**2*SIN^2(theta/2)>>
      tmu2=2.d0*epsn/(1.d0+epsn)
      ymut=SQRT(tmu2)
      lp=NINT((1.d0-ymut)*(n2-2-1)/2.d0) + 2
      do 40 j=2,n2/2
        if (j.ne.lp) then
          ymu0=ymu(j)
          call bavg(ymu0,tmu2,q,tlm,b1,tlp)                           540
          tlam(j)=tlm
          ba1(j)=b1
          tlam(n2+1-j)=tlm
          ba1(n2+1-j)=b1
        else
CC  Accurate calculation near TRAP/PASS boundary  CC
          ymup=(ymu(lp)+ymu(lp+1))/2.
          ymum=(ymu(lp)+ymu(lp-1))/2.
          deltap=DABS(ymum)/DSQRT(tmu2) - 1.d0
          deltam=1.d0 - DABS(ymup)/DSQRT(tmu2)                        550
          delta=deltap+deltam
          tlm=2.d0/pi*DSQRT(1.d0-tmu2)
     &          *(1.539721d0-0.5d0*DLOG(deltam)
     &          -0.5d0*(deltap/delta)*DLOG(deltap/deltam))
          alpha=-0.5d0
          bm2=0.d0
          bm1=1.d0
          tmu2m=tmu2
          tlm=tlm-2.d0/pi*alpha*bm1*tmu2m
          do 49 m=2,mmax                                              560
            tmu2m=tmu2m*tmu2
```

```
          alpha=alpha*DBLE(2*m-3)/DBLE(2*m)
          beta=(2.d0*DBLE(2*m-2)*bm1-DBLE(2*m-3)*bm2)/DBLE(2*m-1)
          dlam=-2.d0/pi*alpha*beta*tmu2m
          tlm=tlm+dlam
          if (DABS(dlam/tlm).lt.ees) goto 50
          bm2=bm1
          bm1=beta
49        continue
          stop                                                          570
50        continue
          tlam(lp)=tlm
          ba1(lp)=-tmu2
          tlam(n2+1-lp)=tlm
          ba1(n2+1-lp)=-tmu2
        endif
        if (j.ne.n2/2) then
          ymuc=(ymu(j)+ymu(j+1))/2.
          call bavg(ymuc,tmu2,q,tlmc,b1c,dummy)
          tlamc(j)=tlmc                                                 580
          ba1c(j)=b1c
          tlamc(n2-j)=tlmc
          ba1c(n2-j)=b1c
        else
          tlamc(j)=0.d0
          ba1c(j)=0.d0
        endif
40    continue
      tlam(1)=tlam(3)
      ba1(1)=ba1(3)                                                     590
      tlamc(1)=tlamc(2)
      ba1c(1)=ba1c(2)
      tlam(n2)=tlam(n2-2)
      ba1(n2)=ba1(n2-2)
      tlamc(n2-1)=tlamc(n2-2)
      ba1c(n2-1)=ba1c(n2-2)
```

```
      return
      end
                                                                      600
      subroutine bavg(ymu,tmu2,q,tlm,b1,tlp)
cccc Calculates the bounce time and other bounce averaged
cccc parameters from a series in elliptic integrals
      implicit double precision (a-h,o-z)
      parameter(mmax=99,ees=1.d-10)
      parameter(pi=3.14159265359)
      double precision ymu,tmu2,tlm,b1
      common /inp/ eps,r0,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass
      alpha=-0.5d0                                                    610
      x=tmu2/(ymu**2)
      itrap=0
      if (x.lt.1.d0) then
        y2=tmu2
      else
        itrap=1
        y2=ymu**2
        x=1.d0/x
        if (x.ge.1.d0) x=.9999999d0
      endif                                                           620
      ajm2=DELK(x)
      ajm=(DELK(x)-DELE(x))/x
          y2m=y2
      tlm=ajm2+alpha*y2m*ajm
      b1=y2m*ajm
      do 35 m=2,mmax
        ajm4=ajm2
        ajm2=ajm
        ajm=(DBLE(2*m-2)*(1.d0+1.d0/x)*ajm2-DBLE(2*m-3)*ajm4/x)
     &       /DBLE(2*m-1)                                           630
        y2m=y2m*y2
        dba1=alpha*y2m*ajm
        b1=b1+dba1
```

```
      alpha=alpha*DBLE(2*m-3)/DBLE(2*m)
      dlam=alpha*y2m*ajm
      tlm=tlm+dlam
      am=DABS(dlam/tlm)
      am2=DABS(dba1/b1)
      if (am2.gt.am) am=am2
      if (am.lt.ees) goto 39                                          640
 35   continue
      stop
 39   b1=-b1/tlm
      tlm=tlm*2.d0/pi
      if (itrap.eq.1) tlm=tlm*DSQRT(x)
      tlp=0.
      return
      end

      subroutine rhh(f,g,n1a,n2a,bth,dpr,tpr)                         650
cc Calculation of theoretical f & g from Rosenbluth, et.al.   cc
      implicit double precision (a-h,o-z)
      parameter(c=2.9979d8,pi=3.14159265359)
      parameter(n2=240,n1=480)
      double precision f(n1a,n2a),g(n1a,n2a)
      common /inp/ eps,r0,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass
      double precision ymu(n2),tlam(n2),ba1(n2),xp(n1)
      double precision tlamc(n2),ba1c(n2)
      common /grid/ ymu,xp,tlam,ba1,tlamc,ba1c                        660
      double precision epsn,ccol,rho,dens,te,ti,q,wpm
      common /surf/ epsn,ccol,rho,dens,te,ti,q,wpm,lp

      do 30 i=1,n1
        vk=xp(i)
        gama=DSQRT(1.d0+vk*vk*bth*bth)
        w=vk*vk/(1.+gama)
c Accurate correction for the relativistic case
        relch=1.5+bth*bth*1.875-bth**4*1.875
```

```
      fnop=(dpr-(relch-w)*tpr)                                          670
     &       *DEXP(-vk*vk/(1.+gama))/(2.*pi)**1.5
      fhat=(-vk*c*bth*fnop/wpm/r0)*tlam(2)/(1.d0+epsn)
      do 10 j=n2/2,2,-1
        ymu0=ymu(j)
        f(i,j)=(-vk*c*bth*fnop/wpm/r0)*ymu0
        if (j.gt.lp) then
          g(i,j)=0.d0
        else
          dmu=ymu(j+1)-ymu(j)
          ymup=(ymu(j+1)+ymu(j))/2.                                     680
          dno=1.d0+ba1c(j)/(ymup**2)
          g(i,j)=gp+fhat/tlamc(j)/dno*dmu
        endif
        lc=n2+1-j
        f(i,lc)=-f(i,j)
        g(i,lc)=-g(i,j)
        gp=g(i,j)
 10     continue
 30   continue
                                                                        690
      return
      end

      subroutine moments(p,n1a,n2a,svm,bth,den,aj,isym)
cccc Calculates the flux-surface averaged density (den) and parallel
cccc flow density (aj) from a distribtion.  ISYM=1 assumes a bounce-
cccc averaged (theta-independent) distribution,
cccc ISYM=-1 is special for f~.
c IN THIS NORMALIZATION, DENS IS IN UNITS OF dens(epsn) c
c AND AJ IS IN UNITS OF dens(epsn)*c*bth(epsn) (m_e times the flux) c    700
      implicit double precision (a-h,o-z)
      parameter (c=2.9979d8,pi=3.14159265359)
      parameter (elec=1.602d-19)
      parameter(n2=240,n1=480)
      double precision p(n1a,n2a),den,aj
```

```
      double precision dl(n1),ajl(n1)
      common /inp/ eps,r0,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass
      double precision ymu(n2),tlam(n2),ba1(n2),xp(n1)
      double precision tlamc(n2),ba1c(n2)                                710
      common /grid/ ymu,xp,tlam,ba1,tlamc,ba1c
      double precision epsn,ccol,rho,dens,te,ti,q,wpm
      common /surf/ epsn,ccol,rho,dens,te,ti,q,wpm,lp

      tmu2=2.d0*epsn/(1.d0+epsn)
      if (isym.eq.1) then
        wtfac2=1.d0/DSQRT(1.-tmu2)
      endif
      gv=(xp(2)-xp(1))/svm
      gm=ymu(3)-ymu(2)                                                   720

      do 20 i=1,n1
        vk=xp(i)/svm
        gama=DSQRT(1.d0+vk*vk*bth*bth)
        dl(i)=0.d0
        ajl(i)=0.d0
        do 10 j=2,n2/2
          ymu0=ymu(j)
          lc=n2+1-j
          exs=1.d0                                                       730
          if (j.eq.2) exs=3.d0/8.d0
          if (j.eq.3) exs=7.d0/6.d0
          if (j.eq.4) exs=23.d0/24.d0
          if (isym.eq.1) then
            wtfac=(1.d0+epsn)*tlam(j)/tlam(n2-1)
          elseif (isym.eq.-1) then
            wtfac=1.d0
            call bavg2(ymu0,tmu2,wtfac2)
          endif
          dl(i)=dl(i)+exs*wtfac*(p(i,j)+p(i,lc))                         740
          ajl(i)=ajl(i)+exs*ymu0*wtfac2*(p(i,j)-p(i,lc))
```

```
10      continue
        dl(i)=2.d0*pi*dl(i)*gm
        ajl(i)=2.d0*pi*ajl(i)*gm/gama
20    continue
      den=(gv**3)*(dl(1)/12.+dl(2)/4.)
      vk=xp(2)/svm
      aj=(gv**4)*(ajl(1)/20.+ajl(2)/5.)
      do 30 i=3,n1,2
        vk=xp(i)/svm                                                    750
        vm=xp(i-1)/svm
        if (i.eq.n1) then
          dip=0.
          ajip=0.
        else
          vp=xp(i+1)/svm
          dip=dl(i+1)
          ajip=ajl(i+1)
        endif
        den=den+gv*(vm*vm*dl(i-1)/3.                                    760
     &        +vk*vk*dl(i)*4./3.+vp*vp*dip/3.)
        aj=aj+gv*(vm**3*ajl(i-1)/3.
     &        +vk**3*ajl(i)*4./3.+vp**3*ajip/3.)
30    continue

      return
      end

      subroutine bavg2(ymu,tmu2,w2)
cccc Another bounce averaging routine (see BAVG), this one calculating  770
cccc bounce averages used in the subroutine MOMENTS.
      implicit double precision (a-h,o-z)
      parameter(pi=3.14159265359)
      parameter(mmax=99,ees=1.d-10)
      double precision ymu,tmu2,w2
      common /inp/ eps,r0,zb,den0,dene,gamn,
     &     te0,ti0,tie,gamt,wc0,qmin,qmax,eq,mass
```

```
      cmu2=(ymu**2)
      x=tmu2/cmu2
      alpha=0.5d0                                                    780
      am1=1.d0
      if (x.lt.1.d0) then
        ajm2=DELK(x)
        ajm=(DELK(x)-DELE(x))/x
      else
        ox=1.d0/x
        if (ox.ge.1.d0) ox=.9999999d0
        ajm2=DELK(ox)/DSQRT(x)
        ajm=(DELK(ox)-DELE(ox))/DSQRT(x)
      endif                                                          790
      tmu2m=tmu2
      w2=2.d0/pi*(ajm2+(alpha-1.d0/cmu2)*tmu2m*ajm)
      do 35 m=2,mmax
        ajm4=ajm2
        ajm2=ajm
        ajm=(DBLE(2*m-2)*(1.d0+1.d0/x)*ajm2-DBLE(2*m-3)*ajm4/x)
     &       /DBLE(2*m-1)
        tmu2m=tmu2m*tmu2
        am1=alpha
        alpha=am1*DBLE(2*m-1)/DBLE(2*m)                              800
        dw2=2.d0/pi*(alpha-am1/cmu2)*tmu2m*ajm
        w2=w2+dw2
        am=DABS(dw2/w2)
        if (am.lt.ees) goto 39
   35 continue
      stop
   39 continue
      return
      end
                                                                     810
      subroutine fastfp(trapo,nits,pmax,te,ti,atmax,z,mass,rmin,rmaj,
     &     v1,v2,ddif,ifw,omc,theta,dperp,dnres,dnpar,
     &     t,n1a,n2a,ipert,bth,dens,tcurr,pabs1,ppr,upi,alimit,dout)
```

```
cccc The original FASTFP code of Shoucri and Shkarofsky, with modifications
cccc for bootstrap current problem, used by the code
cccc All original text is in CAPITAL LETTERS
C     FULLY RELATIVISTIC COLLISIONAL OPERATOR BRAAMS-KARNEY
C     N1 NUMBER OF POINTS IN THE MOMENTUM P
C     N2 NUMBER OF POINTS IN COS(THETA)
c     n1a,n2a Same as N1,N2, but passed from external to the routine                820
C     TRAPO=1. TRAPPING INCLUDED IN THE CALCULATION
C     TRAPO=0. TRAPPING EXCLUDED IN THE CALCULATION
C     MODE=5 USE FIVE POINTS DIAGONAL SOLVER
C     MODE=9 USE NINE POINTS DIAGONAL SOLVER
C     APARAM PARAMETERS FOR CONVERGENCE (USUALLY 60)
C     NITS NUMBER OF ITERATIONS NEEDED
C     PMAX MAXIMUM MOMENTUM
C     TRUNC=1 INCLUDES THE COLLISION TERM FOR MOMENTUM CONSERVATION
C     PMIN MINIMUM VALUE OF THE MOMENTUM ,USUALLY ZERO.
C     TE ELECTRON TEMPERATURE                                                      830
C     TI ION TEMPERATURE
c     atmax  Maximum elapsed time to steady state
C     DT TIME STEP   (calc. from atmax/nits)
C     V1 MINIMUM VALUE OF THE VELOCITY FOR THE LOWER HYBRID SPECTRUM
C     V2 MAXIMUM VALUE OF THE VELOCITY FOR THE LOWER HYBRID SPECTRUM
C     DDIF QUASILINEAR DIFFUSION OPERATOR FOR THE LOWER HYBRID WAVE
C     OMC RATIO OF THE CYCLOTRON FREQUENCY OVER THE WAVE FREQUENCY
C     DPERP COEFFICIENT FOR THE ELECTRON CYCLOTRON DIFFUSION COEFFICIENT
C     DNRES RESONANCE REFRACTIVE INDEX FOR THE ELECTRON CYCLOTRON WAVE
C     DNPAR WIDTH OF THE ELECTRON CYCLOTRON RESONANCE                              840
C     T(I,J) DISTRIBUTION FUNCTION TO BE CALCULATED
C     F(I) MAXWELLIAN DISTRIBUTION
C     DCY(I,J) QUASILINEAR DIFFUSION COEFFICIENT FOR THE ELECTRON
C     CYCLOTRON WAVE
C     EPS ELECTRIC FIELD NORMALIZED TO THE DREICER FIELD
C     RMIN=RADIUS OF THE FLUX SURFACE
C     RMAJ=MAJOR RADIUS OF THE TOKAMAK
C     DELTA IS THE SHAFRANOV SHIFT
c     mass=ion mass (amu)
```

```
c     ifw = Flag for Fast Alfven Wave diffusion (not used)                    850
c     ipert = Flag for different calculations of FASTFP
c        0 Regular RFCD only calculation
c        1 Calculation of g for RF+bootstrap
c       40,41  Spitzer-Harm calc.
c     bth = v_{Te}/c (Output)
c     dens = Normalized density (should be 1 on output)
c     tcurr = Normalized current density from FASTFP (Output)
c     pabs1 = Power density dissipated (Output)
c     ppr = Distribution f~ needed for calculation of g
c     upi = Ion neclassical flow                                              860
c     alimit = Small parameter for code convergence
c              Stop when DELTA(TCURR)/TCURR < alimit
c     dout = Diffusion coefficient on grid (Output)
      IMPLICIT double precision (A-H,O-Z)
      INTEGER N1,N2,N1M,NITS,IFAIL,MODE
      PARAMETER(N1=480,N2=240,N1M=N1)
      double precision T(n1a,n2a),ppr(n1a,n2a),dout(n1a,n2a)
      double precision dumi(n2),wti(n2),so(n1,n2)
      double precision A(N1M,N2),B(N1M,N2),C(N1M,N2),D(N1M,N2)
     &     ,E(N1M,N2),Q(N1M,N2)                                               870
     1,R(N1M,N2),WRKSP1(N1M,N2),WRKSP2(N1M,N2),XP(N1),YMU(N2),
     2DIFF(N1,N2),T1(N1),F(N1),AP1(N1),AP2(N1),AP3(N1),AB1(N1),AB2(N1),
     3AB3(N1),AB4(N1),AB5(N1),AB6(N1),AB7(N1),AB8(N1),AB9(N1),AB10(N1),
     4AB11(N1),AB12(N1),AB13(N1),AB14(N1),SIG(N1),T2(N1),DFDP(N1,N2),
     5DFDM(N1,N2),DE(N1),DF(N1),BI(N1),
     6APH(N1),FPH(N1),FMH(N1),AMH(N1),EJ0(N2),EJ2(N2),DIF(N1,N2),TRP(N2)
     7,TRPP(N2)
      double precision AM(N1M,N2),AP(N1M,N2),EM(N1M,N2),EP(N1M,N2),
     1DCY(N1,N2),WRKSP3(N1M,N2),WRKSP4(N1M,N2),ALPHAMAX
      double precision S11ABF                                                880
      EXTERNAL S11ABF
cccc Initialized parameters - note that many are coming from external
cccc to the routine, and have been commented out here.
cs          TRAPO=1.
            MODE=9
```

```
C     WRITE(*,*) 'APARAM ?'
C     READ(6,*) APARAM
          APARAM=1000.
C     WRITE(*,*) 'NITS ?'
C     READ(6,*) NITS                                                    890
cs        NITS=100
C     WRITE(*,*) 'PMAX ?'
C     READ(6,*) PMAX
cs        PMAX=25.
C     WRITE(*,*) 'TRUNC E-E COLLISION (0 OR 1) ?'
C     READ(6,*) TRUNC
          TRUNC=1.d0
      PMIN=0.
      DP=(PMAX-PMIN)/FLOAT(N1-1)
      DMU=2./FLOAT(N2-2-1)                                              900
C     WRITE(*,*) 'TE ?'
C     READ(6,*) TE
cs    TE=3.015
      BTH=0.04424*DSQRT(TE)
cs        TI=1.
C     WRITE(*,*) 'DT ?'
C     READ(6,*) DTT
cs        DTT=10.0
      dtt=atmax/nits
      DT=DTT                                                            910
C      WRITE(*,*) 'E ?'
C     READ(6,*) EEPS
          EEPS=0.0
      COLLF=0.0
cs    Z=1.0
cs    RMIN=10.
cs    RMAJ=87.
      NPRINT=1
      MPRINT=0
      CC=DSQRT(dfloat(mass)*1840.D0*TE/TI)                              920
      CCC=DSQRT(dfloat(mass)*1840.D0)
```

```
C     PARAMETERS FOR THE LOWER HYBRID WAVE CURRENT DRIVE PROBLEM
C     DIFFUSION MATRIX IN DIF(I,J)
C      WRITE(*,*) 'VMIN ?'
C      READ(6,*) V1
cs        V1=3.
C      WRITE(*,*) 'VMAX ?'
C      READ(6,*) V2
cs        V2=6.
C      WRITE(*,*) 'DQL ?'                                              930
C      READ(6,*) DDIF
cs        DDIF=0.
C     PARAMETERS FOR ELECTRON CYCLOTRON HEATING
cs     COSTHETA=0.5
      costheta=DCOS(theta)
cs     DPERP=0.00
      DELTA=0.0
cs     OMC=1.
cs     DNRES=0.4
cs     DNPAR=0.03                                                      940
C     PARAMETERS FOR THE TRAPPING EFFECT
      RRATIO=RMIN/RMAJ
      TRAP=DSQRT(2.*RRATIO/(1.+RRATIO))
      TRAP2=TRAP*TRAP
      RATIOB=(1.+DELTA/RMAJ+RRATIO)/(1.+DELTA/RMAJ+RRATIO*COSTHETA)
c      OMC=OMC/(1.+DELTA/RMAJ+RRATIO*COSTHETA)
      IF(TRAPO.EQ.0.0D0) RATIOB=1.
cccc Set up the grid
      N1M1=N1-1
      N2M1=N2-1                                                       950
      DO 1 I=1,N1
    1 XP(I)=DP*FLOAT(I-1)+PMIN
      DO 2 J=2,N2M1
    2 YMU(J)=-1.D0+DMU*FLOAT(J-2)
      YMU(1)=YMU(3)
      YMU(N2)=YMU(N2-2)
      jtp1=0
```

```
      jtp2=0
cccc Calculation of trapping time and bounce averaged quantities
cccc At end of loop:                                                    960
cccc     TRPP(J)=lambda/lambda(mu=1) - Norm. bounce time
cccc     TRP(J)=d(ln(lambda))/d(mu)
cccc     EJ0(j)=1/TRPP * sqrt((1+eps)/(1-eps)) - used in f~
cccc     wti(j)=<<mu^2/mu0^2>> / EJ0(j)     also used in f~
cccc     EJ2(j)=<<1/psi-1>>
      DO 46 J=2,N2M1
      ABSM=DABS(YMU(J))
      YMU2=YMU(J)*YMU(J)
      IF(ABSM.LE.TRAP) THEN
      ARG=YMU2/TRAP2                                                    970
      ARG1=1.-ARG
      ELSE
      ARG=TRAP2/YMU2
      ARG1=1.-ARG
      ENDIF
      EK=1.38629436112+0.09666344259*ARG1+0.03590092383*(ARG1**2)
     1+0.03742563713*(ARG1**3)+0.01451196212*(ARG1**4)+(0.5+
     2.12498593597*ARG1+.06880248576*(ARG1**2)+.03328355346*
     3(ARG1**3)+0.00441787012*(ARG1**4))*DLOG(1.D0/ARG1)
      EE=1.+0.44325141463*ARG1+0.0626060122*(ARG1**2)+                  980
     1.04757383546*(ARG1**3)+.01736506451*(ARG1**4)+(.24998368310
     2*ARG1+.09200180037*(ARG1**2)+
     3.04069697526*(ARG1**3)+.00526449639*(ARG1**4))*DLOG(1.D0/ARG1)
      IF(ABSM.LE.TRAP) THEN
      EJ0(J)=EK*DABS(YMU(J)/TRAP)
      EJ2(J)=(EK-EE)*DABS(YMU(J)/TRAP)
      ELSE
      EJ0(J)=EK
      EJ2(J)=(EK-EE)*YMU2/TRAP2
      ENDIF                                                             990
      EJJ=EJ0(J)-0.5*TRAP2*EJ2(J)
      wti(j)=(2./3.1416)*(EJ0(J)+(0.5-1./ymu2)*TRAP2*EJ2(J))
     *     *DSQRT(1.d0-rratio*rratio)/(1.+rratio)
```

```
      TRP(J)=EJJ
      TRPP(J)=EJJ*2.0/3.1416
      EJ2(J)=-TRAP2*(EJ2(J)-.5*(2.*(TRAP2+YMU2)*EJ2(J)-YMU2*EJ0(J))/3.)
     1/EJJ
      EJ0(J)=EJJ
      IF(ABSM.LE.TRAP) THEN
      EJ0(J)=0.                                                          1000
      ELSE
      EJ0(J)=(3.1416/2.)*(1.+RRATIO)/(EJ0(J)*DSQRT(1.D0-RRATIO*RRATIO))
      ENDIF
C     REMOVE TRAPPING
      IF(TRAPO.EQ.0.0D0) EJ0(J)=1.
      IF(TRAPO.EQ.0.0D0) EJ2(J)=0.
c Added to locate the trapped/passing boundaries
      absm1=DABS(ymu(j+1))
      test=(absm1-trap)*(absm-trap)
      if (test.le.0.d0) then                                             1010
        if (jtp2.ne.jtp1) jtp2=j
        if (jtp1.eq.0) jtp1=j
      endif
   46 CONTINUE
      EJ0(1)=EJ0(3)
      EJ0(N2)=EJ0(N2-2)
      EJ2(1)=EJ2(3)
      EJ2(N2)=EJ2(N2-2)
      wti(1)=wti(3)
      wti(n2)=wti(n2-2)                                                  1020
      TRP(1)=TRP(3)
      TRP(N2)=TRP(N2-2)
      TRPP(1)=TRPP(3)
      TRPP(N2)=TRPP(N2-2)
      DO J=2,N2M1
      dumi(J)=(TRP(J+1)-TRP(J-1))/(2.*TRP(J)*DMU)
      ENDDO
      DO J=2,N2M1
      TRP(J)=dumi(J)
```

```
      IF(TRAPO.EQ.0.0D0) TRP(J)=0.                                    1030
      IF(TRAPO.EQ.0.0D0) TRPP(J)=1.
cccc Added to enforce symmetry in YMU
          if (j.gt.n2/2) then
            ymu(j)=-ymu(n2+1-j)
            trpp(j)=trpp(n2+1-j)
            trp(j)=-trp(n2+1-j)
            ej0(j)=ej0(n2+1-j)
            ej2(j)=ej2(n2+1-j)
            wti(j)=wti(n2+1-j)
          endif                                                       1040
      ENDDO
      TRP(1)=TRP(3)
      TRP(N2)=TRP(N2-2)
      TRPP(1)=TRPP(3)
      TRPP(N2)=TRPP(N2-2)
cccc Initial conditions of distribution T and Maxwellian F
cccc Also QL diffusion coefficients calculated
      dens=0.d0
      if (v2.lt.v1) then
        dum=v1                                                        1050
        v1=v2
        v2=dum
      endif
      DO 3 I=1,N1
      DO 3 J=2,N2M1
      ABSM=DABS(YMU(J))
      YMUSQ=YMU(J)*YMU(J)
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
      T(I,J)=DEXP(-XP(I)*XP(I)/(1.+GAMA))/(2.*3.1416)**1.5
          if (ipert.eq.1) t(i,j)=0.d0                                 1060
      F(I)=DEXP(-XP(I)*XP(I)/(1.+GAMA))/(2.*3.1416)**1.5
      if (j.eq.2) then
        dens=dens+f(i)*xp(i)*xp(i)
      endif
c
```

```
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.                                                          1070
      ENDIF
      IF(YMU(J).LT.0.0) CNSTT=-CNSTT
      PPAR=XP(I)*CNSTT
      PPERP=XP(I)*XP(I)-PPAR*PPAR
      IF(PPAR.EQ.0.0) THEN
      RESN=10.
      DCY(I,J)=0.
      ELSE
      RESN=(GAMA-OMC)/(PPAR*BTH)
      DCY(I,J)=PPERP*DPERP*DEXP(-(RESN-DNRES)**2/DNPAR**2)              1080
      ENDIF
      VPAR=PPAR/GAMA
      DIF(I,J)=0.
      delv=0.d0
      dtrans=dp
      IF(VPAR.GE.V1.AND.VPAR.LE.V2) then
        DIF(I,J)=DDIF
c Added to smooth the boundaries of
c the LH diffusion region
      elseif (vpar.lt.v1) then                                          1090
        delv=dabs(vpar-v1)
        if (delv.lt.dtrans) then
          dcay=(1.d0+dcos(3.14159d0*delv/dtrans))/2.d0
          DIF(I,J)=DDIF*dcay
        endif
      elseif (vpar.gt.v2) then
        dtrans=dtrans/ymu(j)**5
        delv=dabs(vpar-v2)
        if (delv.lt.dtrans) then
          dcay=(1.d0+dcos(3.14159d0*delv/dtrans))/2.d0                  1100
          DIF(I,J)=DDIF*dcay
```

```
        endif
      endif
c An artifact of the bounce averaging process results
c in extra factors not included in original FP code
      if (cnstt.ne.0.d0) dcy(i,j)=dcy(i,j)*ymu(j)/cnstt
 3    CONTINUE
      DO 701 J=1,N2
      DIF(1,J)=0.
701   CONTINUE                                                    1110
      DO 702 I=2,N1
      DIF(I,1)=DIF(I,3)
      DIF(I,N2)=DIF(I,N2-2)
702   CONTINUE
      dens=dens*4.*3.1416*dp
      do i=1,n1
        f(i)=f(i)/dens
        do j=2,n2m1
          t(i,j)=t(i,j)/dens
        enddo                                                     1120
      enddo
      DO 44 I=1,N1
      T(I,1)=T(I,3)
      T(I,N2)=T(I,N2-2)
          DCY(I,1)=DCY(I,3)
          DCY(I,N2)=DCY(I,N2-2)
      Q(I,N2)=0.
      Q(I,1)=0.
      A(I,N2)=0.
      B(I,N2)=0.                                                  1130
      C(I,N2)=0.
      D(I,N2)=0.0
      E(I,N2)=0.
      A(I,1)=0.
      B(I,1)=0.
      C(I,1)=0.
      D(I,1)=0.
```

```
c 9 diagonales
            AM(I,1)=0.
            AP(I,1)=0.                                                    1140
            EM(I,1)=0.
            EP(I,1)=0.
 44   E(I,1)=0.
          if (nits.eq.1) dt=1.d-9
      DO 45 J=1,N2
      Q(N1,J)=0.
      Q(1,J)=0.
      A(N1,J)=0.
      B(N1,J)=0.
      C(N1,J)=0.                                                          1150
      D(N1,J)=0.
      E(N1,J)=0.
c 9 diagonales
            AM(N1,J)=0.
            AP(N1,J)=0.
            EM(N1,J)=0.
            EP(N1,J)=0.
      A(1,J)=0.0
      B(1,J)=0.
      C(1,J)=0.                                                           1160
      D(1,J)=0.
c 9 diagonales
      AM(1,J)=0.
            AP(1,J)=0.
            EM(1,J)=0.
            EP(1,J)=0.
 45   E(1,J)=0.
      A(1,1)=0.0
      B(1,1)=0.0
      C(1,1)=0.0                                                          1170
      D(1,1)=0.0
      E(1,1)=0.0
c 9 diagonales
```

```
      AM(1,1)=0.
            AP(1,1)=0.
            EM(1,1)=0.
            EP(1,1)=0.
      IIFAIL=0.
cccc Calculation of Braams-Karney potentials for collision op
      DO 75 I=1,N1                                                    1180
      XSIG=XP(I)*BTH
      SIG(I)=S11ABF(XSIG,IIFAIL)
   75 CONTINUE
      AP1(1)=0.
      AP2(1)=0.
      AP3(N1)=0.
      AP1(2)=4.*3.1416*(F(1)+F(2))*(DP**3)/8.
      AP1(3)=4.*3.1416*(F(1)*XP(1)**2/3.+4.*F(2)*XP(2)**2/3.
     1+F(3)*XP(3)**2/3.)*DP
      AP1(4)=4.*3.1416*(3.*F(1)*XP(1)**2/8.+9.*F(2)*XP(2)**2/8.       1190
     1+9.*F(3)*XP(3)**2/8.)*DP
      AP2(2)=4.*3.1416*BTH*BTH*(F(1)+F(2))*(DP**5)/(12.*5.)
      AP2(3)=4.*3.1416*BTH*BTH*(F(1)*XP(1)**4/3.+4.*F(2)*XP(2)**4/3.
     1+F(3)*XP(3)**4/3.)*DP/6.
      AP2(4)=4.*3.1416*BTH*BTH*(3.*F(1)*XP(1)**4/8.+9.*F(2)
     1*XP(2)**4/8.+9.*F(3)*XP(3)**4/8.)*DP/6.
      AP1(1)=0.
      AP2(1)=0.
      AP3(N1)=0.
      DO 54 I=3,N1                                                    1200
      GAMA=DSQRT(1.+XP(I)*XP(I)*BTH*BTH)
      GAMA1=DSQRT(1.+XP(I-1)*XP(I-1)*BTH*BTH)
      V=XP(I)/GAMA
      VONE=XP(I-1)/GAMA1
      AP1(I)=AP1(I-1)+0.5*XP(I)*F(I)*DP*V*4.*3.1416
     1+0.5*XP(I-1)*F(I-1)*DP*VONE*4.*3.1416
      AP2(I)=AP2(I-1)+0.5*XP(I)*F(I)*DP*V*4.*3.1416*(1.-
     1GAMA*SIG(I)/(BTH*XP(I)))
     2+0.5*XP(I-1)*F(I-1)*DP*VONE*4.*3.1416*(1.-
```

```
   3GAMA1*SIG(I-1)/(BTH*XP(I-1)))                                          1210
54  CONTINUE
    DO 58 I=1,N1M1
    K=N1-I
    AP3(K)=AP3(K+1)+0.5*XP(K)*F(K)*DP*4.*3.1416
   1+0.5*XP(K+1)*F(K+1)*DP*4.*3.1416
58  CONTINUE
    AB1(1)=0.
    AB2(1)=0.
    AB3(1)=0.
    AB4(1)=0.                                                              1220
    AB5(1)=0.
    AB6(1)=0.
    AB7(1)=0.
    AB1(2)=0.
    AB1(2)=4.*3.1416*(F(1)+F(2))*(DP**3)/8.
    AB2(2)=0.
    AB2(2)=4.*3.1416*(F(1)+F(2))*(DP**5)/(32.)
    AB3(2)=0.
    AB4(1)=0.
    AB5(1)=0.                                                              1230
    AB4(2)=0.
    AB5(2)=0.
    DO 59 I=3,N1
    GAMA=DSQRT(1.+XP(I)*XP(I)*BTH*BTH)
    V=XP(I)/GAMA
    GAMA1=DSQRT(1.+XP(I-1)*XP(I-1)*BTH*BTH)
    VONE=XP(I-1)/GAMA1
    AB1(I)=AB1(I-1)+0.5*DP*F(I)*XP(I)*XP(I)*4.*3.1416
   1+0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*4.*3.1416
    AB2(I)=AB2(I-1)+0.5*DP*F(I)*XP(I)*XP(I)*XP(I)*XP(I)*4.*3.1416          1240
   1+0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*XP(I-1)*XP(I-1)*4.*3.1416
    AB3(I)=AB3(I-1)+.5*DP*F(I)*XP(I)*XP(I)*4.*3.1416*(-3.*GAMA+SIG(I)
   1*(3./(BTH*XP(I))+2.*BTH*XP(I)))/GAMA
   2+0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*4.*3.1416*(-3.*GAMA1+SIG(I-1)
   3*(3./(BTH*XP(I-1))+2.*BTH*XP(I-1)))/GAMA1
```

