THREE-DIMENSIONAL MODELING OF PLASMA TRANSPORT IN THE HIDRA STELLARATOR

BY

STEVEN W. MARCINKO

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Nuclear, Plasma, and Radiological Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Master's Committee:

Professor Davide Curreli, Advisor
Professor David Ruzic

Abstract

In the edge region of modern nuclear fusion experiments, the interactions between edge plasmas and the materials which ultimately confine them have become increasingly more important as device sizes and powers trend upwards. Devices such as MPEX and PISCES are built to investigate these interactions at the high energies and particle fluxes ejected by transient edge disruption events, but are ultimately linear devices. To extend the diagnostic environments of the greater fusion community to include a dedicated toroidal plasma-material interface (PMI) device, the Hybrid Illinois Device for Research and Applications (HIDRA) has been dedicated. HIDRA is an l=2, m=5 classical stellarator originally built in in the 1970's for RF heating studies. Its most recent users at the Max Planck Institute for Plasma Physics ran the device as WEGA to test heating schemes and train personnel for the recently-completed W-7X advanced stellarator, after which it was gifted to the University of Illinois at Urbana-Champaign.

To improve the theoretical models of the PMI environment in the edge region of HIDRA, computational tools which apply these models are required. Many simulation tools currently in use focus on tokamak magnetic geometries or high-power, fully-ionized devices, necessitating the creation of an integrated suite of codes to handle partial ionization with more disparate operational power conditions and classical diffusivity unique to HIDRA in contemporary devices. To this end, `HIDRAmod` has been created, encompassing the existing coupled codes Edge Monte Carlo 3D (EMC3) and EIRENE to solve the plasma and neutral transport equations. `FIELDLINES` has been used in the creation of a field-aligned tetrahedral mesh generation utility `TORMESH`, also integrated into `HIDRAmod`. In addition to these established codes, utilities for calculating the limiting surface and tetrahedral mesh intersection and for post-processing have been written. EMC3 has been altered to include a Bohm-like diffusivity to handle the uniquely diffusive plasma in a self-consistent manner.

Preceding operational data on the device, simulations have been run under the context of bounding the incident heat and particle fluxes onto the limiting surface into a region of confidence based on parameters from previous operational campaigns. An outboard midplane limiter, inboard midplane limiter, and 'trench' limiter (along the bottom of the torus) were tested with RF input to core-edge power deposition efficiencies of 10-50% for a 26 kW 2.45 GHz combined RF input discharge. Axial magnetic field strengths of 87.5 mT and 0.5 T were analyzed, corresponding to two heating schemes tested at WEGA. Electron temperatures and densities were seen to match previous WEGA results of 8-10 eV and 1-3 10^{12} cm^{-3} in the edge region respectively. With these results, 26 kW of operational power translates to heat fluxes of up to 1 MW m^{-2} on the inboard limiter, up to 0.2 MW m^{-2} on the outboard limiter, and up to 0.15 MW m^{-2} on the trench limiter. Particle fluxes have been similarly bound by upper limits of 4.7 10^{22}m^{-2} s^{-1}, 5 10^{21} m^{-2} s^{-1}, and 5.6 10^{21} m^{-2} s^{-1} for the inboard, outboard and trench limiters respectively. Scaling laws for peak electron temperature, Bohm-like diffusivity, and heat and particle fluxes have been calculated for both low- and high-field discharges; peak electron temperatures, particle diffusivity, and heat fluxes at the outboard limiter were seen to follow approximately a power-law of type $f(P_{RF}) \propto aP_{RF}^{b}$, with typical exponents in the range $b \sim 0.55 - 0.60$. Higher magnetic fields have the tendency to linearize the heat flux dependence upon the RF power, with exponents in the range of $b \sim 0.75$. Particle fluxes on the outboard limiter are seen to saturate first, and then slightly decline for RF powers in excess of 120 kW in the low-field case and 180 kW in the high-field case.

Finally, extensions and applications of `HIDRAmod` are examined, including a path to a self-consistent full-device model and potential optimization strategies which may be employed to enhance fluxes arriving at the limiting surfaces.

Eir

Acknowledgments

I would like to thank the following individuals, in no particular order (which I would like to believe I have rigorously enforced):

Shane, for his initial work on magnetic meshes and `FIELDLINES` parsing.

Dr. Curreli, for finding something worth spending hundreds of thousands of dollars on. Hopefully some day, I'll find it too.

Rinat, for his insight in statistics and MPI programming.

Peter, Mike, and Filipe, for their moral support.

Dr. Ruzic, for his faultlessly positive attitude.

My parents.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

HIDRA	Hybrid Illinois Device for Research and Applications
WEGA	Wendelstein Experiment in Greifswald fr die Ausbildung
NCSX	National Compact Stellarator Experiment
NSTX	National Spherical Tokamak Experiment
HSX	Helically Symmetric Stellarator
LHD	Large Helical Device
CPMI	Center for Plasma-Material Interactions
PMI	Plasma-Material Interaction
PFC	Plasma-Facing Component
LCMS	Last Closed Magnetic Surface
LCFS	Last Closed Flux Surface
RFLM	Reversible Field Line Mapping
RF	Radio Frequency
EBW	Electron Bernstein Wave
ECRH	Electron Cyclotron Resonance Heating
ICRH	Ion Cyclotron Resonance Heating

List of Symbols

T_e	Electron Temperature
T_i	Ion Temperature
n_e	Electron Density
n_i	Ion Density
D_B	Bohm Diffusivity
B_0	Magnetic Field Magnitude at the Magnetic Axis
θ	Toroidal Angle
ϕ	Poloidal Angle
ι	Rotational Transform

Chapter 1

Introduction

1.1 Challenges on the Road to Fusion

The first commercial fusion reactor, capable of producing many times its own energy consumption in useful power while lacking the ecological consequences of fossil fuels and the long-term radioactivity of modern fission reactors will be a colossal shift in the viability of our species. The high energy density, low spatial footprint, and practically unlimited fuel can satisfy the energy needs of our species efficiently for millions of years. The development of such a device was originally planned to be completed many years ago, but in the course of scaling up small experiments into large devices many additional problems arose.

The fusion energy community has identified a number of challenging areas in the design, construction, and operation of fusion devices:

1. Obtaining funding for the construction and operation of large devices;
2. Disruption and transient mitigation;
3. Lower hybrid current drive efficiency;
4. Plasma material interactions;
5. Steady-state operation with high bootstrap current fractions;
6. A demonstration of burn control with fast alpha particles; and
7. Removal of thermalized alpha particles from the fusing volume.

Of these, the interactions between the edge plasma and the plasma facing components (PFCs) have historically had less emphasis. The elements in the PFCs, their crystalline structure, orientation relative to the field lines, and the plasma conditions at the surface all play a role both in

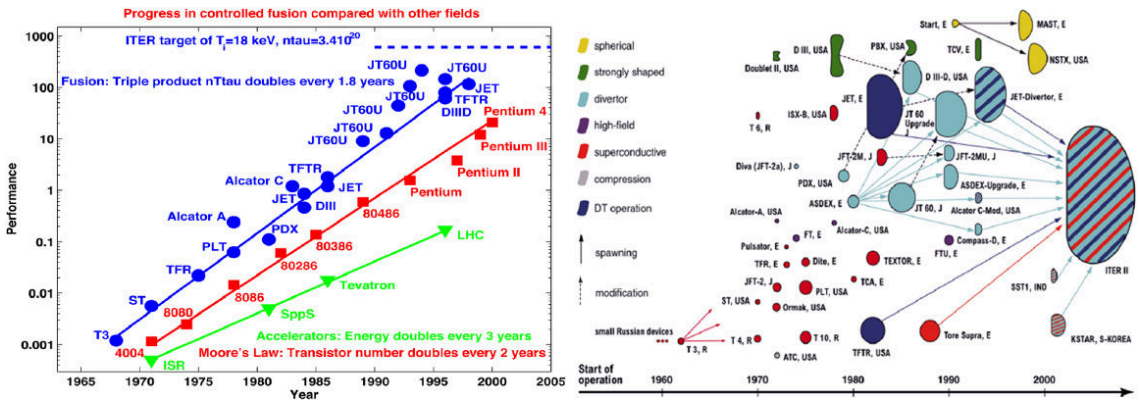


Figure 1.1: From [1], the exponential progress in fusion energy sciences is illustrated by a comparison to Moore's law for semiconductors and the increasing energy of particle accelerators. Also from [1], the increasing size of fusion devices over time is illustrated.

the emission of impurity species into the edge plasma and the recycling of primary plasma ions. Once in the edge the impurities may affect transport parameters or radiate away energy, resulting in changes to plasma temperature and density profiles that propagate back into the core. Recycled neutrals can shield the surface as a significant radiative power loss channel. The design of a commercial fusion reactor must be informed by such considerations.

1.1.1 Disruptions and Transients

Edge Localized Modes

In the pursuit of high-temperature, well-confined, dense plasmas, ever larger research devices were constructed. Exponentially-increasing (figure 1.1) plasma parameters demanded larger volumes and higher magnetic fields. The discovery [2] of an enhanced confinement regime in ASDEX, the H-mode, was accompanied by the discovery [3] of edge localized modes (ELMs), an instability in the plasma edge which substantially erodes confinement and sends high-energy plasma out of the fusing volume and towards the plasma-material interface. Shortly thereafter, other devices reaching the H-mode also began experiencing ELMs [4]. In ASDEX, the heat flux on the PFCs can jump from a nominal 5 MW/m^2 to 20 MW/m^2 [5] in moderate ELMs. If the PFCs are not able to withstand these confinement degradations, a device could potentially destroy itself, causing millions of dollars worth of damage. A commercial reactor at steady-state would need to sustain at least the steady-state energy flux, if not both the steady-state and ELM energy fluxes, for multiple

decades of operation to recuperate the large expected capital costs of the reactor. In an attempt to suppress ELMs, some mitigating techniques have been developed; small pellets of lithium may be injected [6] into the edge to trigger small ELMs before they can build into larger ones, or resonant magnetic perturbation (RMP) coils may be employed [7] to change the stability of the ELMs.

1.1.2 Plasma-Material Interface Integrity

Traditionally, the two parameters in the Lawson criterion (or the three in the plasma triple product, $n\tau_E T$) were seen as independent areas of research. Plasma transport was separately optimized to increase the confinement time, and guarding the plasma against pressure-driven MHD instabilities allowed for higher pressure. The development of heating schemes also took place separately, in devices such as WEGA [8] [9]. As device powers increased, it was realized that plasma-material interactions at the wall were becoming increasingly important to all three disciplines. Simultaneously, the discovery of ELMs and their potential to significantly degrade PFCs increased the urgency on fusion materials development, which was already building from the increased power fluxes across the last closed flux surface.

Many devices are currently available to study heat fluxes resulting from ELMs:

- The Material Plasma Exposure eXperiment (MPEX) at Oak Ridge National Lab has been designed to reach energy fluxes of 10 MW/m^2 in pulses by its second operational phase, and at steady-state by its third operational phase. [10]
- The Plasma Surface Interaction Experimental Facility (PICES-B) at UCSD can reach energy fluxes of up to 10 MW/m^2 at high particle fluxes in steady-state conditions. [11] [12]
- At UIUC, the Divertor Edge and Vapor shielding eXperiment (DEVeX) can produce energy fluxes of up to 45 MW/m^2 with a plasma lasting 100 microseconds. [13]

Although these devices can reach high energy fluxes representative of a predicted ITER discharge, they are linear machines and do not have a magnetic structure representative of a toroidal machine. The connection lengths (the linear distance along the field lines to the next physical surface intersection) which may vary substantially over the plasma profile in a stellarator are uniform

in a linear machine, and the circulation of released impurity species or recycled and re-ionized gases around the torus is not accounted for.

Most major fusion research devices were built with a focus on achieving higher plasma parameters T_e,n_e, and confinement time τ_E and have only a few materials as PFCs: tungsten, carbon, molybdenum, and beryllium PFCs are common, with the occasional lithium treatment to assist in achieving higher densities and temperatures in optimized confinement regimes. At the lower operational power of these machines compared to the expected commercial reactor power levels, it was expected that good PFC material choices would limit the influence of PMI on the bulk plasma. For example, the use of tungsten with a very high sputtering threshold was intended to prevent ions in the low-energy private flux region near the divertor from having a high enough energy to do damage. In the Large Helical Device (LHD) [14], PICES [15] [16], Alcator C-Mod [17], and other smaller chambers [18] [19], the creation of tungsten fuzz has been observed in helium bombardment with ion energies well below the conventional sputter yield. Tungsten fuzz tendrils can break off into the edge region and migrate to the core where they radiate away significant amounts of energy. Although PMI has recently been designated as an important research topic in the plasma community, most larger devices have correspondingly large price tags and an effort is made to keep them clean to hit their design parameters for plasma conditions; only a few larger toroidal devices including the Experimental Advanced Superconducting Tokamak (EAST) in China and the Lithium Tokamak Experiment (LTX) at the Princeton Plasma Physics Laboratory in the USA, are designed or operated with a focus on PMI.

1.1.3 Related Problems

Tritium Breeding and Extraction from a Lithium Blanket

A commercial fusion reactor will likely use tritium as a fuel. Tritium was originally produced by irradiation of water in nuclear fission reactors and is currently produced by irradiation of lithium control rods. The total inventory of tritium in the United States was only 75 kg in 1996 [20] and was mostly used for the fusion stages of thermonuclear weapons. Operation of a fleet of commercial fusion reactors with a small tritium inventory would be impossible. The D-T reaction consumes a tritium atom but releases a neutron, which can interact with lithium-6 or lithium-7 to produce

a new tritium atom. If a lithium blanket were to be constructed around the fusion reactor, the tritium produced by incident neutrons could be collected and fed back into the machine as fuel. Lithium blankets are placed inside the pressure vessel, as the metal walls can absorb substantial amounts of neutrons.