```
      AB4(I)=AB4(I-1)+0.5*DP*F(I)*XP(I)*XP(I)*4.*3.1416*(GAMA-SIG(I)
     1/(BTH*XP(I))-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)/GAMA
     2 +0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*4.*3.1416*(GAMA1-SIG(I-1)
     3/(BTH*XP(I-1))-2.*GAMA1*BTH*BTH*XP(I-1)*XP(I-1)/3.)/GAMA1
      AB5(I)=AB5(I-1)                                                    1250
     1+.5*DP*F(I)*4.*3.1416*XP(I)*XP(I)*(1.-SIG(I)/(GAMA*BTH*XP(I)))
     2+.5*DP*F(I-1)*4.*3.1416*XP(I-1)*XP(I-1)*(1.-SIG(I-1)/(GAMA1*
     3BTH*XP(I-1)))
   59 CONTINUE
      AB8(N1)=0.
      AB9(N1)=0.
      AB10(N1)=0.
      AB8(1)=0.
      AB9(1)=0.
      AB10(1)=0.                                                         1260
      N1M2=N1M1-1
      DO 66 I=1,N1M2
      K=N1-I
      GAMA=DSQRT(1.+XP(K)*XP(K)*BTH*BTH)
      V=XP(K)/GAMA
      GAMA1=DSQRT(1.+XP(K+1)*XP(K+1)*BTH*BTH)
      VONE=XP(K+1)/GAMA1
      AB8(K)=AB8(K+1)+0.5*DP*XP(K)*XP(K)*F(K)*4.*3.1416/V
     1+0.5*DP*XP(K+1)*XP(K+1)*F(K+1)*4.*3.1416/VONE
      AB9(K)=AB9(K+1)+0.5*DP*XP(K)*XP(K)*F(K)*4.*3.1416/(V*GAMA*GAMA)    1270
     1 +.5*DP*XP(K+1)*XP(K+1)*F(K+1)*4.*3.1416/(VONE*GAMA1*GAMA1)
      AB10(K)=AB10(K+1)+0.5*DP*F(K)*XP(K)*XP(K)*4.*3.1416*V
     1 +0.5*DP*F(K+1)*XP(K+1)*XP(K+1)*4.*3.1416*VONE
   66 CONTINUE
      AB8(1)=AB8(2)
      AB9(1)=AB9(2)
      AB10(1)=AB10(2)
      IIFAIL=0
cccc Calculation of A(p),F(p) and B(p) for collision operator
      DO 400 I=2,N1M1                                                    1280
      PPH=XP(I)+DP/2.
```

```
PPH2=PPH*PPH
PMH=XP(I)-DP/2.
PMH2=PMH*PMH
GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
V=XP(I)/GAMA
GAMAPH=DSQRT(1.D0+PPH*PPH*BTH*BTH)
VPH=PPH/GAMAPH
GAMAMH=DSQRT(1.D0+PMH*PMH*BTH*BTH)
VMH=PMH/GAMAMH                                                    1290
XSIG=PPH*BTH
SIGPH=S11ABF(XSIG,IIFAIL)
XSIG=PMH*BTH
SIGMH=S11ABF(XSIG,IIFAIL)
S=V/DSQRT(2.D0)
SPH=VPH/DSQRT(2.D0)
SMH=VMH/DSQRT(2.D0)
EXPPH=DEXP(-PPH2/(1.+GAMAPH))/(2.*3.1416)**1.5
FPH(I)=(AP1(I)+PPH*EXPPH*VPH*0.5*DP*4.*3.1416)/(VPH**2)+(AP2(I)
1+0.5*DP*EXPPH*PPH*VPH*4.*3.1416*(1.-SIGPH*GAMAPH/(BTH*PPH)))/PPH2  1300
2+(AP3(I)-PPH*EXPPH*(DP/2.)*4.*3.1416)*(1.-SIGPH/(BTH*PPH*GAMAPH))
3/VPH
APH(I)=FPH(I)/VPH+Z/(VPH**3*CC*CC)
IF(I.EQ.2) APH(2)=4.*3.1416*F(2)/3.+Z/(VPH**3*CC*CC)
FPH(I)=FPH(I)+Z*TE/(TI*VPH*VPH*CC*CC)
EXPMH=DEXP(-PMH2/(1.+GAMAMH))/(2.*3.1416)**1.5
FMH(I)=(AP1(I)-PMH*EXPMH*VMH*0.5*DP*4.*3.1416)/(VMH**2)+(AP2(I)
1-0.5*DP*EXPMH*PMH*VMH*4.*3.1416*(1.-SIGMH*GAMAMH/(BTH*PMH)))/PMH2
2+(AP3(I)+PMH*EXPMH*(DP/2.)*4.*3.1416)*(1.-SIGMH/(BTH*PMH*GAMAMH))
3/VMH                                                             1310
AMH(I)=FMH(I)/VMH+Z/(VMH**3*CC*CC)
IF(I.EQ.2) AMH(2)=4.*3.1416*F(2)/3.+Z/(VPH**3*CC*CC)
FMH(I)=FMH(I)+Z*TE/(TI*VMH*VMH*CC*CC)
FU2=1.-1./(2.*S*S)
FU2C=1.-1./(2.*S*S*CC*CC)
FU=(S*DEXP(-S*S)+Z*S*DEXP(-S*S*CC*CC)/CC)*2./DSQRT(3.1416D0)
BI(I)=AB1(I)/(2.*V)-AB2(I)/(6.*XP(I)*XP(I)*V)
```

```
     1+AB3(I)/(BTH*BTH*XP(I)*XP(I)*V*GAMA*GAMA*8.)
      BI(I)=BI(I)-
     1AB4(I)/(V*4.*BTH*BTH*XP(I)*XP(I))                                    1320
     2-AB5(I)/(V*4.*GAMA*GAMA)+AB8(I)/2.
      BI(I)=BI(I)+AB9(I)*(-GAMA*GAMA/6.)
      BI(I)=BI(I)+AB9(I)*(-3.*GAMA+SIG(I)*
     1(3./(BTH*XP(I))+2.*BTH*XP(I)))/(8.*GAMA*BTH*BTH*XP(I)*XP(I))
      BI(I)=BI(I)-AB9(I)*GAMA*(GAMA-SIG(I)/(BTH*XP(I))
     1-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)/(4.*BTH*BTH*XP(I)*XP(I))
      BI(I)=BI(I)-AB10(I)*(GAMA-SIG(I)/(BTH*XP(I)))/(4.*GAMA*XP(I)
     1*XP(I))+Z*(1.-1./(CC*CC*V*V))/(2.*V)
 400  CONTINUE
      IIFAIL=0                                                              1330
      IT=1
C     30 CONTINUE
      DO 707 J=1,N2
      DO 707 I=1,N1
      DIFF(I,J)=DIF(I,J)
 707  CONTINUE

cccc MAIN LOOP - calculates the matrix elements for the discrete
cccc equation for the distribution
      DO 4 I=2,N1M1                                                         1340
      PPH=XP(I)+DP/2.
      PPH2=PPH*PPH
      PMH=XP(I)-DP/2.
      PMH2=PMH*PMH
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
          GAMAB=GAMA*RATIOB
      V=XP(I)/GAMA
      GAMAPH=DSQRT(1.D0+PPH*PPH*BTH*BTH)
          GAMAPHB=GAMAPH*RATIOB
      VPH=PPH/GAMAPH                                                        1350
      GAMAMH=DSQRT(1.D0+PMH*PMH*BTH*BTH)
          GAMAMHB=GAMAMH*RATIOB
      VMH=PMH/GAMAMH
```

```
      XSIG=PPH*BTH
      SIGPH=S11ABF(XSIG,IIFAIL)
      XSIG=PMH*BTH
      SIGMH=S11ABF(XSIG,IIFAIL)
      S=V/DSQRT(2.D0)
      SPH=VPH/DSQRT(2.D0)
      SMH=VMH/DSQRT(2.D0)                                              1360
      DO 4 J=2,N2M1
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.
      ENDIF
      DABMU=CNSTT
      IF(YMU(J).LT.0.0) CNSTT=-CNSTT
      YPH=YMU(J)+DMU/2.                                                1370
      IF(J.EQ.N2M1) YPH=YMU(N2M1)-DMU/2.
      DYPH=(1.D0-RATIOB*(1.D0-YPH*YPH))
      IF(DYPH.GT.0.0.AND.DYPH.LE.1.) THEN
      DYPH=DSQRT(DYPH)
      ELSE
      DYPH=0.
      ENDIF
      YMH=YMU(J)-DMU/2.
      IF(J.EQ.2) YMH=YMU(2)+DMU/2.
      DYMH=(1.D0-RATIOB*(1.D0-YMH*YMH))                                1380
      IF(DYMH.GT.0.0.AND.DYMH.LE.1.) THEN
      DYMH=DSQRT(DYMH)
      ELSE
      DYMH=0.
      ENDIF
cccc There seems to be an error here - Corrected 11/27/98  S.S.
c     FBI=1.+EJ2(J)
      fbi=1.+ej2(j)/ymu(j)/ymu(j)
      EPS=EEPS*EJ0(J)
```

```
c      FBI1=1.+0.5*(EJ2(J)+EJ2(J-1))                                    1390
       fbi1=1.+0.5*(EJ2(J)+EJ2(J-1))/ymh/ymh
            if (j.eq.n2/2+1) fbi1=1.
       EPS1=EEPS*(EJ0(J)+EJ0(J-1))*0.5
c      FBI2=1.+0.5*(EJ2(J)+EJ2(J+1))
       fbi2=1.+0.5*(EJ2(J)+EJ2(J+1))/yph/yph
            if (j.eq.n2/2) fbi2=1.
       EPS2=EEPS*(EJ0(J)+EJ0(J+1))*0.5

cccc This section computes factors for a correction to the
cccc derivative in P, making use of the fact that we know the            1400
cccc function is dominated by a decaying Gaussian
       IF(DYMH.NE.0.) THEN
       WW=-EPS1*DMU*XP(I)/(BI(I)*FBI1+.5*(DIFF(I,J)+DIFF(I,J-1))*(1.-
      1YMH*YMH)+0.5*(DCY(I,J)+DCY(I,J-1))*(YMH*GAMAB-(GAMAB-OMC)/YMH)**2
      2/(GAMAB*DYMH*XP(I)))
       ELSE
       WW=-EPS1*DMU*XP(I)/(BI(I)*FBI1+.5*(DIFF(I,J)+DIFF(I,J-1))*
      1(1.-YMH*YMH))
       ENDIF
       IF(WW.NE.0.D0) THEN                                               1410
       DELJ1=1./WW-1./(DEXP(WW)-1.)
       ELSE
       DELJ1=0.5
       END IF
       IF(DYPH.NE.0.) THEN
       WW=-EPS2*DMU*XP(I)/(BI(I)*FBI2+.5*(DIFF(I,J)+DIFF(I,J+1))*(1.-
      1YPH*YPH)+0.5*(DCY(I,J)+DCY(I,J+1))*(YPH*GAMAB-(GAMAB-OMC)/YPH)**2
      2/(GAMAB*DYPH*XP(I)))
       ELSE
       WW=-EPS2*DMU*XP(I)/(BI(I)*FBI2+.5*(DIFF(I,J)+DIFF(I,J+1))*       1420
      1(1.-YPH*YPH))
       ENDIF
       IF(WW.NE.0.D0) THEN
       DELJ2=1./WW-1./(DEXP(WW)-1.)
       ELSE
```

```
      DELJ2=0.5
      END IF
      BF=FPH(I)-EPS*YMU(J)
      IF(DABMU.NE.0) THEN
      CF=APH(I)+0.5*(DIFF(I,J)+DIFF(I+1,J))*YMU(J)*YMU(J)                    1430
     1+0.5*(DCY(I,J)+DCY(I+1,J))*GAMAPHB*(1.-YMU(J)*YMU(J))/(PPH*DABMU)
      ELSE
      CF=APH(I)+0.5*(DIFF(I,J)+DIFF(I+1,J))*YMU(J)*YMU(J)
      ENDIF
      WW=DP*BF/CF
      DELP=1./WW-1./(DEXP(WW)-1.)
      BF=FMH(I)-EPS*YMU(J)
      IF(DABMU.NE.0.) THEN
      CF=AMH(I)+0.5*(DIFF(I,J)+DIFF(I-1,J))*YMU(J)*YMU(J)
     1+0.5*(DCY(I,J)+DCY(I-1,J))*GAMAMHB*(1.-YMU(J)*YMU(J))/(PMH*DABMU)     1440
      ELSE
      CF=AMH(I)+0.5*(DIFF(I,J)+DIFF(I-1,J))*YMU(J)*YMU(J)
      ENDIF
      WW=DP*BF/CF
      DELM=1./WW-1./(DEXP(WW)-1.)
cccc For localized RF fields, the
cccc QL diffusion coefficient is always proportional to 1/lambda (1/TRPP)
      tlb=trpp(J)/trpp(n2m1)

      CMH=0.5*(DIFF(I,J)+DIFF(I,J-1))*(1.-YMH*YMH)/DMU                       1450
      CPH=0.5*(DIFF(I,J+1)+DIFF(I,J))*(1.-YPH*YPH)/DMU
      CDIFF=XP(I)*(DIFF(I,J+1)*YMU(J+1)*(1.-YMU(J+1)*YMU(J+1))
     1-DIFF(I,J-1)*YMU(J-1)*(1.-YMU(J-1)*YMU(J-1)))/(4.*DP*DMU)
      CNTE=(DIFF(I,J)+XP(I)*0.5*(DIFF(I+1,J)-DIFF(I-1,J))/DP)/(2.*DMU)

      CNSTT=1.D0-RATIOB*(1.D0-YMU(J-1)*YMU(J-1))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.                                                               1460
      ENDIF
```

```
 DABM1=CNSTT
 CNSTT=1.D0-RATIOB*(1.D0-YMU(J+1)*YMU(J+1))
 IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
 CNSTT=DSQRT(CNSTT)
 ELSE
 CNSTT=0.
 ENDIF
 DABP1=CNSTT
 DCYPH=(DCY(I+1,J)+DCY(I,J))                                          1470
 DCYMH=(DCY(I,J)+DCY(I-1,J))
 TRPI=0.d0
 TRPJ=-(1.-YMU(J)*YMU(J))*TRP(J)*FBI*BI(I)/(2.*DMU)
 IF(DABMU.NE.0.) THEN
 CNTCY1=-(OMC+GAMAB*(YMU(J)*YMU(J)-1.))*(1.-YMU(J)*YMU(J))*
1(DCY(I+1,J)-DCY(I-1,J))/(4.*DMU*DP*YMU(J)*DABMU)+BTH*BTH*ratiob
2*DCY(I,J)*XP(I)*(1.-YMU(J)*YMU(J))**2/(2.*DMU*YMU(J)*GAMA*DABMU)
 ELSE
 CNTCY1=0.
 ENDIF                                                                1480
 IF(DYMH.NE.0.) THEN
 CNTY2=(1.-YMH*YMH)*0.5*(DCY(I,J-1)+DCY(I,J))*
1(YMH*GAMAB-(GAMAB-OMC)/YMH)**2/(GAMAB*XP(I)*DMU*DMU*DYMH)
 ELSE
 CNTY2=0.
 ENDIF

 A(I,J)=-(1.-YMH*YMH)*(tlb*FBI1*BI(I)/DMU+CMH)/DMU
1+YMU(J)*(1.-YMU(J)*YMU(J))*CNTE+CNTCY1-CNTY2-tlb*TRPJ
 a(i,j)=a(i,j)/tlb                                                    1490

 eff=(rratio-rratio*rratio/2.)/(1.d0+rratio)
 apert=-(1.-YMH*YMH)*BI(I)/DMU/DMU
&     *(1.-eff/ymh/ymu(j-1))/(1+rratio)
&+(-(1.-YMH*YMH)*CMH/DMU+YMU(J)*(1.-YMU(J)*YMU(J))*CNTE
&     +CNTCY1-CNTY2)*dabm1/dabs(ymu(j-1))/ratiob
 apert=apert/tlb
```

```
 CY1=0.
 IF(DABMU.NE.0.) THEN                                                    1500
 IF(DABP1.NE.0.AND.DABM1.NE.0.)
1CY1=(1-YMU(J)*YMU(J))*(YMU(J)*GAMAB-(GAMAB-OMC)/YMU(J))*
2(DCY(I,J+1)-DCY(I,J-1))/(4.*DP*DMU*DABMU)
3+DCY(I,J)*((1.-YMU(J+1)*YMU(J+1))*(GAMAB*YMU(J+1)-(GAMAB-OMC)/
4YMU(J+1))/DABP1-(1.-YMU(J-1)*YMU(J-1))*(GAMAB*YMU(J-1)-
5(GAMAB-OMC)/YMU(J-1))/DABM1)/(4.*DP*DMU)
 ELSE
 CY1=0.
 ENDIF
 IF(DABMU.NE.0.) THEN                                                    1510
 CY2=PMH*.5*DCYMH*GAMAMHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)
 ELSE
 CY2=0.
      ENDIF

 B(I,J)=-(DIFF(I,J)+DIFF(I-1,J))*YMU(J)*YMU(J)
2*PMH2/(2.*DP*DP)+CDIFF-CY1-CY2
&-(AMH(I)/DP-FMH(I)*DELM)*PMH2/DP*tlb-TRPI
 b(i,j)=b(i,j)/tlb
                                                                         1520
 bpert=(-(DIFF(I,J)+DIFF(I-1,J))*YMU(J)*YMU(J)
2*PMH2/(2.*DP*DP)+CDIFF-CY1-CY2)*dabmu/dabs(ymu(j))/ratiob
&-(AMH(I)/DP-FMH(I)*DELM)*PMH2/DP/(1.d0+rratio)
 bpert=bpert/tlb

 CNTCC=+((DIFF(I+1,J)+DIFF(I,J))*PPH2+(DIFF(I,J)+
1DIFF(I-1,J))*PMH2)*YMU(J)*YMU(J)/(2.*DP*DP)
 CNTC=CNTCC-(1.-YPH*YPH)*(-tlb*FBI2*BI(I)/DMU-CPH)/DMU
1+(1.-YMH*YMH)*(tlb*FBI1*BI(I)/DMU+CMH)/DMU
                                                                         1530
 CYY=0.
 IF(DABMU.NE.0.) THEN
 IF(DYMH.NE.0.AND.DYPH.NE.0.)
```

```
1CYY=PPH*.5*DCYPH*GAMAPHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)
2+PMH*.5*DCYMH*GAMAMHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)
3+(1.-YMH*YMH)*0.5*(DCY(I,J-1)+DCY(I,J))*
4(YMH*GAMAB-(GAMAB-OMC)/YMH)**2/(GAMAB*XP(I)*DMU*DMU*DYMH)
5+(1.-YPH*YPH)*0.5*(DCY(I,J+1)+DCY(I,J))*
6(YPH*GAMAB-(GAMAB-OMC)/YPH)**2/(GAMAB*XP(I)*DMU*DMU*DYPH)
 ELSE                                                          1540
 CYY=0.
 ENDIF

 C(I,J)=CNTC+CYY+(-PPH2*(-APH(I)/DP+FPH(I)*DELP)/DP
3+PMH2*(AMH(I)/DP+FMH(I)*(1.-DELM))/DP)*tlb
 c(i,j)=c(i,j)/tlb+XP(I)*XP(I)/DT

 cpert=(CNTCC+CYY+(1.-YPH*YPH)*CPH/DMU
&+(1.-YMH*YMH)*CMH/DMU)*dabmu/dabs(ymu(j))/ratiob
&+(1.-YPH*YPH)*BI(I)/DMU/DMU                                   1550
&     *(1.-eff/yph/ymu(j))/(1+rratio)
&+(1.-YMH*YMH)*BI(I)/DMU/DMU
&     *(1.-eff/ymh/ymu(j))/(1+rratio)
2+(-PPH2*(-APH(I)/DP+FPH(I)*DELP)/DP
3+PMH2*(AMH(I)/DP+FMH(I)*(1.-DELM))/DP)/(1.d0+rratio)
 cpert=cpert/tlb

 IF(DABMU.NE.0.) THEN
 CY22=PPH*.5*DCYPH*GAMAPHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)
 ELSE                                                          1560
 CY22=0.
 ENDIF

 D(I,J)=-(DIFF(I+1,J)+DIFF(I,J))*YMU(J)*YMU(J)
1*PPH2/(2.*DP*DP)-CDIFF+CY1-CY22
2-(APH(I)/DP+FPH(I)*(1.-DELP))*PPH2/DP*tlb+TRPI
 d(i,j)=d(i,j)/tlb

 dpert=(-(DIFF(I+1,J)+DIFF(I,J))*YMU(J)*YMU(J)
```

```
     1*PPH2/(2.*DP*DP)-CDIFF+CY1-CY22)*dabmu/dabs(ymu(j))/ratiob                    1570
     2-(APH(I)/DP+FPH(I)*(1.-DELP))*PPH2/DP/(1.d0+rratio)
      dpert=dpert/tlb

      IF(DYPH.NE.0.) THEN
      CNTY22=(1.-YPH*YPH)*0.5*(DCY(I,J+1)+DCY(I,J))*
     1(YPH*GAMAB-(GAMAB-OMC)/YPH)**2/(GAMAB*XP(I)*DMU*DMU*DYPH)
      ELSE
      CNTY22=0.
      ENDIF
                                                                                   1580
      E(I,J)=-(1.-YPH*YPH)*(tlb*FBI2*BI(I)/DMU+CPH)/DMU
     1-YMU(J)*(1.-YMU(J)*YMU(J))*CNTE-CNTCY1-CNTY22+tlb*TRPJ
      e(i,j)=e(i,j)/tlb

      epert=-(1.-YPH*YPH)*BI(I)/DMU/DMU
     &     *(1.-eff/yph/ymu(j+1))/(1+rratio)
     &+(-(1.-YPH*YPH)*CPH/DMU-YMU(J)*(1.-YMU(J)*YMU(J))*CNTE
     &     -CNTCY1-CNTY22)*dabp1/dabs(ymu(j+1))/ratiob
      epert=epert/tlb
                                                                                   1590
c mode 5
cccc This code has been modified only for the MODE=9 case.  If MODE=5
cccc (5-pt stencil) is tried, the code will stop here.
      IF(MODE.EQ.5) THEN
        write (6,*) 'Code not modified for MODE=5 cases.'
        stop
      END IF
c mode 9 diagonales
      IF (MODE.EQ.9) THEN
           Q(I,J)=T(I,J)*XP(I)*XP(I)/DT                                            1600
      IF(DABMU.NE.0.) THEN
      AM(I,J)=(-0.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)*YMU(J))/(DP*DMU)
     1+DCY(I,J)*(1.-YMU(J)*YMU(J))*((YMU(J)*YMU(J)-1.)*GAMAB/YMU(J)
     2+GAMAB*YMU(J)-(GAMAB-OMC)/YMU(J)+OMC/YMU(J))/(4.*DP*DMU*DABMU)
     &)/tlb
```

```
      AP(I,J)=-1.0*AM(I,J)
      EM(I,J)=-1.0*AM(I,J)
      EP(I,J)=AM(I,J)
      ELSE
      AM(I,J)=-0.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)*YMU(J))/(DP*DMU)          1610
     &/tlb
      AP(I,J)=-1.0*AM(I,J)
      EM(I,J)=-1.0*AM(I,J)
      EP(I,J)=AM(I,J)
      END IF
      ampert=am(i,j)*dabm1/dabs(ymu(j-1))/ratiob
      appert=ap(i,j)*dabm1/dabs(ymu(j-1))/ratiob
      empert=em(i,j)*dabp1/dabs(ymu(j+1))/ratiob
      eppert=ep(i,j)*dabp1/dabs(ymu(j+1))/ratiob
      END IF                                                                   1620

cccc Calculation of "Source term" S from f~ for calculation
cccc of g in RF+bootstrap case
      absm=dabs(ymu(j))
      if ((ipert.eq.1).and.(absm.gt.trap)) then
        so(i,j)=apert*ppr(i,j-1)+bpert*ppr(i-1,j)
     &       +cpert*ppr(i,j)+dpert*ppr(i+1,j)+epert*ppr(i,j+1)
     &       +ampert*ppr(i-1,j-1)+appert*ppr(i+1,j-1)
     &       +empert*ppr(i-1,j+1)+eppert*ppr(i+1,j+1)
      elseif ((ipert/10.eq.4).and.(absm.gt.trap)) then                         1630
        so(i,j)=-xp(i)*xp(i)*xp(i)/gama*ymu(j)/trpp(j)*f(i)
      else
        so(i,j)=0.d0
      endif

  4   CONTINUE

cccc Added to enforce symmetry in the trapped region and
cccc to give correct linking of the trapped, co-passing and counter-
cccc passing regions at the boundary                                           1640
      do 401 i=2,n1m1
```

```
  e(i,jtp1)=e(i,jtp1)/2.d0
  em(i,jtp1)=em(i,jtp1)/2.d0
  ep(i,jtp1)=ep(i,jtp1)/2.d0
  e(i,jtp2)=e(i,jtp2)/2.d0
  em(i,jtp2)=em(i,jtp2)/2.d0
  ep(i,jtp2)=ep(i,jtp2)/2.d0
  a(i,jtp1+1)=a(i,jtp1+1)/2.d0
  am(i,jtp1+1)=am(i,jtp1+1)/2.d0
  ap(i,jtp1+1)=ap(i,jtp1+1)/2.d0                                    1650
  a(i,jtp2+1)=a(i,jtp2+1)/2.d0
  am(i,jtp2+1)=am(i,jtp2+1)/2.d0
  ap(i,jtp2+1)=ap(i,jtp2+1)/2.d0
do 401 j=2,n2m1
absm=dabs(ymu(j))
if (j.le.n2/2) then
  if (absm.lt.trap) then
    a(i,j)=(a(i,j)+e(i,n2+1-j))/2.
    e(i,n2+1-j)=a(i,j)
    b(i,j)=(b(i,j)+b(i,n2+1-j))/2.                                  1660
    b(i,n2+1-j)=b(i,j)
    c(i,j)=(c(i,j)+c(i,n2+1-j))/2.
    c(i,n2+1-j)=c(i,j)
    d(i,j)=(d(i,j)+d(i,n2+1-j))/2.
    d(i,n2+1-j)=d(i,j)
    e(i,j)=(e(i,j)+a(i,n2+1-j))/2.
    a(i,n2+1-j)=e(i,j)
    am(i,j)=(am(i,j)+em(i,n2+1-j))/2.
    em(i,n2+1-j)=am(i,j)
    ap(i,j)=(ap(i,j)+ep(i,n2+1-j))/2.                               1670
    ep(i,n2+1-j)=ap(i,j)
    em(i,j)=(em(i,j)+am(i,n2+1-j))/2.
    am(i,n2+1-j)=em(i,j)
    ep(i,j)=(ep(i,j)+ap(i,n2+1-j))/2.
    ap(i,n2+1-j)=ep(i,j)
  endif
endif
```

```
401  continue

     DO 444 I=1,N1                                                    1680
     Q(I,1)=Q(I,3)
          Q(I,N2)=Q(I,N2-2)
     A(I,1)=0.
     E(I,N2)=0.
     B(I,1)=B(I,3)
     C(I,1)=C(I,3)
     C(I,N2)=C(I,N2-2)
     D(I,1)=D(I,3)
     D(I,N2)=D(I,N2-2)
     A(I,N2)=A(I,N2-2)                                                1690
     E(I,1)=E(I,3)
     so(i,1)=so(i,3)
     so(i,n2)=so(i,n2-2)
c 9 diagonales
     AM(I,1) = AM(I,3)
     AP(I,1) = AP(I,3)
     EM(I,1) = EM(I,3)
     EP(I,1) = EP(I,3)
     AM(I,N2) = AM(I,N2-2)
     AP(I,N2) = AP(I,N2-2)                                            1700
     EM(I,N2) = EM(I,N2-2)
     EP(I,N2) = EP(I,N2-2)
444  CONTINUE

     DO 445 J=1,N2
c The following line has been added for the limiting conditions at the center.
     C(1,J)=1.
     D(1,J)=-F(1)/F(2)
     so(1,j)=0.d0
     so(n1,j)=0.d0                                                    1710
445  CONTINUE

cccc As the comment below states, this is where the loop (to interate
```

```
cccc the distribution as a function of time) begins.
C     IT=1
C     FOR LHCD ALONE (NO FAST WAVE), IF DIFF(I,J) AND EPS ARE CONSTANT
C     START LOOP HERE
      time=0.d0
      tsave=0.d0
 30   CONTINUE                                                    1720
      time=time+dt

cccc Calculation of the l=1 component of the distribution
cccc function - to be put into the momentum conservation quantity
cccc C(F(maxwellian),F(l=1))
      DINT=0.
      DINT2=0.
      DO 151 I=1,N1
      DINT=0.
      DO 150 J=2,N2M1                                             1730
          if (it.ne.1) then
      DINT=DINT+T(I,J)*YMU(J)
          else
cccc For inclusion of f~ in l=1 component of collision operator
      dint=dint+ppr(i,j)*wti(j)*ymu(j)
          endif
      DINT2=DINT2+T(I,J)*(3.*YMU(J)*YMU(J)-1.)
 150  CONTINUE
      T2(I)=DINT2*DMU*5./4.
      T1(I)=DINT*DMU*3./2.                                        1740
C     IF(T1(I).LT.0.0D0) T1(I)=0.0D0
C     IF(T2(I).LT.0.0D0) T2(I)=0.0D0
 151  CONTINUE

cccc Braams-Karney calculation of the same l=1 component scattering
cccc potential
      AB1(1)=0.
      AB2(1)=0.
      AB3(1)=0.
```

```
 AB4(1)=0.                                                                  1750
 AB5(1)=0.
 AB6(1)=0.
 AB7(1)=0.
 AB8(1)=0.
 AB9(1)=0.
 AB10(1)=0.
 AB1(2)=0.
 AB2(2)=0.
 AB3(2)=0.
 AB4(2)=0.                                                                  1760
 AB5(2)=0.
 AB6(2)=0.
 AB7(2)=0.
 AB8(2)=0.
 AB9(2)=0.
 AB10(2)=0.
 DO 152 I=3,N1
 XP2=XP(I)*XP(I)
 GAMA=DSQRT(1.+XP2*BTH*BTH)
 V=XP(I)/GAMA                                                               1770
 GAMA1=DSQRT(1.+XP(I-1)*XP(I-1)*BTH*BTH)
 VONE=XP(I-1)/GAMA1
 AB1(I)=AB1(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416
1+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416
 AB2(I)=AB2(I-1)+0.5*DP*T1(I)*(XP(I)**5)*4.*3.1416/GAMA
1+0.5*DP*T1(I-1)*(XP(I-1)**5)*4.*3.1416/GAMA1
 AB3(I)=AB3(I-1)+0.5*DP*T1(I)*XP(I)*4.*3.1416*(GAMA-SIG(I)
1/(BTH*XP(I)))/GAMA
1+0.5*DP*T1(I-1)*XP(I-1)*4.*3.1416*(GAMA1-SIG(I-1)
1/(BTH*XP(I-1)))/GAMA1                                                      1780
 AB4(I)=AB4(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416/GAMA
1+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416/GAMA1
 AB5(I)=AB5(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416*(GAMA-SIG(I)
1/(BTH*XP(I))-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)
2/(GAMA*BTH*BTH*XP(I)*XP(I))
```