To test this system, tritium must be extracted from lithium inside the vacuum chamber and fed back into the plasma, for example by a neutral beam injector (NBI). Due to the rarity, expense, and handling difficulty of tritium, deuterium or hydrogen deposited into the lithium blanket could be used to test the extraction and re-injection processes. In doing so the lithium must be exposed to the plasma, and the amount of hydrogen or deuterium injected into the plasma must be known and controlled so that the system is representative of what could be expected in a commercial fusion reactor. This full system has never been tested in a running reactor.

Surface Morphology and Response (Surface Retention and Recycling)

Particle fluxes are also indicative of damage done to surfaces in PMI applications. Higher fluences can lead to dramatically different surface morphologies which react differently to untreated surfaces under plasma bombardment. Energy and angular distributions of particles as they impact the surface can significantly affect the surface morphology in addition to the particle fluences. Over time, surfaces which may be initially favorable can lose their desirable properties in unpredicted ways.

Although large devices can test some different materials, due to the capital cost fewer risks are taken with novel materials or morphologies. A toroidal device that can reproduce the energy and angular distributions of the particles arriving at the PFCs of a commercial reactor design with an emphasis on the PMI of many different materials could assist in finding innovative solutions to high-fluence damages over long exposure times. Such a device would require a long operational time, unlike the traditionally pulsed discharges of contemporary large fusion devices, and with moderate to high instantaneous particle fluxes.

1.2 The new HIDRA Facility at Illinois

The Hybrid Illinois Device for Research and Applications (HIDRA) has had a long career, originally as a joint German-French radio frequency heating experiment located in France (1972-1982) before being relocated to the University of Stuttgart (1982-2000) and finally being moved to Greifswald (2001-2013) where it was known as the Wendelstein Experiment in Greifswald fr die Ausbildung (WEGA). During its latest tenure in Germany, the device was used for testing diagnostics and training personnel for the recently-onlined W7-X stellarator. Having served its purpose, the device was donated [21] to the Center for Plasma-Material Interactions (CPMI) at the University of Illinois at Urbana-Champaign (UIUC). The UIUC College of Engineering covered the costs of shipping the device across the Atlantic, as well as installation costs for adding additional power lines and installing a cooling system for the magnetic coils. Presently, the device is located at the Nuclear Radiation Lab as in figure 1.2 and is being prepared for its first experimental campaign in Illinois following the first plasma event in April 2016.

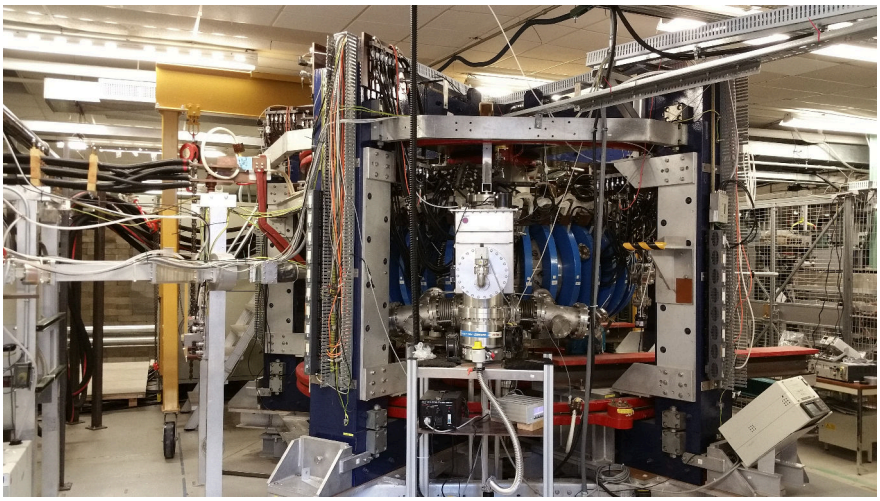


Figure 1.2: The Hybrid Illinois Device for Research and Applications (HIDRA). Clearly visible are the blue toroidal coils (center), red vertical coils (top, bottom), vacuum pump manifold (center), 2.45 GHz microwave sources (left), electrical wiring for the coil assembly (left, top), and some of the transformers (behind).

HIDRA is a classical l=2, m=5 stellarator, with a major radius of 72 cm and a minor radius of 19 cm. The most striking feature of HIDRA is the ease with which it may be disassembled; the vacuum vessel is made of two otherwise continuous halves, one of which is mounted on rails for

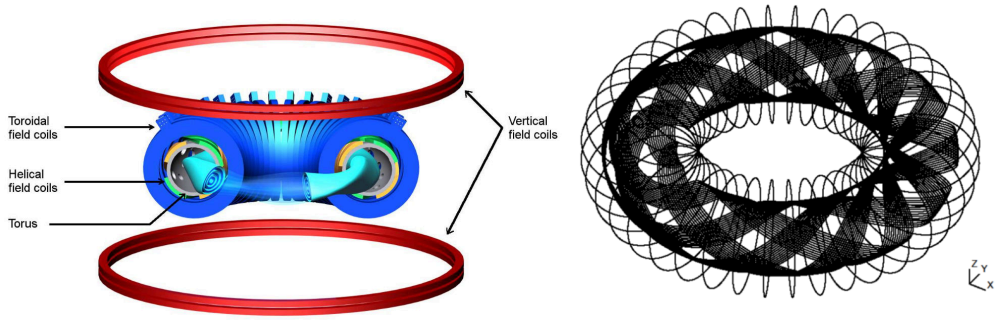


Figure 1.3: Two representations of HIDRA's coil systems (left from [22]).

easy removal. When properly assembled, the one ton vacuum vessel half may be slid along the rails by one or two individuals.

HIDRA is equipped with 26 kW of 2.45 GHz microwave heating, separated into a 20 kW and a 6 kW source. As of the time of writing, the 6 kW source has an Electron Bernstein Wave (EBW) antenna, and the 20 kW source has an O-mode antenna. Operational vacuums are sustained by up to two Varian 1000HT turbopumps backed by Varian DS402 rough pumps. The turbopumps are located on periods 3 and 5, with the latter currently installed. HIDRA has a number of default diagnostics, including Langmuir probes, current probes, Hall probes, $\dot{\mathrm{B}}$ probes, and Rowgowski coils, and is easily extensible to new diagnostics due to the original design focus which called for ease of access to the plasma volume.

The toroidal assembly is comprised of forty coils, each of thirteen turns of copper wire. Four packets of fourteen filaments each wrap helically around the vacuum vessel making five turns in the poloidal direction per two turns in the toroidal direction. Each packet is connected in series and current flows in alternating directions around the device, providing a dipole-like perturbation to the toroidal field. Due to the separation in the vacuum vessel, each half of the torus is fully wound independently and are connected by a bridge. Four vertical field coils are also present, two above and two below. All three coil systems are shown together in figure 1.3.

HIDRAs coil system is made of water-cooled copper wires, and its operational time is principally limited by the efficiency of the heat-removal system and the current flowing through the coils. In the current configuration, two operational modes have been proposed: a low-field discharge with an axial field strength of 87.5 mT corresponding to the electron cyclotron resonance at the magnetic axis, and a high-field discharge with an axial field strength of up to 0.5 T corresponding to the

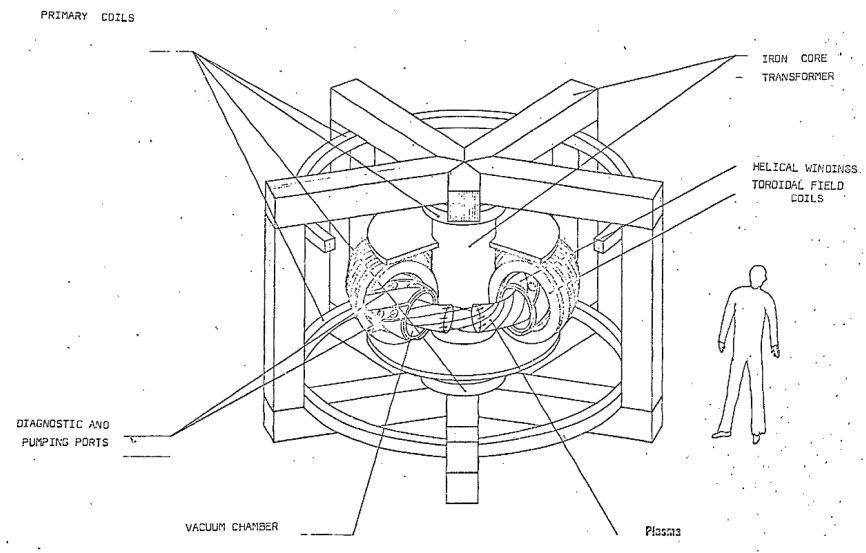


Figure 1.4: A schematic of WEGA, including the ohmic heating system and a person to scale.

second electron cyclotron harmonic. In the high-field discharge the cooling capacity of the reservoirs is sufficient to run the device up to half an hour.

Magnetic fields configurations in a given stellarator may be compared by introducing the rotational transform, or the relative strengths of the poloidal to toroidal fields (chapter 3). Since the ratio is dimensionless a given rotational transform may be achieved in either high- or low-field configurations, limited only by physical constraints. The achievable rotational transforms are determined by the transformers; the toroidal coils have thirteen turns while the helical coils have only two turns, and both coil sets have transformers of similar specifications. HIDRA may be operated with rotational transforms of 0 to 0.4 in high-field operation as demonstrated in WEGA rotational transform scans [23], and potentially as high as desired in low-field operation. Higher rotational transforms generally lead to smaller plasma volumes and worse confinement, but can be used to change the angle of incidence of the plasma onto a prospective surface or otherwise maneuver the plasma about the vacuum vessel.

In addition to the stellarator mode of operation, HIDRA may be (and originally was) operated as a tokamak. A coil to drive current for ohmic heating is installed as in figure 1.4, and the vertical field coils may be used to stabilize the plasma. The tokamak mode may optionally use a second vacuum vessel with thinner walls to facilitate quicker field permeation for enhanced stability. In stellarator mode, ohmic heating may be used for a temporary injection of energy into the plasma

at the cost of changing the plasma equilibrium by introducing a significant toroidal current.

HIDRAs scientific mission is:

1. To educate the next generation of fusion and PMI researches on the operation of a medium-sized toroidal device
2. To act as a liquid metal PFC test facility
3. To act as the first complete test of a lithium blanket hydrogen isotope extraction and re-injection system
4. To synthesize materials under the influence of a toroidal plasma
5. To assist in refining and validating plasma and PMI models which may inform on the operation of other devices
6. To assist in developing diagnostics, especially of surfaces, which may be used on other devices

Creating an appealing testbed to the greater plasma physics community, especially for the testing of novel surface compositions and morphologies, requires knowledge of the conditions which may be expected at the PFCs in a typical HIDRA discharge. A tool must be developed to apply theoretical insights to the complex 3D plasmas encountered in stellarators, both to assist in the design and operation of experiments and to analyze the data in the context of a predictive analysis.

1.3 `HIDRAmod`: Establishing a 3D Simulation Code for HIDRA

Producing a high-fidelity three-dimensional simulation of HIDRA's plasma environment has a number of challenges:

- As a consequence of the complex magnetic fields encountered in the stellarator, a fully-3D simulation is a requirement.
- The importance placed on the plasma-material interactions necessitates a robust plasma-material interaction model, and hints at the presence of dense impurities.

- As a classical stellarator with low-power modes of operation, and due to the few heating schemes available, the plasma may have significant neutral fractions in most of scenarios.

A fully-featured code to solve these challenges requires many interconnected modules, each one handling a piece of the physics of the device. Such a simulation framework for HIDRA, named `HIDRAmod`, is schematically represented in figure 1.5. As conceptualized, `HIDRAmod` has the following modules:

1. A 3D equilibrium magnetic field solver, handling the creation of a field-aligned magnetic mesh for the simulation domain. The stochasticity of HIDRA's B-field outside the Last Closed Magnetic Surface requires a dedicated treatment.

2. A core plasma transport solver, handling the central regions of power deposition extending outwards to the edge boundary, in the context of significant neutrals.

3. A full-wave electromagnetic solver, handling the absorption and reflection of the electromagnetic waves in the plasma volume and up to the vacuum vessel.

4. An edge transport solver, handling the transport of plasma and impurities in the ergodic and the chaotic 3D edge region of a classical stellarator, in the context of significant neutrals.

5. A robust plasma-material interaction module, including a magnetized sheath, for the determination of the material emission processes consequent to surface irradiation.