```
     3+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416*(GAMA1-SIG(I-1)
     4/(BTH*XP(I-1))-2.*GAMA1*BTH*BTH*XP(I-1)*XP(I-1)/3.)
     5/(GAMA1*BTH*BTH*XP(I-1)*XP(I-1))
      AB6(I)=AB6(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416
     1*(-3.*GAMA*SIG(I)/(BTH*BTH*XP(I)*XP(I))+3./(BTH*XP(I))                    1790
     2+BTH*XP(I))/(6.*GAMA*BTH*XP(I))
     3+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416
     4*(-3.*GAMA1*SIG(I-1)/(BTH*BTH*XP(I-1)*XP(I-1))+3./(BTH*XP(I-1))
     5+BTH*XP(I-1))/(6.*GAMA1*BTH*XP(I-1))
      AB8(I)=AB8(I-1)+.5*DP*T1(I)*4.*3.1416*(XP(I)**3)*(-3.*GAMA+SIG(I)
     1*(3./(BTH*XP(I))+2.*BTH*XP(I)))/(2.*GAMA*BTH*BTH*XP(I)*XP(I))
     2+0.5*DP*T1(I-1)*4.*3.1416*(XP(I-1)**3)*(-3.*GAMA1+SIG(I-1)*(3./
     3(BTH*XP(I-1))+2.*BTH*XP(I-1)))/(2.*GAMA1*BTH*BTH*XP(I-1)*XP(I-1))
      AB7(I)=AB7(I-1)+.5*DP*T1(I)*4.*3.1416*(XP(I)**3)*(-3.*GAMA*SIG(I)
     1/(BTH*BTH*XP(I)*XP(I))+3./(BTH*XP(I))+BTH*XP(I)-2.*(BTH*XP(I))            1800
     2**3/5.)/(2.*GAMA*BTH*BTH*BTH*XP(I))
     3+.5*DP*T1(I-1)*4.*3.1416*(XP(I-1)**3)*(-3.*GAMA1*SIG(I-1)
     4/(BTH*BTH*XP(I-1)*XP(I-1))+3./(BTH*XP(I-1))+BTH*XP(I-1)-2.*(BTH*
     5XP(I-1))**3/5.)/(2.*GAMA1*BTH*BTH*BTH*XP(I-1))
      AB9(I)=AB9(I-1)+0.5*DP*T1(I)*XP(I)*4.*3.1416*(GAMA*SIG(I)
     1/(BTH*XP(I))-1.)/GAMA
     2+0.5*DP*T1(I-1)*XP(I-1)*4.*3.1416*(GAMA1*SIG(I-1)
     3/(BTH*XP(I-1))-1.)/GAMA1
      AB10(I)=AB10(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416*(GAMA*SIG(I)
     1*(15./(BTH*XP(I))**2+6.)-15./(BTH*XP(I))-11.*BTH*XP(I))/                  1810
     2(12.*GAMA*BTH*BTH*BTH*XP(I))
     3+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416*(GAMA1*SIG(I-1)
     4*(15./(BTH*XP(I-1))**2+6.)-15./(BTH*XP(I-1))-11.*BTH*XP(I-1))/
     5(12.*GAMA1*BTH*BTH*BTH*XP(I-1))
152  CONTINUE
      AB11(N1)=0.
      AB12(N1)=0.
      AB13(N1)=0.
      AB14(N1)=0.
      DO 155 I=1,N1M1                                                           1820
      K=N1-I
```

```
      XP2=XP(K)*XP(K)
      GAMA=DSQRT(1.+XP2*BTH*BTH)
      GAMA1=DSQRT(1.+XP(K+1)*XP(K+1)*BTH*BTH)
      AB11(K)=AB11(K+1)+0.5*DP*T1(K)*4.*3.1416/GAMA
     1+0.5*DP*T1(K+1)*4.*3.1416/GAMA1
      AB12(K)=AB12(K+1)+0.5*DP*T1(K)*4.*3.1416
     1+0.5*DP*T1(K+1)*4.*3.1416
      AB13(K)=AB13(K+1)+0.5*DP*T1(K)*(XP(K)**2)*4.*3.1416
     1+0.5*DP*T1(K+1)*(XP(K+1)**2)*4.*3.1416                          1830
      AB14(K)=AB14(K+1)+0.5*DP*T1(K)*(XP(K)**2)*4.*3.1416/GAMA
     1+.5*DP*T1(K+1)*(XP(K+1)**2)*4.*3.1416/GAMA1
  155 CONTINUE

cccc Calculation of the right-hand side of the matrix equation, Q
      DO 502 I=2,N1M1
      XP2=XP(I)*XP(I)
      GAMA=DSQRT(1.+XP2*BTH*BTH)
      GAMAB=GAMA*RATIOB
      DO 502 J=2,N2M1                                                 1840
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.
      ENDIF
      DABMU=CNSTT
c mode 9 diagonales
      IF (MODE.EQ.9) THEN
          Q(I,J)=T(I,J)*XP(I)*XP(I)/DT                                1850
        if ((j.eq.jtp1).or.(j.eq.jtp2)) then
          q(i,j)=q(i,j)-e(i,j)*t(i,n2-j)
     &           -em(i,j)*t(i-1,n2-j)
     &           -ep(i,j)*t(i+1,n2-j)
        endif
        if ((j.eq.jtp1+1).or.(j.eq.jtp2+1)) then
          q(i,j)=q(i,j)-a(i,j)*t(i,n2+2-j)
```

```
     &          -am(i,j)*t(i-1,n2+2-j)
     &          -ap(i,j)*t(i+1,n2+2-j)
        endif                                                          1860
      ENDIF
  502 CONTINUE
c Put the l=1 conservation term into the RHS of the matrix equ. (Q)
      DO 503 I=2,N1M1
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
      V=XP(I)/GAMA
           CNT=-2.*AB1(I)*GAMA/3.+AB2(I)*GAMA/5.+AB3(I)+AB4(I)/3.
     1+(-AB5(I)+AB7(I)+AB8(I)-AB10(I))*GAMA
      CNT=CNT+AB6(I)*(GAMA*XP(I)*XP(I)-5.)+AB9(I)*XP(I)*XP(I)
      CNT=CNT/(GAMA*XP(I)*XP(I))                                       1870
      CNT1=+AB11(I)*(1./3.+(GAMA-SIG(I)/(BTH*XP(I)))/(XP(I)*XP(I))
     1-5.*(-3.*GAMA*SIG(I)
     2/(BTH*BTH*XP(I)*XP(I))+3./(BTH*XP(I))+BTH*XP(I))/(6.*BTH*XP(I)))
      CNT1=CNT1+AB12(I)*(-2.*GAMA/3.+XP(I)*XP(I)/5.-(GAMA-SIG(I)
     1/(BTH*XP(I))-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)/(BTH*BTH*XP(I)
     2*XP(I)))
      CNT1=CNT1+AB12(I)*(-3.*GAMA*SIG(I)/(BTH*BTH*XP(I)*XP(I))+3.
     1/(BTH*XP(I))+BTH*XP(I)-2.*(BTH*XP(I))**3/5.)/(2.*BTH*BTH*BTH
     2*XP(I))
      CNT1=CNT1+AB12(I)*(-3.*GAMA+SIG(I)*(3./(BTH*XP(I))+2.*BTH*XP(I)))/  1880
     1(2.*BTH*BTH*XP(I)*XP(I))
      CNT1=CNT1-AB12(I)*(GAMA*SIG(I)*(15./(BTH*BTH*XP(I)*XP(I))+6.)
     1-15./(BTH*XP(I))-11.*BTH*XP(I))/(12.*BTH*XP(I)*BTH*BTH)
      CNT1=CNT1+AB13(I)*(-3.*GAMA*SIG(I)/(BTH*BTH*XP(I)*XP(I))+3.
     1/(BTH*XP(I))+BTH*XP(I))/(6.*BTH*XP(I))
      CNT1=CNT1+AB14(I)*(GAMA*SIG(I)/(BTH*XP(I))-1.)/(XP(I)*XP(I))
      CNT1=CNT1*XP(I)/GAMA
      CNT=(CNT*F(I)+CNT1*F(I))+4.*3.1416*F(I)*T1(I)/GAMA
      CNT=CNT*XP(I)*XP(I)
      DO 503 J=2,N2M1                                                  1890
         if (it.eq.1) then
           so(i,j)=so(i,j)-EJ0(J)*CNT*YMU(J)*TRUNC
         endif
```

```
      Q(I,J)=Q(I,J)+EJ0(J)*CNT*YMU(J)*TRUNC
ccc Put in friction with the ion neoclassical flow
     &        +z*ymu(j)*upi*f(i)*trpp(n2m1)/trpp(j)
 503  CONTINUE
      DO 513 I=1,N1
      Q(I,1)=Q(I,3)
      Q(I,N2)=Q(I,N2-2)                                              1900
 513  CONTINUE

cccc Reduction of the matrix equation to the difference --
cccc Instead of M*T(step+1)=B, the reduced equation is
cccc          M*(T(step+1)-T(step))=Bprime=B-M*T(step)
      DO 80 J=2,N2M1
      DO 60 I=2,N1M1
      IF (MODE.EQ.9) THEN
c mode 9 diagonales
      IF(C(I,J).NE.0.D0) THEN                                        1910
      R(I,J)=Q(I,J)-A(I,J)*T(I,J-1)-B(I,J)*T(I-1,J)-C(I,J)*T(I,J)-
     1D(I,J)*T(I+1,J)-E(I,J)*T(I,J+1)
     2-COLLF*XP(I)*XP(I)*T(I,J)
     3-AM(I,J)*T(I-1,J-1)-AP(I,J)*T(I+1,J-1)-EM(I,J)*T(I-1,J+1)
     4-EP(I,J)*T(I+1,J+1)
      ELSE
      R(I,J)=Q(I,J)-T(I,J)*XP(I)*XP(I)/DT
      END IF
      r(i,j)=r(i,j)-so(i,j)
      END IF                                                         1920
 60   CONTINUE
 80   CONTINUE
      DO 81 I=2,N1M1
      IF(C(I,N2).NE.0.0D0) THEN
      R(I,1)=R(I,3)
      R(I,N2)=R(I,N2-2)
      ELSE
      r(i,1)=q(i,1)-t(i,1)*xp(i)*xp(i)/dt-so(i,1)
      R(I,N2)=Q(I,N2)-T(I,N2)*XP(I)*XP(I)/DT-so(i,n2)
```

```
      END IF                                                              1930
   81 CONTINUE
   55 FORMAT(2I10,2F16.6)
      DO 500 J=1,N2
      Q(N1,J)=T(N1,J)*XP(N1)*XP(N1)/DT
      Q(1,J)=0.
      R(1,J)=0.
      R(N1,J)=0.
  500 CONTINUE

cccc MATRIX EQUATION SOLVE - uses sip9d, with the Strongly Implicit       1940
cccc Method
      IFAIL=0
      IF (MODE.EQ.9) THEN
            ALPHAMAX = 1.- 2.*APARAM/(N1M1*N1M1+N2M1*N2M1)
      call sip9d(N1,N2,AM,A,AP,B,C,D,EM,E,EP,R,IT,ALPHAMAX,WRKSP1,WRKSP2
     1,WRKSP3,WRKSP4)
      ENDIF

      MPRINT=MPRINT+1
      DO 6 J=1,N2                                                         1950
      DO 6 I=1,N1
      T(I,J)=T(I,J)+R(I,J)
    6 CONTINUE
cccc Enforce the symmetry in the trapped region
      DO 501 I=1,N1
      T(I,1)=T(I,3)
  501 T(I,N2)=T(I,N2-2)
      IF(TRAPO.EQ.0.0D0) GO TO 1700
      N2H=N2/2
      DO I=1,N1                                                           1960
      DO J=1,N2H
      ABSM=DABS(YMU(J))
      CNST=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNST.GT.0.0.AND.CNST.LE.1.) THEN
      CNST=DSQRT(CNST)
```

```
      ELSE
      CNST=0.
      ENDIF
      IF(ABSM.LE.TRAP) T(I,N2+1-J)=(T(I,N2+1-J)+T(I,J))/2.
      IF(ABSM.LE.TRAP) T(I,J)=T(I,N2+1-J)                              1970
      ENDDO
      ENDDO
 1700 CONTINUE

      IT=IT+1
      IF(IT.EQ.NITS) GO TO 101
      IF(MPRINT.LT.NPRINT) GO TO 100
      MPRINT=0.
   51 FORMAT(5X,4F16.8,I10)
                                                                       1980
cccc Original routine to calculate density and parallel flow
  101 DENS=0.0
      CUR=0.
      CURR=0.
      CURRR=0.
      DO 5 J=2,N2M1
      DO 5 I=2,N1M1
      ABSM=DABS(YMU(J))
      XPH=XP(I)-DP/2.
      GAMAH=DSQRT(1.D0+XPH*XPH*BTH*BTH)                                1990
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.
      ENDIF
      IF(YMU(J).LT.0.0) CNSTT=-CNSTT
      PPAR=XP(I)*CNSTT
      VPAR=PPAR/GAMA                                                   2000
c Correction
```

```
      ppar=xp(i)*ymu(j)*DSQRT((1.d0+rratio)/(1.d0-rratio))

           IF(TRAPO.EQ.0.) THEN
      CUR=CUR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           ELSE
           IF(ABSM.GE.TRAP.AND.YMU(J).LT.0.)
     1CURRR=CURRR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           IF(ABSM.GE.TRAP.AND.YMU(J).GT.0.)
     1 CUR=CUR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA                              2010
           IF(ABSM.LT.TRAP)
     1 CURR=CURR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           ENDIF
c     DENS=DENS+2.*T(I,J)*XP(I)*XP(I)+2.*(T(I,J)+T(I-1,J))*XPH*XPH
      exs=1.d0
      if ((j.eq.2).or.(j.eq.n2m1)) exs=3.d0/8.d0
      if ((j.eq.3).or.(j.eq.n2m1-1)) exs=7.d0/6.d0
      if ((j.eq.4).or.(j.eq.n2m1-2)) exs=23.d0/24.d0
      dens=dens+t(i,j)*xp(i)*xp(i)*(1.d0+rratio)
     &     *trpp(j)/trpp(n2m1)*exs                                        2020
    5 CONTINUE

C     CALCULATE DERIVATIVE OF THE DISTRIBUTION FUNCTION USING CUBIC SPLINE
      DO 506 J=1,N2
      DE(1)=0.
      DF(1)=0.
        CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
        IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
          CNSTT=DSQRT(CNSTT)
        ELSE                                                             2030
          CNSTT=0.
        ENDIF
        cfac=cnstt/DABS(ymu(j))/ratiob
      DO 507 I=2,N1M1
      DFI=3.*(T(I+1,J)-T(I-1,J))/DP
     &     +3.*(ppr(i+1,j)-ppr(i-1,j))/dp*cfac
      DE(I)=-1./(DE(I-1)+4.)
```

```
 507  DF(I)=(DF(I-1)-DFI)*DE(I)
      DFDP(N1,J)=0.0
      DO 508 I=1,N1M1                                                    2040
      K=N1-I
 508  DFDP(K,J)=DE(K)*DFDP(K+1,J)+DF(K)
 506  CONTINUE
      DO 526 I=1,N1
      DE(1)=0.
      DF(1)=0.0
      DO 527 J=2,N2M1
        CNSTT=1.D0-RATIOB*(1.D0-YMU(J-1)*YMU(J-1))
        IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
          CNSTT=DSQRT(CNSTT)                                             2050
        ELSE
          CNSTT=0.
        ENDIF
        cfac1=cnstt/DABS(ymu(j-1))/ratiob
        CNSTT=1.D0-RATIOB*(1.D0-YMU(J+1)*YMU(J+1))
        IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
          CNSTT=DSQRT(CNSTT)
        ELSE
          CNSTT=0.
        ENDIF                                                            2060
        cfac2=cnstt/DABS(ymu(j+1))/ratiob
      DFI=3.*(T(I,J+1)-T(I,J-1))/DMU
     &      +3.*(ppr(i,j+1)*cfac2-ppr(i,j-1)*cfac1)/dmu
      DE(J)=-1./(DE(J-1)+4.)
 527  DF(J)=(DF(J-1)-DFI)*DE(J)
      DFDM(I,N2)=0.0
      DO 528 J=1,N2M1
      K=N2-J
 528  DFDM(I,K)=DE(K)*DFDM(I,K+1)+DF(K)
 526  CONTINUE                                                           2070

cccc Calculation of power deposition
      PABS1=0.
```

```
      PABSR=0.
      PABSRR=0.
      PABSRRR=0.
    DO 505 J=2,N2M1
    DO 505 I=2,N1M1
    ABSM=DABS(YMU(J))
    GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)                                    2080
    GAMAB=GAMA*RATIOB
    CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
    IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
    CNSTT=DSQRT(CNSTT)
    ELSE
    CNSTT=0.
    ENDIF
    IF(YMU(J).LT.0.0) CNSTT=-CNSTT
    PPAR=XP(I)*CNSTT
    PPERP=XP(I)*XP(I)-PPAR*PPAR                                             2090
    VPAR=PPAR/GAMA
    IF(PPAR.EQ.0.) GO TO 3000
c Correction by S. Schultz
    cfac=(1.d0+rratio)/trpp(j)
      IF(TRAPO.EQ.0.) THEN
    PABSR=PABSR+(XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+
   2(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))/GAMA+(XP(I)**3)
   3*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB
   4*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
    ELSE                                                                    2100
      IF(ABSM.GE.TRAP.AND.YMU(J).LT.0.)
   1PABSRRR=PABSRRR+((XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+
   1(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))*TRPP(J)/GAMA+TRPP(J)*(XP(I)
   1**3)*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB
   1*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
   &)*cfac
      IF(ABSM.GE.TRAP.AND.YMU(J).GT.0.)
   1PABSR=PABSR+((XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+
   2(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))*TRPP(J)/GAMA+TRPP(J)*(XP(I)
```

```
     3**3)*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB          2110
     4*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
     7)*cfac
          IF(ABSM.LT.TRAP)
     1PABSRR=PABSRR+((XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+
     2(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))*TRPP(J)/GAMA+TRPP(J)*(XP(I)
     3**3)*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB
     4*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
     7)*cfac
          ENDIF
 3000 CONTINUE                                                            2120
 505  CONTINUE

 518  FORMAT(5X,F12.5,2G12.5)
c      DENS=DENS*2.*3.1416*DP*DMU/6.
      dens=dens*2.*3.1416*dp*dmu
      CUR=CUR*2.*3.1416*DP*DMU
      CURR=CURR*2.*3.1416*DP*DMU
      CURRR=CURRR*2.*3.1416*DP*DMU
      TCURR=CUR+CURR+CURRR
      PABSR=-PABSR*2.*3.1416*DP*DMU                                       2130
      PABSRR=-PABSRR*2.*3.1416*DP*DMU
      PABSRRR=-PABSRRR*2.*3.1416*DP*DMU
      PABS1=PABSR+PABSRR+PABSRRR

      IF(PABS1.EQ.0.) GO TO 5181
      RJP1=TCURR/PABS1
 5181 CONTINUE

C     TO RENORMALIZE THE SOLUTION TO KEEP DENSITY STRICTLY 1
      if (ipert.eq.0) then                                                2140
      DO 300 J=1,N2
      DO 300 I=1,N1
  300 T(I,J)=T(I,J)/DENS
C     RENORMALIZED VALUES
      CUR=CUR/DENS
```

```
      CURR=CURR/DENS
      CURRR=CURRR/DENS
      TCURR=TCURR/DENS
      PABSR=PABSR/DENS
      PABSRR=PABSRR/DENS                                          2150
      PABSRRR=PABSRRR/DENS
      PABS1=PABS1/DENS
      DENS=DENS/DENS
      else
cccc 'Boundary' condition for calculation of g -- assume g=0
cccc at ppar=0
        do 301 i=1,n1
          do 301 j=1,n2
            absm=DABS(ymu(j))
            if (absm.lt.trap) then                                2160
              t(i,j)=0.d0
            endif
 301      continue
        endif

  100 CONTINUE
   50  FORMAT(I4,2X,4F15.6)

cccc Check convergence criterion at every eighth time step
          if (mod(it-1,8).eq.0) then                              2170
            chk=DABS((tcurr-tsave)/tcurr)/(8*dt)
          WRITE(13,1001) IT,TCURR,PABS1,CHK
 1001      format(i5,3(1x,e10.4))
            if (chk.lt.alimit) it=nits+1
            tsave=tcurr
          endif
      IF(IT.LE.NITS) GO TO 30
      write (6,*) time,atmax
      dt=dtt
                                                                  2180
cs  56  FORMAT(5X,6F14.8)
```

```
C     WRITE THE DISTRIBUTION FUNCTION IN A FILE
cs      OPEN(UNIT=11,FILE='distr.dat',FORM='UNFORMATTED')
C           OPEN(UNIT=11,FILE='distr.dat',STATUS='UNKNOWN')
cs      WRITE(11) ((T(I,J),I=1,N1),J=2,N2M1)
cs      CLOSE(11)
 5000 CONTINUE
cs      STOP

cccc Output diffusion coefficients of N1xN2 grid to study                    2190
cccc by Matlab
      do i=1,n1m1
        do j=1,n2
          dout(i,j)=diff(i,j)
              if (dperp.gt.1.d-12) dout(i,j)=dcy(i,j)
        enddo
      enddo

cccc Renormalization of Spitzer-Harm distribution
      if (ipert.eq.40) then                                                  2200
        do i=1,n1m1
          do j=1,n2
            t(i,j)=t(i,j)/f(i)
          enddo
        enddo
      endif

      return
      END
                                                                             2210

      SUBROUTINE sip9d(m,n,a1,a2,a3,a4,cc,a5,a6,a7,a8,r,kt,alphamax,w1
     &,w2,w3,w4)
cccc This routine solves the matrix equation using the strongly implicit
cccc method, for a matrix with 9 diagonals
            integer n,m,kt,iacc
            integer itc,itc00,itc01,itc10,itc11
```

```
      integer i,ii,j,jj
      integer it,ib,jb
      integer z1,z2,z3,z4                                                    2220
      double precision a1(m,n),a2(m,n),a3(m,n),a4(m,n),a5(m,n),a6(m,n)
      double precision a7(m,n),a8(m,n),cc(m,n),r(m,n),w1(m,n)
      double precision alpha,alphamax,alp(9),w2(m,n),w3(m,n),w4(m,n)
      double precision fibjb,fijb,fitjb,gibjb,gijb,gitjb,hibjb
      double precision hijb,hitjb,iibjb,iijb,iitjb,ribjb,rijb,ritjb
      double precision fibj,gibj,hibj,iibj,ribj
      double precision sa,sb,sc,sd,se
c
      data alp(1),alp(2),alp(3),alp(4),alp(5),alp(6),alp(7),alp(8)
     &,alp(9)/1.0D0,0.625D0,0.25D0,0.875D0,0.5D0,0.125D0,0.75D0,             2230
     &0.375D0,0.0D0/
c

      n1 = n + 1
      m1 = m + 1

c     Alternate direction technique

      itc = mod(kt-1,4) + 1
      if (itc.eq.0) itc = 4                                                  2240

      if (itc.eq.1) then
            itc00 = 1
            itc01 = 0
            itc10 = 1
            itc11 = 0
      endif

      if (itc.eq.2) then
            itc00 = 0                                                        2250
            itc01 = 1
            itc10 = 1
            itc11 = 0
```

```
      endif

      if (itc.eq.3) then
            itc00 = 1
            itc01 = 0
            itc10 = 0
            itc11 = 1                                                    2260
      endif

      if (itc.eq.4) then
            itc00 = 0
            itc01 = 1
            itc10 = 0
            itc11 = 1
      endif

      z1 = itc00*itc10                                                   2270
      z2 = itc01*itc10
      z3 = itc00*itc11
      z4 = itc01*itc11

c Acceleration parameter

      iacc = mod(kt-1,36)
      if (iacc.lt.0) iacc = iacc +36

      iacc = iacc/4 + 1                                                  2280
      alpha = 1.0 - (1.0 - alphamax)**alp(iacc)

c Approximate factorisation and inversion of the lower triangular matrix

      do 10 jj=1,n

            j = itc00*jj + itc01*(n1-jj)
            jb = j - itc00 + itc01
```

```
do 10 ii=1,m                                                    2290

      i = itc10*ii + itc11*(m1-ii)
      ib = i - itc10 + itc11
      it = i + itc10 - itc11

      if (cc(i,j).eq.0.0) goto 100

      if ((jb.eq.0).or.(jb.eq.n1)) then
            fibjb = 0.0
            fijb = 0.0                                          2300
            fitjb = 0.0
            gibjb = 0.0
            gijb = 0.0
            gitjb = 0.0
            hibjb = 0.0
            hijb = 0.0
            hitjb = 0.0
            iibjb = 0.0
            iijb = 0.0
            iitjb = 0.0                                         2310
            ribjb = 0.0
            rijb = 0.0
            ritjb = 0.0
      else
            if ((ib.eq.0).or.(ib.eq.m1)) then
                  fibjb = 0.0
                  gibjb = 0.0
                  hibjb = 0.0
                  iibjb = 0.0
                  ribjb = 0.0                                   2320
            else
                  fibjb = w1(ib,jb)
                  gibjb = w2(ib,jb)
                  hibjb = w3(ib,jb)
                  iibjb = w4(ib,jb)
```

```
                ribjb = r(ib,jb)
            endif

            if ((it.eq.0).or.(it.eq.m1)) then
                fitjb = 0.0                                          2330
                gitjb = 0.0
                hitjb = 0.0
                iitjb = 0.0
                ritjb = 0.0
            else
                fitjb = w1(it,jb)
                gitjb = w2(it,jb)
                hitjb = w3(it,jb)
                iitjb = w4(it,jb)
                ritjb = r(it,jb)                                     2340
            endif

            fijb = w1(i,jb)
            gijb = w2(i,jb)
            hijb = w3(i,jb)
            iijb = w4(i,jb)
            rijb = r(i,jb)
        endif

        if ((ib.eq.0).or.(ib.eq.m1)) then                            2350
            fibj = 0
            gibj = 0
            hibj = 0
            iibj = 0
            ribj = 0
        else
            fibj = w1(ib,j)
            gibj = w2(ib,j)
            hibj = w3(ib,j)
            iibj = w4(ib,j)                                          2360
            ribj = r(ib,j)
```

```
                endif

                sa = a1(i,j)*z1 + a6(i,j)*z2 + a3(i,j)*z3 + a8(i,j)*z4
                sb = (a2(i,j)*z1 + a7(i,j)*z2 + a2(i,j)*z3 + a7(i,j)*z4)
     & - sa*fibjb
                sc = ((a3(i,j)*z1 + a8(i,j)*z2 + a1(i,j)*z3+a6(i,j)*z4)
     & - sb*fijb)          /(1+alpha*fitjb)
                sd = ((a4(i,j)*z1 + a4(i,j)*z2 + a5(i,j)*z3 +a5(i,j)*z4)
     & - sb*gijb - sa*hibjb - 2*alpha*sa*gibjb)/(1 + alpha*gibj)                2370
                se = cc(i,j) - sd*fibj - sc*gitjb - sb*hijb - sa*iibjb +
     & alpha*(sc*fitjb + sa*gibjb + sc*iitjb + sd*gibj)

                w1(i,j) = ((a5(i,j)*z1 + a5(i,j)*z2 + a4(i,j)*z3 +
     &a4(i,j)*z4) - sb*iijb - sc*hitjb - alpha*sc*(2*iitjb + fitjb))/se
                w2(i,j) = ((a6(i,j)*z1 + a1(i,j)*z2 + a8(i,j)*z3 +
     &a3(i,j)*z4) - sd*hibj - alpha*sd*gibj)/se
                w3(i,j) = ((a7(i,j)*z1 + a2(i,j)*z2 + a7(i,j)*z3 +
     &a2(i,j)*z4) - sd*iibj)/se
                w4(i,j) = (a8(i,j)*z1 + a3(i,j)*z2 + a6(i,j)*z3 +              2380
     &a1(i,j)*z4)/se


c Inversion of the lower triangular matrix

                r(i,j) = (r(i,j) - sa*ribjb - sb*rijb - sc*ritjb
     & - sd*ribj)/se

                goto 10

                                                                               2390
  100     w1(i,j) = 0.0
          w2(i,j) = 0.0
                w3(i,j) = 0.0
          w4(i,j) = 0.0

   10 continue
```

```
c Inversion of the upper triangular matrix */
                                                                      2400
      do 20 jj=1,n

       j = itc00*(n1-jj) + itc01*jj
       jb = j + itc00 - itc01

       do 20 ii=1,m

          i = itc10*(m1-ii) + itc11*ii
          ib = i + itc10 - itc11
          it = i - itc10 + itc11                                      2410

          if ((jb.ne.0).and.(jb.ne.n1)) then

             if ((ib.ne.0).and.(ib.ne.m1)) then
                r(i,j) = r(i,j) - w4(i,j)*r(ib,jb)
             endif

             if ((it.ne.0).and.(it.ne.m1)) then
                r(i,j) = r(i,j) - w2(i,j)*r(it,jb)
             endif                                                    2420

             r(i,j) = r(i,j) - w3(i,j)*r(i,jb)
          endif



          if ((ib.ne.0).and.(ib.ne.m1)) then
             r(i,j) = r(i,j) - w1(i,j)*r(ib,j)
          endif

   20 continue                                                        2430

       return
       end
```

```
      DOUBLE PRECISION FUNCTION S11ABF(X,IFAIL)
c Calculates inverse hyperbolic sine
C     MARK 5A REVISED - NAG COPYRIGHT 1976
C     MARK 11.5(F77) REVISED. (SEPT 1985.)
C     ARCSINH(X)                                                        2440
C
C     .. Scalar Arguments ..
      DOUBLE PRECISION          X
      INTEGER                   IFAIL
C     .. Local Scalars ..
      DOUBLE PRECISION          LN2, T, XHI, Y
C     .. Intrinsic Functions ..
      INTRINSIC                 ABS, LOG, SIGN, SQRT
C     .. Data statements ..
C     PRECISION DEPENDENT CONSTANTS                                     2450
C08   DATA XHI,LN2/1.0D+5,6.9314718D-1/
C12   DATA XHI,LN2/1.0D+7,6.93147180560D-1/
C14   DATA XHI,LN2/1.0D+8,6.9314718055995D-1/
      DATA XHI,LN2/1.0D+9,6.931471805599453D-1/
C18   DATA XHI,LN2/1.0D+10,6.9314718055994309D-1/
C     .. Executable Statements ..
C
C     NO FAILURE EXITS
      IFAIL = 0
      T = ABS(X)                                                        2460
C     TEST LARGE RANGE
      IF (T.GT.XHI) GO TO 20
      T = X*X
C
C     TEST FOR MIDDLE RANGE
      IF (T.GT.1.0D0) GO TO 40
C     EXPANSION ARGUMENT
      T = 2.0D0*T - 1.0D0
C
```

```
C     * EXPANSION (0012) *                                                    2470
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 08E
C08   Y = (((((((+2.5373657D-6)*T-8.8982681D-6)*T+2.6962688D-5)
C08  *    *T-1.0390000D-4)*T+4.2316523D-4)*T-1.8337913D-3)
C08  *    *T+9.0042679D-3)*T-5.7366614D-2)*T + 9.3122986D-1
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 12E
C12   Y = ((((((((((((-6.08021584775D-9)*T+1.98638973110D-8)
C12  *    *T-4.57508091809D-8)*T+1.58893891259D-7)
C12  *    *T-5.83039413992D-7)*T+2.05819238929D-6)                           2480
C12  *    *T-7.40199001696D-6)*T+2.74058634726D-5)
C12  *    *T-1.05071718894D-4)*T+4.23002689824D-4)
C12  *    *T-1.83345864783D-3)*T+9.00428855313D-3)
C12  *    *T-5.73666392630D-2)*T + 9.31229859453D-1
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 14E
C14   Y = ((((((((((((((-5.8308026585469D-10)*T+1.8762742119651D-9)
C14  *    *T-3.8936648507940D-9)*T+1.3296937569163D-8)
C14  *    *T-4.9030635676319D-8)*T+1.6792346090382D-7)
C14  *    *T-5.8053399097437D-7)*T+2.0520358645332D-6)                       2490
C14  *    *T-7.4030149627429D-6)*T+2.7408018256297D-5)
C14  *    *T-1.0507150365535D-4)*T+4.2300233069384D-4)
C14  *    *T-1.8334586677591D-3)*T+9.0042885755717D-3)
C14  *    *T-5.7366639262479D-2)*T + 9.3122985945271D-1
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 16E
      Y = ((((((((((((((+1.810792296549804D-11
     *    *T-5.731943029121004D-11)*T+1.008344962167889D-10)
     *    *T-3.394726871170490D-10)*T+1.299779213740398D-9)
     *    *T-4.319978113584910D-9)*T+1.432753532351304D-8)                   2500
     *    *T-4.863477336087045D-8)*T+1.670117348345774D-7)
     *    *T-5.807433412373489D-7)*T+2.052474396638805D-6)
     *    *T-7.402952157663977D-6)*T+2.740790473603819D-5)
     *    *T-1.050715136470630D-4)*T+4.230023450529706D-4)
     *    *T-1.833458667045431D-3)*T+9.004288574881897D-3
```

```
     Y = (Y*T-5.736663926249348D-2)*T + 9.312298594527122D-1
C
C    EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 18E
C18   Y = ((((((((((((((+1.83000936086660710D-12
C18  *    *T-5.74532833323253760D-12)*T+8.95787616116500070D-12)                2510
C18  *    *T-3.00291207083554898D-11)*T+1.20278345675996627D-10)
C18  *    *T-3.94053306282758078D-10)*T+1.27690409672956527D-9)
C18  *    *T-4.26028056137241518D-9)*T+1.43437981020129308D-8)
C18  *    *T-4.86735767698085752D-8)*T+1.67004579212037421D-7)
C18  *    *T-5.80728097040980303D-7)*T+2.05247631332341414D-6)
C18  *    *T-7.40295567555544635D-6)*T+2.74079044411636393D-5)
C18  *    *T-1.05071513207326607D-4)*T+4.23002345076007640D-4
C18   Y = (((Y*T-1.83345866707041640D-3)*T+9.00428857488119850D-3)
C18  *    *T-5.73666392624930609D-2)*T + 9.31229859452712177D-1
C                                                                              2520
C
      S11ABF = X*Y
      RETURN
C
C     LARGE X
   20 S11ABF = SIGN(DLOG(T)+LN2,X)
      RETURN
C
C     MIDDLE X
   40 S11ABF = SIGN(DLOG(SQRT(T+1.0D0)+ABS(X)),X)                              2530
      RETURN
C
      END


      double precision FUNCTION delk(ak2)
cccc Complete Elliptic Integral K(k)
      double precision ak2
CU    USES drf
      double precision drf
      delk=drf(0.d0,1.d0-ak2,1.d0)                                            2540
      return
```