6. A robust coupling methodology between the various modules.

The intrinsic multi-scale and multi-physics nature of the `HIDRAmod` framework requires the development of several independent modules and the testing of the interfaces among the individual components. Such a large effort is outside the scope of a single Thesis; in the present work some of the critical components of `HIDRAmod` will be developed and used for a characterization of the edge conditions of HIDRA. Furthermore, some model components may be adapted from existing codes, for use in `HIDRAmod`. `FIELDLINES`, a part of the STELLOPT suite of codes, is a field line tracer which has been used to produce visualizations of W7-X, NCSX, HSX, and other major devices. `FIELDLINES` will be adopted for the tracing of the magnetic field lines, an operation necessary

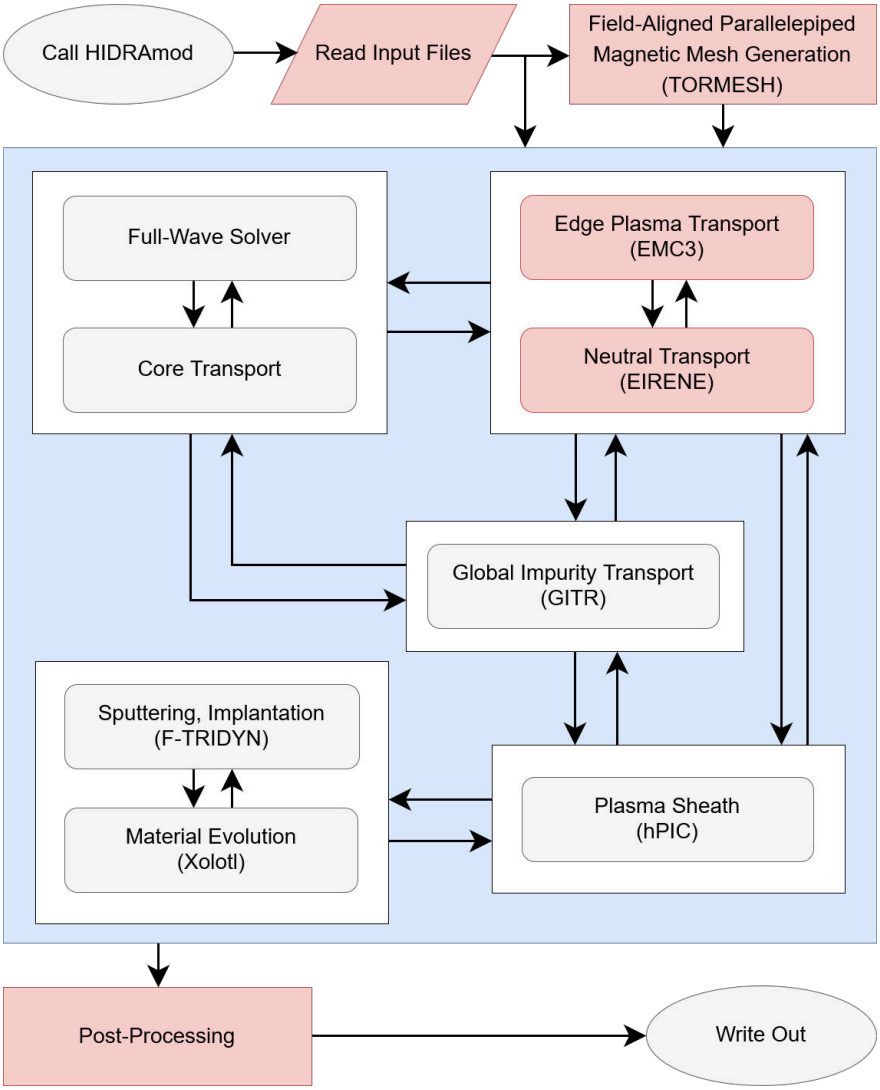


Figure 1.5: A schematic diagram of `HIDRAmod`. The sheath model and surface physics comprise the PMI model. Along with a core plasma, edge plasma, and global impurity model, these four components comprise the core of `HIDRAmod`. In the present work, the red boxes have been implemented.

for the creation of a 3D magnetic mesh. We have used FIELDLINES inside our newly developed 3D mesh module, created as part of this thesis. EMC3-EIRENE is a 3D edge transport code with impurities (EMC3) coupled with a neutral solver (EIRENE) that can handle regions outside of the power deposition region up to the plasma wall. Opportunely combined and extended, the three codes could provide a critical subset of the `HIDRAmod` environment. Independently of `HIDRAmod` the two could also be used to form a self-consistent picture of the edge plasma, given inputs from material models and an uncoupled core solver.

1.4 Thesis overview

This thesis aims to establish the field solver and edge transport modules of `HIDRAmod` and to use them in a preliminary characterization of expected HIDRA edge operating conditions. Extrapolation from WEGA operational data and HIDRA hardware configurations will provide missing core plasma information, and the built-in impurity module of EMC3-EIRENE will be used in place of a more robust PMI module.

Specifically, the thesis aims to accomplish the following:

- Prepare documentation for the usage of FIELDLINES and EMC3-EIRENE on HIDRA.
- Adapt EMC3-EIRENE to a classical stellarator by adding a Bohm-like diffusivity to permit intrinsic calculation of transport coefficients.
- Show that the edge simulations give physically plausible results in the absence of additional modules.
- Perform an elementary analysis of different limiter positions in the HIDRA edge plasma that may inform PMI experiments.
- Create software utilities to facilitate the rapid investigation of such conditions.

A preliminary introduction to the theory behind EMC3 and EIRENE will be followed by the additions to EMC3 to handle a Bohm-like diffusivity. The theory behind `FIELDLINES` and its application to the specific case of the classical stellarator HIDRA will be presented. Simulation results from the implemented modules will be presented, with a discussion on the physical relevance

and relation to established WEGA data. A general description of components of `HIDRAmod` will be located in appendix A, and a description of added code for a Bohm-like diffusivity in EMC3-EIRENE will be attached in appendix B. In appendices C and D, a brief description of the input and output files of EMC3 are presented.

Chapter 2

Plasma Transport Equations

A brief overview of the theoretical model used in describing HIDRA's partially-ionized plasma is presented herein. Currently, the models are implemented in the coupled codes EMC3-EIRENE, with EMC3 handling edge plasma and ionized impurity transport and EIRENE handling neutral transport. Some specific information related to the implementation is presented in the final section to aid with grid generation as documentation of EMC3's mesh checking routines and the transparency of the coupling between the two codes has significant room for improvement, though both are of fundamental importance for the successful completion of a simulation. For reference on the fluid models in EMC3, the reader is also referred to [24], [25], [26], [27], and [28]; for reference on the neutral models in EIRENE, the reader is referred to the EIRENE user manual [29] and [30]; for reference on the Reversible Field-line Mapping boundary condition, the reader is referred to [31], [32], and [33]. Appendices C and D Contain a further brief overview on the input and output files of EMC3.

2.1 Braginskii Model

Braginskii was the first to write down the transport equations which are named after him. His original derivation neglects elastic collisions, and his closure of the transport equations by the assumption of sonic flows hinders applicability to more stationary discharges [34]. His original derivation [24] is presented in this section, and the resulting equations in EMC3 summarized in the next section.

The motion and interaction of charged particles in a plasma may be specified by a distribution function $f_i(\boldsymbol{r}, \boldsymbol{v}, t)$, of which each species i has its own unique instance. The quantity $f_i(\boldsymbol{r}, \boldsymbol{v}, t)\ \mathrm{d}\boldsymbol{r}\ \mathrm{d}\boldsymbol{v}$ describes the density of particles in a continuous, arbitrarily precise range of posi-

tion ($\boldsymbol{r}$) and velocities ($\boldsymbol{v}$) at a given time (t), and an integration of the distribution function will provide the total number of particles in the given range. The propagation and development of the distribution functions in time, space, and velocity and the interactions between coexisting distribution functions in the same region of space may be described by the Boltzmann kinetic transport equation (BKTE),

$$\frac{\partial f_i}{\partial t} + \nabla_{\boldsymbol{r}} \left(\boldsymbol{v} \cdot f_i \right) + \nabla_{\boldsymbol{v}} \left(\frac{\boldsymbol{F}}{m} \cdot f_i \right) = C_i + S_i. \tag{2.1}$$

In equation 2.1, the terms on the left-hand side represent the time evolution, convection, and acceleration of the distribution function, while all collisions that d not transform the species that a given distribution may encounter are abstracted in the quantity $C_i = \sum_j C_{ij}(f_i, f_j)$ and all sources in the source term S_i. The plasma transport equations for particles, momentum, and energy may be derived by integrating the BKTE over the zeroth, first, and second order moments in the velocity space.

2.1.1 Derivation of the Plasma Transport Equations

The integral of the zeroth, first, and second order moments in the velocity space may be written as

$$\begin{aligned}
&\int_{\boldsymbol{v}} \frac{\partial f}{\partial t} d\boldsymbol{v} + \int_{\boldsymbol{v}} \nabla_{\boldsymbol{r}} \left(\boldsymbol{v} \cdot f \right) d\boldsymbol{v} + \int_{\boldsymbol{v}} \nabla_{\boldsymbol{v}} \left(\frac{\boldsymbol{F}}{m} \cdot f \right) d\boldsymbol{v} \\
&= \int_{\boldsymbol{v}} C \ \ d\boldsymbol{v} + \int_{\boldsymbol{v}} S \ \ d\boldsymbol{v} \\
&\int_{\boldsymbol{v}} m\boldsymbol{v} \frac{\partial f}{\partial t} d\boldsymbol{v} + \int_{\boldsymbol{v}} m\boldsymbol{v} \nabla_{\boldsymbol{r}} \left(\boldsymbol{v} \cdot f \right) d\boldsymbol{v} + \int_{\boldsymbol{v}} m\boldsymbol{v} \nabla_{\boldsymbol{v}} \left(\frac{\boldsymbol{F}}{m} \cdot f \right) d\boldsymbol{v} \\
&= \int_{\boldsymbol{v}} m\boldsymbol{v} \ \ C \ \ d\boldsymbol{v} + \int_{\boldsymbol{v}} m\boldsymbol{v} \ \ S \ \ d\boldsymbol{v} \\
&\int_{\boldsymbol{v}} \frac{m\boldsymbol{v}^2}{2} \frac{\partial f}{\partial t} d\boldsymbol{v} + \int_{\boldsymbol{v}} \frac{m\boldsymbol{v}^2}{2} \nabla_{\boldsymbol{r}} \left(\boldsymbol{v} \cdot f \right) d\boldsymbol{v} + \int_{\boldsymbol{v}} \frac{m\boldsymbol{v}^2}{2} \nabla_{\boldsymbol{v}} \left(\frac{\boldsymbol{F}}{m} \cdot f \right) d\boldsymbol{v} \\
&= \int_{\boldsymbol{v}} \frac{m\boldsymbol{v}^2}{2} C \ \ d\boldsymbol{v} + \int_{\boldsymbol{v}} \frac{m\boldsymbol{v}^2}{2} S \ \ d\boldsymbol{v}.
\end{aligned} \tag{2.2}$$

Setting aside collisional and source terms, the transport terms may be simplified by recognizing a key aspect of distribution functions; for any unit volume in physical space, a distribution function may be integrated over all of velocity space to provide a 'bulk value' of the given quantity for that

location. Corresponding to the three moments, these three quantities are

$$
\begin{aligned}
n_i(t,\boldsymbol{r}) &= \int f_i(t,\boldsymbol{r},\boldsymbol{v}\ \ d\boldsymbol{v}, \\
\boldsymbol{V}_i(t,\boldsymbol{r}) &= \frac{1}{n_i}\int v f_i(t,\boldsymbol{r},\boldsymbol{v})\ \ d\boldsymbol{v}, \\
T_i(t,\boldsymbol{r}) &= \frac{1}{n_i}\int \frac{m_i}{3}(\boldsymbol{v}-V_i)^2 f_i(t,\boldsymbol{r},\boldsymbol{v})\ \ d\boldsymbol{v}.
\end{aligned}
\tag{2.3}
$$

In the second expression, the quantity $\boldsymbol{V}_i(t,\boldsymbol{r})$ may also be recognized as the quantity $\langle \boldsymbol{v} \rangle$, the drift velocity; in the last expression the expected integration factor of energy $\frac{1}{2}m\boldsymbol{v}^2$ is instead replaced by the factor $\frac{m_i}{3}(\boldsymbol{v}-V_i)$ as the former is only valid in Maxwellian distributions.

Using these expressions, the zeroth, first, and second order moments may be simplified by interchanging the order of integration and differentiation. In order to facilitate readability, the species subscript i has been omitted except when necessary; subscripts a and b refer to coordinate indicies. The first two terms of each moment equation produce the following relations:

$$
\begin{aligned}
&\int_{\boldsymbol{v}} \frac{\partial f}{\partial t} d\boldsymbol{v} = \frac{\partial}{\partial t}\int_{\boldsymbol{v}} f\ \ d\boldsymbol{v} = \frac{\partial n}{\partial t}, \\
&\int_{\boldsymbol{v}} \nabla_{\boldsymbol{r}}\left(\boldsymbol{v}\cdot f\right) d\boldsymbol{v} = \nabla_{\boldsymbol{r}}\int_{\boldsymbol{v}} \left(\boldsymbol{v}\cdot f\right) d\boldsymbol{v} = \nabla_{\boldsymbol{r}}(n\boldsymbol{V}), \\
&\int_{\boldsymbol{v}} m\boldsymbol{v}\frac{\partial f}{\partial t} d\boldsymbol{v} = m\frac{\partial}{\partial t}\int_{\boldsymbol{v}} \boldsymbol{v} f\ \ d\boldsymbol{v} = m\frac{\partial(n\boldsymbol{V})}{\partial t}, \\
&\int_{\boldsymbol{v}} m\boldsymbol{v}\nabla_{\boldsymbol{r}}\left(\boldsymbol{v}\cdot f\right) d\boldsymbol{v} = m\nabla_{\boldsymbol{r}}\int_{\boldsymbol{v}} \boldsymbol{v}\left(\boldsymbol{v}\cdot f\right) d\boldsymbol{v} = m\nabla_{\boldsymbol{r}}(n\langle \boldsymbol{v}_a\boldsymbol{v}_b\rangle), \\
&\int_{\boldsymbol{v}} \frac{m\boldsymbol{v}^2}{2}\frac{\partial f}{\partial t} d\boldsymbol{v} = \frac{m}{2}\frac{\partial}{\partial t}\int_{\boldsymbol{v}} \boldsymbol{v}^2 f\ \ d\boldsymbol{v} = \frac{m}{2}\frac{\partial(n\langle \boldsymbol{v}^2\rangle)}{\partial t}, \\
&\int_{\boldsymbol{v}} \frac{m\boldsymbol{v}^2}{2}\nabla_{\boldsymbol{r}}\left(\boldsymbol{v}\cdot f\right) d\boldsymbol{v} = \frac{m}{2}\nabla_{\boldsymbol{r}}\int_{\boldsymbol{v}} \boldsymbol{v}^2\left(\boldsymbol{v}\cdot f\right) d\boldsymbol{v} = \frac{m}{2}\nabla_{\boldsymbol{r}}(n\langle \boldsymbol{v}^2\boldsymbol{v}\rangle).
\end{aligned}
\tag{2.4}
$$