```
      END
      double precision FUNCTION dele(ak2)
cccc Complete Elliptic Integral E(k)
      double precision ak2
CU    USES drd,drf
      double precision drd,drf
      dele=drf(0.d0,1.d0-ak2,1.d0)-ak2*drd(0.d0,1.d0-ak2,1.d0)/3.d0
      return
      END                                                              2550
      double precision FUNCTION drf(x,y,z)
cccc Partial Elliptic Integral
      double precision x,y,z,ERRTOL,TINY,BIG,THIRD,C1,C2,C3,C4
      PARAMETER (ERRTOL=.08,TINY=1.5d-38,BIG=3.D37,THIRD=1./3.,
     *C1=1./24.,C2=.1,C3=3./44.,C4=1./14.)
      double precision alamb,ave,delx,dely,delz,e2,e3,sqrtx,sqrty
     *,sqrtz,xt,yt,zt
      if(min(x,y,z).lt.0..or.min(x+y,x+z,y+z).lt.TINY.or.max(x,y,
     *z).gt.BIG)pause 'invalid arguments in drf'
      xt=x                                                             2560
      yt=y
      zt=z
1     continue
        sqrtx=dsqrt(xt)
        sqrty=dsqrt(yt)
        sqrtz=dsqrt(zt)
        alamb=sqrtx*(sqrty+sqrtz)+sqrty*sqrtz
        xt=.25*(xt+alamb)
        yt=.25*(yt+alamb)
        zt=.25*(zt+alamb)                                              2570
        ave=THIRD*(xt+yt+zt)
        delx=(ave-xt)/ave
        dely=(ave-yt)/ave
        delz=(ave-zt)/ave
      if(max(dabs(delx),dabs(dely),dabs(delz)).gt.ERRTOL)goto 1
      e2=delx*dely-delz**2
      e3=delx*dely*delz
```

```
      drf=(1.+(C1*e2-C2-C3*e3)*e2+C4*e3)/dsqrt(ave)
      return
      END                                                              2580
      double precision FUNCTION drd(x,y,z)
cccc Partial Elliptic Integral
      double precision x,y,z,ERRTOL,TINY,BIG,C1,C2,C3,C4,C5,C6
      PARAMETER (ERRTOL=.05,TINY=1.d-25,BIG=4.5D21,C1=3./14.,C2=1./6.,
     *C3=9./22.,C4=3./26.,C5=.25*C3,C6=1.5*C4)
      double precision alamb,ave,delx,dely,delz,ea,eb,ec,ed,ee,fac,
     *sqrtx,sqrty,sqrtz,sum,xt,yt,zt
      if(min(x,y).lt.0..or.min(x+y,z).lt.TINY.or.max(x,y,
     *z).gt.BIG)pause 'invalid arguments in drd'
      xt=x                                                             2590
      yt=y
      zt=z
      sum=0.
      fac=1.
1     continue
        sqrtx=dsqrt(xt)
        sqrty=dsqrt(yt)
        sqrtz=dsqrt(zt)
        alamb=sqrtx*(sqrty+sqrtz)+sqrty*sqrtz
        sum=sum+fac/(sqrtz*(zt+alamb))                                 2600
        fac=.25*fac
        xt=.25*(xt+alamb)
        yt=.25*(yt+alamb)
        zt=.25*(zt+alamb)
        ave=.2*(xt+yt+3.*zt)
        delx=(ave-xt)/ave
        dely=(ave-yt)/ave
        delz=(ave-zt)/ave
      if(max(dabs(delx),dabs(dely),dabs(delz)).gt.ERRTOL)goto 1
      ea=delx*dely                                                     2610
      eb=delz*delz
      ec=ea-eb
      ed=ea-6.*eb
```

```
      ee=ed+ec+ec
      drd=3.*sum+fac*(1.+ed*(-C1+C5*ed-C6*delz*ee)+delz*(C2*ee+delz*
     &(-C3*ec+delz*C4*ea)))/(ave*dsqrt(ave))
      return
      END
```

D.2 The FASTFP Fokker-Planck Code

```
C     FULLY RELATIVISTIC COLLISIONAL OPERATOR BRAAMS-KARNEY
C     N1 NUMBER OF POINTS IN THE MOMENTUM P
C     N2 NUMBER OF POINTS IN COS(THETA)
C     TRAPO=1. TRAPPING INCLUDED IN THE CALCULATION
C     TRAPO=0. TRAPPING EXCLUDED IN THE CALCULATION
C     MODE=5 USE FIVE POINTS DIAGONAL SOLVER
C     MODE=9 USE NINE POINTS DIAGONAL SOLVER
C     APARAM PARAMETERS FOR CONVERGENCE (USUALLY 60)
C     NITS NUMBER OF ITERATIONS NEEDED
C     PMAX MAXIMUM MOMENTUM                                          10
C     TRUNC=1 INCLUDES THE COLLISION TERM FOR MOMENTUM CONSERVATION
C     PMIN MINIMUM VALUE OF THE MOMENTUM ,USUALLY ZERO.
C     TE ELECTRON MOMENTUM
C     DT TIME STEP
C     MINIMUM VALUE OF THE VELOCITY FOR THE LOWER HYBRID SPECTRUM
C     MAXIMUM VALUE OF THE VELOCITY FOR THE LOWER HYBRID SPECTRUM
C     DDIF QUASILINEAR DIFFUSION OPERATOR FOR THE LOWER HYBRID WAVE
C     OMC RATIO OF THE CYCLOTRON FREQUENCY OVER THE WAVE FREQUENCY
C     DPERP COEFFICIENT FOR THE ELECTRON CYCLOTRON DIFFUSION COEFFICIENT
C     DNRES RESONANCE REFRACTIVE INDEX FOR THE ELECTRON CYCLOTRON WAVE
C     DNPAR WIDTH OF THE ELECTRON CYCLOTRON RESONANCE
C     T(I,J) DISTRIBUTION FUNCTION TO BE CALCULATED
C     F(I) MAXWELLIAN DISTRIBUTION
C     DCY(I,J) QUASILINEAR DIFFUSION COEFFICIENT FOR THE ELECTRON
C     CYCLOTRON WAVE
C     EPS ELECTRIC FIELD NORMALIZED TO THE DREICER FIELD
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER N1,N2,N1M,NITS,IFAIL,MODE
```

```
      PARAMETER(N1=400,N2=200,N1M=N1)
      REAL*8 A(N1M,N2),B(N1M,N2),C(N1M,N2),D(N1M,N2),E(N1M,N2),Q(N1M,N2)
     1,R(N1M,N2),T(N1M,N2),WRKSP1(N1M,N2),WRKSP2(N1M,N2),XP(N1),YMU(N2),
     2DIFF(N1,N2),T1(N1),F(N1),AP1(N1),AP2(N1),AP3(N1),AB1(N1),AB2(N1),
     3AB3(N1),AB4(N1),AB5(N1),AB6(N1),AB7(N1),AB8(N1),AB9(N1),AB10(N1),
     4AB11(N1),AB12(N1),AB13(N1),AB14(N1),SIG(N1),T2(N1),DFDP(N1,N2),
     5DFDM(N1,N2),DE(N1),DF(N1),DFS(N1,N2),ER(N1,N2),DFW(N1,N2),BI(N1),
     6APH(N1),FPH(N1),FMH(N1),AMH(N1),EJ0(N1),EJ2(N1),DIF(N1,N2),TRP(N2)
     7,TRPP(N2)
      REAL*8 AM(N1M,N2),AP(N1M,N2),EM(N1M,N2),EP(N1M,N2),DCY(N1,N2),
     1WRKSP3(N1M,N2),WRKSP4(N1M,N2),ALPHAMAX
C     FULLY RELATIVISTIC COLLISIONAL OPERATOR BRAAMS-KARNEY
      REAL*8 S11ABF
      EXTERNAL S11ABF
c     EXTERNAL D03UAF
         TRAPO=1.
         open(unit=6,status='unknown',file='fpnew.log')^M
C     WRITE(*,*) 'MODE 5 or 9 ?'
C     READ(6,*) MODE
         MODE=9
C     WRITE(*,*) 'APARAM ?'
C     READ(6,*) APARAM
         APARAM=1000.
C     WRITE(*,*) 'NITS ?'
C     READ(6,*) NITS
         NITS=10000
C     WRITE(*,*) 'PMAX ?'
C     READ(6,*) PMAX
         PMAX=30.
C     WRITE(*,*) 'TRUNC E-E COLLISION (0 OR 1) ?'
C     READ(6,*) TRUNC
         TRUNC=1.
      PMIN=0.
      DP=(PMAX-PMIN)/FLOAT(N1-1)
      DMU=2./FLOAT(N2-2-1)
C     WRITE(*,*) 'TE ?'
```

```
C     READ(6,*) TE
      TE=4.0
      BTH=0.04424*DSQRT(TE)
          TI=1.0
      TRAT=TE/TI
C     WRITE(*,*) 'DT ?'                                              70
C     READ(6,*) DTT
          DTT=.1
      DT=DTT
C     WRITE(*,*) 'E ?'
C     READ(6,*) EEPS
          EEPS=0.0
c     EEPS=0.0000
      COLLF=0.0
      Z=1.0
C     RMIN=RADIUS OF THE FLUX SURFACE                                80
      RMIN=5.
C     RMAJ=MAJOR RADIUS OF THE TOKAMAK
      RMAJ=83.
      NPRINT=1
      MPRINT=0
      CC=DSQRT(1840.D0*TE/TI)
      CCC=DSQRT(1840.D0)
C     PARAMETERS FOR THE LOWER HYBRID WAVE CURRENT DRIVE PROBLEM
C     DIFFUSION MATRIX IN DIF(I,J)
C     WRITE(*,*) 'VMIN ?'                                            90
C     READ(6,*) V1
          V1=3.
C     WRITE(*,*) 'VMAX ?'
C     READ(6,*) V2
          V2=6.
C     WRITE(*,*) 'DQL ?'
C     READ(6,*) DDIF
          DDIF=0.5
C     PARAMETERS FOR ELECTRON CYCLOTRON HEATING
      HARMON=1.                                                     100
```

```
      COSTHETA=DSQRT(3.d0)/2.
      THE=DACOS(COSTHETA)
      DTHE=THE*180./3.1416
      WRITE(6,*) DTHE
      DPERP=0.01
C     DELTA IS THE SHAFRANOV SHIFT
      DELTA=0.0
      OMC=1.
      DNRES=0.6
      DNPAR=0.03                                                    110
C     PARAMETERS FOR THE TRAPPING EFFECT
      RRATIO=RMIN/RMAJ
      TRAP=DSQRT(2.*RRATIO/(1.+RRATIO))
      THE=DACOS(TRAP)
      DTHE=THE*180./3.1416
      WRITE(6,*) DTHE,THE
      TRAP2=TRAP*TRAP
          WRITE(6,*) 'TRAP=',TRAP,' TRAP2=',TRAP2
      RATIOB=(1.+DELTA/RMAJ+RRATIO)/(1.+DELTA/RMAJ+RRATIO*COSTHETA)
      OMC=OMC/(1.+DELTA/RMAJ+RRATIO*COSTHETA)                       120
      IF(TRAPO.EQ.0.0D0) RATIOB=1.
          DTRA=DSQRT(1.D0/RATIOB)
          DTRA=DASIN(DTRA)
      WRITE(6,*) RATIOB,DTRA
      WRITE(6,*) OMC
      N1M1=N1-1
      N2M1=N2-1
      DO 1 I=1,N1
    1 XP(I)=DP*FLOAT(I-1)+PMIN
      DO 2 J=2,N2M1                                                 130
    2 YMU(J)=-1.D0+DMU*FLOAT(J-2)
      YMU(1)=YMU(3)
      YMU(N2)=YMU(N2-2)
      CONST=2./DSQRT(3.1416D0)
      DO 46 J=2,N2M1
      ABSM=DABS(YMU(J))
```

```
      YMU2=YMU(J)*YMU(J)
      IF(ABSM.LE.TRAP) THEN
      ARG=YMU2/TRAP2
      ARG1=1.-ARG                                                    140
      ELSE
      ARG=TRAP2/YMU2
      ARG1=1.-ARG
      ENDIF
      EK=1.38629436112+0.09666344259*ARG1+0.03590092383*(ARG1**2)
     1+0.03742563713*(ARG1**3)+0.01451196212*(ARG1**4)+(0.5+
     2.12498593597*ARG1+.06880248576*(ARG1**2)+.03328355346*
     3(ARG1**3)+0.00441787012*(ARG1**4))*DLOG(1.D0/ARG1)
      EE=1.+0.44325141463*ARG1+0.0626060122*(ARG1**2)+
     1.04757383546*(ARG1**3)+.01736506451*(ARG1**4)+(.24998368310   150
     2*ARG1+.09200180037*(ARG1**2)+
     3.04069697526*(ARG1**3)+.00526449639*(ARG1**4))*DLOG(1.D0/ARG1)
      IF(ABSM.LE.TRAP) THEN
      EJ0(J)=EK*DABS(YMU(J)/TRAP)
      EJ2(J)=(EK-EE)*DABS(YMU(J)/TRAP)
      ELSE
      EJ0(J)=EK
      EJ2(J)=(EK-EE)*YMU2/TRAP2
      ENDIF
      EJJ=EJ0(J)-0.5*TRAP2*EJ2(J)                                    160
          TRP(J)=EJJ
          TRPP(J)=EJJ*2.0/3.1416
          WRITE(6,*) J,TRPP(J)
      EJ2(J)=-TRAP2*(EJ2(J)-.5*(2.*(TRAP2+YMU2)*EJ2(J)-YMU2*EJ0(J))/3.)
     1/EJJ
      EJ0(J)=EJJ
C     IF(YMU2.GT.TRAP2) THEN
      IF(ABSM.LE.TRAP) THEN
      EJ0(J)=0.
      ELSE                                                           170
      EJ0(J)=(3.1416/2.)*(1.+RRATIO)/(EJ0(J)*DSQRT(1.D0-RRATIO*RRATIO))
      ENDIF
```

```
C     REMOVE TRAPPING
      IF(TRAPO.EQ.0.0D0) EJ0(J)=1.
      IF(TRAPO.EQ.0.0D0) EJ2(J)=0.
   46 CONTINUE
      EJ0(1)=EJ0(3)
      EJ0(N2)=EJ0(N2-2)
      EJ2(1)=EJ2(3)
      EJ2(N2)=EJ2(N2-2)                                                  180
          TRP(1)=TRP(3)
          TRP(N2)=TRP(N2-2)
          TRPP(1)=TRPP(3)
          TRPP(N2)=TRPP(N2-2)
      DO J=2,N2M1
      APH(J)=(TRP(J+1)-TRP(J-1))/(2.*TRP(J)*DMU)
      ENDDO
      DO J=2,N2M1
      TRP(J)=APH(J)
      IF(TRAPO.EQ.0.0D0) TRP(J)=0.                                       190
      IF(TRAPO.EQ.0.0D0) TRPP(J)=1.
      ENDDO
      TRP(1)=TRP(3)
      TRP(N2)=TRP(N2-2)
      TRPP(1)=TRPP(3)
      TRPP(N2)=TRPP(N2-2)
      DO 3 I=1,N1
      DO 3 J=2,N2M1
      ABSM=DABS(YMU(J))
      YMUSQ=YMU(J)*YMU(J)                                                200
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
      T(I,J)=DEXP(-XP(I)*XP(I)/(1.+GAMA))/(2.*3.1416)**1.5
      F(I)=DEXP(-XP(I)*XP(I)/(1.+GAMA))/(2.*3.1416)**1.5
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.
```

```
      ENDIF
      IF(YMU(J).LT.0.0) CNSTT=-CNSTT                                    210
      PPAR=XP(I)*CNSTT
      PPERP=XP(I)*XP(I)-PPAR*PPAR
      IF(PPAR.EQ.0.0) THEN
      RESN=10.
      DCY(I,J)=0.
      ELSE
      RESN=(GAMA-OMC)/(PPAR*BTH)
      DCY(I,J)=PPERP*DPERP*DEXP(-(RESN-DNRES)**2/DNPAR**2)
      ENDIF
      VPAR=PPAR/GAMA                                                    220
      DIF(I,J)=0.
      IF(VPAR.GE.V1.AND.VPAR.LE.V2) DIF(I,J)=DDIF
    3 CONTINUE
      DO 44 I=1,N1
      T(I,1)=T(I,3)
      T(I,N2)=T(I,N2-2)
          DCY(I,1)=DCY(I,3)
          DCY(I,N2)=DCY(I,N2-2)
      Q(I,N2)=0.
      Q(I,1)=0.                                                         230
      A(I,N2)=0.
      B(I,N2)=0.
      C(I,N2)=0.
      D(I,N2)=0.0
      E(I,N2)=0.
      A(I,1)=0.
      B(I,1)=0.
      C(I,1)=0.
      D(I,1)=0.
c 9 diagonales                                                          240
            AM(I,1)=0.
            AP(I,1)=0.
            EM(I,1)=0.
            EP(I,1)=0.
```

```
 44  E(I,1)=0.
     DO 45 J=1,N2
     Q(N1,J)=0.
     Q(1,J)=0.
     A(N1,J)=0.
     B(N1,J)=0.                                                    250
     C(N1,J)=0.
     D(N1,J)=0.
     E(N1,J)=0.
c 9 diagonales
          AM(N1,J)=0.
          AP(N1,J)=0.
          EM(N1,J)=0.
          EP(N1,J)=0.
     A(1,J)=0.0
     B(1,J)=0.                                                     260
     C(1,J)=0.
     D(1,J)=0.
c 9 diagonales
     AM(1,J)=0.
          AP(1,J)=0.
          EM(1,J)=0.
          EP(1,J)=0.
 45  E(1,J)=0.
     A(1,1)=0.0
     B(1,1)=0.0                                                    270
     C(1,1)=0.0
     D(1,1)=0.0
     E(1,1)=0.0
c 9 diagonales
     AM(1,1)=0.
          AP(1,1)=0.
          EM(1,1)=0.
          EP(1,1)=0.
     IIFAIL=0.
     DO 75 I=1,N1                                                  280
```

```
      XSIG=XP(I)*BTH
      SIG(I)=S11ABF(XSIG,IIFAIL)
   75 CONTINUE
      AP1(1)=0.
      AP2(1)=0.
      AP3(N1)=0.
C     DO 54 I=2,N1
C     GAMA=DSQRT(1.+XP(I)*XP(I)*BTH*BTH)
C     V=XP(I)/GAMA
C     AP1(I)=AP1(I-1)+XP(I)*F(I)*DP*V*4.*3.1416                    290
C     AP2(I)=AP2(I-1)+XP(I)*F(I)*DP*V*4.*3.1416*(1.-
C     1GAMA*SIG(I)/(BTH*XP(I)))
C  54 CONTINUE
      AP1(2)=4.*3.1416*(F(1)+F(2))*(DP**3)/8.
      AP1(3)=4.*3.1416*(F(1)*XP(1)**2/3.+4.*F(2)*XP(2)**2/3.
     1+F(3)*XP(3)**2/3.)*DP
      AP1(4)=4.*3.1416*(3.*F(1)*XP(1)**2/8.+9.*F(2)*XP(2)**2/8.
     1+9.*F(3)*XP(3)**2/8.)*DP
      AP2(2)=4.*3.1416*BTH*BTH*(F(1)+F(2))*(DP**5)/(12.*5.)
      AP2(3)=4.*3.1416*BTH*BTH*(F(1)*XP(1)**4/3.+4.*F(2)*XP(2)**4/3.  300
     1+F(3)*XP(3)**4/3.)*DP/6.
      AP2(4)=4.*3.1416*BTH*BTH*(3.*F(1)*XP(1)**4/8.+9.*F(2)
     1*XP(2)**4/8.+9.*F(3)*XP(3)**4/8.)*DP/6.
      AP1(1)=0.
      AP2(1)=0.
      AP3(N1)=0.
      DO 54 I=3,N1
      GAMA=DSQRT(1.+XP(I)*XP(I)*BTH*BTH)
      GAMA1=DSQRT(1.+XP(I-1)*XP(I-1)*BTH*BTH)
      V=XP(I)/GAMA                                                 310
      V1=XP(I-1)/GAMA1
      AP1(I)=AP1(I-1)+0.5*XP(I)*F(I)*DP*V*4.*3.1416
     1+0.5*XP(I-1)*F(I-1)*DP*V1*4.*3.1416
      AP2(I)=AP2(I-1)+0.5*XP(I)*F(I)*DP*V*4.*3.1416*(1.-
     1GAMA*SIG(I)/(BTH*XP(I)))
     2+0.5*XP(I-1)*F(I-1)*DP*V1*4.*3.1416*(1.-
```

```
     3GAMA1*SIG(I-1)/(BTH*XP(I-1)))
  54 CONTINUE
     DO 58 I=1,N1M1
     K=N1-I                                                              320
     AP3(K)=AP3(K+1)+0.5*XP(K)*F(K)*DP*4.*3.1416
     1+0.5*XP(K+1)*F(K+1)*DP*4.*3.1416
  58 CONTINUE
     AB1(1)=0.
     AB2(1)=0.
     AB3(1)=0.
     AB4(1)=0.
     AB5(1)=0.
     AB6(1)=0.
     AB7(1)=0.                                                           330
     AB1(2)=0.
     AB1(2)=4.*3.1416*(F(1)+F(2))*(DP**3)/8.
     AB2(2)=0.
     AB2(2)=4.*3.1416*(F(1)+F(2))*(DP**5)/(32.)
     AB3(2)=0.
     AB4(1)=0.
     AB5(1)=0.
     AB4(2)=0.
     AB5(2)=0.
     DO 59 I=3,N1                                                        340
     GAMA=DSQRT(1.+XP(I)*XP(I)*BTH*BTH)
     V=XP(I)/GAMA
     GAMA1=DSQRT(1.+XP(I-1)*XP(I-1)*BTH*BTH)
     V1=XP(I-1)/GAMA1
     AB1(I)=AB1(I-1)+0.5*DP*F(I)*XP(I)*XP(I)*4.*3.1416
     1+0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*4.*3.1416
     AB2(I)=AB2(I-1)+0.5*DP*F(I)*XP(I)*XP(I)*XP(I)*XP(I)*4.*3.1416
     1+0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*XP(I-1)*XP(I-1)*4.*3.1416
     AB3(I)=AB3(I-1)+.5*DP*F(I)*XP(I)*XP(I)*4.*3.1416*(-3.*GAMA+SIG(I)
     1*(3./(BTH*XP(I))+2.*BTH*XP(I)))/GAMA                               350
     2+0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*4.*3.1416*(-3.*GAMA1+SIG(I-1)
     3*(3./(BTH*XP(I-1))+2.*BTH*XP(I-1)))/GAMA1
```

```
      AB4(I)=AB4(I-1)+0.5*DP*F(I)*XP(I)*XP(I)*4.*3.1416*(GAMA-SIG(I)
     1/(BTH*XP(I))-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)/GAMA
     2 +0.5*DP*F(I-1)*XP(I-1)*XP(I-1)*4.*3.1416*(GAMA1-SIG(I-1)
     3/(BTH*XP(I-1))-2.*GAMA1*BTH*BTH*XP(I-1)*XP(I-1)/3.)/GAMA1
      AB5(I)=AB5(I-1)+
     1+.5*DP*F(I)*4.*3.1416*XP(I)*XP(I)*(1.-SIG(I)/(GAMA*BTH*XP(I)))
     2+.5*DP*F(I-1)*4.*3.1416*XP(I-1)*XP(I-1)*(1.-SIG(I-1)/(GAMA1*
     3BTH*XP(I-1)))                                                    360
   59 CONTINUE
      AB8(N1)=0.
      AB9(N1)=0.
      AB10(N1)=0.
      AB8(1)=0.
      AB9(1)=0.
      AB10(1)=0.
      N1M2=N1M1-1
      DO 66 I=1,N1M2
      K=N1-I                                                           370
      GAMA=DSQRT(1.+XP(K)*XP(K)*BTH*BTH)
      V=XP(K)/GAMA
      GAMA1=DSQRT(1.+XP(K+1)*XP(K+1)*BTH*BTH)
      V1=XP(K+1)/GAMA1
      AB8(K)=AB8(K+1)+0.5*DP*XP(K)*XP(K)*F(K)*4.*3.1416/V
     1+0.5*DP*XP(K+1)*XP(K+1)*F(K+1)*4.*3.1416/V1
      AB9(K)=AB9(K+1)+0.5*DP*XP(K)*XP(K)*F(K)*4.*3.1416/(V*GAMA*GAMA)
     1 +.5*DP*XP(K+1)*XP(K+1)*F(K+1)*4.*3.1416/(V1*GAMA1*GAMA1)
      AB10(K)=AB10(K+1)+0.5*DP*F(K)*XP(K)*XP(K)*4.*3.1416*V
     1 +0.5*DP*F(K+1)*XP(K+1)*XP(K+1)*4.*3.1416*V1                     380
   66 CONTINUE
      AB8(1)=AB8(2)
      AB9(1)=AB9(2)
      AB10(1)=AB10(2)
      IIFAIL=0
      DO 400 I=2,N1M1
      PPH=XP(I)+DP/2.
      PPH2=PPH*PPH
```

```
PMH=XP(I)-DP/2.
PMH2=PMH*PMH                                                          390
GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
V=XP(I)/GAMA
GAMAPH=DSQRT(1.D0+PPH*PPH*BTH*BTH)
VPH=PPH/GAMAPH
GAMAMH=DSQRT(1.D0+PMH*PMH*BTH*BTH)
VMH=PMH/GAMAMH
XSIG=PPH*BTH
SIGPH=S11ABF(XSIG,IIFAIL)
XSIG=PMH*BTH
SIGMH=S11ABF(XSIG,IIFAIL)                                             400
S=V/DSQRT(2.D0)
SPH=VPH/DSQRT(2.D0)
SMH=VMH/DSQRT(2.D0)
EXPPH=DEXP(-PPH2/(1.+GAMAPH))/(2.*3.1416)**1.5
FPH(I)=(AP1(I)+PPH*EXPPH*VPH*0.5*DP*4.*3.1416)/(VPH**2)+(AP2(I)
1+0.5*DP*EXPPH*PPH*VPH*4.*3.1416*(1.-SIGPH*GAMAPH/(BTH*PPH)))/PPH2
2+(AP3(I)-PPH*EXPPH*(DP/2.)*4.*3.1416)*(1.-SIGPH/(BTH*PPH*GAMAPH))
3/VPH
APH(I)=FPH(I)/VPH+Z/(VPH**3*CC*CC)
IF(I.EQ.2) APH(2)=4.*3.1416*F(2)/3.+Z/(VPH**3*CC*CC)                  410
FPH(I)=FPH(I)+Z*TE/(TI*VPH*VPH*CC*CC)
EXPMH=DEXP(-PMH2/(1.+GAMAMH))/(2.*3.1416)**1.5
FMH(I)=(AP1(I)-PMH*EXPMH*VMH*0.5*DP*4.*3.1416)/(VMH**2)+(AP2(I)
1-0.5*DP*EXPMH*PMH*VMH*4.*3.1416*(1.-SIGMH*GAMAMH/(BTH*PMH)))/PMH2
2+(AP3(I)+PMH*EXPMH*(DP/2.)*4.*3.1416)*(1.-SIGMH/(BTH*PMH*GAMAMH))
3/VMH
AMH(I)=FMH(I)/VMH+Z/(VMH**3*CC*CC)
IF(I.EQ.2) AMH(2)=4.*3.1416*F(2)/3.+Z/(VPH**3*CC*CC)
FMH(I)=FMH(I)+Z*TE/(TI*VMH*VMH*CC*CC)
FU2=1.-1./(2.*S*S)                                                    420
FU2C=1.-1./(2.*S*S*CC*CC)
FU=(S*DEXP(-S*S)+Z*S*DEXP(-S*S*CC*CC)/CC)*CONST
BI(I)=AB1(I)/(2.*V)-AB2(I)/(6.*XP(I)*XP(I)*V)
1+AB3(I)/(BTH*BTH*XP(I)*XP(I)*V*GAMA*GAMA*8.)
```

```
      BI(I)=BI(I)-
     1AB4(I)/(V*4.*BTH*BTH*XP(I)*XP(I))-AB5(I)/(V*4.*GAMA*GAMA)+AB8(I)/2.
      BI(I)=BI(I)+AB9(I)*(-GAMA*GAMA/6.)
      BI(I)=BI(I)+AB9(I)*(-3.*GAMA+SIG(I)*
     1(3./(BTH*XP(I))+2.*BTH*XP(I)))/(8.*GAMA*BTH*BTH*XP(I)*XP(I))
      BI(I)=BI(I)-AB9(I)*GAMA*(GAMA-SIG(I)/(BTH*XP(I))                    430
     1-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)/(4.*BTH*BTH*XP(I)*XP(I))
      BI(I)=BI(I)-AB10(I)*(GAMA-SIG(I)/(BTH*XP(I)))/(4.*GAMA*XP(I)
     1*XP(I))+Z*(1.-1./(CC*CC*V*V))/(2.*V)
  400 CONTINUE
      IIFAIL=0
      DO 701 J=1,N2
      DIF(1,J)=0.
  701 CONTINUE
      DO 702 I=2,N1
      DIF(I,1)=DIF(I,3)                                                   440
      DIF(I,N2)=DIF(I,N2-2)
  702 CONTINUE
      IT=1
C   30 CONTINUE
      DO 707 J=1,N2
      DO 707 I=1,N1
      DIFF(I,J)=DIF(I,J)
  707 CONTINUE
      DO 4 I=2,N1M1
      PPH=XP(I)+DP/2.                                                     450
      PPH2=PPH*PPH
      PMH=XP(I)-DP/2.
      PMH2=PMH*PMH
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
          GAMAB=GAMA*RATIOB
      V=XP(I)/GAMA
      GAMAPH=DSQRT(1.D0+PPH*PPH*BTH*BTH)
          GAMAPHB=GAMAPH*RATIOB
      VPH=PPH/GAMAPH
      GAMAMH=DSQRT(1.D0+PMH*PMH*BTH*BTH)                                  460
```

```
      GAMAMHB=GAMAMH*RATIOB
    VMH=PMH/GAMAMH
    XSIG=PPH*BTH
    SIGPH=S11ABF(XSIG,IIFAIL)
    XSIG=PMH*BTH
    SIGMH=S11ABF(XSIG,IIFAIL)
    S=V/DSQRT(2.D0)
    SPH=VPH/DSQRT(2.D0)
    SMH=VMH/DSQRT(2.D0)
    DO 4 J=2,N2M1                                                    470
    CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
    IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
    CNSTT=DSQRT(CNSTT)
    ELSE
    CNSTT=0.
    ENDIF
    DABMU=CNSTT
    IF(YMU(J).LT.0.0) CNSTT=-CNSTT
    YPH=YMU(J)+DMU/2.
    IF(J.EQ.N2M1) YPH=YMU(N2M1)-DMU/2.                               480
    DYPH=(1.D0-RATIOB*(1.D0-YPH*YPH))
    IF(DYPH.GT.0.0.AND.DYPH.LE.1.) THEN
    DYPH=DSQRT(DYPH)
    ELSE
    DYPH=0.
    ENDIF
    YMH=YMU(J)-DMU/2.
    IF(J.EQ.2) YMH=YMU(2)+DMU/2.
    DYMH=(1.D0-RATIOB*(1.D0-YMH*YMH))
    IF(DYMH.GT.0.0.AND.DYMH.LE.1.) THEN                              490
    DYMH=DSQRT(DYMH)
    ELSE
    DYMH=0.
    ENDIF
c    FBI=1.+EJ2(J)
    fbi=1.+ej2(j)/ymu(j)/ymu(j)
```

```
      EPS=EEPS*EJ0(J)
c      FBI1=1.+0.5*(EJ2(J)+EJ2(J-1))
      fbi1=1.+0.5*(EJ2(J)+EJ2(J-1))/ymh/ymh
      if (j.eq.n2/2+1) fbi1=1.d0                                          500
      EPS1=EEPS*(EJ0(J)+EJ0(J-1))*0.5
c      FBI2=1.+0.5*(EJ2(J)+EJ2(J+1))
      fbi2=1.+0.5*(EJ2(J)+EJ2(J+1))/yph/yph
      if (j.eq.n2/2) fbi2=1.d0
      EPS2=EEPS*(EJ0(J)+EJ0(J+1))*0.5
      IF(DYMH.NE.0.) THEN
      WW=-EPS1*DMU*XP(I)/(BI(I)*FBI1+.5*(DIFF(I,J)+DIFF(I,J-1))*(1.-
     1YMH*YMH)+0.5*(DCY(I,J)+DCY(I,J-1))*(YMH*GAMAB-(GAMAB-OMC)/YMH)**2
C     2/(GAMA*DABS(YMH)*XP(I)))
     2/(GAMAB*DYMH*XP(I)))                                                510
CB     2(GAMA*DABS(YMH)*XP(I)))
CM     2GAMAB)
      ELSE
      WW=-EPS1*DMU*XP(I)/(BI(I)*FBI1+.5*(DIFF(I,J)+DIFF(I,J-1))*
     1(1.-YMH*YMH))
      ENDIF
      IF(WW.NE.0.D0) THEN
      DELJ1=1./WW-1./(DEXP(WW)-1.)
      ELSE
      DELJ1=0.5                                                            520
      END IF
      IF(DYPH.NE.0.) THEN
      WW=-EPS2*DMU*XP(I)/(BI(I)*FBI2+.5*(DIFF(I,J)+DIFF(I,J+1))*(1.-
     1YPH*YPH)+0.5*(DCY(I,J)+DCY(I,J+1))*(YPH*GAMAB-(GAMAB-OMC)/YPH)**2
C     2/(GAMA*DABS(YPH)*XP(I)))
     2/(GAMAB*DYPH*XP(I)))
CB     2/(GAMA*DABS(YPH)*XP(I)))
CM     2GAMAB)
      ELSE
      WW=-EPS2*DMU*XP(I)/(BI(I)*FBI2+.5*(DIFF(I,J)+DIFF(I,J+1))*           530
     1(1.-YPH*YPH))
      ENDIF
```

```
      IF(WW.NE.0.D0) THEN
      DELJ2=1./WW-1./(DEXP(WW)-1.)
      ELSE
      DELJ2=0.5
      END IF
      BF=FPH(I)-EPS*YMU(J)
      IF(DABMU.NE.0) THEN
      CF=APH(I)+0.5*(DIFF(I,J)+DIFF(I+1,J))*YMU(J)*YMU(J)                    540
     1+0.5*(DCY(I,J)+DCY(I+1,J))*GAMAPHB*(1.-YMU(J)*YMU(J))/(PPH*DABMU)
      ELSE
      CF=APH(I)+0.5*(DIFF(I,J)+DIFF(I+1,J))*YMU(J)*YMU(J)
      ENDIF
      WW=DP*BF/CF
      DELP=1./WW-1./(DEXP(WW)-1.)
      BF=FMH(I)-EPS*YMU(J)
      IF(DABMU.NE.0.) THEN
      CF=AMH(I)+0.5*(DIFF(I,J)+DIFF(I-1,J))*YMU(J)*YMU(J)
     1+0.5*(DCY(I,J)+DCY(I-1,J))*GAMAMHB*(1.-YMU(J)*YMU(J))/(PMH*DABMU)     550
      ELSE
      CF=AMH(I)+0.5*(DIFF(I,J)+DIFF(I-1,J))*YMU(J)*YMU(J)
      ENDIF
      WW=DP*BF/CF
      DELM=1./WW-1./(DEXP(WW)-1.)
      CMH=0.5*(DIFF(I,J)+DIFF(I,J-1))*(1.-YMH*YMH)/DMU
      CPH=0.5*(DIFF(I,J+1)+DIFF(I,J))*(1.-YPH*YPH)/DMU
      CDIFF=XP(I)*(DIFF(I,J+1)*YMU(J+1)*(1.-YMU(J+1)*YMU(J+1))
     1-DIFF(I,J-1)*YMU(J-1)*(1.-YMU(J-1)*YMU(J-1)))/(4.*DP*DMU)
      CNTE=(DIFF(I,J)+XP(I)*0.5*(DIFF(I+1,J)-DIFF(I-1,J))/DP)/(2.*DMU)       560
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J-1)*YMU(J-1))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.
      ENDIF
      DABM1=CNSTT
C     DABM1=DABS(YMU(J-1))
```

```
 CNSTT=1.D0-RATIOB*(1.D0-YMU(J+1)*YMU(J+1))
 IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN                                  570
 CNSTT=DSQRT(CNSTT)
 ELSE
 CNSTT=0.
 ENDIF
 DABP1=CNSTT
 DCYPH=(DCY(I+1,J)+DCY(I,J))
 DCYMH=(DCY(I,J)+DCY(I-1,J))
 IF(DABMU.NE.0.) THEN
 TRPI=(1.-YMU(J)*YMU(J))*TRP(J)*(DCY(I,J)*((YMU(J)*YMU(J)
1-1.)*GAMAB+OMC)/(YMU(J)*2.*DP*DABMU)-DIFF(I,J)*XP(I)*YMU(J)/          580
2(2.*DP))
 TRPJ=(1.-YMU(J)*YMU(J))*TRP(J)*(-DCY(I,J)*((YMU(J)*YMU(J)-1.)*
1GAMAB+OMC)**2/(GAMAB*2.*DMU*XP(I)*DABMU*YMU(J)*YMU(J))-DIFF(I,J)
2*(1.-YMU(J)*YMU(J))/(2.*DMU))-(1.-YMU(J)*YMU(J))*TRP(J)*FBI*BI(I)
3 / (2.*DMU)
 CNTCY1=-(OMC+GAMAB*(YMU(J)*YMU(J)-1.))*(1.-YMU(J)*YMU(J))*
1(DCY(I+1,J)-DCY(I-1,J))/(4.*DMU*DP*YMU(J)*DABMU)+BTH*BTH
2*DCY(I,J)*XP(I)*(1.-YMU(J)*YMU(J))**2/(2.*DMU*YMU(J)*GAMAB*DABMU)
 ELSE
 TRPI=(1.-YMU(J)*YMU(J))*TRP(J)*(-DIFF(I,J)*XP(I)*YMU(J)/(2.*DP))      590
 TRPJ=(1.-YMU(J)*YMU(J))*TRP(J)*(-DIFF(I,J)*(1.-
2YMU(J)*YMU(J))/(2.*DMU))-(1.-YMU(J)*YMU(J))*TRP(J)*FBI*BI(I)/
3  (2.*DMU)
 CNTCY1=0.
 ENDIF
 IF(DYMH.NE.0.) THEN
 CNTY2=(1.-YMH*YMH)*0.5*(DCY(I,J-1)+DCY(I,J))*
1(YMH*GAMAB-(GAMAB-OMC)/YMH)**2/(GAMAB*XP(I)*DMU*DMU*DYMH)
 ELSE
 CNTY2=0.                                                               600
 ENDIF
 A(I,J)=-(1.-YMH*YMH)*(FBI1*BI(I)/DMU+EPS1*XP(I)*DELJ1+CMH)/DMU
1+YMU(J)*(1.-YMU(J)*YMU(J))*CNTE+CNTCY1-CNTY2-TRPJ
 CY1=0.
```

```
      IF(DABMU.NE.0.) THEN
      IF(DABP1.NE.0.AND.DABM1.NE.0.)
     1CY1=(1-YMU(J)*YMU(J))*(YMU(J)*GAMAB-(GAMAB-OMC)/YMU(J))*
     2(DCY(I,J+1)-DCY(I,J-1))/(4.*DP*DMU*DABMU)
     3+DCY(I,J)*((1.-YMU(J+1)*YMU(J+1))*(GAMAB*YMU(J+1)-(GAMAB-OMC)/
     4YMU(J+1))/DABP1-(1.-YMU(J-1)*YMU(J-1))*(GAMAB*YMU(J-1)-          610
     5(GAMAB-OMC)/YMU(J-1))/DABM1)/(4.*DP*DMU)
      ELSE
      CY1=0.
      ENDIF
      IF(DABMU.NE.0.) THEN
      CY2=PMH*.5*DCYMH*GAMAMHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)
      ELSE
      CY2=0.
          ENDIF
      B(I,J)=-PMH2*YMU(J)*EPS*DELM/DP-TRPI                              620
     1-(DIFF(I,J)+DIFF(I-1,J))*YMU(J)*YMU(J)
     2*PMH2/(2.*DP*DP)-(AMH(I)/DP-FMH(I)*DELM)*PMH2/DP+CDIFF-CY1-CY2
      CNTCC=+((DIFF(I+1,J)+DIFF(I,J))*PPH2+(DIFF(I,J)+
     1DIFF(I-1,J))*PMH2)*YMU(J)*YMU(J)/(2.*DP*DP)
      CNTC=CNTCC-(1.-YPH*YPH)*(-FBI2*BI(I)/DMU-EPS2*XP(I)*DELJ2-CPH)/DMU
     1+(1.-YMH*YMH)*(FBI1*BI(I)/DMU-EPS1*XP(I)*(1.-DELJ1)+CMH)/DMU
      CYY=0.
      IF(DABMU.NE.0.) THEN
      IF(DYMH.NE.0.AND.DYPH.NE.0.)
     1CYY=PPH*.5*DCYPH*GAMAPHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)        630
     2+PMH*.5*DCYMH*GAMAMHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)
     3+(1.-YMH*YMH)*0.5*(DCY(I,J-1)+DCY(I,J))*
     4(YMH*GAMAB-(GAMAB-OMC)/YMH)**2/(GAMAB*XP(I)*DMU*DMU*DYMH)
     5+(1.-YPH*YPH)*0.5*(DCY(I,J+1)+DCY(I,J))*
     6(YPH*GAMAB-(GAMAB-OMC)/YPH)**2/(GAMAB*XP(I)*DMU*DMU*DYPH)
      ELSE
      CYY=0.
      ENDIF
      C(I,J)=EPS*YMU(J)*(PPH2*DELP-PMH2*(1.-DELM))/DP
     2+CNTC-PPH2*(-APH(I)/DP+FPH(I)*DELP)/DP                           640
```

```
     3+PMH2*(AMH(I)/DP+FMH(I)*(1.-DELM))/DP+CYY+XP(I)*XP(I)/DT
     4+XP(I)*(1.-YMU(J)*YMU(J))*EPS*TRP(J)
      IF(DABMU.NE.0.) THEN
      CY22=PPH*.5*DCYPH*GAMAPHB*(1.-YMU(J)*YMU(J))/(DABMU*DP*DP)
      ELSE
      CY22=0.
      ENDIF
      D(I,J)=PPH2*YMU(J)*(EPS*(1.-DELP)-(DIFF(I+1,J)+DIFF(I,J))*YMU(J)
     1/(2.*DP))/DP-PPH2*(APH(I)/DP+FPH(I)*(1.-DELP))/DP-CDIFF+CY1-CY22
     2+TRPI                                                              650
      IF(DYPH.NE.0.) THEN
      CNTY22=(1.-YPH*YPH)*0.5*(DCY(I,J+1)+DCY(I,J))*
c     1(YPH*GAMA-(GAMA-OMC)/YPH)**2/(GAMA*XP(I)*DMU*DMU*DYPH)
     1(YPH*GAMAB-(GAMAB-OMC)/YPH)**2/(GAMAB*XP(I)*DMU*DMU*DYPH)
      ELSE
      CNTY22=0.
      ENDIF
      E(I,J)=-(1.-YPH*YPH)*(FBI2*BI(I)/DMU-EPS2*XP(I)*(1.-DELJ2)+CPH)
     1/DMU-YMU(J)*(1.-YMU(J)*YMU(J))*CNTE-CNTCY1-CNTY22+TRPJ
c mode 5 diagonales                                                      660
          IF(MODE.EQ.5) THEN
      IF(DABMU.NE.0.) THEN
      Q(I,J)=T(I,J)*XP(I)*XP(I)/DT+.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)
     1*YMU(J))*(T(I+1,J+1)-T(I+1,J-1)-T(I-1,J+1)+T(I-1,J-1))/(DP*DMU)
     2-DCY(I,J)*(1.-YMU(J)*YMU(J))*((YMU(J)*YMU(J)-1.)*GAMAB/YMU(J)
     3+GAMAB*YMU(J)-(GAMAB-OMC)/YMU(J)+OMC/YMU(J))*(T(I+1,J+1)-
     4T(I+1,J-1)-T(I-1,J+1)+T(I-1,J-1))/(4.*DP*DMU*DABMU)
      ELSE
      Q(I,J)=T(I,J)*XP(I)*XP(I)/DT+.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)
     1*YMU(J))*(T(I+1,J+1)-T(I+1,J-1)-T(I-1,J+1)+T(I-1,J-1))/(DP*DMU)     670
      END IF
      END IF
c mode 9 diagonales
      IF (MODE.EQ.9) THEN
          Q(I,J)=T(I,J)*XP(I)*XP(I)/DT
      IF(DABMU.NE.0.) THEN
```

```
      AM(I,J)=-0.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)*YMU(J))/(DP*DMU)
     1+DCY(I,J)*(1.-YMU(J)*YMU(J))*((YMU(J)*YMU(J)-1.)*GAMAB/YMU(J)
     2+GAMAB*YMU(J)-(GAMAB-OMC)/YMU(J)+OMC/YMU(J))/(4.*DP*DMU*DABMU)
      AP(I,J)=-1.0*AM(I,J)                                              680
      EM(I,J)=-1.0*AM(I,J)
      EP(I,J)=AM(I,J)
      ELSE
      AM(I,J)=-0.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)*YMU(J))/(DP*DMU)
      AP(I,J)=-1.0*AM(I,J)
      EM(I,J)=-1.0*AM(I,J)
      EP(I,J)=AM(I,J)
      END IF
      END IF
    4 CONTINUE                                                          690
      DO 444 I=1,N1
      Q(I,1)=Q(I,3)
           Q(I,N2)=Q(I,N2-2)
      A(I,1)=0.
      E(I,N2)=0.
      B(I,1)=B(I,3)
      C(I,1)=C(I,3)
      C(I,N2)=C(I,N2-2)
      D(I,1)=D(I,3)
      D(I,N2)=D(I,N2-2)                                                 700
      A(I,N2)=A(I,N2-2)
      E(I,1)=E(I,3)
c 9 diagonales
      AM(I,1) = AM(I,3)
      AP(I,1) = AP(I,3)
      EM(I,1) = EM(I,3)
      EP(I,1) = EP(I,3)
      AM(I,N2) = AM(I,N2-2)
      AP(I,N2) = AP(I,N2-2)
      EM(I,N2) = EM(I,N2-2)                                             710
      EP(I,N2) = EP(I,N2-2)
  444 CONTINUE
```

```
      DO 445 J=1,N2
c la ligne suivante a ete ajoute pour la condition aux limites au centre
      C(1,J)=1.
      D(1,J)=-F(1)/F(2)
 445  CONTINUE
C     IT=1
C     FOR LHCD ALONE (NO FAST WAVE), IF DIFF(I,J) AND EPS ARE CONSTANT
C     START LOOP HERE                                                    720
 30   CONTINUE
      DINT=0.
      DINT2=0.
      DO 151 I=1,N1
      DINT=0.
      DO 150 J=2,N2M1
      DINT=DINT+T(I,J)*YMU(J)
      DINT2=DINT2+T(I,J)*(3.*YMU(J)*YMU(J)-1.)
 150  CONTINUE
      T2(I)=DINT2*DMU*5./4.                                              730
      T1(I)=DINT*DMU*3./2.
C     IF(T1(I).LT.0.0D0) T1(I)=0.0D0
C     IF(T2(I).LT.0.0D0) T2(I)=0.0D0
 151  CONTINUE
      AB1(1)=0.
      AB2(1)=0.
      AB3(1)=0.
      AB4(1)=0.
      AB5(1)=0.
      AB6(1)=0.                                                          740
      AB7(1)=0.
      AB8(1)=0.
      AB9(1)=0.
      AB10(1)=0.
      AB1(2)=0.
      AB2(2)=0.
      AB3(2)=0.
      AB4(2)=0.
```

```
AB5(2)=0.
AB6(2)=0.                                                              750
AB7(2)=0.
AB8(2)=0.
AB9(2)=0.
AB10(2)=0.
DO 152 I=3,N1
XP2=XP(I)*XP(I)
GAMA=DSQRT(1.+XP2*BTH*BTH)
V=XP(I)/GAMA
GAMA1=DSQRT(1.+XP(I-1)*XP(I-1)*BTH*BTH)
V1=XP(I-1)/GAMA1                                                       760
AB1(I)=AB1(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416
1+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416
AB2(I)=AB2(I-1)+0.5*DP*T1(I)*(XP(I)**5)*4.*3.1416/GAMA
1+0.5*DP*T1(I-1)*(XP(I-1)**5)*4.*3.1416/GAMA1
AB3(I)=AB3(I-1)+0.5*DP*T1(I)*XP(I)*4.*3.1416*(GAMA-SIG(I)
1/(BTH*XP(I)))/GAMA
1+0.5*DP*T1(I-1)*XP(I-1)*4.*3.1416*(GAMA1-SIG(I-1)
1/(BTH*XP(I-1)))/GAMA1
AB4(I)=AB4(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416/GAMA
1+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416/GAMA1                          770
AB5(I)=AB5(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416*(GAMA-SIG(I)
1/(BTH*XP(I))-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)
2/(GAMA*BTH*BTH*XP(I)*XP(I))
3+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416*(GAMA1-SIG(I-1)
4/(BTH*XP(I-1))-2.*GAMA1*BTH*BTH*XP(I-1)*XP(I-1)/3.)
5/(GAMA1*BTH*BTH*XP(I-1)*XP(I-1))
AB6(I)=AB6(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416
1*(-3.*GAMA*SIG(I)/(BTH*BTH*XP(I)*XP(I))+3./(BTH*XP(I))
2+BTH*XP(I))/(6.*GAMA*BTH*XP(I))
3+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416                                780
4*(-3.*GAMA1*SIG(I-1)/(BTH*BTH*XP(I-1)*XP(I-1))+3./(BTH*XP(I-1))
5+BTH*XP(I-1))/(6.*GAMA1*BTH*XP(I-1))
AB8(I)=AB8(I-1)+.5*DP*T1(I)*4.*3.1416*(XP(I)**3)*(-3.*GAMA+SIG(I)
1*(3./(BTH*XP(I))+2.*BTH*XP(I)))/(2.*GAMA*BTH*BTH*XP(I)*XP(I))
```

```
     2+0.5*DP*T1(I-1)*4.*3.1416*(XP(I-1)**3)*(-3.*GAMA1+SIG(I-1)*(3./
     3(BTH*XP(I-1))+2.*BTH*XP(I-1)))/(2.*GAMA1*BTH*BTH*XP(I-1)*XP(I-1))
      AB7(I)=AB7(I-1)+.5*DP*T1(I)*4.*3.1416*(XP(I)**3)*(-3.*GAMA*SIG(I)
     1/(BTH*BTH*XP(I)*XP(I))+3./(BTH*XP(I))+BTH*XP(I)-2.*(BTH*XP(I))
     2**3/5.)/(2.*GAMA*BTH*BTH*BTH*XP(I))
     3+.5*DP*T1(I-1)*4.*3.1416*(XP(I-1)**3)*(-3.*GAMA1*SIG(I-1)                  790
     4/(BTH*BTH*XP(I-1)*XP(I-1))+3./(BTH*XP(I-1))+BTH*XP(I-1)-2.*(BTH*
     5XP(I-1))**3/5.)/(2.*GAMA1*BTH*BTH*BTH*XP(I-1))
      AB9(I)=AB9(I-1)+0.5*DP*T1(I)*XP(I)*4.*3.1416*(GAMA*SIG(I)
     1/(BTH*XP(I))-1.)/GAMA
     2+0.5*DP*T1(I-1)*XP(I-1)*4.*3.1416*(GAMA1*SIG(I-1)
     3/(BTH*XP(I-1))-1.)/GAMA1
      AB10(I)=AB10(I-1)+0.5*DP*T1(I)*(XP(I)**3)*4.*3.1416*(GAMA*SIG(I)
     1*(15./(BTH*XP(I))**2+6.)-15./(BTH*XP(I))-11.*BTH*XP(I))/
     2(12.*GAMA*BTH*BTH*BTH*XP(I))
     3+0.5*DP*T1(I-1)*(XP(I-1)**3)*4.*3.1416*(GAMA1*SIG(I-1)                     800
     4*(15./(BTH*XP(I-1))**2+6.)-15./(BTH*XP(I-1))-11.*BTH*XP(I-1))/
     5(12.*GAMA1*BTH*BTH*BTH*XP(I-1))
152  CONTINUE
      AB11(N1)=0.
      AB12(N1)=0.
      AB13(N1)=0.
      AB14(N1)=0.
      DO 155 I=1,N1M1
      K=N1-I
      XP2=XP(K)*XP(K)                                                            810
      GAMA=DSQRT(1.+XP2*BTH*BTH)
      GAMA1=DSQRT(1.+XP(K+1)*XP(K+1)*BTH*BTH)
      AB11(K)=AB11(K+1)+0.5*DP*T1(K)*4.*3.1416/GAMA
     1+0.5*DP*T1(K+1)*4.*3.1416/GAMA1
      AB12(K)=AB12(K+1)+0.5*DP*T1(K)*4.*3.1416
     1+0.5*DP*T1(K+1)*4.*3.1416
      AB13(K)=AB13(K+1)+0.5*DP*T1(K)*(XP(K)**2)*4.*3.1416
     1+0.5*DP*T1(K+1)*(XP(K+1)**2)*4.*3.1416
      AB14(K)=AB14(K+1)+0.5*DP*T1(K)*(XP(K)**2)*4.*3.1416/GAMA
     1+.5*DP*T1(K+1)*(XP(K+1)**2)*4.*3.1416/GAMA1                               820
```

```
155 CONTINUE
    DO 502 I=2,N1M1
    XP2=XP(I)*XP(I)
    GAMA=DSQRT(1.+XP2*BTH*BTH)
    GAMAB=GAMA*RATIOB
    DO 502 J=2,N2M1
    CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
    IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
    CNSTT=DSQRT(CNSTT)
    ELSE                                                          830
    CNSTT=0.
    ENDIF
    DABMU=CNSTT
c mode 5 diagonales
    IF (MODE.EQ.5) THEN
    IF(DABMU.NE.0.) THEN
    Q(I,J)=T(I,J)*XP(I)*XP(I)/DT+.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)
   1*YMU(J))*(T(I+1,J+1)-T(I+1,J-1)-T(I-1,J+1)+T(I-1,J-1))/(DP*DMU)
   2-DCY(I,J)*(1.-YMU(J)*YMU(J))*((YMU(J)*YMU(J)-1.)*GAMAB/YMU(J)
   3+GAMAB*YMU(J)-(GAMAB-OMC)/YMU(J)+OMC/YMU(J))*(T(I+1,J+1)-      840
   4-T(I+1,J-1)-T(I-1,J+1)+T(I-1,J-1))/(4.*DP*DMU*DABMU)
    ELSE
    Q(I,J)=T(I,J)*XP(I)*XP(I)/DT+.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)*
   1YMU(J))*(T(I+1,J+1)-T(I+1,J-1)-T(I-1,J+1)+T(I-1,J-1))/(DP*DMU)
    END IF
    END IF
c mode 9 diagonales
    IF (MODE.EQ.9) THEN
        Q(I,J)=T(I,J)*XP(I)*XP(I)/DT
    IF(DABMU.NE.0.) THEN                                          850
    AM(I,J)=-0.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)*YMU(J))/(DP*DMU)
   1+DCY(I,J)*(1.-YMU(J)*YMU(J))*((YMU(J)*YMU(J)-1.)*GAMAB/YMU(J)
   2+GAMAB*YMU(J)-(GAMAB-OMC)/YMU(J)+OMC/YMU(J))/(4.*DP*DMU*DABMU)
    AP(I,J)=-1.0*AM(I,J)
    EM(I,J)=-1.0*AM(I,J)
    EP(I,J)=AM(I,J)
```

```
      ELSE
      AM(I,J)=-0.5*DIFF(I,J)*XP(I)*YMU(J)*(1.-YMU(J)*YMU(J))/(DP*DMU)
      AP(I,J)=-1.0*AM(I,J)
      EM(I,J)=-1.0*AM(I,J)                                                    860
      EP(I,J)=AM(I,J)
      END IF
      ENDIF
  502 CONTINUE
      DO 503 I=2,N1M1
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
      V=XP(I)/GAMA
           CNT=-2.*AB1(I)*GAMA/3.+AB2(I)*GAMA/5.+AB3(I)+AB4(I)/3.
     1+(-AB5(I)+AB7(I)+AB8(I)-AB10(I))*GAMA
      CNT=CNT+AB6(I)*(GAMA*XP(I)*XP(I)-5.)+AB9(I)*XP(I)*XP(I)                 870
      CNT=CNT/(GAMA*XP(I)*XP(I))
      CNT1=+AB11(I)*(1./3.+(GAMA-SIG(I)/(BTH*XP(I)))/(XP(I)*XP(I))
     1-5.*(-3.*GAMA*SIG(I)
     2/(BTH*BTH*XP(I)*XP(I))+3./(BTH*XP(I))+BTH*XP(I))/(6.*BTH*XP(I)))
      CNT1=CNT1+AB12(I)*(-2.*GAMA/3.+XP(I)*XP(I)/5.-(GAMA-SIG(I)
     1/(BTH*XP(I))-2.*GAMA*BTH*BTH*XP(I)*XP(I)/3.)/(BTH*BTH*XP(I)
     2*XP(I)))
      CNT1=CNT1+AB12(I)*(-3.*GAMA*SIG(I)/(BTH*BTH*XP(I)*XP(I))+3.
     1/(BTH*XP(I))+BTH*XP(I)-2.*(BTH*XP(I))**3/5.)/(2.*BTH*BTH*BTH
     2*XP(I))                                                                 880
      CNT1=CNT1+AB12(I)*(-3.*GAMA+SIG(I)*(3./(BTH*XP(I))+2.*BTH*XP(I)))/
     1(2.*BTH*BTH*XP(I)*XP(I))
      CNT1=CNT1-AB12(I)*(GAMA*SIG(I)*(15./(BTH*BTH*XP(I)*XP(I))+6.)
     1-15./(BTH*XP(I))-11.*BTH*XP(I))/(12.*BTH*XP(I)*BTH*BTH)
      CNT1=CNT1+AB13(I)*(-3.*GAMA*SIG(I)/(BTH*BTH*XP(I)*XP(I))+3.
     1/(BTH*XP(I))+BTH*XP(I))/(6.*BTH*XP(I))
      CNT1=CNT1+AB14(I)*(GAMA*SIG(I)/(BTH*XP(I))-1.)/(XP(I)*XP(I))
      CNT1=CNT1*XP(I)/GAMA
      CNT=(CNT*F(I)+CNT1*F(I))+4.*3.1416*F(I)*T1(I)/GAMA
      CNT=CNT*XP(I)*XP(I)                                                     890
      DO 503 J=2,N2M1
      Q(I,J)=Q(I,J)+EJ0(J)*CNT*YMU(J)*TRUNC
```

```
503  CONTINUE
     DO 513 I=1,N1
     Q(I,1)=Q(I,3)
     Q(I,N2)=Q(I,N2-2)
c 9 diagonales
     AM(I,1) = AM(I,3)
     AP(I,1) = AP(I,3)
     EM(I,1) = EM(I,3)                                                    900
     EP(I,1) = EP(I,3)
     AM(I,N2) = AM(I,N2-2)
     AP(I,N2) = AP(I,N2-2)
     EM(I,N2) = EM(I,N2-2)
     EP(I,N2) = EP(I,N2-2)
513  CONTINUE
     DO 80 J=2,N2M1
     DO 60 I=2,N1M1
     IF (MODE.EQ.5) THEN
c mode 5 diagonales                                                       910
     IF(C(I,J).NE.0.D0) THEN
     R(I,J)=Q(I,J)-A(I,J)*T(I,J-1)-B(I,J)*T(I-1,J)-C(I,J)*T(I,J)-
    1D(I,J)*T(I+1,J)-E(I,J)*T(I,J+1)
    2-COLLF*XP(I)*XP(I)*T(I,J)
     ELSE
     R(I,J)=Q(I,J)-T(I,J)*XP(I)*XP(I)/DT
     END IF
     END IF
     IF (MODE.EQ.9) THEN
c mode 9 diagonales                                                       920
     IF(C(I,J).NE.0.D0) THEN
     R(I,J)=Q(I,J)-A(I,J)*T(I,J-1)-B(I,J)*T(I-1,J)-C(I,J)*T(I,J)-
    1D(I,J)*T(I+1,J)-E(I,J)*T(I,J+1)
    2-COLLF*XP(I)*XP(I)*T(I,J)
    3-AM(I,J)*T(I-1,J-1)-AP(I,J)*T(I+1,J-1)-EM(I,J)*T(I-1,J+1)
    4-EP(I,J)*T(I+1,J+1)
     ELSE
     R(I,J)=Q(I,J)-T(I,J)*XP(I)*XP(I)/DT
```

```
      END IF
      END IF                                                          930
   60 CONTINUE
   80 CONTINUE
      DO 81 I=1,N1M1
      IF(C(I,N2).NE.0.0D0) THEN
      R(I,1)=R(I,3)
      R(I,N2)=R(I,N2-2)
      ELSE
      R(I,N2)=Q(I,N2)-T(I,N2)*XP(I)*XP(I)/DT
      END IF
   81 CONTINUE                                                        940
   55 FORMAT(2I10,2F16.6)
      DO 500 J=1,N2
      Q(N1,J)=T(N1,J)*XP(N1)*XP(N1)/DT
      Q(1,J)=0.
      R(1,J)=0.
      R(N1,J)=0.
  500 CONTINUE
      IFAIL=0
c     CALL D03UAF(N1,N2,N1M,A,B,C,D,E,APARAM,IT,R,WRKSP1,WRKSP2,IFAIL)
            IF (MODE.EQ.5) THEN                                       950
            ALPHAMAX = 1.- 2.*APARAM/(N1M1*N1M1+N2M1*N2M1)
c           DO 446 J=1,N2
c      DO 446 I=1,N1
c           write(*,*) I,J,A(I,J),B(I,J),C(I,J),D(I,J),E(I,J)
c  446 CONTINUE
      call sip5d(N1,N2,A,B,C,D,E,R,IT,ALPHAMAX,WRKSP1,WRKSP2)
      END IF
      IF (MODE.EQ.9) THEN
            ALPHAMAX = 1.- 2.*APARAM/(N1M1*N1M1+N2M1*N2M1)
      call sip9d(N1,N2,AM,A,AP,B,C,D,EM,E,EP,R,IT,ALPHAMAX,WRKSP1,WRKSP2  960
     1,WRKSP3,WRKSP4)
      ENDIF
      MPRINT=MPRINT+1
      IT=IT+1
```

```
C     IF YOU WANT DT TO REMAIN CONSTANT AT EACH ITERATION, COMMENT THE
C     FOLLOWING CARD
C     DT=DTT+(IT-1)*DTT/NITS
      DO 6 J=1,N2
      DO 6 I=1,N1
      T(I,J)=T(I,J)+R(I,J)                                              970
    6 CONTINUE
      DO 501 I=1,N1
      T(I,1)=T(I,3)
  501 T(I,N2)=T(I,N2-2)
      IF(TRAPO.EQ.0.0D0) GO TO 1700
      N2H=N2/2
      DO I=1,N1
      DO J=1,N2H
      ABSM=DABS(YMU(J))
      CNST=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))                             980
      IF(CNST.GT.0.0.AND.CNST.LE.1.) THEN
      CNST=DSQRT(CNST)
      ELSE
      CNST=0.
      ENDIF
      IF(ABSM.LE.TRAP) T(I,N2+1-J)=(T(I,N2+1-J)+T(I,J))/2.
      IF(ABSM.LE.TRAP) T(I,J)=T(I,N2+1-J)
CC    IF(CNST.EQ.0.0) T(I,J)=T(I,N2+1-J)
      ENDDO
      ENDDO                                                             990
          GO TO 1700
      DO J=2,N2H
      ABSM=DABS(YMU(J))
      ABSM1=DABS(YMU(J+1))
      CNST=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNST.GT.0.0.AND.CNST.LE.1.) THEN
      CNST=DSQRT(CNST)
      ELSE
      CNST=0.
      ENDIF                                                             1000
```

```
      CNST1=1.D0-RATIOB*(1.D0-YMU(J+1)*YMU(J+1))
      IF(CNST1.GT.0.0.AND.CNST1.LE.1.) THEN
      CNST1=DSQRT(CNST1)
      ELSE
      CNST1=0.
      ENDIF
      DO I=1,N1M1
      IF(ABSM1.LE.TRAP.AND.ABSM.GT.TRAP) T(I,J+1)=(T(I,J+1)+T(I,J))/2.
      IF(ABSM1.LE.TRAP.AND.ABSM.GT.TRAP) T(I,J)=(T(I,J+1)+T(I,J))/2.
      IF(ABSM1.LE.TRAP.AND.ABSM.GT.TRAP) T(I,J-1)=(T(I,J-1)+T(I,J))/2.   1010
      ENDDO
      ENDDO
 1700 CONTINUE
      IF(IT.EQ.NITS) GO TO 101
      IF(MPRINT.LT.NPRINT) GO TO 100
      MPRINT=0.
   51 FORMAT(5X,4F16.8,I10)
  101 DENS=0.0
      CUR=0.
      CURR=0.                                                            1020
      CURRR=0.
      DO 5 J=2,N2M1
      DO 5 I=2,N1M1
      ABSM=DABS(YMU(J))
      XPH=XP(I)-DP/2.
      GAMAH=DSQRT(1.D0+XPH*XPH*BTH*BTH)
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN
      CNSTT=DSQRT(CNSTT)                                                 1030
      ELSE
      CNSTT=0.
      ENDIF
C     IF(YMU(J).LT.0.0.AND.ABSM.GT.TRAP) CNSTT=-CNSTT
      IF(YMU(J).LT.0.0) CNSTT=-CNSTT
      PPAR=XP(I)*CNSTT
```