To simplify the third term of each moment containing the generalized force, it is necessary to integrate by parts. It is also prudent to specialize the force to the case of plasma physics, using the Lorentz force

$$
\boldsymbol{F} = q\boldsymbol{E} + q\boldsymbol{v}\times\boldsymbol{B} \tag{2.5}
$$

for a given particle charge q, electric field $\boldsymbol{E}$, and magnetic field $\boldsymbol{B}$ for further simplifications. In

these expressions, the rapid decay of the distribution function for high velocities must be assumed:

$$\begin{aligned}
&\int_{\boldsymbol{v}} \nabla_{\boldsymbol{v}} \left(\frac{\boldsymbol{F}}{m} \cdot f \right) d\boldsymbol{v} = 0, \\
&\int_{\boldsymbol{v}} \boldsymbol{v} \nabla_{\boldsymbol{v}} \left(\boldsymbol{F} \cdot f \right) d\boldsymbol{v} = - \int_{\boldsymbol{v}} \nabla_{\boldsymbol{v}} \boldsymbol{v} \left(\boldsymbol{F} \cdot f \right) d\boldsymbol{v} = en \left(\boldsymbol{E} + \boldsymbol{V} \times \boldsymbol{B} \right), \\
&\int_{\boldsymbol{v}} \frac{\boldsymbol{v}^2}{2} \nabla_{\boldsymbol{v}} \left(\boldsymbol{F} \cdot f \right) d\boldsymbol{v} = - \int_{\boldsymbol{v}} \frac{\nabla_{\boldsymbol{v}} \boldsymbol{v}^2}{2} \left(\boldsymbol{F} \cdot f \right) d\boldsymbol{v} = en \ \boldsymbol{E} \cdot \boldsymbol{V}.
\end{aligned} \tag{2.6}$$

To further simplify the average velocity terms the velocity may be written as a mean velocity plus a random velocity component unique to a given particle,

$$\boldsymbol{v}_i = \boldsymbol{V} + (\boldsymbol{v} - \boldsymbol{V}) = \boldsymbol{V} + \boldsymbol{v}'. \tag{2.7}$$

With the knowledge that $\langle \boldsymbol{v}' \rangle$ is zero by definition, the higher-order velocity-average terms will then simplify with the following relations:

$$\begin{aligned}
\langle \boldsymbol{v}_a \boldsymbol{v}_b \rangle &= \langle \boldsymbol{v}'_a \boldsymbol{v}'_b + \boldsymbol{v}'_a \boldsymbol{V}_b + \boldsymbol{v}'_b \boldsymbol{V}_a + \boldsymbol{V}_a \boldsymbol{V}_b \rangle = \langle \boldsymbol{v}'_a \boldsymbol{v}'_b \rangle + \boldsymbol{V}_a \boldsymbol{V}_b, \\
\langle \boldsymbol{v}_a^2 \rangle &= \langle \boldsymbol{v}'_a \boldsymbol{v}'_a + \boldsymbol{v}'_a \boldsymbol{V}_a + \boldsymbol{v}'_a \boldsymbol{V}_a + \boldsymbol{V}_a \boldsymbol{V}_a \rangle = \langle \boldsymbol{v}_a'^2 \rangle + \boldsymbol{V}_a^2, \\
\langle \boldsymbol{v}_a^2 \boldsymbol{v}_b \rangle &= \langle (\boldsymbol{V}_a + \boldsymbol{v}'_a)^2 (\boldsymbol{V}_b + \boldsymbol{v}'_b) \rangle = \boldsymbol{V}_a^2 \boldsymbol{V}_b + \boldsymbol{V}_b \langle \boldsymbol{v}_a'^2 \rangle + 2 \boldsymbol{V}_a \langle \boldsymbol{v}'_a \boldsymbol{v}'_b \rangle + \langle \boldsymbol{v}_a'^2 \boldsymbol{v}'_b \rangle.
\end{aligned} \tag{2.8}$$

Finally, it may be recognized that the scalar pressure, pressure tensor, and the stress tensor are found in these expressions in the forms

$$\begin{aligned}
p &= \frac{nm \langle \boldsymbol{v}'^2 \rangle}{3}, \\
\boldsymbol{P}_{ab} &= nm \langle \boldsymbol{v}'_a \boldsymbol{v}'_b \rangle = p \boldsymbol{\delta}_{ab} + \boldsymbol{\pi}_{ab}, \\
\boldsymbol{\pi}_{ab} &= nm \langle \boldsymbol{v}'_a \boldsymbol{v}'_b - (\boldsymbol{v}_a'^2) \boldsymbol{\delta}_{ab} \rangle.
\end{aligned} \tag{2.9}$$

Collecting terms, the moment equations may be written as

$$\begin{aligned}
&\frac{\partial n_i}{\partial t} + \nabla_r \cdot (n_i \boldsymbol{V}_a) = C_{0,i} + S_{0,i}, \\
&\frac{\partial}{\partial t} (m_i n_i \boldsymbol{V}_a) + \nabla_r \cdot (m_i n_i \langle \boldsymbol{v}_a \boldsymbol{v}_b \rangle) - q_i n_i \left(E_a + [\boldsymbol{V} \boldsymbol{B}]_a \right) = C_{2,i} + S_{1,i}, \\
&\frac{\partial}{\partial t} \left(\frac{m_i n_i}{2} \langle \boldsymbol{v}^2 \rangle \right) + \nabla_r \cdot \left(\frac{m_i n_i}{2} \langle \boldsymbol{v}^2 \boldsymbol{v} \rangle \right) - q_i n_i \boldsymbol{E} \cdot \boldsymbol{V} = C_{2,i} + S_{2,i},
\end{aligned} \tag{2.10}$$

with $C_{0-2,i}$ representing the collision integrals and $S_{0-2,i}$, which will be expressed in the next subsection. Using the pressure and stress tensor relations, the expectation values of the various

velocity combinations may be rewritten in the form used by EMC3, as will be discussed in section 2.2.1.

2.1.2 Elastic Collisions

In the high-density fusion test facilities EMC3-EIRENE is used for simulating, collisions are non-negligible. Due to the Monte-Carlo model used and the requirements of computational scalability to large physical dimensions, the collision integral must be used in a form suitable to the numerical scheme. The forms used in EMC3 have been explained in detail in [24] and [35], and are summarized here.

For the first elastic collision integral the contribution is trivially zero, since elastic collisions do not result in particle transformations. As a result,

$$C_{0,i} = \int_{\boldsymbol{v}} C_i \; d\boldsymbol{v} = 0. \tag{2.11}$$

The second elastic collision integral was expressed by Braginskii as the combination of a friction force and a thermal force, due to relative particle velocities and temperature gradients respectively. Thus (with i,j over the species),

$$C_{1,i} = \int_{\boldsymbol{v}} m_i \boldsymbol{v}_i C_i \; d\boldsymbol{v} = \boldsymbol{R}_{ij} = \boldsymbol{R}_{ij}^{\Delta \boldsymbol{v}} + \boldsymbol{R}_{ij}^{\nabla T}. \tag{2.12}$$

Similarly, the third elastic collision integral is given by

$$C_{2,i} = \int_{\boldsymbol{v}} \frac{m_i \boldsymbol{v}_i^2}{2} C_i \; d\boldsymbol{v} = \boldsymbol{v}_i \cdot \boldsymbol{R}_{ij} + Q_{ij}, \tag{2.13}$$

with R_{ij} as in the second elastic collision integral and Q_{ij} as

$$Q_{ij} = 3 n_i \frac{\mu_{ij}}{m_i + m_j} \frac{1}{\tau_{ij}} (T_j - T_i). \tag{2.14}$$

2.2 Fokker-Planck Model and Monte Carlo Interpretation

As derived in [27], for weakly-coupled plasmas Baginskii's fluid equations may be generalized into a Fokker-Planck form

$$\frac{\partial g}{\partial t} + \nabla \cdot [\boldsymbol{V} g - \boldsymbol{D} \cdot \nabla g] = S. \tag{2.15}$$

In this the quantity of interest g represents either the density, moment, or electron or ion temperatures. The Fokker-Planck equation as originally derived describes the evolution of a probability distribution and has its origin in Brownian motion; correspondingly it is readily solvable by a Monte Carlo method, bypassing the usual limitations on mesh size for finite-difference, finite-element, and direct matrix inversion methods for a system of linear equations.

A Monte Carlo method in the general sense reconstructs the quantity of interest (in this case, the fluid moments) by following so-called 'test particles' as they wander about the domain and sampling the distribution where they land. Reconstructing the trajectory and thus the moments occurs by integrating over the sum of the random motions of all test particles with a weight function which varies to capture the topology of the domain.

To relate the macroscopic Fokker-Planck equation to the microscopic motion of individual particles, one may consider a Markov process defined by

$$g(\boldsymbol{r}, t + \Delta t) = \int T(\boldsymbol{r}, \boldsymbol{r} + \Delta t | \boldsymbol{r}', t) \; g(\boldsymbol{r}' t) d\boldsymbol{r}'. \tag{2.16}$$

At any time $t + \delta t$, the new state is determined only by the transition probability T and the current state. The transition probability may be rewritten in terms of $\delta \boldsymbol{r} = \boldsymbol{r} - \boldsymbol{r}'$ and expanded in a Taylor series around δt and δr to find

$$\frac{\partial g}{\partial t} + \nabla \cdot \left\{ \frac{g}{\Delta t} \int \Delta \boldsymbol{r} \; T(\boldsymbol{r} + \Delta \boldsymbol{r}, \boldsymbol{r} + \Delta t | \boldsymbol{r}, t) \; d\Delta \boldsymbol{r} - \nabla \cdot \left[\frac{g}{2\Delta t} \int \Delta \boldsymbol{r} \Delta \boldsymbol{r} \; T(\boldsymbol{r} + \Delta \boldsymbol{r}, \boldsymbol{r} + \Delta t | \boldsymbol{r}, t) \; d\Delta \boldsymbol{r} \right] \right\} \tag{2.17}$$

As long as

$$\begin{aligned} \frac{1}{\Delta t}\int \Delta \boldsymbol{r}\ T(\boldsymbol{r}+\Delta\boldsymbol{r},\boldsymbol{r}+\Delta t|\boldsymbol{r},t)\ d\Delta\boldsymbol{r} = \boldsymbol{V}+\nabla\cdot\boldsymbol{D} \quad \text{and} \\ \frac{1}{2\Delta t}\int \Delta\boldsymbol{r}\Delta\boldsymbol{r}\ T(\boldsymbol{r}+\Delta\boldsymbol{r},\boldsymbol{r}+\Delta t|\boldsymbol{r},t)\ d\Delta\boldsymbol{r} = \boldsymbol{D}, \end{aligned} \tag{2.18}$$

the microscopic Markov process and macroscopic Fokker-Planck equation are identical. When relating the macroscopic Fokker-Planck equation to the macroscopic Braginskii transport equations, the dependence of the quantities g, $\boldsymbol{V}$, and $\boldsymbol{D}$ in the Fokker-Plank equation solely on macroscopic scales is evident. Since the transition probability is likewise only dependent on the microscopic variable $\delta\boldsymbol{r}$ for a given $\boldsymbol{r}$, the dynamics of the Markov process may be analyzed in the context of a local coordinate system independent of the global coordinates. By separating out the parallel and perpendicular components of $\boldsymbol{D}$ as $\boldsymbol{D} = D_{\parallel}\boldsymbol{bb} + D_{\perp}(\boldsymbol{I}-\boldsymbol{bb})$ and then expressing the quantity $\boldsymbol{V}+\nabla\cdot\boldsymbol{D}$ in terms of the orthonormal unit vectors

$$\begin{aligned} \hat{\boldsymbol{x}} = \boldsymbol{e}_1 = -\frac{(\boldsymbol{b}\cdot\nabla)\boldsymbol{b}}{|(\boldsymbol{b}\cdot\nabla\boldsymbol{v})|}, \\ \hat{\boldsymbol{y}} = \boldsymbol{e}_2 = \boldsymbol{b}\times\boldsymbol{e}_1, \\ \hat{\boldsymbol{z}} = \boldsymbol{e}_3 = \boldsymbol{b}, \end{aligned} \tag{2.19}$$

it may be seen that

$$\boldsymbol{V}+\nabla\cdot\boldsymbol{D} = \begin{bmatrix} \boldsymbol{e}_1\cdot(\boldsymbol{V}+\nabla D_{\perp}) - (D_{\parallel}-D_{\perp})|(\boldsymbol{e}_3\cdot\nabla)\boldsymbol{e}_3| \\ \boldsymbol{e}_2\cdot(\boldsymbol{V}+\nabla D_{\perp}) \\ \boldsymbol{e}_3\cdot(\boldsymbol{V}+\nabla D_{\parallel}) + (D_{\parallel}-D_{\perp})\nabla\cdot\boldsymbol{e}_3 \end{bmatrix}. \tag{2.20}$$

To solve the resulting expressions, the transition probability for a random walk in three dimensions may be written as

$$T(\Delta x,\Delta y,\Delta z,\Delta t) = \frac{1}{MNL}\sum_{m,n,l}\delta_m(\Delta x-\Delta x_m)\delta_n(\Delta y-\Delta y_n)\delta_l(\Delta z-\Delta z_l), \tag{2.21}$$

inserted into equation 2.18 and 2.18, and integrated. In doing so the values for M, N, and L must be specialized to the case where $M=N=L=2$, which will ultimately set the requirement for a mesh made of tetrahedrons. Indeed, there is no solution for any $M,N,L\leq 2$. Due to the previous

diagonalization of the diffusion tensor, only a few terms will remain in the form of

$$\Delta i_{1,2} = \Delta t\ (\boldsymbol{V} + \nabla \cdot \boldsymbol{D})_i \pm \sqrt{2D_{ii}\Delta t} \quad \text{for } i = x, y, z. \tag{2.22}$$

The transition function in equation 2.21 restricts the particle to movement in constant steps, with each direction determined independently by either the positive or negative sign in equation 2.22. By replacing the plus-minus sign with a random variable ξ_i with equal chance to be either positive or negative unity, it is trivial to recognize the final form of the Monte Carlo step in the local coordinate system,

$$\begin{aligned} \Delta i &= \Delta t\ \ (\boldsymbol{V} + \nabla \cdot \boldsymbol{D})_i + \xi_i \sqrt{2D_{ii}\Delta t}, \\ \boldsymbol{r}(t + \Delta t) &= \boldsymbol{r}(t) + \sum_{i\,=x,y,z} \Delta i(\xi_i)\hat{\boldsymbol{i}}. \end{aligned} \tag{2.23}$$

2.2.1 Final EMC3 Fluid Plasma Equations and Monte Carlo Solution

EMC3 has implemented a set of equations slightly simplified from the full Braginskii expressions; the particle, momentum, energy, and ion equations used are respectively

$$\begin{aligned} &\nabla_{\parallel} \cdot (nV_{\parallel}) + \nabla_{\perp} \cdot (-D\nabla_{\perp} n) = S_p, \\ &\nabla_{\parallel} \cdot (mnV_{\parallel}V_{\parallel} - \eta_{\parallel}\nabla_{\parallel}V_{\parallel}) + \nabla_{\perp} \cdot (-mV_{\parallel}D\nabla_{\perp} - \eta_{\perp}\nabla_{\perp}V_{\parallel}) = -\nabla_{\parallel} p + S_m, \\ &\nabla_{\parallel} \cdot \left(-\kappa_i \nabla_{\parallel} T_i + \frac{5}{2} n T_i V_{\parallel}\right) + \nabla_{\perp} \cdot \left(-\chi_i n \nabla_{\perp} T_i - \frac{5}{2} T_i D \nabla_{\perp} n\right) = k_B(T_e - T_i) + S_{ei}, \\ &\nabla_{\parallel} \cdot \left(-\kappa_e \nabla_{\parallel} T_e + \frac{5}{2} n T_e V_{\parallel}\right) + \nabla_{\perp} \cdot \left(-\chi_e n \nabla_{\perp} T_e - \frac{5}{2} T_e D \nabla_{\perp} n\right) = k_B(T_i - T_e) + S_{ee}. \end{aligned} \tag{2.24}$$

The Monte Carlo solution given in equation 2.23 will then have the coefficients given in table 2.1 for the parallel ($\hat{\boldsymbol{z}}$) direction and in table 2.2 for the perpendicular ($\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$) directions. Source terms which do not show up directly in the particle movement manifest instead as scale and weight factors.

Table 2.1: Coefficients for the computation of the parallel component of $\boldsymbol{r}(t)$.

f	n	$V_{\parallel}$	$T_{e,i}$
$(\boldsymbol{V} + \nabla \cdot \boldsymbol{D})_{\parallel}$	$V_{\parallel}$	$m_i n V_{\parallel} + \nabla_{\parallel}\eta_{\parallel}$	$\frac{5}{2} n V_{\parallel} + \nabla_{\parallel}\kappa_{e,i}$
$D_{\parallel}$	0	$\eta_{\parallel}$	$\kappa_{e,i}$

Table 2.2: Coefficients for the computation of the perpendicular component of $\boldsymbol{r}(t)$.

f	n	$V_\parallel$	$T_{e,i}$
$(\boldsymbol{V} + \nabla \cdot \boldsymbol{D})_\perp$	0	0	$(\chi_{e,i} - \frac{5}{2}D)\nabla_\perp n$
$D_\perp$	D	$m_i n D$	$n\chi_{e,i}$

2.2.2 Trace Impurity Model

In lieu of a full set of fluid equations for each impurity species, EMC3 treats all impurities as trace impurities. The temperature is assumed to be equivalent to the ion temperature, with the particle and momentum balances given by [35] and [28]

$$\begin{aligned} &\nabla_\parallel \cdot (n_Z V_{Z\parallel}) + \nabla_\perp \cdot (-D_{imp} \nabla_\perp n_z) = S_Z, \\ &-\frac{1}{n_Z}\frac{\mathrm{d}p_Z}{\mathrm{d}s} + ZeE_\parallel + m_Z f_{friction}\frac{V_\parallel - V_Z\ \parallel}{\tau_S} + 0.71Z^2\frac{\mathrm{d}T_e}{\mathrm{d}s} + 2.6 f_{ion} Z^2 \frac{\mathrm{d}T_i}{\mathrm{d}s} = m_Z \frac{\mathrm{d}V_{Z\parallel}}{\mathrm{d}t}. \end{aligned} \tag{2.25}$$

The momentum balance equates the motion of the particle to the force balance between the impurity pressure gradient, the electrostatic force, the friction force, and the electron and ion thermal forces respectively. Due to the trace assumption, the influence of the impurities is completely decoupled from the rest of the plasma with the exception of the impurity energy sink, which is added to the energy balance sources in the main plasma fluid equations.

2.3 Neutral Gas Coupling: The EIRENE Model

To solve for the neutrals gas profiles, EMC3 is coupled to EIRENE, a Monte Carlo kinetic Boltzmann transport code. EIRENE has been extensively coupled to various codes, the most famous of which is the B2 plasma fluid code used in tokamak simulations, and has been found to be in good agreement [30] with its contemporary neutral gas codes DEGAS and NIMBUS. The three codes differed mostly by statistical estimators and geometric options specific to their use case when written, but may have diverged since.

EIRENE works on a similarly structured grid to EMC3, but does not impose requirements on the mesh related to magnetic field alignments, so a grid for EMC3 will certainly work for EIRENE; EMC3 handles the conversion of the simulation domain into an EIRENE-compliant format au-

tomatically and opaquely to the end user. Atomic, molecular, and surface reflection information come from user-specified databases, but the code also requires many boundary conditions to be set independently from the mesh in addition to particle information, chemical reactions, diagnostic information, and dozens of variables to tailor the underlying models in the code, among others. As a result, although the grid is taken care of automatically there are still hundreds of variables which must be manually set in Fortran 77-style fixed-format-width fields. For a more thorough analysis than presented in this chapter the user is referred to the EIRENE manual [29] for specifics, with a recommendation to cross-reference the manual with the source code.

The coupling of the two codes is formed by the passing arrays of output vales of respective quantities between them. EIRENE requires the plasma background as an input and outputs neutral particles as sources and sinks for the plasma transport equations in EMC3. A schematic overview of the coupling is given in figure 2.1.

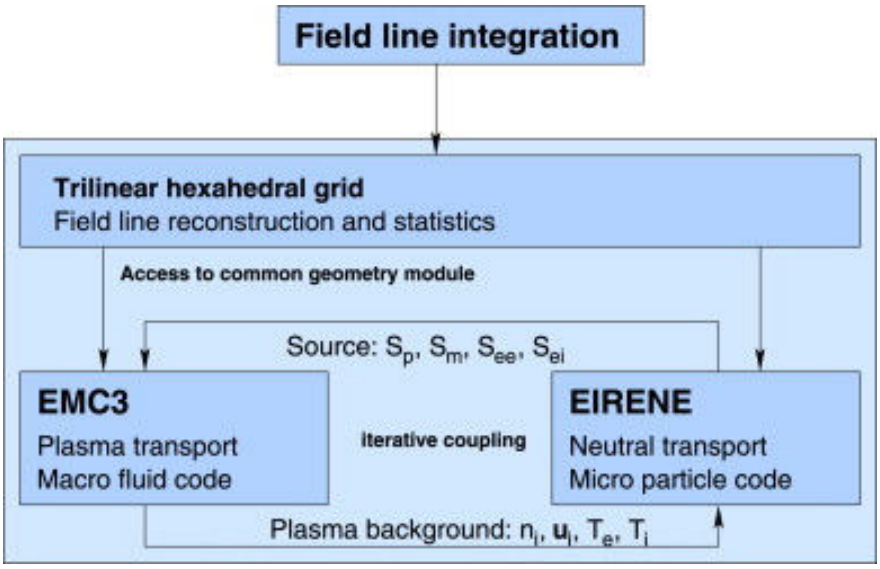


Figure 2.1: A schematic of the coupling of the EMC3-EIRENE code, from [35]. The two codes communicate by passing arrays filled with sources and sinks, iterating for a fixed number of iterations.

2.3.1 The EIRENE Model

EIRENE solves a set of coupled Boltzmann equations, one for each species, in the given geometry and outputs a set of user-defined tallies. From [29], the Boltzmann equation for the species i is

$$\frac{\partial f_i(\boldsymbol{r},\boldsymbol{v},t)}{\partial t} + \boldsymbol{v}\cdot\nabla_{\boldsymbol{r}} f_i(\boldsymbol{r},\boldsymbol{v},t) + \frac{\boldsymbol{F}(\boldsymbol{r},\boldsymbol{v},t)}{m}\cdot\nabla_{\boldsymbol{v}} f_i(\boldsymbol{r},\boldsymbol{v},t)$$
$$= \int\int\int \sigma(\boldsymbol{v}',\boldsymbol{V}';\boldsymbol{v},\boldsymbol{V})|\boldsymbol{v}'-\boldsymbol{V}'|f(\boldsymbol{v}')f_b(\boldsymbol{V}') - \int\int\int \sigma(\boldsymbol{v},\boldsymbol{V};\boldsymbol{v}',\boldsymbol{V}')|\boldsymbol{v}-\boldsymbol{V}|f(\boldsymbol{v})f_b(\boldsymbol{V}) + Q(\boldsymbol{r},\boldsymbol{v}), \tag{2.26}$$

with the source term broken up into a collisional gain, collisional loss, and external source term respectively for a given velocity interval. The two integrals clearly correspond to the classical $\frac{\partial f}{\partial t}_{\text{coll}}$, and would thus need to be summed over every collision j each with its own cross section $\sigma_{coll,j}$.

The Monte Carlo solution for the Boltzmann equation may be reconstructed from Markov chains similarly to EMC3. Using the initial distribution $Q(\boldsymbol{r},\boldsymbol{v},t)$ and the transition probability $L = T\cdot C$ with C given as a function of the condition probability densities C_k for each collision k as

$$C(\boldsymbol{r}',\boldsymbol{v}',i'\to\boldsymbol{v},i) = \sum_k \frac{\Sigma_k}{\Sigma_{tot}} C_k(\boldsymbol{r}';\boldsymbol{v}',i'\to\boldsymbol{v},i) \tag{2.27}$$

and T given along the travel distance l as

$$l < l_{max}: \qquad T(l) = \Sigma_{tot}(l)\,\exp\left[-\int_0^l \Sigma_{tot}(s)\,ds\right] \tag{2.28}$$

$$l \geq l_{max}: \qquad T(l) = \delta(l-l_{max})\,\exp\left[-\int_0^{l_{max}} \Sigma_{tot}(s)\,ds\right] \tag{2.29}$$

with $\Sigma_{tot} = \sum \Sigma_k$. The profiles may then be reconstructed with a variety of included estimators X_g evaluated at each saved history ω_i in the Markov chain by the expression

$$R \approx \tilde{R} = \frac{1}{N}\sum_{i=1}^{N} X_g(\omega_i) \tag{2.30}$$

and the exact magnitudes and dimensions recovered with a scaling factor.

2.4 Mesh Checking and Boundary Conditions

2.4.1 EMC3 Mesh Checking Routines

An EMC3-EIRENE mesh is made up of a hierarchy of structures whose opportune use is key to reducing simulation setup overhead and computational costs while maximizing accuracy of the resulting mesh.

The lowest level of organization is the 'physical cell'. A physical cell is a single field-aligned parallelepiped, consisting of eight vertices with three values each corresponding to the typical toroidal coordinate system (r, θ, ϕ) where θ represents the toroidal angle and ϕ the poloidal angle. Each physical cell has eight values for the magnetic field strength, with the direction set by the field-aligned requirement.

Multiple physical cells may be joined together into a 'plasma cell'. EMC3 computes all results and saves all outputs on the plasma cell, drastically shortening the computation time for large meshes with regions of low variability. A plasma cell typically extends in the toroidal direction most strongly, but can also be allocated manually to have a radial or poloidal extent. The default plasma cell allocation is one plasma cell to one physical cell in the r and ϕ directions and multiple physical cells to one plasma cell in the θ direction corresponding to a magnetic flux tube. HIDRA meshes are small by fusion test device standards, and as a result the default which would tend to overestimate the degree of physical cell concatenation works sufficiently well without increasing simulation times.

The highest level is the plasma 'zone', a toroidal block of perfectly-connected field-aligned parallelepipeds. Simple plasma structures may have as few as one zone for a complete simulation, although most devices use many. Boundary conditions are set per-zone, giving more fidelity where needed for plasma-facing structures, heat deposition patterns across the device, or other requirements.