```
      VPAR=PPAR/GAMA
           IF(TRAPO.EQ.0.) THEN
      CUR=CUR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           ELSE                                                    1040
C     CUR=CUR+2.*T(I,J)*PPAR*XP(I)*XP(I)/GAMA
C     1+2.*(T(I,J)+T(I-1,J))*PPARH*XPH*XPH/GAMAH
CC       IF(CNSTT.LT.0.0) CURRR=CURRR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           IF(ABSM.GE.TRAP.AND.YMU(J).LT.0.)
     1CURRR=CURRR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
CC       IF(CNSTT.GT.0.0) CUR=CUR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           IF(ABSM.GE.TRAP.AND.YMU(J).GT.0.)
     1 CUR=CUR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
CC       IF(CNSTT.EQ.0.0) CURR=CURR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           IF(ABSM.LT.TRAP)                                        1050
     1 CURR=CURR+T(I,J)*PPAR*XP(I)*XP(I)/GAMA
           ENDIF
      DENS=DENS+2.*T(I,J)*XP(I)*XP(I)+2.*(T(I,J)+T(I-1,J))*XPH*XPH
    5 CONTINUE
C     CALCULATE DERIVATIVE OF THE DISTRIBUTION FUNCTION USING CUBIC SPLINE
      DO 506 J=1,N2
      DE(1)=0.
      DF(1)=0.
      DO 507 I=2,N1M1
      DFI=3.*(T(I+1,J)-T(I-1,J))/DP                                1060
      DE(I)=-1./(DE(I-1)+4.)
  507 DF(I)=(DF(I-1)-DFI)*DE(I)
      DFDP(N1,J)=0.0
      DO 508 I=1,N1M1
      K=N1-I
  508 DFDP(K,J)=DE(K)*DFDP(K+1,J)+DF(K)
  506 CONTINUE
      DO 526 I=1,N1
      DE(1)=0.
      DF(1)=0.0                                                    1070
      DO 527 J=2,N2M1
      DFI=3.*(T(I,J+1)-T(I,J-1))/DMU
```

```
      DE(J)=-1./(DE(J-1)+4.)
  527 DF(J)=(DF(J-1)-DFI)*DE(J)
      DFDM(I,N2)=0.0
      DO 528 J=1,N2M1
      K=N2-J
  528 DFDM(I,K)=DE(K)*DFDM(I,K+1)+DF(K)
  526 CONTINUE
      PABS1=0.                                                          1080
          PABSR=0.
          PABSRR=0.
          PABSRRR=0.
      DO 505 J=2,N2M1
      DO 505 I=2,N1M1
      ABSM=DABS(YMU(J))
      GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)
      GAMAB=GAMA*RATIOB
      CNSTT=1.D0-RATIOB*(1.D0-YMU(J)*YMU(J))
      IF(CNSTT.GT.0.0.AND.CNSTT.LE.1.) THEN                             1090
      CNSTT=DSQRT(CNSTT)
      ELSE
      CNSTT=0.
      ENDIF
      IF(YMU(J).LT.0.0) CNSTT=-CNSTT
      PPAR=XP(I)*CNSTT
      PPERP=XP(I)*XP(I)-PPAR*PPAR
      VPAR=PPAR/GAMA
      IF(PPAR.EQ.0.) GO TO 3000
          IF(TRAPO.EQ.0.) THEN                                          1100
      PABSR=PABSR+(XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+
     2(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))/GAMA
     3+(XP(I)**3)*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB
     4*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
      ELSE
CC        IF(CNSTT.LT.0.0)
          IF(ABSM.GE.TRAP.AND.YMU(J).LT.0.)
     1PABSRRR=PABSRRR+(XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+
```

```
     1(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))*TRPP(J)/GAMA+TRPP(J)
     1*(XP(I)**3)*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB  1110
     1*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
CC      IF(CNSTT.GT.0.0)
          IF(ABSM.GE.TRAP.AND.YMU(J).GT.0.)
     1PABSR=PABSR+(XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+
     2(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))*TRPP(J)/GAMA+TRPP(J)
     3*(XP(I)**3)*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB
     4*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
CC    IF(CNSTT.EQ.0.0)
          IF(ABSM.LT.TRAP)
     1PABSRR=PABSRR+(XP(I)**3)*DIFF(I,J)*YMU(J)*(YMU(J)*DFDP(I,J)+          1120
     2(1.-YMU(J)*YMU(J))*DFDM(I,J)/XP(I))*TRPP(J)/GAMA+TRPP(J)
     3*(XP(I)**3)*DCY(I,J)*(1.-YMU(J)*YMU(J))*(GAMAB*DFDP(I,J)-(OMC-GAMAB
     4*(1.-YMU(J)*YMU(J)))*DFDM(I,J)/(XP(I)*YMU(J)))/(GAMA*DABS(PPAR))
          ENDIF
 3000 CONTINUE
  505  CONTINUE
C     IF YOU WANT TO WRITE THE SOLUTION T(I,J) AT THE DIFFERENT COORDINATE
C     POINTS
C     DO 515 I=1,N1
C     DO 515 J=1,N2                                                        1130
C     WRITE(6,51) XP(I),YMU(J),T(I,J)
C 515 CONTINUE
C     THE FOLLOWING LOOP CALCULATE AND WRITE THE RUNAWAY RATE AT DIFFERENT
C     RADII RAD
C     DO 517 I=40,N1,5
C     EPS=EEPS*EJ0(J)
C     RUN1=0.
C     RUN=0.
C     RAD=XP(I)
C     GAMA=DSQRT(1.D0+XP(I)*XP(I)*BTH*BTH)                                 1140
C     GAMAPH=GAMA
C     V=XP(I)/GAMA
C     VPH=V
C     PPH=XP(I)
```

```
C     XSIG=PPH*BTH
C     SIGPH=S11ABF(XSIG,IIFAIL)
C     PPH2=PPH*PPH
C     SPH=V/DSQRT(2.D0)
C     EXPPH=DEXP(-PPH2/(1.+GAMAPH))/(2.*3.1416)**1.5
C     FPHI=AP1(I)/(VPH**2)+AP2(I)/PPH2+                                  1150
C    1AP3(I)*(1.-SIGPH/(BTH*PPH*GAMAPH))/VPH
C     APHI=FPHI/VPH+Z/(VPH**3*CC*CC)
C     FPHI=FPHI+Z*TE/(TI*VPH*VPH*CC*CC)
C     DO 516 J=2,N2M1
C     RUN=RUN+(EPS*YMU(J)*T(I,J)-
C    1APHI*(T(I+1,J)-T(I-1,J))/(2.*DP)-FPHI*T(I,J))*DMU
C     RUN1=RUN1+(EPS*YMU(J)*T(I,J)-
C    1APHI*DFDP(I,J)-FPHI*T(I,J))*DMU
C 516 CONTINUE
C     RUN=-RUN*2.*3.1416*RAD*RAD                                         1160
C     RUN1=-RUN1*2.*3.1416*RAD*RAD
CC     WRITE(6,518) RAD,RUN,RUN1
C 517 CONTINUE
 518  FORMAT(5X,F12.5,2G12.5)
      DENS=DENS*2.*3.1416*DP*DMU/6.
C      CUR=CUR*2.*3.1416*DP*DMU/6.
      CUR=CUR*2.*3.1416*DP*DMU
      CURR=CURR*2.*3.1416*DP*DMU
      CURRR=CURRR*2.*3.1416*DP*DMU
      TCURR=CUR+CURR+CURRR                                               1170
      PABSR=-PABSR*2.*3.1416*DP*DMU
      PABSRR=-PABSRR*2.*3.1416*DP*DMU
      PABSRRR=-PABSRRR*2.*3.1416*DP*DMU
      PABS1=PABSR+PABSRR+PABSRRR
      IF(PABS1.EQ.0.) GO TO 5181
      RJP1=TCURR/PABS1
 5181 CONTINUE
C     TO RENORMALIZE THE SOLUTION TO KEEP DENSITY STRICTLY 1
      DO 300 J=1,N2
      DO 300 I=1,N1                                                      1180
```

```
 300 T(I,J)=T(I,J)/DENS
C     RENORMALIZED VALUES
      CUR=CUR/DENS
      CURR=CURR/DENS
      CURRR=CURRR/DENS
      TCURR=TCURR/DENS
      PABSR=PABSR/DENS
      PABSRR=PABSRR/DENS
      PABSRRR=PABSRRR/DENS
      PABS1=PABS1/DENS                                              1190
      DENS=DENS/DENS
 100 CONTINUE
      WRITE(6,*) IT,DENS,TCURR,PABS1,RJP1
cs    WRITE(6,*) CUR,CURR,CURRR
  50  FORMAT(I4,2X,4F15.6)
      IF(IT.LT.NITS) GO TO 30
  56  FORMAT(5X,6F14.8)
C     WRITE THE DISTRIBUTION FUNCTION IN A FILE
cs    OPEN(UNIT=11,FILE='distr.dat',FORM='UNFORMATTED')
C         OPEN(UNIT=11,FILE='distr.dat',STATUS='UNKNOWN')           1200
cs    WRITE(11) ((T(I,J),I=1,N1),J=2,N2M1)
cs    CLOSE(11)
 5000 CONTINUE
      STOP
      END



      SUBROUTINE sip5d(m,n,a1,a2,cc,a3,a4,r,kt,alphamax,w1,w2)
                                                                    1210
            integer m,n,kt,iacc
            integer k,itc,itc0,itc1
            integer i,ii,j,jj,ij,itj,ijb,ibj,ib,it,jb,n1,m1
      double precision a1(m,n),a2(m,n),a3(m,n),a4(m,n),cc(m,n),r(m,n)
            double precision alpha,alphamax,alp(9),w1(m,n),w2(m,n)
      double precision eijb,eibj,fijb,fibj,rijb,ribj,sb,sc,sd
```

```
      double precision sbeijb,scfibj
c
      data alp(1),alp(2),alp(3),alp(4),alp(5),alp(6),alp(7),alp(8)
     &,alp(9)/1.0D0,0.625D0,0.25D0,0.875D0,0.5D0,0.125D0,0.75D0,
     &0.375D0,0.0D0/
c
            n1 = n + 1
            m1 = m + 1
c
            itc = mod(kt-1,2) + 1
            if (itc.eq.0)       itc = 2

            itc0 = 2 - itc
            itc1 = itc - 1

c Acceleration parameter

            iacc = mod(kt-1,18)
            if (iacc.lt.0) iacc = iacc + 18

            iacc = iacc/2 + 1
            alpha = 1.0 - (1.0 - alphamax)**alp(iacc)

c Approximate factorisation and inversion of the lower triangular matrix

            do 10 jj=1,n

                  j = itc0*jj + itc1*(n1-jj)
                  jb = j - itc0 + itc1

                  do 10 i=1,m

                        ib = i - 1
```

```
                if (cc(i,j).eq.0.0) goto 100

                if ((jb.eq.0).or.(jb.eq.n1)) then
                        eijb = 0.0D0
                        fijb = 0.0D0
                        rijb = 0.0D0
                else
                        eijb = w1(i,jb)                                    1260
                        fijb = w2(i,jb)
                        rijb = r(i,jb)
                endif

                if (ib.eq.0) then
                        eibj = 0.0D0
                        fibj = 0.0D0
                        ribj = 0.0D0
                else
                        eibj = w1(ib,j)                                    1270
                        fibj = w2(ib,j)
                        ribj = r(ib,j)
                endif

                sb = (itc0*a1(i,j) + itc1*a4(i,j))/(1 + alpha*eijb)
                sc = a2(i,j)/(1 + alpha*fibj)

                sbeijb = sb*eijb
                scfibj = sc*fibj
                                                                           1280
c               sd = 1.0D0 - sb*fijb - sc*eibj + alpha*(sbeijb + scfibj)
      sd = cc(i,j) - sb*fijb - sc*eibj + alpha*(sbeijb + scfibj)

                w1(i,j) = (a3(i,j) - alpha*sbeijb)/sd
                w2(i,j) = (itc0*a4(i,j) + itc1*a1(i,j) - alpha*scfibj)/sd


c Inversion of the lower triangular matrix
```

```
                r(i,j) = (r(i,j) - sb*rijb - sc*ribj)/sd                    1290

                goto 10

100     w1(i,j) = 0.0
        w2(i,j) = 0.0

 10 continue

c Inversion of the upper triangular matrix

                                                                            1300
        do 20 jj=1,n

            j = itc0*(n1-jj) + itc1*jj
            jb = j + itc0 - itc1

            do 20 ii=1,m

                i = m1 - ii
                it = i + 1

                                                                            1310
                if ((jb.ne.0).and.(jb.ne.n1))           then
                   r(i,j) = r(i,j) - w2(i,j)*r(i,jb)
                endif

                if (it.ne.m1) then
                        r(i,j) = r(i,j) - w1(i,j)*r(it,j)
              endif

 20 continue

                                                                            1320
   return
   end
```

```
      SUBROUTINE sip9d(m,n,a1,a2,a3,a4,cc,a5,a6,a7,a8,r,kt,alphamax,w1
     &,w2,w3,w4)
            integer n,m,kt,iacc
            integer itc,itc00,itc01,itc10,itc11
            integer i,ii,j,jj,ij
            integer it,itj,ijb,ibj,ib,jb                                  1330
            integer ibjb,itjb
            integer z1,z2,z3,z4
            double precision a1(m,n),a2(m,n),a3(m,n),a4(m,n),a5(m,n),a6(m,n)
            double precision a7(m,n),a8(m,n),cc(m,n),r(m,n),w1(m,n)
            double precision alpha,alphamax,alp(9),w2(m,n),w3(m,n),w4(m,n)
            double precision fibjb,fijb,fitjb,gibjb,gijb,gitjb,hibjb
            double precision hijb,hitjb,iibjb,iijb,iitjb,ribjb,rijb,ritjb
            double precision fibj,gibj,hibj,iibj,ribj
            double precision sa,sb,sc,sd,se
c                                                                         1340
      data alp(1),alp(2),alp(3),alp(4),alp(5),alp(6),alp(7),alp(8)
     &,alp(9)/1.0D0,0.625D0,0.25D0,0.875D0,0.5D0,0.125D0,0.75D0,
     &0.375D0,0.0D0/
c
            n1 = n + 1
            m1 = m + 1

c     Alternate direction technique
                                                                          1350
            itc = mod(kt-1,4) + 1
            if (itc.eq.0) itc = 4

            if (itc.eq.1) then
                  itc00 = 1
                  itc01 = 0
                  itc10 = 1
                  itc11 = 0
            endif
                                                                          1360
```

```
      if (itc.eq.2) then
            itc00 = 0
            itc01 = 1
            itc10 = 1
            itc11 = 0
      endif

      if (itc.eq.3) then
            itc00 = 1
            itc01 = 0                                                    1370
            itc10 = 0
            itc11 = 1
      endif

      if (itc.eq.4) then
            itc00 = 0
            itc01 = 1
            itc10 = 0
            itc11 = 1
      endif                                                              1380

      z1 = itc00*itc10
      z2 = itc01*itc10
      z3 = itc00*itc11
      z4 = itc01*itc11

c Acceleration parameter

      iacc = mod(kt-1,36)
      if (iacc.lt.0) iacc = iacc +36                                     1390

      iacc = iacc/4 + 1
      alpha = 1.0 - (1.0 - alphamax)**alp(iacc)

c Approximate factorisation and inversion of the lower triangular matrix
```

```
      do 10 jj=1,n

         j = itc00*jj + itc01*(n1-jj)
         jb = j - itc00 + itc01                                    1400

         do 10 ii=1,m

            i = itc10*ii + itc11*(m1-ii)
            ib = i - itc10 + itc11
            it = i + itc10 - itc11

            if (cc(i,j).eq.0.0) goto 100

            if ((jb.eq.0).or.(jb.eq.n1)) then                      1410
               fibjb = 0.0
               fijb = 0.0
               fitjb = 0.0
               gibjb = 0.0
               gijb = 0.0
               gitjb = 0.0
               hibjb = 0.0
               hijb = 0.0
               hitjb = 0.0
               iibjb = 0.0                                         1420
               iijb = 0.0
               iitjb = 0.0
               ribjb = 0.0
               rijb = 0.0
               ritjb = 0.0
            else
               if ((ib.eq.0).or.(ib.eq.m1)) then
                  fibjb = 0.0
                  gibjb = 0.0
                  hibjb = 0.0                                      1430
                  iibjb = 0.0
                  ribjb = 0.0
```

```
      else
            fibjb = w1(ib,jb)
            gibjb = w2(ib,jb)
            hibjb = w3(ib,jb)
            iibjb = w4(ib,jb)
            ribjb = r(ib,jb)
      endif
                                                                  1440
      if ((it.eq.0).or.(it.eq.m1)) then
            fitjb = 0.0
            gitjb = 0.0
            hitjb = 0.0
            iitjb = 0.0
            ritjb = 0.0
      else
            fitjb = w1(it,jb)
            gitjb = w2(it,jb)
            hitjb = w3(it,jb)                                     1450
            iitjb = w4(it,jb)
            ritjb = r(it,jb)
      endif

      fijb = w1(i,jb)
      gijb = w2(i,jb)
      hijb = w3(i,jb)
      iijb = w4(i,jb)
      rijb = r(i,jb)
endif                                                             1460

if ((ib.eq.0).or.(ib.eq.m1)) then
      fibj = 0
      gibj = 0
      hibj = 0
      iibj = 0
      ribj = 0
else
```

```
                    fibj = w1(ib,j)
                    gibj = w2(ib,j)                                          1470
                    hibj = w3(ib,j)
                    iibj = w4(ib,j)
                    ribj = r(ib,j)
                endif

                sa = a1(i,j)*z1 + a6(i,j)*z2 + a3(i,j)*z3 + a8(i,j)*z4
                sb = (a2(i,j)*z1 + a7(i,j)*z2 + a2(i,j)*z3 + a7(i,j)*z4)
     & - sa*fibjb
                sc = ((a3(i,j)*z1 + a8(i,j)*z2 + a1(i,j)*z3+a6(i,j)*z4)
     & - sb*fijb)       /(1+alpha*fitjb)                                      1480
                sd = ((a4(i,j)*z1 + a4(i,j)*z2 + a5(i,j)*z3 +a5(i,j)*z4)
     & - sb*gijb - sa*hibjb - 2*alpha*sa*gibjb)/(1 + alpha*gibj)
                se = cc(i,j) - sd*fibj - sc*gitjb - sb*hijb - sa*iibjb +
     & alpha*(sc*fitjb + sa*gibjb + sc*iitjb + sd*gibj)

                w1(i,j) = ((a5(i,j)*z1 + a5(i,j)*z2 + a4(i,j)*z3 +
     &a4(i,j)*z4) - sb*iijb - sc*hitjb - alpha*sc*(2*iitjb + fitjb))/se
                w2(i,j) = ((a6(i,j)*z1 + a1(i,j)*z2 + a8(i,j)*z3 +
     &a3(i,j)*z4) - sd*hibj - alpha*sd*gibj)/se
                w3(i,j) = ((a7(i,j)*z1 + a2(i,j)*z2 + a7(i,j)*z3 +           1490
     &a2(i,j)*z4) - sd*iibj)/se
                w4(i,j) = (a8(i,j)*z1 + a3(i,j)*z2 + a6(i,j)*z3 +
     &a1(i,j)*z4)/se


c Inversion of the lower triangular matrix

                r(i,j) = (r(i,j) - sa*ribjb - sb*rijb - sc*ritjb
     & - sd*ribj)/se
                                                                             1500
                goto 10

100     w1(i,j) = 0.0
        w2(i,j) = 0.0
```

```
                    w3(i,j) = 0.0
        w4(i,j) = 0.0

   10 continue

                                                                    1510
c Inversion of the upper triangular matrix */

      do 20 jj=1,n

        j = itc00*(n1-jj) + itc01*jj
        jb = j + itc00 - itc01

        do 20 ii=1,m

          i = itc10*(m1-ii) + itc11*ii                              1520
          ib = i + itc10 - itc11
          it = i - itc10 + itc11

          if ((jb.ne.0).and.(jb.ne.n1)) then

            if ((ib.ne.0).and.(ib.ne.m1)) then
              r(i,j) = r(i,j) - w4(i,j)*r(ib,jb)
            endif

            if ((it.ne.0).and.(it.ne.m1)) then                      1530
              r(i,j) = r(i,j) - w2(i,j)*r(it,jb)
            endif

            r(i,j) = r(i,j) - w3(i,j)*r(i,jb)
          endif


          if ((ib.ne.0).and.(ib.ne.m1)) then
            r(i,j) = r(i,j) - w1(i,j)*r(ib,j)
          endif                                                     1540
```

```
 20 continue

    return
    end


    DOUBLE PRECISION FUNCTION S11ABF(X,IFAIL)
C    MARK 5A REVISED - NAG COPYRIGHT 1976
C    MARK 11.5(F77) REVISED. (SEPT 1985.)                              1550
C    ARCSINH(X)
C
C    .. Scalar Arguments ..
     DOUBLE PRECISION          X
     INTEGER                   IFAIL
C    .. Local Scalars ..
     DOUBLE PRECISION          LN2, T, XHI, Y
C    .. Intrinsic Functions ..
     INTRINSIC                 ABS, LOG, SIGN, SQRT
C    .. Data statements ..                                             1560
C    PRECISION DEPENDENT CONSTANTS
C08  DATA XHI,LN2/1.0D+5,6.9314718D-1/
C12  DATA XHI,LN2/1.0D+7,6.93147180560D-1/
C14  DATA XHI,LN2/1.0D+8,6.9314718055995D-1/
     DATA XHI,LN2/1.0D+9,6.931471805599453D-1/
C18  DATA XHI,LN2/1.0D+10,6.93147180559945309D-1/
C    .. Executable Statements ..
C
C    NO FAILURE EXITS
     IFAIL = 0                                                         1570
     T = ABS(X)
C    TEST LARGE RANGE
     IF (T.GT.XHI) GO TO 20
     T = X*X
C
C    TEST FOR MIDDLE RANGE
```

```
      IF (T.GT.1.0D0) GO TO 40
C     EXPANSION ARGUMENT
      T = 2.0D0*T - 1.0D0
C                                                                       1580
C     * EXPANSION (0012) *
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 08E
C08   Y = (((((((+2.5373657D-6)*T-8.8982681D-6)*T+2.6962688D-5)
C08  *    *T-1.0390000D-4)*T+4.2316523D-4)*T-1.8337913D-3)
C08  *    *T+9.0042679D-3)*T-5.7366614D-2)*T + 9.3122986D-1
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 12E
C12   Y = ((((((((((((-6.08021584775D-9)*T+1.98638973110D-8)
C12  *    *T-4.57508091809D-8)*T+1.58893891259D-7)                     1590
C12  *    *T-5.83039413992D-7)*T+2.05819238929D-6)
C12  *    *T-7.40199001696D-6)*T+2.74058634726D-5)
C12  *    *T-1.05071718894D-4)*T+4.23002689824D-4)
C12  *    *T-1.83345864783D-3)*T+9.00428855313D-3)
C12  *    *T-5.73666392630D-2)*T + 9.31229859453D-1
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 14E
C14   Y = ((((((((((((((-5.8308026585469D-10)*T+1.8762742119651D-9)
C14  *    *T-3.8936648507940D-9)*T+1.3296937569163D-8)
C14  *    *T-4.9030635676319D-8)*T+1.6792346090382D-7)                 1600
C14  *    *T-5.8053399097437D-7)*T+2.0520358645332D-6)
C14  *    *T-7.4030149627429D-6)*T+2.7408018256297D-5)
C14  *    *T-1.0507150365535D-4)*T+4.2300233069384D-4)
C14  *    *T-1.8334586677591D-3)*T+9.0042885755717D-3)
C14  *    *T-5.7366639262479D-2)*T + 9.3122985945271D-1
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 16E
      Y = ((((((((((((((+1.810792296549804D-11
     *    *T-5.731943029121004D-11)*T+1.008344962167889D-10)
     *    *T-3.394726871170490D-10)*T+1.299779213740398D-9)             1610
     *    *T-4.319978113584910D-9)*T+1.432753532351304D-8)
     *    *T-4.863477336087045D-8)*T+1.670117348345774D-7)
```