Using these structures, an EMC3-compliant mesh must satisfy three conditions:

1. The magnetic flux Φ over the physical cell flux tube from the first bounding toroidal surface to the last for a single zone must not vary more than 1% from its average value. The magnetic

flux through a given toroidal surface is calculated as

$$\Phi = \frac{1}{4}\theta_{\mathrm{p}}(A_1 + A_2)(B_1 + B_2) \tag{2.31}$$

where A_1 and A_2 correspond to the area of one of the constant-θ surfaces on the physical cell (A_1 and A_2 refer to the two such surfaces in a cell), B_1 and B_2 correspond to the average field of the four points which contribute to the area, and θ_p is the pitch angle between the normals of A_1 and A_2 to correct for toroidal geometry. Then the average flux conservation is given simply by the difference between the maximum and minimum values over the flux tube divided by the average,

$$\frac{\Delta\Phi}{\overline{\Phi}} = \frac{max(\Phi) - min(\Phi)}{mean(\Phi)}. \tag{2.32}$$

2. The magnetic flux must not deviate more than 10%. That is, the highest and lowest flux values for a given flux tube must be within 10% of each other.

3. The mesh must be linear. The exact qualifier for nonlinearity refers to the trilinear hexahedra interpolation map being well-behaved (convex [35]), but non-linearity has not been a concern with EMC3-EIRENE meshes containing the proper boundary toroidal boundary conditions as enumerated in chapter 3; a nonlinearity check failure usually indicates a severe problem with the generation of the mesh on the part of the user, such as mislabeled toroidal surfaces, a bad magnetic field equilibrium, incorrect boundary conditions, or other more obvious problems.

2.4.2 EMC3-EIRENE Boundary Conditions: The Reversible Field-Line Mapping Technique

In EMC3-EIRENE, there are six general boundary conditions to apply: periodic, up-down symmetric, Reversible Field-Line Mapping, Bohm velocity, set-flux, and decay. The physical domain at the coarsest level of the plasma zone will have many separate boundary conditions which need to be set: Each toroidal and poloidal surface needs one, for a total of four. Any additional radial surfaces corresponding to limiters or divertors require a Bohm velocity boundary condition. Radi-

ally, one needs input power, momentum, and particle fluxes for every species in the plasma, and an associated decay length on the outside of the domain to represent scrape-off layer or last closed magnetic surface losses. These boundary conditions are summarized visually in 2.2.

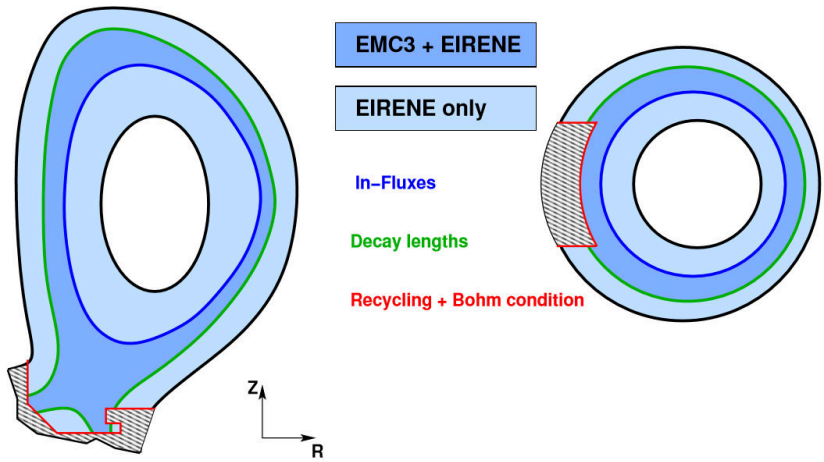


Figure 2.2: A schematic of the boundary conditions and operational zones for EMC3-EIRENE, from [35].

The periodic boundary condition is, at least in the case of HIDRA magnetic meshes, exclusively used for the two poloidal boundaries to connect the domain into a torus. At the edges of the domain which must be up-down symmetric surfaces, the up-down symmetric boundary condition is used. If the domain extends instead to a full period, or if the domain is comprised of multiple zones, the Reversible Field-Line Mapping technique must be used.

Computing the magnetic field at every point in space as the Monte Carlo particle travels around would be too computationally intensive to use for large-scale simulations. To counter this, the magnetic field is stored on a field-aligned grid, and in preprocessing the magnetic field value is integrated forwards and backwards throughout the plasma zone and stored in a 2D array representing the field lines at a point inside the sub-domain of a given physical cell [33] [31]. As particles move about the domain, the magnetic field is interpolated at their location to a high degree of accuracy. The reversible field-line mapping boundary condition indicates to the code that the two meshes of magnetic field lines stored in the code need to have their local coordinates mapped together to facilitate particle transport throughout the domain, as illustrated in figure 2.3.

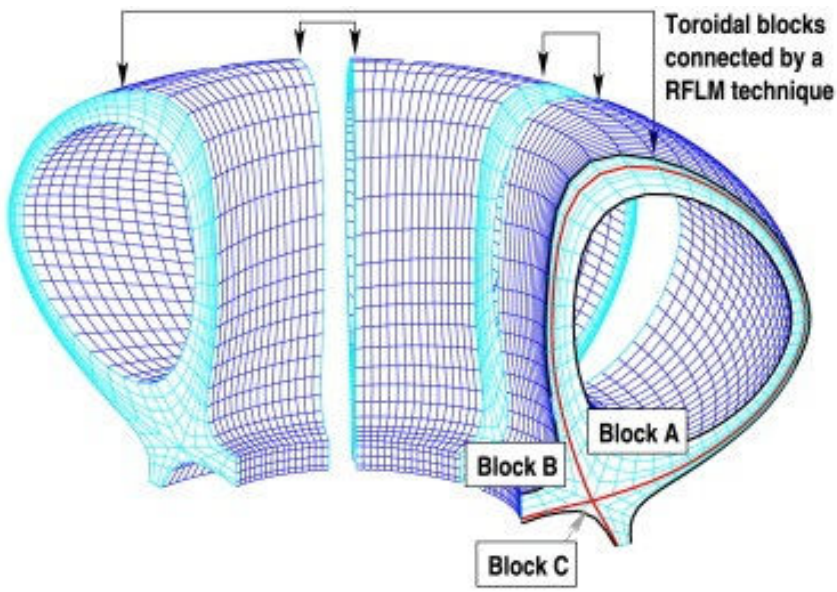


Figure 2.3: A schematic of the zone division in a typical plasma mesh, with the reversible field-line mapping boundary surfaces highlighted, from [31] [35].

2.4.3 Additional Required Mesh Inputs for EIRENE

A simulation domain for EIRENE is not limited by the same mesh requirements as EMC3 is. An EIRENE domain typically covers not only the entire EMC3 mesh but also extends out past the last closed magnetic surface to the wall and inwards into the heated region in the core. In these regions where EMC3 is not active, a plasma background must be set for EIRENE in the input files on a per-physical-cell basis, although some shorthand expressions are available to reduce the workload. Crucially this plasma background is not allowed to vary over the iteration. These external cells, shown in 2.2 as the light blue regions, will de-activate any test particle entering them. In chapter 3, this will be seen to have a consequence for the mapping losses as a result of misaligned reversible field-line mapping surfaces.

As a final note, since an EIRENE region usually extends out to the wall, information on the material type must be present in the setting of radial boundary conditions instead of flux and decay information. For toroidal, poloidal, and additional surface boundary conditions, EMC3 automatically passes its boundary conditions.

Chapter 3

Magnetic Mesh

In this chapter, the methods employed for the generation of field-aligned 3D magnetic meshes for use in the EMC3-EIREN code are discussed. The domain in which a given simulation takes place has a dominating effect on the accuracy and wall clock time of any given simulation. In EMC3-EIRENE, the simulation domain, referred to as a magnetic mesh, is particularly important as a result of constraints arising from the use of a local field-aligned coordinate system. In consideration of a magnetic mesh for EMC3, the main findings from chapter 2 are restated:

- An EMC3 magnetic mesh must be as linear as possible, with the definition given in Chapter 2.
- An EMC3 magnetic mesh must conserve the magnetic flux through the subsequent magnetic cells, according to 2.32.
- An EMC3 magnetic mesh must have perfectly-represented up/down and periodic boundary conditions. Every mesh point and magnetic field on an up/down symmetric surface must be, to computer precision, up/down symmetric. Every mesh point and magnetic field on each of a pair of matching periodic surfaces must match to computer precision. In the case where exact up/down or periodic surfaces cannot occur, the reversible field-line mapping (RFLM) technique must be used.

HIDRA as a PMI test bed will be expected to produce conditions at the plasma-material interface which may vary from experiment to experiment. The angle of incidence of the plasma arriving at the surface is influenced heavily by the incident magnetic field line inclination, potentially requiring many different magnetic configurations over the course of a single test. To account for these, a well-automated method must be devised to create the requisite magnetic meshes for use

in EMC3 to provide a computational comparison for experimental results.

In pursuit of this goal a new mesh generation utility TORMESH has been developed using the magnetic field line tracing utility `FIELDLINES` and a HIDRA-specific automatic or semi-automatic edge recognition technique. The new automatic edge recognition finds the edge by taking the gradient of the connection lengths. Due to the constraints on an EMC3-EIRENE mesh that may make these automatic methods unsuitable, a hook is provided in TORMESH to adjust the degree of ergodicity permitted in the final mesh. TORMESH then filters or splines the Poincaré section of the magnetic field lines, producing the required field-aligned magnetic mesh. TORMESH is also capable of identifying surfaces containing islands and applying different splining or filtering techniques to those surfaces in particular, although in practice forward-backward Butterworth filtering was seen to be sufficiently accurate for all types of surfaces.

The confining magnetic fields in a stellarator are present before the plasma is started and the contribution from the plasma current is much weaker than a tokamak. Thus, the concept of a vacuum magnetic field is important for an analysis of the resulting plasma equilibrium and subsequent discretization into a magnetic mesh. Although larger stellarators with higher plasma densities and temperatures like W7-AS [36] have significant deviations from these vacuum magnetic fields due to the presence of bootstrap currents, devices operating with lower densities and temperatures as HIDRA is expected to do not. If one can determine the vacuum magnetic field of HIDRA, one has the information necessary to depict the shape of the plasma equilibrium on the confining magnetic fields, which leads directly to the generation of a simulation domain for plasma simulation codes.

3.1 Vacuum Magnetic Field

3.1.1 The Rotational Transform in a Stellarator

A well-known consequence of the virial theorem [37] applied to an ideal magnetohydrodynamic fluid is that is not possible to create a fusion device that is sustained entirely by fields generated in the plasma; some external magnetic field must be applied, generated by sets of coils positioned around the volume of the plasma.

Tokamak confining fields are created chiefly by toroidal field coils positioned on constant-θ

planes around the torus, with θ as the toroidal angle. The number of coils is decided by a desire to minimize the magnetic ripples generated by the coils [38]. In classical stellarators, these fields are created by the same large toroidal coils in conjunction with smaller coils wound helically around the vacuum vessel. The smaller magnetic field from these helical coils act mostly perpendicular to the large field created by the toroidal coils, resulting in a perturbation to the magnetic field structure. These perturbations introduce a rotation, traditionally expressed by a rotational transform indicating how much the plasma is rotating in the ϕ direction (around the magnetic axis) in each pass around the torus in the θ direction. This rotation acts to stabilize the plasma against particle drifts by ensuring that magnetically-confined particles spend time on both the high-field and low-field side of the device and experience the forces in both regions.

The rotational transform in a stellarator is given by [39] as

$$\iota(r_0) = 2\pi \frac{l-1}{N} \left(\frac{R_0 B_h}{a B_0}\right)^2 \left(\frac{r_0}{a}\right)^{2l-4}. \tag{3.1}$$

with l and N the poloidal periodicity and number of helical periods respectively, R_0 and a_0 the major and minor radius of the device, B_h and B_0 the helical and toroidal magnetic fields, and r_0 the distance of a point in space from the magnetic axis. With $l = 2$, the profile is constant along the chamber (higher order terms can add back a radial contribution per [40]), and the rotational transform becomes a useful quantity for characterizing the magnetic profiles of varying discharges.

3.1.2 Representing a Magnetic Coil Assembly as a Sequence of Current-Carrying Filaments

Coil designs with analytical solutions, such as circular current loops, were used in classical stellarators and early tokamak designs but have been superseded by supercomputer-optimized non-planar coils and high-aspect-ratio D-shaped coils respectively. Though classical stellarators like HIDRA have circular toroidal coils, the rotational-transform-inducing helical coils do not have an analytical solution in toroidal geometry. For simulating these complex device classes, it is necessary to discretize the coils into easier shapes with an analytical solution that may be handled by numerical integration.

The simplest shape for building coils is a line segment; the magnetic field from a line segment of length dl can be easily found from the Bio-Savart law,

$$d\boldsymbol{B} = \frac{\mu_0}{4\pi}\frac{I d\boldsymbol{l} \times \boldsymbol{r}}{|\boldsymbol{r}|^3}, \tag{3.2}$$

where I is the current, $\boldsymbol{r}$ is the vector from the center of the segment to the point at which the field is evaluated, and μ_0 is the usual vacuum permeability. In `HIDRAmod`, the line segments are discretized by connecting segments between points evaluated on the surface of the curve corresponding to a given coil, and equation 3.2 is used directly as

$$\boldsymbol{B}_i = \frac{\mu_0}{4\pi}\frac{I \boldsymbol{L}_i \times \boldsymbol{r}}{|\boldsymbol{r}|^3} \tag{3.3}$$

where $\boldsymbol{L}_i$ is now the multiple of the segment orientation vector with its length. In this discretization, the contributions from each line segment are summed to find the total field as $\boldsymbol{B} = \sum_i \boldsymbol{B}_i$; the discretization errors come from the finite length of each line segment and an offset between the true radial distance to the curve and the distance to the finite-sized line segment. The exact discritization error depends on the geometry of the coil, which may have a complicated shape, but an approximation may be obtained by analyzing the difference in the fields as the number of segments increases.