```
     *    *T-5.807433412373489D-7)*T+2.052474396638805D-6)
     *    *T-7.402952157663977D-6)*T+2.740790473603819D-5)
     *    *T-1.050715136470630D-4)*T+4.230023450529706D-4)
     *    *T-1.833458667045431D-3)*T+9.004288574881897D-3
          Y = (Y*T-5.736663926249348D-2)*T + 9.312298594527122D-1
C
C     EXPANSION (0012) EVALUATED AS Y(T)  --PRECISION 18E
C18   Y = ((((((((((((((+1.83000936086660710D-12                                1620
C18  *    *T-5.74532833323253760D-12)*T+8.95787616116500070D-12)
C18  *    *T-3.00291207083554898D-11)*T+1.20278345675996627D-10)
C18  *    *T-3.94053306282758078D-10)*T+1.27690409672956527D-9)
C18  *    *T-4.26028056137241518D-9)*T+1.43437981020129308D-8)
C18  *    *T-4.86735767698085752D-8)*T+1.67004579212037421D-7)
C18  *    *T-5.80728097040980303D-7)*T+2.05247631332341414D-6)
C18  *    *T-7.40295567555544635D-6)*T+2.74079044411636393D-5)
C18  *    *T-1.05071513207326607D-4)*T+4.23002345076007640D-4
C18   Y = (((Y*T-1.83345866707041640D-3)*T+9.00428857488119850D-3)
C18  *    *T-5.73666392624930609D-2)*T + 9.31229859452712177D-1              1630
C
C
      S11ABF = X*Y
      RETURN
C
C     LARGE X
   20 S11ABF = SIGN(DLOG(T)+LN2,X)
      RETURN
C
C     MIDDLE X                                                               1640
   40 S11ABF = SIGN(DLOG(SQRT(T+1.0D0)+ABS(X)),X)
      RETURN
C
      END
```

Appendix E

Bounce Averaging of Collision Operators

Here, we show the details of the bounce averaging integrals for the collision operators used in Chapters 3 and 4. The general form of the linearized operator for collisions with a background of Maxwellian electrons and ions, normalized to the thermal collision frequency, is

$$\begin{aligned} C(f) &= -\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\left(A(p)\frac{\partial f}{\partial p}+F(p)f\right)\right] \\ &\quad -\frac{B_t(p)}{p^2}\frac{\partial}{\partial \xi}\left[(1-\xi^2)\frac{\partial f}{\partial \xi}\right]+I_1(f_M,f_{l=1})\xi \end{aligned} \tag{E.1}$$

as described in Section 2.3.2. The potentials A, F, and B_t and the function I_1 are given in Appendix A, and the l=1 Legendre component of f is

$$f_{l=1} \equiv \frac{3}{2}\int_{-1}^{1} \xi f \, d\xi. \tag{E.2}$$

It is natural to separate (E.1) into three parts:

$$C(f) = C_I(f) + C_{II}(f) + C_{III}(f)$$

$$
\begin{aligned}
C_I(f) &\equiv -\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\left(A(p)\frac{\partial f}{\partial p}+F(p)f\right)\right] \\
C_{II}(f) &\equiv -\frac{B_t(p)}{p^2}\frac{\partial}{\partial \xi}\left[(1-\xi^2)\frac{\partial f}{\partial \xi}\right] \\
C_{III}(f) &\equiv I_1(f_M, f_{l=1})\xi.
\end{aligned} \tag{E.3}
$$

We note that C_{II} has a form identical to the Lorentz electron-ion collision operator used in Section 3.2 in the limit $Z_i \gg 1$.

The bounce average is given by (2.64),

$$
\{\mathcal{A}\} = \frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_{\sigma}\right]_T \mathcal{A} \tag{E.4}
$$

where θ_c equals π for circulating orbits and θ_T for trapped orbits, and the sum over $\sigma = \xi/|\xi|$ is included only for trapped orbits. The quantity λ is the normalized bounce time (2.65),

$$
\lambda = \int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}. \tag{E.5}
$$

The momentum phase space is chosen to have coordinates (p, ξ_0) which are independent of θ, and ξ is implicitly dependent on θ through (2.28),

$$
\xi = \sigma\sqrt{1-\Psi(r,\theta)\,(1-\xi_0^2)} \tag{E.6}
$$

with $\Psi = (1+\epsilon)/(1+\epsilon\cos\theta)$ as given in (2.26). Partial derivatives with respect to ξ are replaced by taking the differential of (E.6),

$$
\xi\, d\xi = \Psi\xi_0\, d\xi_0 \tag{E.7}
$$

Therefore

$$
\frac{\partial}{\partial \xi} = \frac{\xi}{\Psi\xi_0}\frac{\partial}{\partial \xi_0} \tag{E.8}
$$

E.1 Distributions Without θ Dependence

First, we consider the case where the distribution function f is independent of the poloidal angle θ. We note that such a function must be *symmetric* in the trapped region of phase space,

$$f(\sigma = +1) = f(\sigma = -1); \tag{E.9}$$

see Section 3.2 for details.

Part I

We observe that in this case $C_I(f)$ is completely independent of θ, and therefore

$$\{C_I(f)\} = -\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\left(A(p)\frac{\partial f}{\partial p} + F(p)f\right)\right] \tag{E.10}$$

Part II

To bounce average $C_{II}(f)$, we first need to replace the partial derivatives with respect to ξ using (E.8):

$$C_{II}(f) = -\frac{B_t(p)}{p^2}\frac{\xi}{\Psi\xi_0}\frac{\partial}{\partial\xi_0}\left[\Psi(1-\xi_0^2)\frac{\xi}{\Psi\xi_0}\frac{\partial f}{\partial\xi_0}\right] \tag{E.11}$$

Then

$$\begin{aligned}
\{C_{II}(f)\} &= -\frac{B_t(p)}{p^2}\frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_\sigma\right]_T\frac{\xi}{\Psi\xi_0}\frac{\partial}{\partial\xi_0}\left[\Psi(1-\xi_0^2)\frac{\xi}{\Psi\xi_0}\frac{\partial f}{\partial\xi_0}\right] \\
&= -\frac{B_t(p)}{p^2}\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[(1-\xi_0^2)\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\left[\frac{1}{2}\sum_\sigma\right]_T\frac{\xi}{\Psi\xi_0}\frac{\partial f}{\partial\xi_0}\right] \\
&= -\frac{B_t(p)}{p^2}\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[(1-\xi_0^2)\lambda\left\{\frac{\xi^2}{\Psi\xi_0^2}\right\}\frac{\partial f}{\partial\xi_0}\right] \\
&= -\frac{B_t(p)}{p^2}\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[(1-\xi_0^2)\lambda\left\{\frac{1-\Psi(1-\xi_0^2)}{\Psi\xi_0^2}\right\}\frac{\partial f}{\partial\xi_0}\right] \\
&= -\frac{B_t(p)}{p^2}\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[(1-\xi_0^2)\lambda\left(1-\frac{\Delta}{\xi_0^2}\right)\frac{\partial f}{\partial\xi_0}\right]
\end{aligned} \tag{E.12}$$

where

$$\Delta \equiv \left\{1 - \frac{1}{\Psi}\right\} = \frac{\{1-\cos\theta\}\epsilon}{(1+\epsilon)}. \tag{E.13}$$

The function Δ can be calculated numerically with elliptic integrals.

Part III

Making the substitution (E.8) in the integral (E.2), we get

$$f_{l=1} = \frac{3}{2}\int_{-1}^{1} \xi d\xi\ f = \frac{3}{2}\int_{-1}^{-\sqrt{1-1/\Psi}} \Psi\xi_0 d\xi_0\ f + \frac{3}{2}\int_{\sqrt{1-1/\Psi}}^{1} \Psi\xi_0 d\xi_0\ f. \tag{E.14}$$

The limits of integration come from (E.6),

ξ	-1	0^-	0^+	1
ξ_0	-1	$-\sqrt{1-1/\Psi}$	$\sqrt{1-1/\Psi}$	1

However, the region of phase space $\xi_0 \in (-\sqrt{1-1/\Psi}, \sqrt{1-1/\Psi})$ consists of trapped orbits, and the symmetry of f for trapped orbits means that

$$\frac{3}{2}\int_{-\sqrt{1-1/\Psi}}^{\sqrt{1-1/\Psi}} \Psi\xi_0 d\xi_0\ f = 0 \tag{E.15}$$

and thus

$$f_{l=1} = \frac{3}{2}\int_{-1}^{1} \Psi\xi_0 d\xi_0\ f \tag{E.16}$$

This can be written as $f_{l=1} = \Psi f_{l=1,0}$, where $f_{l=1,0}$ is the Legendre integral at θ=0 and is independent of θ. Then, since the linear operator I_1 and the Maxwellian distribution f_M are independent of θ, the bounce average is

$$\begin{aligned}
\{C_{III}(f)\} &= \{I_1(f_M, \Psi f_{l=1,0})\xi\} \\
&= I_1(f_M, f_{l=1,0})\{\Psi\xi\} \\
&= I_1(f_M, f_{l=1,0})\frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_\sigma\right]_T \Psi\xi \\
&= I_1(f_M, f_{l=1,0})\frac{1}{\lambda}\left[\frac{1}{2}\sum_\sigma\right]_T \xi_0 \int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{1+\epsilon}{1+\epsilon\cos\theta}
\end{aligned} \tag{E.17}$$

Since σ is the sign of ξ_0,

$$\left[\sum_{\sigma}\right]_T \xi_0 = 0 \tag{E.18}$$

and trapped particles are excluded. Then we can perform the integral over θ (see, e.g., Equation **3.613** 1 in Reference [96]),

$$\{C_{III}(f)\} \;=\; H(|\xi_0| - \xi_{0T})\; I_1(f_M, f_{l=1,0})\xi_0\; \frac{1}{\lambda}\sqrt{\frac{1+\epsilon}{1-\epsilon}} \tag{E.19}$$

where H is the Heaviside unit step function and $\xi_{0T} = \sqrt{2\epsilon/(1+\epsilon)}$ is the trapped-passing boundary in phase space.

E.2 Distributions of the Form $\tilde{f}$

In Section 2.4.3 we defined the function $\tilde{f}$:

$$\tilde{f} \equiv -\frac{v_\parallel}{\Omega_\theta}\frac{\partial f_0}{\partial r} = -\frac{v\xi}{\Omega_\theta}\frac{\partial f_0}{\partial r} \tag{E.20}$$

for a known, θ-independent distribution function f_0. Since f_0 and therefore $\partial f_0/\partial r$ are symmetric in the trapped region of phase space, $\tilde{f}$ is observed to be *antisymmetric*,

$$\tilde{f}(\sigma = +1) = \; -\tilde{f}(\sigma = -1), \tag{E.21}$$

because of the extra factor of ξ.

The function $\tilde{f}$ contains an implicit θ dependence from two sources: $\xi = \sigma\sqrt{1 + \Psi(1 - \xi_0^2)}$and $\Omega_\theta \propto B_\theta \propto \Psi$ since the poloidal magnetic field is inversely proportional to the major radius as shown in Section 2.2.1. We define $\tilde{f}^{(0)}$ as the value of $\tilde{f}$ at θ=0,

$$\tilde{f}^{(0)} = \frac{v\xi_0}{\Omega_{\theta,min}}\frac{\partial f_0}{\partial r} \tag{E.22}$$

where $\Omega_{\theta,min} \equiv \Omega_\theta/\Psi$, then separate the θ-dependent part from $\tilde{f}$ by writing

$$\tilde{f} = \left(\frac{\xi}{\Psi\xi_0}\right)\tilde{f}^{(0)}. \tag{E.23}$$

Just like $\tilde{f}$, the function $\tilde{f}^{(0)}$ is antisymmetric in the trapped region.

Part I

Because p is independent of θ, finding $\{C_I(\tilde{f})\}$ is simple,

$$\left\{C_I(\tilde{f})\right\} = -\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\left(A(p)\frac{\partial\{\tilde{f}\}}{\partial p} + F(p)\{\tilde{f}\}\right)\right]. \tag{E.24}$$

We find the bounce average of $\tilde{f}$, as follows,

$$\begin{aligned}\left\{\tilde{f}\right\} &= \frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_\sigma\right]_T\left(\frac{\xi}{\Psi\xi_0}\right)\tilde{f}^{(0)} \\ &= \frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{1+\epsilon\cos\theta}{1+\epsilon}\left[\frac{1}{2}\sum_\sigma\right]_T\tilde{f}^{(0)} \\ &= H(|\xi_0| - \xi_{0T})\,\frac{1}{\lambda(1+\epsilon)}\tilde{f}^{(0)}\end{aligned} \tag{E.25}$$

where we used the fact that $\tilde{f}^{(0)}$ is antisymmetric for trapped orbits. Therefore

$$\left\{C_I(\tilde{f})\right\} = -H(|\xi_0| - \xi_{0T})\,\frac{1}{\lambda(1+\epsilon)}\frac{1}{p^2}\frac{\partial}{\partial p}\left[p^2\left(A(p)\frac{\partial\tilde{f}^{(0)}}{\partial p} + F(p)\tilde{f}^{(0)}\right)\right]. \tag{E.26}$$

Part II

First, the antisymmetry of $\tilde{f}$ is once again used to show that $\{C_{II}(\tilde{f})\}$ is zero for trapped orbits:

$$C_{II}(\tilde{f}) = -\frac{B_t(p)}{p^2}\frac{\partial}{\partial\xi}\left[(1-\xi^2)\frac{\partial\tilde{f}}{\partial\xi}\right] \tag{E.27}$$

$$\left\{C_{II}(\tilde{f})\right\} = -\frac{B_t(p)}{p^2}\frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_\sigma\right]_T\frac{\partial}{\partial\xi}\left[(1-\xi^2)\frac{\partial\tilde{f}}{\partial\xi}\right]$$

$$
\begin{aligned}
&= -\frac{B_t(p)}{p^2}\frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\frac{\partial}{\partial|\xi|}\left[(1-|\xi|^2)\frac{\partial}{\partial|\xi|}\left[\frac{1}{2}\sum_\sigma\right]_T\tilde{f}\right] \\
&= -H(|\xi_0|-\xi_{0T})\,\frac{B_t(p)}{p^2}\frac{1}{\lambda}\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\frac{\partial}{\partial\xi}\left[(1-\xi^2)\frac{\partial\tilde{f}}{\partial\xi}\right] \qquad \text{(E.28)}
\end{aligned}
$$

We can now find $\{C_{II}(\tilde{f})\}$ by algebraic calculation with (E.6) and (E.8):

$$
\begin{aligned}
C_{II}(\tilde{f}) &= -H(|\xi_0|-\xi_{0T})\,\frac{B_t(p)}{p^2}\frac{1}{\lambda}\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\frac{\xi}{\Psi\xi_0}\frac{\partial}{\partial\xi_0}\left[\Psi(1-\xi_0^2)\left(\frac{\partial\tilde{f}}{\partial\xi}\right)\right] \\
&= -H(|\xi_0|-\xi_{0T})\,\frac{B_t(p)}{p^2}\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[(1-\xi_0^2)\left(\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\frac{\partial\tilde{f}}{\partial\xi}\right)\right] \qquad \text{(E.29)}
\end{aligned}
$$

We manipulate $\partial\tilde{f}/\partial\xi$, using (E.6) and (E.8) as necessary, and algebraic substitution:

$$
\begin{aligned}
\frac{\partial\tilde{f}}{\partial\xi} &= \frac{\partial}{\partial\xi}\left(\frac{\xi}{\Psi\xi_0}\tilde{f}^{(0)}\right) \\
&= \frac{1}{\Psi}\left[\frac{1}{\xi_0}\tilde{f}^{(0)}+\xi\frac{\xi}{\Psi\xi_0}\frac{\partial}{\partial\xi_0}\left(\frac{1}{\xi_0}\tilde{f}^{(0)}\right)\right] \\
&= \frac{1}{\Psi}\left[\frac{1}{\xi_0}\tilde{f}^{(0)}+\frac{1-\Psi(1-\xi_0^2)}{\Psi\xi_0}\frac{\partial}{\partial\xi_0}\left(\frac{1}{\xi_0}\tilde{f}^{(0)}\right)\right] \\
&= \frac{1}{\Psi}\left[\frac{1}{\xi_0}\tilde{f}^{(0)}+\xi_0\frac{\partial}{\partial\xi_0}\left(\frac{1}{\xi_0}\tilde{f}^{(0)}\right)+\frac{1-\Psi}{\Psi\xi_0}\frac{\partial}{\partial\xi_0}\left(\frac{1}{\xi_0}\tilde{f}^{(0)}\right)\right] \\
&= \frac{1}{\Psi}\left[\frac{\partial\tilde{f}^{(0)}}{\partial\xi_0}-\left(1-\frac{1}{\Psi}\right)\frac{1}{\xi_0}\frac{\partial}{\partial\xi_0}\left(\frac{1}{\xi_0}\tilde{f}^{(0)}\right)\right]. \qquad \text{(E.30)}
\end{aligned}
$$

This gives the form

$$
\begin{aligned}
\frac{\partial\tilde{f}}{\partial\xi} &= \frac{1}{\Psi}(L_1+L_2) \\
L_1 &\equiv \frac{\partial\tilde{f}^{(0)}}{\partial\xi_0} \qquad \text{(E.31)} \\
L_2 &\equiv -\left(1-\frac{1}{\Psi}\right)\frac{1}{\xi_0}\frac{\partial}{\partial\xi_0}\left(\frac{1}{\xi_0}\tilde{f}^{(0)}\right). \qquad \text{(E.32)}
\end{aligned}
$$

This form of $\partial\tilde{f}/\partial\xi$ is useful for the RF-bootstrap current problem for two reasons: the term L_2 is small in the limit $\epsilon \ll 1$, since $1-1/\Psi = \epsilon(1-\cos\theta)/(1+\epsilon)$; and L_2 is identically zero when f_0 is independent of ξ_0, as we shall show below.

The final expression for $\{C_{II}(\tilde{f})\}$ is

$$\left\{C_{II}(\tilde{f})\right\} = -H(|\xi_0| - \xi_{0T}) \frac{B_t(p)}{p^2} \frac{1}{\lambda(1+\epsilon)} \frac{\partial}{\partial \xi_0} \left[(1-\xi_0^2)\left(\bar{L}_1 + \bar{L}_2\right)\right] \quad \text{(E.33)}$$

where we define

$$\begin{aligned} \bar{L}_1 &\equiv (1+\epsilon) \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} \frac{1}{\Psi} L_1 \\ &= (1+\epsilon) \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} \frac{1+\epsilon\cos\theta}{1+\epsilon} \frac{\partial \tilde{f}^{(0)}}{\partial \xi_0} \\ &= \frac{\partial \tilde{f}^{(0)}}{\partial \xi_0} \end{aligned} \quad \text{(E.34)}$$

and

$$\begin{aligned} \bar{L}_2 &\equiv (1+\epsilon) \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} \frac{1}{\Psi} L_2 \\ &= -(1+\epsilon) \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} \frac{1}{\Psi} \left(1 - \frac{1}{\Psi}\right) \frac{1}{\xi_0} \frac{\partial}{\partial \xi_0} \left(\frac{1}{\xi_0} \tilde{f}^{(0)}\right) \\ &= -(1+\epsilon) \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} \frac{1+\epsilon\cos\theta}{1+\epsilon} \frac{\epsilon(1-\cos\theta)}{1+\epsilon} \frac{1}{\xi_0} \frac{\partial}{\partial \xi_0} \left(\frac{1}{\xi_0} \tilde{f}^{(0)}\right) \\ &= -\frac{\epsilon - \epsilon^2/2}{1+\epsilon} \frac{1}{\xi_0} \frac{\partial}{\partial \xi_0} \left(\frac{1}{\xi_0} \tilde{f}^{(0)}\right) \end{aligned} \quad \text{(E.35)}$$

This expression is particularly simple for the case where f_0 is independent of ξ_0, e.g. the standard neoclassical transport case where f_0 is a Maxwellian. We note that $\tilde{f}^{(0)}$ in (E.22) is proportional to ξ_0, and

$$\frac{\partial}{\partial \xi_0} \left(\frac{1}{\xi_0} \tilde{f}^{(0)}\right) = \frac{\partial}{\partial \xi_0} \left(\frac{v}{\Omega_{\theta,min}} \frac{\partial f_0}{\partial r}\right) = 0, \quad \text{(E.36)}$$

and therefore L_2 in (E.32) is zero. Then we get

$$\begin{aligned} \left\{C_{II}(\tilde{f})\right\} &= -H(|\xi_0| - \xi_{0T}) \frac{B_t(p)}{p^2} \frac{1}{\lambda(1+\epsilon)} \frac{\partial}{\partial \xi_0} \left[(1-\xi_0^2) \frac{\partial \tilde{f}^{(0)}}{\partial \xi_0}\right] \\ &= -H(|\xi_0| - \xi_{0T}) \frac{B_t(p)}{p^2} \frac{1}{\lambda(1+\epsilon)} \frac{\partial}{\partial \xi_0} \left[(1-\xi_0^2)\right] \frac{v}{\Omega_{\theta,min}} \frac{\partial f_0}{\partial r} \end{aligned}$$

$$= -H(|\xi_0| - \xi_{0T})\ \frac{B_t(p)}{p^2}\frac{1}{\lambda(1+\epsilon)}(-2\xi_0)\frac{v}{\Omega_{\theta,min}}\frac{\partial f_0}{\partial r}$$
$$= 2H(|\xi_0| - \xi_{0T})\ \frac{B_t(p)}{p^2}\frac{1}{\lambda(1+\epsilon)}\tilde{f}^{(0)}. \tag{E.37}$$

Part III

The integral (E.2) in this case is

$$\begin{aligned}\tilde{f}_{l=1} &= \frac{3}{2}\int_{-1}^{1} \xi d\xi\ \tilde{f} \\ &= \frac{3}{2}\int_{-1}^{-\sqrt{1-1/\Psi}} \Psi\xi_0 d\xi_0\ \left(\frac{\xi}{\Psi\xi_0}\right)\tilde{f}^{(0)} + \frac{3}{2}\int_{\sqrt{1-1/\Psi}}^{1} \Psi\xi_0 d\xi_0\ \left(\frac{\xi}{\Psi\xi_0}\right)\tilde{f}^{(0)} \\ &= \frac{3}{2}\int_{-1}^{1} d\xi_0\ H(|\xi_0| - \xi_{0T})\ \xi\tilde{f}^{(0)}. \end{aligned} \tag{E.38}$$

In order to find $\{C_{III}(\tilde{f})\}$, we write it in the form

$$C_{III}(\tilde{f}) = I_1(f_M, \tilde{f}_{l=1}\xi) \tag{E.39}$$

from (2.48). Then, with I_1 and f_M independent of θ, we get

$$\left\{C_{III}(\tilde{f})\right\} = I_1\left(f_M, \{\tilde{f}_{l=1}\xi\}\right). \tag{E.40}$$

$$\{\tilde{f}_{l=1}\xi\} = \frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_{\sigma}\right]_T \xi\ \tilde{f}_{l=1} \tag{E.41}$$

But $\tilde{f}_{l=1}$ is independent of σ, since ξ_0 is integrated out of (E.38). and $\left[\frac{1}{2}\sum_{\sigma}\right]_T \xi$=0 for trapped orbits. So we continue with circulating orbits only,

$$\begin{aligned}\{\tilde{f}_{l=1}\xi\} &= H(|\xi_0| - \xi_{0T})\ \frac{1}{\lambda}\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\tilde{f}_{l=1}\xi \\ &= H(|\xi_0| - \xi_{0T})\ \frac{1}{\lambda}\xi_0\int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\left[\frac{3}{2}\int_{-1}^{1} d\xi_0\ H(|\xi_0| - \xi_{0T})\ \xi\tilde{f}^{(0)}\right] \\ &= H(|\xi_0| - \xi_{0T})\ \frac{1}{\lambda}\xi_0\left[\frac{3}{2}\int_{-1}^{1} d\xi_0\ H(|\xi_0| - \xi_{0T})\ \int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\xi\ \tilde{f}^{(0)}\right] \\ &= H(|\xi_0| - \xi_{0T})\ \frac{1}{\lambda}\xi_0\left[\frac{3}{2}\int_{-1}^{1} d\xi_0\ H(|\xi_0| - \xi_{0T})\ \int_{-\pi}^{\pi}\frac{d\theta}{2\pi}\frac{\xi}{\xi_0}\ \xi_0\tilde{f}^{(0)}\right]\end{aligned}$$

$$
\begin{aligned}
&= H(|\xi_0| - \xi_{0T})\ \frac{1}{\lambda}\xi_0 \left[\frac{3}{2}\int_{-1}^{1}\ d\xi_0\ H(|\xi_0| - \xi_{0T})\ \lambda \left\{\frac{\xi^2}{\xi_0^2}\right\}\ \xi_0 \tilde{f}^{(0)}\right] \\
&= H(|\xi_0| - \xi_{0T})\ \frac{1}{\lambda}\xi_0 \left[\frac{3}{2}\int_{-1}^{1}\ d\xi_0\ H(|\xi_0| - \xi_{0T})\ \lambda \frac{1 - \{\Psi\}(1 - \xi_0^2)}{\xi_0^2}\ \xi_0 \tilde{f}^{(0)}\right]
\end{aligned}
\tag{E.42}
$$

Thus we can express $\{C_{III}(\tilde{f})\}$ as

$$
\begin{aligned}
\{C_{III}(f)\} &= H(|\xi_0| - \xi_{0T})\ I_1(f_M, f_{avg})\xi_0 \frac{1}{\lambda} \\
f_{avg} &\equiv \frac{3}{2}\int_{-1}^{1}\ d\xi_0\ W\xi_0 \tilde{f}^{(0)} \\
W &\equiv H(|\xi_0| - \xi_{0T})\ \lambda \frac{1 - \{\Psi\}(1 - \xi_0^2)}{\xi_0^2}.
\end{aligned}
\tag{E.43}
$$

Appendix F

Bounce Averaging of Quasilinear Operators

The analysis of Hammett [26] on the subject of quasilinear operators in inhomogeneous (specifically toroidal) plasmas is used to guide the bounce averaging of the quasilinear operator for the FASTFP-NC Fokker-Planck code. Hammett shows that using the form of the quasilinear operator in the homogeneous plasma limit from Kennel and Engelmann [21] and bounce averaging gives a result identical to the quasilinear operator calculated in an inhomogeneous plasma using a time average [23, 24, 25].

We perform the bounce average of the quasilinear operator derived by Lerche [22] in the relativistic limit. First in Section F.1 we take the bounce average of a general quasilinear operator. In Section F.2 we consider the specific case of lower hybrid waves, and Section F.3 shows the derivation for electron cyclotron waves.

F.1 General Bounce Average

The quasilinear operator used here is (2.50-2.52). We take a single value of n, where $\omega \approx n\Omega$. It is useful to be able to rewrite this operator so that the integral over $\mathbf{k}$ is inside the differential operator $\mathcal{L}$. Sine $\mathcal{L}$ in (2.51) is a function of $k_\parallel$, this requires a substitution: using the delta function for the resonance condition $\omega - k_\parallel v_\parallel - n\Omega = 0$

we get

$$\frac{k_{\|}p}{\gamma m\omega} = \frac{1}{\xi}\left(1 - \frac{n\Omega}{\omega}\right) \tag{F.1}$$

since $p_{\|} = p\xi$. Then (2.50) becomes

$$Q(f) = \mathcal{L}'\left[p^2(1-\xi^2)\ \bar{D}(p,\xi)\mathcal{L}(f)\right] \tag{F.2}$$

$$\mathcal{L}(\mathcal{A}) = \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(1 - \xi^2 - \frac{n\Omega}{\omega}\right)\frac{1}{p^2\xi}\frac{\partial \mathcal{A}}{\partial \xi} \tag{F.3}$$

$$\mathcal{L}'(\mathcal{A}) = \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\mathcal{A}) + \frac{\partial}{\partial \xi}\left[\left(1 - \xi^2 - \frac{n\Omega}{\omega}\right)\frac{\mathcal{A}}{\xi}\right]\right) \tag{F.4}$$

$$\bar{D}(p,\xi) \equiv \lim_{V\to\infty}\frac{1}{V}\int \frac{d^3k}{(2\pi)^3} D_{\mathbf{k}}\delta(\omega - k_{\|}v_{\|} - n\Omega) \tag{F.5}$$

In order to perform a bounce average, we change from the local momentum-space coordinates (p,ξ) to the coordinates (p,ξ_0) which are constants of the motion. From (2.28) we have $\xi = \sigma\sqrt{1 - \Psi(r,\theta)\,(1-\xi_0^2)}$ with $\Psi = (1+\epsilon)/(1+\epsilon\cos\theta)$, and by taking the derivative we get

$$\frac{\partial}{\partial \xi} = \frac{\xi}{\Psi\xi_0}\frac{\partial}{\partial \xi_0}. \tag{F.6}$$

With this expressions we can rewrite

$$\begin{aligned}\mathcal{L}(\mathcal{A}) &= \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(\Psi(1-\xi_0^2) - \frac{n\Omega}{\omega}\right)\frac{1}{p^2\Psi\xi_0}\frac{\partial \mathcal{A}}{\partial \xi_0} \\ &= \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(1 - \xi_0^2 - \frac{n\Omega}{\omega\Psi}\right)\frac{1}{p^2\xi_0}\frac{\partial \mathcal{A}}{\partial \xi_0}\end{aligned} \tag{F.7}$$

$$\begin{aligned}\mathcal{L}'(\mathcal{A}) &= \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\mathcal{A}) + \frac{\xi}{\Psi\xi_0}\frac{\partial}{\partial \xi_0}\left[\left(\Psi(1-\xi_0^2) - \frac{n\Omega}{\omega}\right)\frac{\mathcal{A}}{\xi}\right]\right) \\ &= \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\mathcal{A}) + \frac{\xi}{\xi_0}\frac{\partial}{\partial \xi_0}\left[\left(1 - \xi_0^2 - \frac{n\Omega}{\omega\Psi}\right)\frac{\mathcal{A}}{\xi}\right]\right).\end{aligned} \tag{F.8}$$

In these expressions ξ is implicitly a function of ξ_0 and θ. We note the appearance in these expressions of the ratio $n\Omega/\omega\Psi$ which is independent of θ since both Ω and Ψ are proportional to the magnetic field.

Now the bounce average is performed according to (2.64),

$$\{\mathcal{A}\} = \frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_{\sigma}\right]_T \mathcal{A} \tag{F.9}$$

where θ_c equals π for circulating orbits and θ_T for trapped orbits, and the sum over $\sigma = \xi/|\xi|$ is included only for trapped orbits. The quantity λ is the normalized bounce time (2.65).

We take the bounce average of the differential operator $\mathcal{L}'$ first:

$$\begin{aligned}
\{\mathcal{L}'(\mathcal{A})\} &= \frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\frac{\xi_0}{\xi}\left[\frac{1}{2}\sum_\sigma\right]_T\frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\mathcal{A})+\frac{\xi}{\xi_0}\frac{\partial}{\partial\xi_0}\left[\left(1-\xi_0^2-\frac{n\Omega}{\omega\Psi}\right)\frac{\mathcal{A}}{\xi}\right]\right)\\
&= \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\{\mathcal{A}\})+\frac{1}{\lambda}\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\left[\frac{1}{2}\sum_\sigma\right]_T\frac{\partial}{\partial\xi_0}\left[\left(1-\xi_0^2-\frac{n\Omega}{\omega\Psi}\right)\frac{\mathcal{A}}{\xi}\right]\right)\\
&= \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\{\mathcal{A}\})+\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[\left(1-\xi_0^2-\frac{n\Omega}{\omega\Psi}\right)\int_{-\theta_c}^{\theta_c}\frac{d\theta}{2\pi}\left[\frac{1}{2}\sum_\sigma\right]_T\frac{\mathcal{A}}{\xi}\right]\right)\\
&= \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\{\mathcal{A}\})+\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[\left(1-\xi_0^2-\frac{n\Omega}{\omega\Psi}\right)\frac{\lambda\{\mathcal{A}\}}{\xi_0}\right]\right) \qquad \text{(F.10)}
\end{aligned}$$

So the quasilinear operator becomes

$$\{Q(f)\} = \bar{\mathcal{L}}'\left\{p^2(1-\xi^2)\,\bar{D}(p,\xi)\mathcal{L}(f)\right\} \qquad \text{(F.11)}$$

where

$$\bar{\mathcal{L}}'(\mathcal{A}) = \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\mathcal{A})+\frac{1}{\lambda}\frac{\partial}{\partial\xi_0}\left[\left(1-\xi_0^2-\frac{n\Omega}{\omega\Psi}\right)\frac{\lambda\mathcal{A}}{\xi_0}\right]\right). \qquad \text{(F.12)}$$

and $\mathcal{L}$ is still given by (F.7).

F.2 Lower Hybrid Diffusion

For lower-hybrid waves, we have n=0 in the quasilinear operator, and the diffusion coefficent can be written

$$\bar{D} \equiv \frac{p_\parallel^2}{p_\perp^2}D_{LH} = \frac{p^2\xi^2}{p^2(1-\xi)^2}D_{LH}. \qquad \text{(F.13)}$$

The factor of $p_\parallel^2/p_\perp^2$ is necessary to put the operator into the traditional form (3.27); it also appears in the diffusion coefficient (2.52) when we consider lower hybrid waves

with $E_{\mathbf{k},\parallel}$ only. Then (F.11) becomes

$$\{Q(f)\} = \bar{\mathcal{L}}'\left\{p^2\xi^2\ D_{LH}(p,\xi)\mathcal{L}(f)\right\} \tag{F.14}$$

We use the model for the diffusion coefficient discussed in Section 3.3.1:

$$\begin{aligned} D_{LH} &= D_0, \quad v_1 < v_\parallel < v_2 \\ &= 0 \qquad v_\parallel < v_1, v_\parallel > v_2, \end{aligned}$$

where D_0 is a constant greater than 1 when normalized to the collisional diffusion coefficient $\nu_e p_{Te}^2$. We note that the region of momentum space over which D_{LH} is nonzero covers primarily circulating electrons. This can be observed by noting that v_1 is typically 3 to 4 times v_{Te}. Since trapped electrons have

$$\xi = \frac{v_\parallel}{v} < \xi_{0T} = \sqrt{\frac{2\epsilon}{1+\epsilon}}$$

trapped electrons interacting with the waves have v greater than $(3/\sqrt{2\epsilon})v_{Te}$. This effect is even stronger when relativity is including, since the resonant trapped electrons would need p greater than $(3/\sqrt{2\epsilon})\gamma p_{Te}$ with the extra factor γ greater than 1. For plasmas which are approximately Maxwellian in their core, there very few trapped electrons with enough momentum to interact with the LH waves.

For this reason we approximate the quasilinear operator by taking $\xi \approx \xi_0$ in (F.14), which simplifies the bounce averaging:

$$\{Q(f)\} = \bar{\mathcal{L}}'\left[p^2\xi_0^2\ D_{LH}(p,\xi_0)\mathcal{L}(\{f\})\right] \tag{F.15}$$

$$\bar{\mathcal{L}}'(\mathcal{A}) = \frac{1}{p^2}\left(\frac{\partial}{\partial p}(p\mathcal{A}) + \frac{1}{\lambda}\frac{\partial}{\partial \xi_0}\left[\left(1-\xi_0^2\right)\frac{\lambda\mathcal{A}}{\xi_0}\right]\right) \tag{F.16}$$

$$\mathcal{L}(\mathcal{A}) = \frac{1}{p}\frac{\partial \mathcal{A}}{\partial p} + \left(1-\xi_0^2\right)\frac{1}{p^2\xi_0}\frac{\partial \mathcal{A}}{\partial \xi_0} \tag{F.17}$$

$$D_{LH} = D_0, \quad v_1 < v_{\parallel,0} < v_2$$

$$= \quad 0 \qquad v_{\|} < v_1, v_{\|,0} > v_2,$$

F.3 Electron Cyclotron Diffusion

As mentioned in Section 4.2 electron cyclotron waves propagate in the plasma in the form of a well-collimated beam. We take this beam to cross the flux surface r at $\theta = \theta_b$ and $\zeta = \zeta_b$. The beam also has an extent of $\Delta\theta$ and $\Delta\zeta$ in these two respective directions.

We note that this ζ-dependence contradicts the assumption of axisymmetry. However, because the bounce average is an average over the orbit of an electron, a circulating electron will repeatedly orbit in ζ, so that the effect is also to average over ζ. We must explicitly perform the average by adding an integral to (F.11):

$$\begin{aligned} \{Q(f)\} &= \int_0^{2\pi} \frac{d\zeta}{2\pi} \bar{\mathcal{L}}' \left\{ p^2(1-\xi^2)\ \bar{D}(p,\xi)\mathcal{L}(f) \right\} \\ &= \bar{\mathcal{L}}' \left[\frac{1}{\lambda} \int_0^{2\pi} \frac{d\zeta}{2\pi} \int_{-\pi}^{\pi} \frac{d\theta}{2\pi} \frac{\xi_0}{\xi} p^2(1-\xi^2)\ \bar{D}(p,\xi)\mathcal{L}(f) \right] \end{aligned} \tag{F.18}$$

By doing this we are restricted to cases where only circulating electrons are resonant with the waves, which fortunately is the case for current drive cases where the Ohkawa effect [37] on trapped electrons is avoided.

If the widths $\Delta\theta$ and $\Delta\zeta$ are small, we approximate the integration by evaluating the arguments at the point where the beam intersects the flux surface:

$$\begin{aligned} \{Q(f)\} &= \bar{\mathcal{L}}' \left[\frac{1}{\lambda} \frac{\Delta\zeta}{2\pi} \frac{\Delta\theta}{2\pi} \frac{\xi_0}{\xi} p^2(1-\xi^2)\ \bar{D}(p,\xi)\mathcal{L}(f) \right]_{\theta_b} \\ &= \frac{(\Delta\zeta)(\Delta\theta)}{4\pi^2} \bar{\mathcal{L}}' \left[\frac{1}{\lambda} \frac{\xi_0}{\xi} p^2(1-\xi^2)\ \bar{D}(p,\xi)\mathcal{L}(f) \right]_{\theta_b} \end{aligned} \tag{F.19}$$

The diffusion coefficient is evaluated by assuming a Gaussian envelope for the EC beam, which means that when we take the Fourier transform the spectrum in $k_{\|}$ is

also Gaussian and the diffusion coefficient has the form (3.57)

$$D_{\mathbf{k}} = D_{\mathbf{k},0}\, p_\perp^2 \exp\left[-\frac{(k_\parallel - k_{\parallel,0})^2}{(\Delta k_\parallel)^2}\right] \tag{F.20}$$

where

$$D_{\mathbf{k},0} \equiv q_e^2 2\sqrt{\frac{\mu_0}{\epsilon_0}} p_{\mathbf{k},0} \left(N_\perp \frac{p_{Te}}{m_e c}\frac{\omega}{2\Omega_0}\right)^2 \tag{F.21}$$

for peak power flow density $p_{\mathbf{k},0}$. When we perform the integral (F.5), the delta function integration gives the result

$$\bar{D} = D_0\, p_\perp^2 \frac{1}{|v_\parallel|} \exp\left[-\frac{(k_{\parallel,res} - k_{\parallel,0})^2}{(\Delta k_\parallel)^2}\right] \tag{F.22}$$

where

$$D_0 = \lim_{V\to\infty} \frac{1}{V} \int \frac{d^2 k_\perp}{(2\pi)^3}\, q_e^2 2\sqrt{\frac{\mu_0}{\epsilon_0}} p_{\mathbf{k},0} \left(N_\perp \frac{p_{Te}}{m_e c}\frac{\omega}{2\Omega_0}\right)^2 \Bigg|_{k_\parallel = k_{\parallel,res}} . \tag{F.23}$$

and

$$k_{\parallel,res} \equiv \frac{\omega - n\Omega}{v_\parallel} = \frac{\gamma - n\Omega_0/\omega}{p_\parallel/m_e c}\frac{\omega}{c} \tag{F.24}$$

We use a simple model where this D_0 is taken to be a constant.

Substituting this expression into the quasilinear operator (F.19) gives

$$\{Q(f)\} = \frac{(\Delta\zeta)(\Delta\theta)}{4\pi^2}\bar{\mathcal{L}}' \left[\frac{1}{\lambda}\frac{\xi_0}{\xi} p^4 (1-\xi^2)^4\, D_0 \frac{1}{|v_\parallel|} \exp\left[-\frac{(k_{\parallel,res} - k_{\parallel,0})^2}{(\Delta k_\parallel)^2}\right] \mathcal{L}(f)\right]_{\theta_b} . \tag{F.25}$$

We define a new, reduced diffusion coefficient

$$\text{New } D_0 \equiv \frac{(\Delta\zeta)(\Delta\theta)}{4\pi^2} D_0 = \frac{R_0(\Delta\zeta)\, r(\Delta\theta)}{4\pi^2 r R_0} D_0. \tag{F.26}$$

Note that this contains the ratio of the area of the beam $R_0(\Delta\zeta)r(\Delta\theta)$ to the total area of the flux surface. Therefore adding this factor has the same effect as redistributing the wave power throughout the flux surface; this effect was first noted by Cairns and

Lashmore-Davies [79].

Using the subscript b to denote quantities evaluated at the location of the beam θ_b, we get the quasilinear operator

$$\{Q(f)\} = \bar{\mathcal{L}}' \left[\frac{1}{\lambda} \frac{\xi_0}{\xi_b} p^4 \Psi_b^2 (1-\xi_0^2)^4 \; D_0 \frac{1}{\left| v_{\parallel,b} \right|} \exp \left[-\frac{(k_{\parallel,res} - k_{\parallel,0})^2}{(\Delta k_\parallel)^2} \right] \mathcal{L}(f_b) \right] \quad \text{(F.27)}$$

$$\mathcal{L}(\mathcal{A}) = \frac{1}{p} \frac{\partial \mathcal{A}}{\partial p} + \left(1 - \xi_0^2 - \frac{n\Omega}{\omega \Psi} \right) \frac{1}{p^2 \xi_0} \frac{\partial \mathcal{A}}{\partial \xi_0} \quad \text{(F.28)}$$

$$\bar{\mathcal{L}}'(\mathcal{A}) = \frac{1}{p^2} \left(\frac{\partial}{\partial p}(p\mathcal{A}) + \frac{1}{\lambda} \frac{\partial}{\partial \xi_0} \left[\left(1 - \xi_0^2 - \frac{n\Omega}{\omega \Psi} \right) \frac{\lambda \mathcal{A}}{\xi_0} \right] \right). \quad \text{(F.29)}$$

Appendix G

Matrix Form of Collision and Quasilinear Operators for FASTFP and FASTFP-NC Codes

The Fokker-Planck and quasilinear equation is solved by the FASTFP and FASTFP-NC codes on a grid in (p, ξ_0) space. From Section 4.2, the Fokker-Planck equation is solved in the discretized form $f_{i,j} = f(p = p_i, \xi_0 = \xi_j)$ on an $N_1 \times N_2$ grid of points in momentum space:

$$p_i = (i-1)\Delta p, \qquad i = 1, ..., N_1, \tag{G.1}$$

$$\xi_j = -1 + (j-2)\Delta\xi, \qquad j = 2, ..., N_2 - 1, \tag{G.2}$$

where

$$\Delta p = \frac{p_{max}}{(N_1 - 1)}, \qquad \Delta\xi = \frac{2}{(N_2 - 2 - 1)}.$$

(Note that in this section we use the notation $\xi_0 = \xi_j$ rather than ξ_{0j}, to reduce the number of subscripts and make the expressions easier to read.)

The general form of the Fokker-Planck equation solved in these codes is

$$\{C(f)\} + \{Q(f)\} = S \tag{G.3}$$

where the unknown distribution function f (either f_0 or g) is independent of θ. We

solve the problem by a relaxation method, introducing a time-like variable τ and replacing (G.3) with

$$\frac{\partial f}{\partial \tau} + \{C(f)\} + \{Q(f)\} = S. \tag{G.4}$$

In discretized form, this problem is solved as a matrix equation (4.12)

$$\frac{\mathbf{f}(\tau + \Delta\tau) - \mathbf{f}(\tau)}{\Delta\tau} - \bar{M} \cdot \mathbf{f}(\tau + \Delta\tau) = \mathbf{S} \tag{G.5}$$

where $\mathbf{f}$ and $\mathbf{S}$ are the distribution and source as vectors, and $\bar{M}$ is a matrix form of the collision and quasilinear operators.

We let the distribution evolve in "time" (τ) until it converges. There are a number of possible convergence criteria we could use (e.g. the vector norm of $\mathbf{f}$), but since we are ultimately seeking the current we use a criterion based on the parallel flux of electrons $\Gamma_{e,\|}(\mathbf{f})$: when

$$\frac{\Gamma_{e,\|}(\tau + \Delta\tau) - \Gamma_{e,\|}(\tau)}{\Delta\tau} < \varepsilon \tag{G.6}$$

for a small value ε, the code stops.

The bounce-averaged collision operator in (p, ξ_0) coordinates is (4.2), and the bounce-averaged quasilinear operators for lower-hybrid and electron-cyclotron waves respectively are (4.3) and (4.6). Using finite differencing for the derivatives we write the operators in the form of a matrix:

$$\begin{aligned} \{C(f)\} &= \bar{M}^{(C)} \cdot \mathbf{f} \\ \{Q(f)\} &= \bar{M}^{(LH)} \cdot \mathbf{f} + \bar{M}^{(EC)} \cdot \mathbf{f} \end{aligned} \tag{G.7}$$

Each matrix $\bar{M}$ has 9 diagonals since the partial derivatives in momentum space at each grid point are calculated using the eight "neighboring" grid points: each matrix $\bar{M}$ can be written in the form

$$\begin{aligned} \bar{M} \cdot \mathbf{f} = {} & a_{ij} f_{i,j-1} + b_{ij} f_{i-1,j} + c_{ij} f_{i,j} \\ & + d_{ij} f_{i+1,j} + e_{ij} f_{i,j+1} + am_{ij} f_{i-1,j-1} \end{aligned}$$

$$+ ap_{ij} f_{i+1,j-1} + em_{ij} f_{i-1,j+1} + ep_{ij} f_{i+1,j+1}. \tag{G.8}$$

Collision Operator

The bounce averaged collision operator consists of a linear differential operator corresponding to $\{C(f, f_M) + C(f, f_i)\}$, which was derived in (E.10) and (E.12), and an integral operator corresponding to $\{C(f_M, f)\}$ in (E.19). The former is written as a matrix form $\bar{M}^{(C)} \cdot \mathbf{f}$ and the latter is included as a source term in the Fokker-Planck equation $\mathbf{S}^{(C)}$, as we show below.

The diagonal elements of $\bar{M}^{(C)}$ in the form of (G.8) are

$$\begin{aligned}
p_i^2\, a_{ij} &= -(1-\xi_{j-1/2}^2)\left(1-\frac{\Delta_{j-1/2}}{\xi_{j-1/2}^2}\right)\frac{B_{ti}}{(\Delta\xi)^2} \\
&\quad + (1-\xi_j^2)\frac{\lambda_{j+1/2}-\lambda_{j-1/2}}{(\Delta\xi)\lambda_j}\left(1-\frac{\Delta_j}{\xi_j^2}\right)\frac{B_{ti}}{(\Delta\xi)} && \text{(G.9)} \\
p_i^2\, b_{ij} &= \frac{p_{i-1/2}^2}{(\Delta p)}\left(-\frac{A_{i-1/2}}{(\Delta p)} + F_{i-1/2}\delta_{i-1/2}\right) && \text{(G.10)} \\
p_i^2\, c_{ij} &= -\frac{p_{i+1/2}^2}{(\Delta p)}\left(-\frac{A_{i+1/2}}{(\Delta p)} + F_{i+1/2}\delta_{i+1/2}\right) + \frac{p_{i-1/2}^2}{(\Delta p)}\left(\frac{A_{i-1/2}}{(\Delta p)} + F_{i-1/2}(1-\delta_{i-1/2})\right) \\
&\quad + (1-\xi_{j-1/2}^2)\frac{B_{ti}}{(\Delta\xi)^2} + (1-\xi_{j+1/2}^2)\frac{B_{ti}}{(\Delta\xi)^2} && \text{(G.11)} \\
p_i^2\, d_{ij} &= -\frac{p_{i+1/2}^2}{(\Delta p)}\left(\frac{A_{i+1/2}}{(\Delta p)} + F_{i+1/2}(1-\delta_{i+1/2})\right) && \text{(G.12)} \\
p_i^2\, e_{ij} &= -(1-\xi_{j+1/2}^2)\left(1-\frac{\Delta_{j+1/2}}{\xi_{j+1/2}^2}\right)\frac{B_{ti}}{(\Delta\xi)^2} \\
&\quad - (1-\xi_j^2)\frac{\lambda_{j+1/2}-\lambda_{j-1/2}}{(\Delta\xi)\lambda_j}\left(1-\frac{\Delta_j}{\xi_j^2}\right)\frac{B_{ti}}{(\Delta\xi)} && \text{(G.13)}
\end{aligned}$$

and the matrix elements am_{ij}, ap_{ij}, em_{ij}, and ep_{ij} are all zero for the collision operator. We make use of the notation $i \pm 1/2$ and $j \pm 1/2$ for the intermediate points on the (p_i, ξ_j) grid:

$$\begin{aligned}
p_{i\pm1/2} &= \frac{1}{2}(p_i + p_{i\pm1}) \\
\xi_{j\pm1/2} &= \frac{1}{2}(\xi_j + \xi_{j\pm1}).
\end{aligned}$$

The quantities A_i, F_i, and B_{ti} are the values of the potentials $A(p), F(p)$, and $B_t(p)$ (see Appendix A) at p_i. The quantities λ_j and Δ_j are (2.65) and (E.13) respectively, at the grid point ξ_j.

The discretization of f at the half-grid points $p_{i\pm1/2}$ is performed in the manner of Chang and Cooper [97] to take into account the expected exponential decay of f in p. They perform the averaging of f_i and f_{i+1} according to

$$f_{i+1/2,j} = (1 - \delta_{i+1/2}) f_{i+1,j} + \delta_{i+1/2} f_{i,j}. \tag{G.14}$$

The values of $\delta_{i+1/2}$ are given by Chang and Cooper [97] to be

$$\delta_{i+1/2} = \frac{1}{W} - \frac{1}{\exp(W) - 1} \tag{G.15}$$

where W equals Δp times the expected value of $|d \ln f/dp|$. We calculate this by assuming a Maxwellian form of f [98, 97]. The value of $\delta_{i+1/2}$ varies from $1/2$ to 0 as W varies from 0 to infinity.

We also consider the source term of the collision operator corresponding to $\{C(f_M, f)\}$ or (E.19). This is

$$S_{ij}^{(C)} = -H(|\xi_j| - \xi_{0T}) \ \frac{1}{\lambda_j} \sqrt{\frac{1+\epsilon}{1-\epsilon}} (I_1)_i \xi_j \tag{G.16}$$

where $(I_1)_i = I_1(f_M, f_{l=1,0})$ at p_i is given by (A.8), and we find $f_{l=1,0}$ by numerically integrating:

$$f_{l=1} \equiv \frac{3}{2} \int_{-1}^{1} \xi f \, d\xi = \frac{3}{2} \sum_{j=2}^{N_2-1} \xi_j f_{i,j} (\Delta\xi). \tag{G.17}$$

Lower Hybrid Quasilinear Operator

The discretized lower hybrid operator $\bar{M}^{(LH)}$ is calculated from $\{Q(f)\}$ in (4.3) and has the following elements:

$$\begin{aligned} p_i^2 \, a_{ij} \ = \ & -(1 - \xi_{j-1/2}^2)^2 \frac{D_{i,j}^{(LH)} + D_{i,j-1}^{(LH)}}{2} \frac{1}{(\Delta\xi)^2} \\ & + \xi_j (1 - \xi_j^2) \left(D_{i,j}^{(LH)} + p_i \frac{D_{i+1,j}^{(LH)} - D_{i-1,j}^{(LH)}}{2(\Delta p)} \right) \frac{1}{2(\Delta\xi)} \end{aligned} \tag{G.18}$$

$$
\begin{aligned}
p_i^2\ b_{ij} &= -p_{i-1/2}^2\ \xi_j^2 \frac{D_{i,j}^{(LH)} + D_{i-1,j}^{(LH)}}{2} \frac{1}{(\Delta p)^2} \\
&\quad + p_i \frac{\xi_{j+1}(1-\xi_{j+1}^2) D_{i,j+1}^{(LH)} - \xi_{j-1}(1-\xi_{j-1}^2) D_{i,j-1}^{(LH)}}{4(\Delta p)(\Delta \xi)} \qquad \text{(G.19)} \\
p_i^2\ c_{ij} &= \xi_j^2 \left(p_{i+1/2}^2 \frac{D_{i+1,j}^{(LH)} + D_{i,j}^{(LH)}}{2} + p_{i-1/2}^2 \frac{D_{i,j}^{(LH)} + D_{i-1,j}^{(LH)}}{2} \right) \frac{1}{(\Delta p)^2} \qquad \text{(G.20)} \\
p_i^2\ d_{ij} &= -p_{i+1/2}^2\ \xi_j^2 \frac{D_{i,j}^{(LH)} + D_{i+1,j}^{(LH)}}{2} \frac{1}{(\Delta p)^2} \\
&\quad - p_i \frac{\xi_{j+1}(1-\xi_{j+1}^2) D_{i,j+1}^{(LH)} - \xi_{j-1}(1-\xi_{j-1}^2) D_{i,j-1}^{(LH)}}{4(\Delta p)(\Delta \xi)} \qquad \text{(G.21)} \\
p_i^2\ e_{ij} &= -(1-\xi_{j+1/2}^2)^2 \frac{D_{i,j}^{(LH)} + D_{i,j+1}^{(LH)}}{2} \frac{1}{(\Delta \xi)^2} \\
&\quad - \xi_j(1-\xi_j^2) \left(D_{i,j}^{(LH)} + p_i \frac{D_{i+1,j}^{(LH)} - D_{i-1,j}^{(LH)}}{2(\Delta p)} \right) \frac{1}{2(\Delta \xi)} \qquad \text{(G.22)} \\
p_i^2\ am_{ij} &= -p_i^2\ ap_{ij} = -p_i^2\ em_{ij} = p_i^2\ ep_{ij} \\
&= -p_i\ \xi_j(1-\xi_j^2) D_{i,j}^{(LH)} \frac{1}{2(\Delta p)(\Delta \xi)} \qquad \text{(G.23)}
\end{aligned}
$$

In each case $D_{i,j}^{(LH)}$ is the value of the diffusion coefficent at (p_i, ξ_j), given

$$
\begin{aligned}
D_{i,j}^{(LH)} &= D_0, \qquad v_1 < v_{\|,0} < v_2 \\
&= 0 \qquad\quad v_\| < v_1, v_{\|,0} > v_2.
\end{aligned}
$$

Electron Cyclotron Quasilinear Operator

We note that $\{Q(f)\}$ in (4.6) involves quantities evaluated at the location θ_b at which the EC beam crosses the flux surface. So we first define $\Psi_b \equiv (1+\epsilon)/(1+\epsilon\cos\theta_b)$ and $\xi_{b,j} = \sqrt{1-\Psi_b(1-\xi_j^2)}$. We also use the shorthand $\hat{\Omega} \equiv n\gamma\Omega/\omega$ which is independent of p and ξ.

The discretized lower hybrid operator $\bar{M}^{(EC)}$ has the following elements:

$$
\begin{aligned}
p_i^2\ \lambda_j a_{ij} &= -(1-\xi_j^2) \frac{D^{(EC)}{}_{i+1,j} - D_{i-1,j}^{(EC)}}{2(\Delta p)\,|\xi_{b,j}|} \frac{\hat{\Omega} + \gamma_i \Psi_b(\xi_j^2 - 1)}{\xi_j} \frac{1}{2(\Delta \xi)} \\
&\quad + (1-\xi_j^2) \frac{D_{i,j}^{(EC)}}{|\xi_{b,j}|} \frac{p_i (v_{Te}/c)^2 \Psi_b (1-\xi_j^2)}{\gamma_i \xi_j} \frac{1}{2(\Delta \xi)}
\end{aligned}
$$

$$- (1-\xi_{j-1/2}^2)\frac{D_{i,j-1}^{(EC)}+D_{i,j}^{(EC)}}{2\left|\xi_{b,j-1/2}\right|}\frac{[\hat{\Omega}+\gamma_i\Psi_b(\xi_{j-1/2}^2-1)]^2}{\gamma_i\Psi_b p_i\ \xi_{j-1/2}^2}\frac{1}{(\Delta\xi)^2} \quad \text{(G.24)}$$

$$\begin{aligned} p_i^2\ \lambda_j b_{ij} &= -(1-\xi_j^2)\frac{D^{(EC)}{}_{i,j+1}-D_{i,j-1}^{(EC)}}{2(\Delta\xi)\left|\xi_{b,j}\right|}\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_j^2-1)}{\xi_j}\frac{1}{2(\Delta p)} \\ &\quad + D_{i,j}^{(EC)}\frac{1}{2(\Delta\xi)}\left[(1-\xi_{j+1}^2)\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_{j+1}^2-1)}{\xi_{j+1}\left|\xi_{b,j+1}\right|}\right. \\ &\quad \left. -(1-\xi_{j-1}^2)\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_{j-1}^2-1)}{\xi_{j-1}\left|\xi_{b,j-1}\right|}\right]\frac{1}{2(\Delta p)} \\ &\quad - p_{i-1/2}(1-\xi_j^2)\gamma_{i-1/2}\Psi_b\frac{D_{i,j}^{(EC)}+D_{i-1,j}^{(EC)}}{2\left|\xi_{b,j}\right|}\frac{1}{(\Delta p)^2} \end{aligned} \quad \text{(G.25)}$$

$$\begin{aligned} p_i^2\ \lambda_j c_{ij} &= p_{i-1/2}(1-\xi_j^2)\gamma_{i-1/2}\Psi_b\frac{D_{i,j}^{(EC)}+D_{i-1,j}^{(EC)}}{2\left|\xi_{b,j}\right|}\frac{1}{(\Delta p)^2} \\ &\quad + p_{i+1/2}(1-\xi_j^2)\gamma_{i+1/2}\Psi_b\frac{D_{i,j}^{(EC)}+D_{i+1,j}^{(EC)}}{2\left|\xi_{b,j}\right|}\frac{1}{(\Delta p)^2} \\ &\quad + (1-\xi_{j-1/2}^2)\frac{D_{i,j-1}^{(EC)}+D_{i,j}^{(EC)}}{2\left|\xi_{b,j-1/2}\right|}\frac{[\hat{\Omega}+\gamma_i\Psi_b(\xi_{j-1/2}^2-1)]^2}{\gamma_i\Psi_b p_i\ \xi_{j-1/2}^2}\frac{1}{(\Delta\xi)^2} \\ &\quad + (1-\xi_{j+1/2}^2)\frac{D_{i,j+1}^{(EC)}+D_{i,j}^{(EC)}}{2\left|\xi_{b,j+1/2}\right|}\frac{[\hat{\Omega}+\gamma_i\Psi_b(\xi_{j+1/2}^2-1)]^2}{\gamma_i\Psi_b p_i\ \xi_{j+1/2}^2}\frac{1}{(\Delta\xi)^2} \end{aligned} \quad \text{(G.26)}$$

$$\begin{aligned} p_i^2\ \lambda_j d_{ij} &= (1-\xi_j^2)\frac{D^{(EC)}{}_{i,j+1}-D_{i,j-1}^{(EC)}}{2(\Delta\xi)\left|\xi_{b,j}\right|}\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_j^2-1)}{\xi_j}\frac{1}{2(\Delta p)} \\ &\quad - D_{i,j}^{(EC)}\frac{1}{2(\Delta\xi)}\left[(1-\xi_{j+1}^2)\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_{j+1}^2-1)}{\xi_{j+1}\left|\xi_{b,j+1}\right|}\right. \\ &\quad \left. -(1-\xi_{j-1}^2)\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_{j-1}^2-1)}{\xi_{j-1}\left|\xi_{b,j-1}\right|}\right]\frac{1}{2(\Delta p)} \\ &\quad - p_{i+1/2}(1-\xi_j^2)\gamma_{i+1/2}\Psi_b\frac{D_{i,j}^{(EC)}+D_{i+1,j}^{(EC)}}{2\left|\xi_{b,j}\right|}\frac{1}{(\Delta p)^2} \end{aligned} \quad \text{(G.27)}$$

$$\begin{aligned} p_i^2\ \lambda_j e_{ij} &= (1-\xi_j^2)\frac{D^{(EC)}{}_{i+1,j}-D_{i-1,j}^{(EC)}}{2(\Delta p)\left|\xi_{b,j}\right|}\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_j^2-1)}{\xi_j}\frac{1}{2(\Delta\xi)} \\ &\quad - (1-\xi_j^2)\frac{D_{i,j}^{(EC)}}{\left|\xi_{b,j}\right|}\frac{p_i(v_{Te}/c)^2\Psi_b(1-\xi_j^2)}{\gamma_i\xi_j}\frac{1}{2(\Delta\xi)} \\ &\quad - (1-\xi_{j+1/2}^2)\frac{D_{i,j+1}^{(EC)}+D_{i,j}^{(EC)}}{2\left|\xi_{b,j+1/2}\right|}\frac{[\hat{\Omega}+\gamma_i\Psi_b(\xi_{j+1/2}^2-1)]^2}{\gamma_i\Psi_b p_i\ \xi_{j+1/2}^2}\frac{1}{(\Delta\xi)^2} \end{aligned} \quad \text{(G.28)}$$

$$p_i^2\ \lambda_j am_{ij} = -p_i^2\ \lambda_j ap_{ij} = -p_i^2\ \lambda_j em_{ij} = p_i^2\ \lambda_j ep_{ij}$$

$$= (1-\xi_j^2)\frac{D^{(EC)}{}_{i,j}}{|\xi_{b,j}|}\frac{\hat{\Omega}+\gamma_i\Psi_b(\xi_j^2-1)}{\xi_j}\frac{1}{4(\Delta p)(\Delta\xi)} \tag{G.29}$$

In each case $D_{i,j}^{(EC)}$ is the value of the diffusion coefficent at (p_i, ξ_j), and $\theta = \theta_b$:

$$D_{i,j}^{(EC)} = p_i^2\Psi_b(1-\xi_j^2)D_0\exp\left[-\left(\frac{\gamma\omega - n\Omega_0}{p_\parallel/m} - k_{\parallel,0}\right)_b^2/(\Delta k_\parallel)^2\right]. \tag{G.30}$$

Bibliography

[1] M. Shoucri and I. Shkarofsky. *Comp. Phys. Comm* **82**, 287 (1994).

[2] M.Y. Hsiao, D.A. Ehst, and K Evans. *Nuc. Fusion* **29**, 49 (1989).

[3] A. Bers and Ram A.K. In *Proc. of the IAEA Tech. Meeting on Fast Wave Current in Reactor Scale Tokamaks*, Arles, France, September 1991.

[4] A.A. Galeev and R.Z. Sagdeev. *Zh. Eksp. Teor. Fiz.* **53**, 348 (1967). [Sov. Phys.-JETP **26**, 233 (1968)].

[5] R.J. Bickerton, J.W. Connor, and J.B. Taylor. *Nature Phys. Sci.* **229**, 110 (1971).

[6] M.C. Zarnstorff, M.G. Bell, and M. Bitter. *Phys. Rev. Lett.* **60**, 1306 (1988).

[7] M.N. Rosenbluth, R.D. Hazeltine, and F.L. Hinton. *Phys. Fluids* **15**, 116 (1972).

[8] F.L. Hinton and R.D. Hazeltine. *Rev. Mod. Phys.* **48**, 239 (1976).

[9] S.P. Hirshman and D.J. Sigmar. *Nuc. Fusion* **21**, 1079 (1981).

[10] C.E. Kessel. *Nuc. Fusion* **34**, 1221 (1994).

[11] T.M. Jr. Antonsen and K. Yoshioka. *Phys. Fluids* **29**, 2235 (1986).

[12] A. Bécoulet, V. Basiuk, E. Joffrin, and G.T. Hoang. *Plasma Phys. and Cont. Fusion* **40**, A157 (1998).

[13] M. Taguchi. *Proc. 21st Eur. Conf. on Cont. Fusion and Plasma Phys.*, 1142, (1994).

[14] T. Hellsten, J. Carlsson, and L.G. Eriksson. *Phys. Rev. Lett.* **74**, 3612 (1995).

[15] J.A. Heikkinen and S.A. Sipila. *Nuc. Fusion* **37**, 835 (1997).

[16] P. Helander, R.J. Hastie, and J.W. Connor. *Phys. Plasmas* **4**, 3211 (1997).

[17] T.H. Stix. *The Theory of Plasma Waves.* McGraw-Hill, Inc., New York, 1962.

[18] T.H. Stix. *Waves in Plasmas.* Amer. Inst. of Physics, New York, 1992.

[19] W.P. Allis, S.J. Buchsbaum, and A. Bers. *Waves in Anisotropic Plasmas.* MIT Press, Cambridge, Massachusetts, 1963.

[20] L.D. Landau. *J. Phys. U.S.S.R.* **10**, 25 (1946).

[21] C.F. Kennel and F. Englemann. *Phys. Fluids* **9**, 2377 (1966).

[22] I. Lerche. *Phys. Fluids* **11**, 1720 (1968).

[23] I.B. Bernstein and D.C. Baxter. *Phys. Fluids* **24**, 108 (1981).

[24] H.L. Berk. *J. Plasma Phys.* **20**, 205 (1978).

[25] G.D. Kerbel and M.G. McCoy. *Phys. Fluids* **28**, 3629 (1985).

[26] G.W. Hammett. *Fast Ion Studies of Ion Cyclotron Heating in the PLT Tokamak.* PhD thesis, Princeton University, 1986.

[27] J. Killeen, G.D. Kerbel, M.G. McCoy, and A.A. Mirin. *Computational Methods for Kinetic Models of Magnetically Confined Plasmas.* Springer-Verlag, New York, 1986.

[28] D.J.H Wort. *Plasma Phys.* **13**, 258 (1971).

[29] N.J. Fisch. *Phys. Rev. Lett.* **41**, 873 (1978).

[30] A. Bers. In *Proc. 3rd Topical Conf. on RF Plasma Heating*, Pasadena, California, 1978.

[31] A. Bers and N.J. Fisch. In *Proc. 3rd Topical Conf. on RF Plasma Heating*, Pasadena, California, 1978.

[32] R.R. Parker. *Massachusetts Institute of Technology, Research Laboratory of Electronics Report QPR* **102**, 97 (1971).

[33] F. Troyon and F.W. Perkins. In *Proc. 2nd Topical Conf. on RF Plasma Heating*, Lubbock, Texas, 1974.

[34] V.E. Golant. *Zh. Tekh. Fiz.* **41**, 2492 (1971). [Sov. Phys. Tech. Phys. **16**, 1980 (1972)].

[35] P.M. Bellan and M. Porkolab. *Phys. Fluids* **19**, 995 (1976).

[36] N.J. Fisch and A.H. Boozer. *Phys. Rev. Lett.* **45**, 720 (1980).

[37] T. Ohkawa. *Gneral Atomics Company Report No. A13847* (1976).

[38] J.G. Cordey, T. Edlington, and D.F.H. Start. *Plasma Phys. and Cont. Fusion* **24**, 73 (1982).

[39] M. Taguchi. *J. Phys. Soc. Jpn.* **52**, 2035 (1983).

[40] A. Ferriera, M.R. O'Brien, and D.F.H. Start. *Plasma Phys. and Cont. Fusion* **26**, 1565 (1984).

[41] K. Yoshioka and T.M. Jr. Antonsen. *Nuc. Fusion* **26**, 839 (1986).

[42] N.J. Fisch. *Rev. Mod. Phys.* **59**, 175 (1987).

[43] I.E. Tamm and A.D. Sakharov. Theory of magnetic thermonuclear fusion reactor. *Plasma Phys. and Problems of Cont. Fusion* **1**, 3 (1958).

[44] L.A. Artsimovich, G.A. Bobrovsky, E.P. Gorbunov, D.P. Ivanov, V.D. Kirillov, E.I. Kuznetsov, S.V. Mirnov, M.P. Petrov, K.A. Razumova, Strelkov V.S., and Shcheglov D.A. *Nuclear Fusion Special Suppliment, IAEA Conf., Novosibirsk*, 17 (1969).

[45] J. Sheffield. *Rev. Mod. Phys.* **66**, 1015 (1994).

[46] B.B. Kadomstev and V.D. Shafranov. In *Proc. 4th Int. Conf. on Plasma Phys. and Cont. Nuc. Fusion, Madison, WI*, 1971. Nuclear Fusion Special Suppliment p.209.

[47] F. Troyon and R. Gruber. *Phys. Lett.* **110A**, 29 (1985).

[48] C. Kessel, J. Manickam, G. Rewoldt, and W.M. Tang. *Phys. Rev. Lett.* **72**, 1212 (1994).

[49] F. Najmabdai and The ARIES Team. Assessment of tokamak plasma operation modes as fusion power plants: the Starlite study. In *Proc. 16th Int. Conf. Fusion Energy Agency, Montreal, Canada*, Vienna, 1997. IAEA.

[50] F. Najmabadi and The ARIES Team. *Fusion Engineering and Design* **38**, 3 (1997).

[51] M. Kikuchi. *Plasma Phys. and Cont. Fusion* **35**, B39 (1993).

[52] M.D. Kruskal and R.M. Kulsrud. *Phys. Fluids* **1**, 265 (1958).

[53] V.D. Shafranov. Plasma equilibrium in a magnetic field. In M.A. Leontovich, editor, *Reviews of Plasma Physics*, volume 2, pages 103–151. Consultants Bureau, New York, 1966.

[54] H. Grad. *Phys. Fluids* **10**, 137 (1967).

[55] A.I. Morozov and L.S. Solov'ev. Motion of charged particles in electromagnetic fields. In M.A. Leontovich, editor, *Reviews of Plasma Physics*, volume 2, pages 201–297. Consultants Bureau, New York, 1966.

[56] H. Alfvén. *Cosmical Electrodynamics.* Oxford University Press, Fairlawn, N.J., 1950.

[57] L.D. Landau. *Phys. Z. Sowjetunion* **10**, 154 (1936).

[58] S. Chandrasekhar. *Astrophys. Journal* **97**, 255 (1943).

[59] C.F.F. Karney and N.J Fisch. *Phys. Fluids* **28**, 116 (1985).

[60] S.I. Braginskii. Transport processes in a plasma. In M.A. Leontovich, editor, *Reviews of Plasma Physics*, volume 1, pages 205–311. Consultants Bureau, New York, 1965.

[61] B.J. Braams and C.F.F. Karney. *Phys. Fluids* **B1**, 1355 (1989).

[62] M.N. Rosenbluth, W.M. MacDonald, and D.L. Judd. *Phys. Rev.* **107**, 1 (1957).

[63] S.T. Beliaev and G.I. Budker. *Dokl. Akad. Nauk SSSR* **107**, 807 (1956).

[64] A.A. Galeev and R.Z. Sagdeev. Theory of neoclassical diffusion. In M.A. Leontovich, editor, *Reviews of Plasma Physics*, volume 7. Consultants Bureau, New York, 1979.

[65] J.G. Cordey, C.D. Challis, and P.M. Stubberfield. *Plasma Phys. and Cont. Fusion* **30**, 1625 (1988).

[66] P.L. Colestock and J.L. Kulp. *IEEE Trans. Plasma Sci.* **PS-8**, 71 (1980).

[67] P.T. Bonoli and R.C. Englade. *Phys. Fluids* **29**, 2937 (1986).

[68] K. Theilhaber. *Nuc. Fusion* **22**, 363 (1982).

[69] G.P. Leclert, C.F.F. Karney, A. Bers, and D.J. Kaup. *Phys. Fluids* **22**, 1545 (1979).

[70] A. Bers. In *3rd Symposium on Plasma Heating in Toroidal Devices*, Varenna-Como, Italy, 1976.

[71] A.A. Vedenov. In M.A. Leontovich, editor, *Reviews of Plasma Physics*, volume 3. Consultants Bureau, New York, 1967.

[72] C.F.F. Karney and N.J. Fisch. *Phys. Fluids* **22**, 1817 (1979).

[73] V. Fuchs, R.A. Cairns, M. Shoucri, K. Hizanidis, and A. Bers. *Phys. Fluids* **28**, 3619 (1985).

[74] T.M. Jr. Antonsen and K.R. Chu. *Phys. Fluids* **25**, 1295 (1982).

[75] S.P. Hirshman. *Phys. Fluids* **23**, 1238 (1980).

[76] L. Spitzer and R. Härm. *Phys. Rev.* **89**, 977 (1953).

[77] M. Porkolab. Advanced tokamak operation of Alcator C-Mod with lower hybrid current drive profile control. Presented at the Alcator C-Mod Program Advisory Committee Meeting, MIT, Feb. 7, 1999.

[78] V.S. Chan, S.C. Chiu, J.Y. Hsu, and S.K. Wong. *Nuc. Fusion* **22**, 787 (1982).

[79] R.A. Cairns and C. Lashmore-Davies. *Plasma Phys. and Cont. Fusion* **28**, 1539 (1986).

[80] M. Shoucri and I. Shkarofsky. *CCFM Report 467* (1996).

[81] M. Shoucri, I. Shkarofsky, and Y. Peysson. *CCFM Report 465* (1996).

[82] I. Shkarofsky. *Phys. Plasmas* **4**, 2464 (1997).

[83] Y. Peysson and M. Shoucri. *Comp. Phys. Comm.* **109**, 55 (1998).

[84] M. Shoucri, I. Shkarofsky, Y. Peysson, Y. Demers, A. Côté, and C. Côté. *CCFM Report 479* (1998).

[85] Robert D. Richtmyer and K.W. Morton. *Difference Methods for Initial-Value Problems.* Interscience Publishers, New York, 1957.

[86] R. Courant, K.O. Friedrichs, and H. Lewy. *Math. Ann.* **100**, 32 (1928).

[87] H.J. Stone. *SIAM Journal on Numerical Analysis* **5**, 530 (1968).

[88] B.B Kadomtsev. *Tokamak Plasma: A Complex Physical System.* Plasma Physics Series. Institute of Physics Publishing, Bristol, UK and Philadelphia, PA, 1992. Translation Editor: E.W. Laing.

[89] L. Demeio and F. Engelmann. *Plasma Phys. and Cont. Fusion* **28**, 1851 (1986).

[90] M.R. O'Brien, M. Cox, and D.F.H. Start. *Nuc. Fusion* **26**, 1625 (1986).

[91] S.P. Hirshman. *Phys. Fluids* **31**, 3150 (1988).

[92] O. Sauter, C. Angioni, and Y.R. Lin-Liu. *Phys. of Plasmas* **6**, 2834 (1999).

[93] T.C. Luce. Electron cyclotron current drive in DIII-D. In Stefano Bernabei, editor, *Proc. of 13th Topical Conference on Radio Frequency Power in Plasmas*, Annapolis, MD, 1999.

[94] G.T. Hoang. *Nuc. Fusion* **38**, 117 (1998).

[95] K.C. Shaing, R.D. Hazeltine, and M.C. Zarnstorff. *Phys. Plasmas* **4**, 771 (1997).

[96] I.S. Gradshteyn and I.M. Ryzhik. *Table of Integrals, Series, and Products.* Academic Press, New York, 1965.

[97] J.S. Chang and G. Cooper. *J. Comp. Phys.* **6**, 1 (1970).

[98] C.F.F. Karney. *Comput. Phys. Rep.* **4**, 183 (1986).