A collection of line segments with the current directed appropriately can represent an arbitrarily complex 3D shape when the individual segments are sufficiently small. Smaller line segments can more accurately represent curved shapes, but longer segments can avoid additional storage and computational costs. The trade-off must be determined on a per-device basis by analyzing the coil dimensions and the required accuracy of the field solver. By representing all current-carrying structures as a sequence of current filaments, and providing the user with a means to easily input complex coil geometries and have them seamlessly converted into the requisite geometry, `HIDRAmod` simplifies the analytical calculus problem to one of geometry while maintaining the ability to generalize to other devices or to complex additional magnetic fields if necessary.

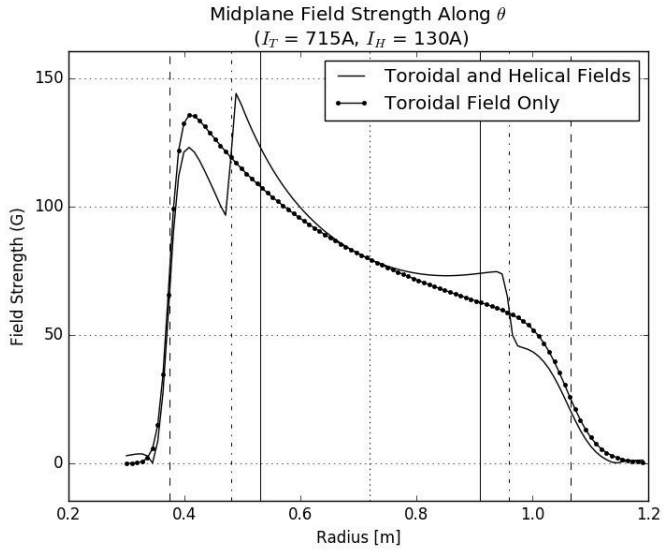


Figure 3.1: The toroidal magnetic field at the HIDRA midplane with and without helical field contributions. The figure displays: the vacuum vessel minor radius (solid vertical lines), the helical coil minor radius (dash-dot vertical lines), and the toroidal coil minor radius (dashed vertical lines). The axial magnetic field is equal to the toroidal-coil-only magnetic field, allowing a free choice of rotational transform with peaked energy deposition on the magnetic axis as long as the axial magnetic field strength remains at the cyclotron resonance.

3.1.3 HIDRA's Magnetic Field at the Mid-Plane

The magnetic field in HIDRA is chiefly determined by the toroidal field, especially in the center of the chamber, as illustrated in figure 3.1. The 2.45 GHz magnetron heating sources on HIDRA are most efficiently coupled to the core plasma when they match the electron cyclotron frequency, corresponding to an axial magnetic field of 87.5 mT. Having fixed the axial field with the toroidal field strength, any helical current is permissible within the physical constraints described in section 3 of this chapter. Operation of the high-field discharge provides identical profiles for an identical rotational transform with the magnitude scaling accordingly. Individual components of the magnetic field are presented in figure 3.2.

3.2 Field Line Tracing

3.2.1 The Magnetic Field Line ODE

An ODE for the tracing of magnetic field lines about a torus is commonly given as

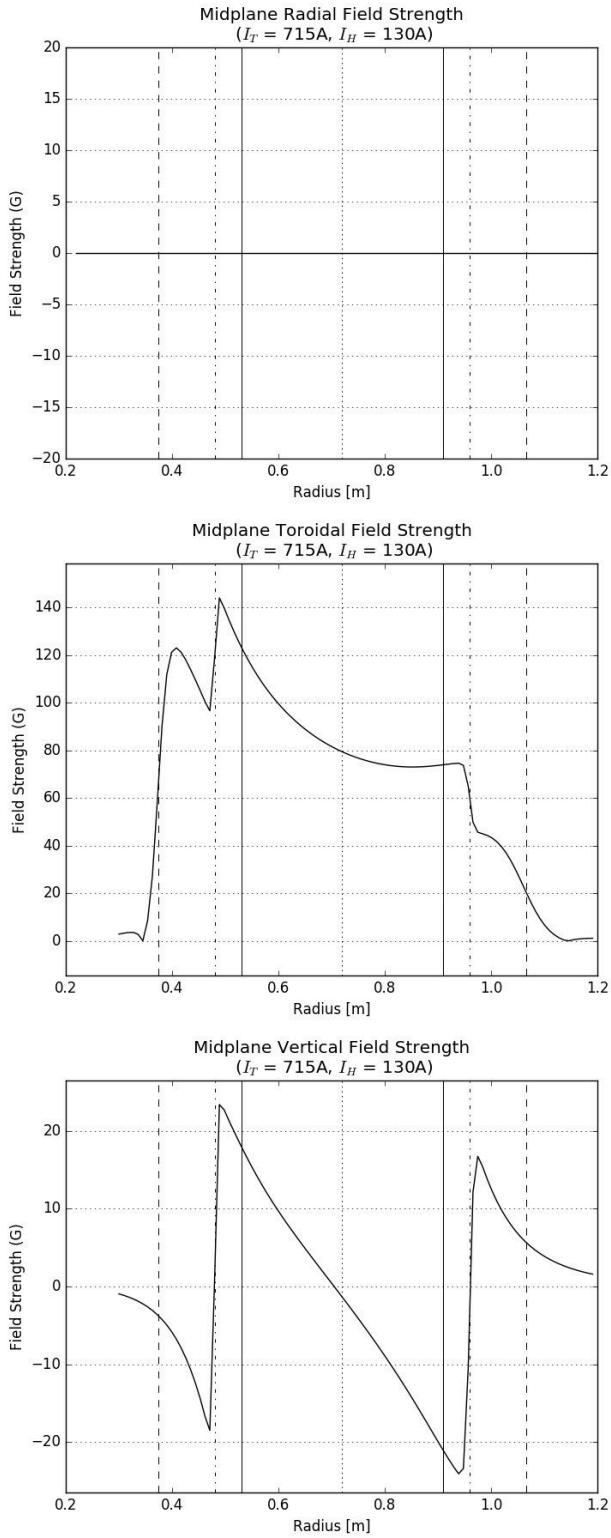


Figure 3.2: Vacuum magnetic fields in HIDRA on the midplane at a toroidal coil. This position also has up-down symmetric helical coil assemblies, resulting in mostly-canceled radial field components. The rotational transform is represented by the significant vertical field component at this location. High slope areas result from singular field calculations near coils. Information on the meaning of vertical lines is given in 3.1.

$$\frac{dR}{d\phi} = R\frac{B_R}{B_\phi}, \tag{3.4}$$

$$\frac{dZ}{d\phi} = R\frac{B_Z}{B_\phi}. \tag{3.5}$$

Provided with an initial position (referred to as a seed point), these coupled ODEs describe the motion of a magnetic field line in three dimensions in toroidal geometry. By following many such seed points the magnetic structure of a given device may be visualized. To numerically compute the trajectory of the field lines, the magnetic field and necessarily the equation itself must be discretized.

3.2.2 `FIELDLINES` and Three-dimensional Splining for Fast Field Reconstruction

To solve for the vacuum magnetic field, `HIDRAmod` has included the `FIELDLINES` code from the STELLOPT suite of plasma physics codes. `FIELDLINES` is a package for following magnetic field lines originating from a prescribed magnetic coil geometry, with user-defined field line ODE integration methods and control over step sizes, ODE integration tolerances, and the ability to add a controlled amount of magnetic diffusion to simulate magnetic field stochasticity.

Similarly to the reversible field-line mapping technique used in EMC3, a magnetic field solver for the large volume of a stellarator needs some considerations to computational efficiency in order to run in a timely fashion. The particle analog is the magnetic field line, following the magnetic fields around the vacuum vessel.

3.2.3 Methods of Discretization

To follow the magnetic field lines as they move around the vacuum vessel, `FIELDLINES` incorporates three different methods for integrating the ODE problem of equations 3.4 and 3.5: LSODE (Livermore Solver for Ordinary Differential Equations), RKH68 (Runge-Kutta-Hutta sixth-order eight-step), and NAG D02CJF (a variable-order variable-step Adams method). As D02CJF requires a license, its usage is not desirable from a portability standpoint and was omitted from the

development of `HIDRAmod`.

LSODE

The LSODE method [41] solves the LU factorization of a matrix defined by an Adams method (nonstiff) or backward differentiation formula method (stiff) for the ODE. LSODE is a variable-order, variable-step method which automatically adjusts both to maintain a user-specified accuracy. The general linear multistep method is given by

$$\boldsymbol{Y}_n = \sum_{j=1}^{K_1} \alpha_j \boldsymbol{Y}_{n-j} + h_n \sum_{j=0}^{K_2} \beta_j \boldsymbol{f}_{n-j} \tag{3.6}$$

where $\boldsymbol{Y}$ is the solution vector, $\boldsymbol{f}$ is an approximate derivative to $\boldsymbol{Y}$, and the rest of the coefficients are set per-method. For order q, the Adams method has $K_1 = 1$ and $K_2 = q - 1$, and for the backward differentiation formula $K_1 = q$ and $K_2 = 0$. The coefficients α and β are given by [42] for $q \leq 6$, and given in [41] for the rest in terms of the final implemented equations. The Adams method may vary up to order 12, while the backward differentiation formula may vary up to order 5. LSODE attempts to set the maximum step size for a given order in order to maximize efficiency, and the error correspondingly varies with the order as

$$\boldsymbol{d}_n = C_{q+1} h_n^{q+1} \boldsymbol{y}^{q+1}(x_n) + O(h_n^{q+2}), \tag{3.7}$$

with x_n the position vector and the constants C_{q+1} are given in [42]. Since it is a variable order method, the local truncation error may change each step to keep the required precision.

RKH68

The RKH68 method [43] is a Fortran implementation of an improved Runge-Kutta sixth-order eight-step method, solving the ODE with the expression

$$y(x_0+h) = y_0 + \frac{1}{840}\left[41(k_0+k_7) + 216(k_2+k_6) + 27(k_3+k_5) + 272k_4\right], \tag{3.8}$$

with

$$k_0 = hf[x_0,\ y_0], \tag{3.9}$$

$$k_1 = hf\left[x_0 + \frac{1}{9}h,\ y_0 + \frac{1}{9}k_0\right], \tag{3.10}$$

$$k_2 = hf\left[x_0 + \frac{1}{6}h,\ y_0 + \frac{1}{24}(k_0 + 3k_1)\right], \tag{3.11}$$

$$k_3 = hf\left[x_0 + \frac{1}{3}h,\ y_0 + \frac{1}{6}(k_0 - 3k_1 + 4k_2)\right], \tag{3.12}$$

$$k_4 = hf\left[x_0 + \frac{1}{2}h,\ y_0 + \frac{1}{8}(k_0 + 3k_3)\right], \tag{3.13}$$

$$k_5 = hf\left[x_0 + \frac{2}{3}h,\ y_0 + \frac{1}{3}(-4k_0 - 21k_1 + 46k_2 - 29k_3 + 10k_4)\right], \tag{3.14}$$

$$k_6 = hf\left[x_0 + \frac{5}{6}h,\ y_0 + \frac{1}{72}(-8k_0 + 99k_1 - 84k_2 + 44k_4 + 9k_5)\right], \tag{3.15}$$

$$k_7 = hf\left[x_0 + h,\ y_0 + \frac{1}{82}(107k_0 - 243k_1 + 354k_3 - 172k_4 - 36k_5 + 72k_6)\right], \tag{3.16}$$

and with an error of

$$\frac{9}{1400}\left(\frac{h}{6}\right)^9 f^{(8)}(\overline{x}). \tag{3.17}$$

LSODE and RKH68 may be used interchangeably for `HIDRAmod`. Performance for similarly-sized tasks is indistinguishable for the purposes `FIELDLINES` is utilized for. Given the higher potential truncation order and adaptable error, the LSODE method is preferable and was used during the development of `HIDRAmod`.

Both integrators used are multi-step integrators, which are non-symplectic (dissipative). For the sensitive task of finding magnetic flux surfaces, the error must be set sufficiently low to prevent altering of the flux surfaces. Because of the normalization to the toroidal field in equations 3.4 and 3.5, insufficient accuracy only results in a dissipative integration in the helical direction, and a

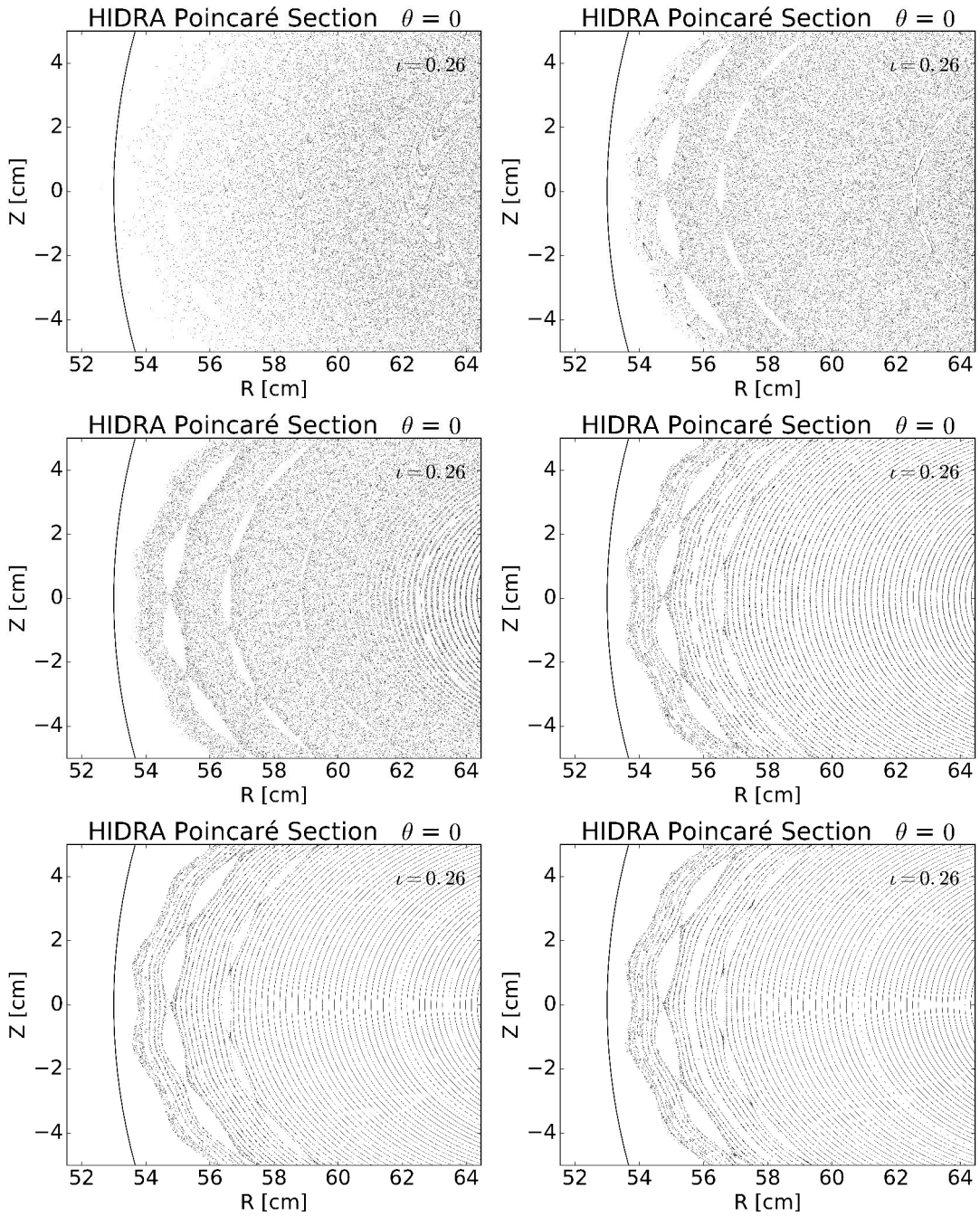


Figure 3.3: An illustration of the errors encountered by tracing flux lines with an insufficiently tolerant LSODE method. From left to right and top to bottom, a tolerance of 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}, and 10^{-11}. The final two Poincaré sections show no significant difference after processing; the 10^{-8} Poincaré section has degraded surfaces from dissipation, which become apparent after the thousands of passes through the same plane. In an EMC3 mesh which requires less than a full revolution about the chamber, such dissipation may be noticeable if the mesh size is small. Improving the accuracy from 10^{-9} to 10^{-10} results in a minor increase in accuracy. For the meshes presented herein, a tolerance of 10^{-9} was used, corresponding to the lowest tolerance which gives well-defined surfaces.

tolerance an order larger than used to construct the grid produces a sufficiently well-behaved grid as per 3.3.

3.3 Poincaré Section

A Poincaré map is a map of a periodic orbit in some higher-dimensional state space to a lower-dimensional subspace. This lower-dimensional subspace, the Poincaré section, retains information about the trajectory as the orbiting entity returns over the course of many orbits, showing the allowed regions of the state space that the entity may inhabit. The Poincaré section is perpendicular to the direction of motion of the field lines, which in the case of toroidal fusion devices with predominantly toroidal fields implies that it is a toroidal cross-section of the device.

Due to the influence of the helical fields in the case of classical stellarators, or due to the complex shaping of the coils in the case of advanced stellarators, magnetic field lines that are followed around the vacuum vessel move around the poloidal plane at different toroidal positions. The application of a Poincaré map to these ergodic field lines allows the extraction of nested flux surfaces which serve as opportune seed points for the tracing of field lines for a field-aligned magnetic mesh. Flux surfaces are named such as they satisfy the condition of zero perpendicular flux, and thus the flux enclosed by them remains constant. Of particular importance to EMC3, the field-aligned grid generated by connecting seed points on neighboring magnetic flux surfaces correctly preserves parallel and perpendicular local coordinate separation.

Creating a magnetic grid that fulfills the required boundary conditions requires knowledge of certain poloidal planes that correspond to symmetry in the plasma. The act of extracting a poloidal plane at a given toroidal angle, including many passes through the plane of each individual line, creates a Poincaré Section. Certain Poincaré sections will have vertical symmetry and may be used to create the symmetry surfaces of a magnetic grid.

Each successive flux surface also gives an indication of the magnetic structure seen by the plasma as it is transported outwards, and the quality of the surfaces limit the confinement of the plasma. The magnetic axis, a line around which the flux surfaces are nested, is the region of highest confinement. Radially outwards from the magnetic axis, the flux surfaces gradually get thicker and less well-defined. Eventually the structure of the surfaces break down and the field

lines are unconfined. In stellarators there may also be surfaces where the field lines bifurcate or are nested around a line which is not the magnetic axis. Termed magnetic islands, the existence and mitigation of these structures is important to understanding the confinement quality of a stellarator discharge.

3.3.1 Accessible Rotational Transforms and Poincaré Sections in HIDRA

As reported in [23], the high-field operation is limited to $\iota = 0.4$ due to current and heating constraints. This suggests that in the low-field case where the current supply is not an issue, any ι which leads to a reasonably confined discharge may be simulated, and comparisons between similar ι with differing toroidal field intensity are valid so long as the ι of the discharge is below the permissible limit in the high-field case.

In the present work, tests of `HIDRAmod` have been run at a more relaxed $\iota = 0.318$. For comparison, three Poincaré sections will be presented for each ι, corresponding to toroidal angles $\theta = 0$, $\theta = \pi/10$, and $\theta = \pi/5$ of a single period. As HIDRA is an $l = 2$, $m = 5$ stellarator, the remaining evenly spaced period locations of $\theta = 3\pi/10$ and $\theta = 4\pi/10$ correspond up-down symmetrically to the $\theta = \pi/10$ and $\theta = 0$ profiles. Poincaré sections for $\iota = 0.26$ (the smallest discharge confined within the vacuum vessel), $\iota = 0.318$, and $\iota = 0.4$ are presented in figure 3.4.

3.4 Splining and Filtering Techniques

Magnetic surfaces on the Poincaré sections have some thickness which must be understood in the context of a discrete magnetic grid. In order to reduce the ergodic volume of the flux surface to a single smooth line in the Poincaré section as in figure 3.5, splining and filtering techniques are necessary.

`HIDRAmod`, as a Python code, currently uses the splining and filtering routines in the scipy [44] package which were found to sufficiently approximate the plasma structure on a relevant length scale for the simulation of a HIDRA discharge (hereafter the implementation of the two techniques shall collectively be referred to as surface processing). Surface processing only occurs at the plane which provides the seed points for a given magnetic field trace to preserve the flux-aligned structure of the magnetic mesh naturally resulting from tracing the field lines.

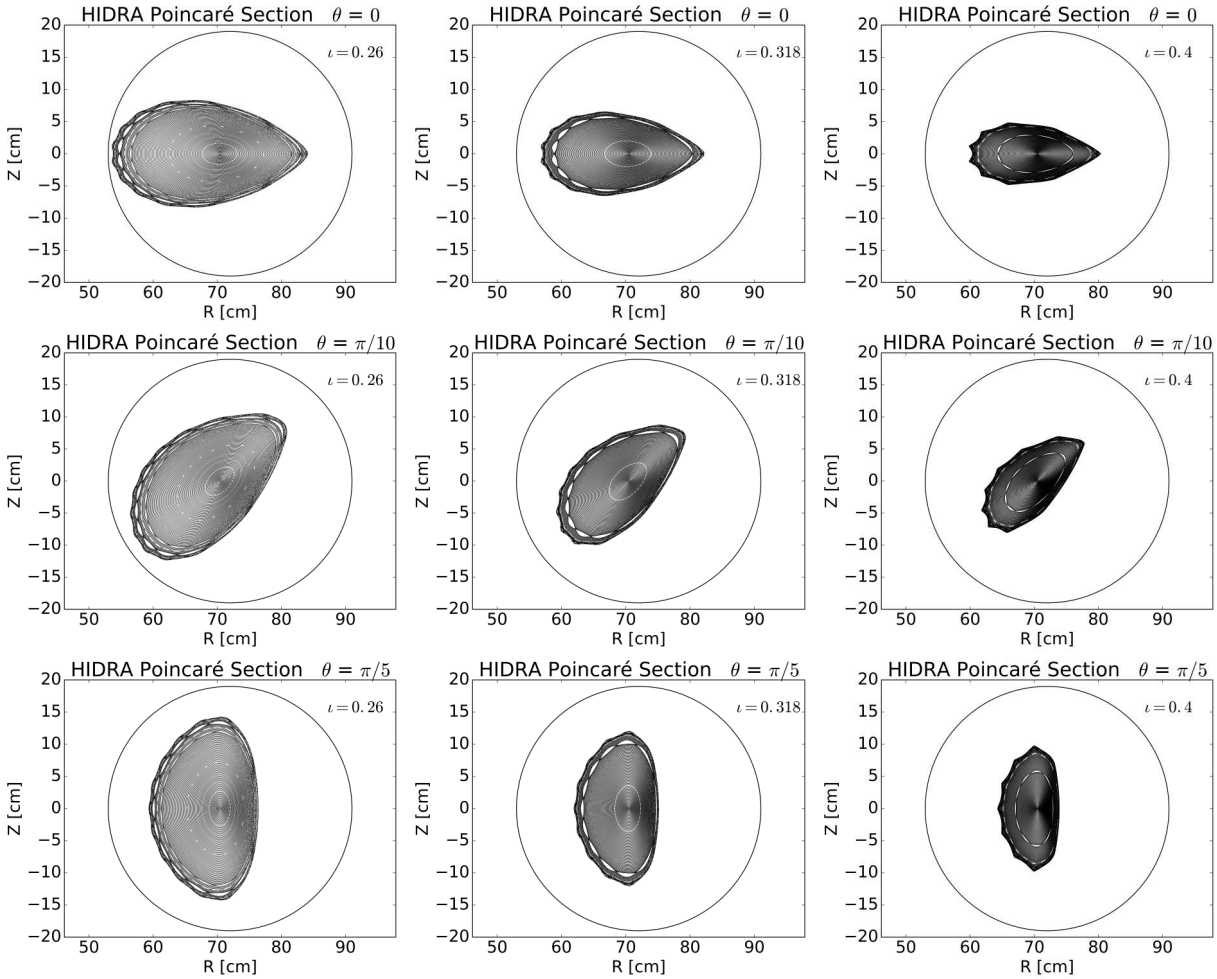


Figure 3.4: Poincaré sections for $\iota = 0.26$ (left), 0.318 (middle), and 0.4 (right). The minimum ι is set by a desire to fully confine all flux surfaces.

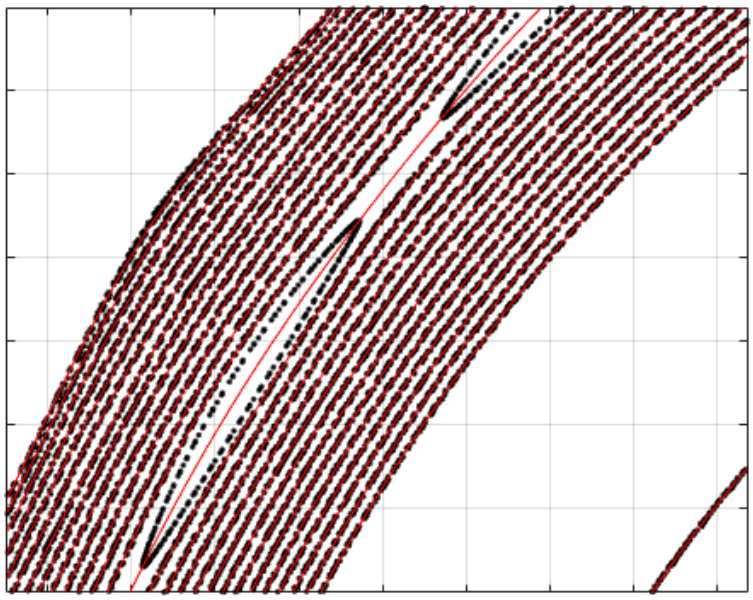


Figure 3.5: A close view of the resulting flux surface from a Poincaré section. Larger, more strongly shaped islands tend to exist only near or past the ergodic-stochastic transition. Smaller islands as pictured here do not cause issues in `HIDRAmod`.

3.4.1 Filters

A filter in a general sense removes some unwanted information from a dataset or feed of information. A single surface on a Poincaré set can be seen to have a low-frequency and high-frequency component as the shape of the magnetic surface and the oscillation of the points about that shape respectively. In this sense the solution to the problem of finding the shape of the magnetic surface is as trivial as removing only the high-frequency component. The solution lends itself to obvious discovery by filtering.

Forward-Backward Filtering

A forward-backward filter is a zero-phase filter implemented by recursively filtering a dataset both forward and backward. In doing so the phase response is eliminated and the amplitude response is squared. An n-th order Butterworth filter with gain function [45]

$$G(\omega) = \frac{1}{\sqrt{1+\omega^{2n}}} \tag{3.18}$$

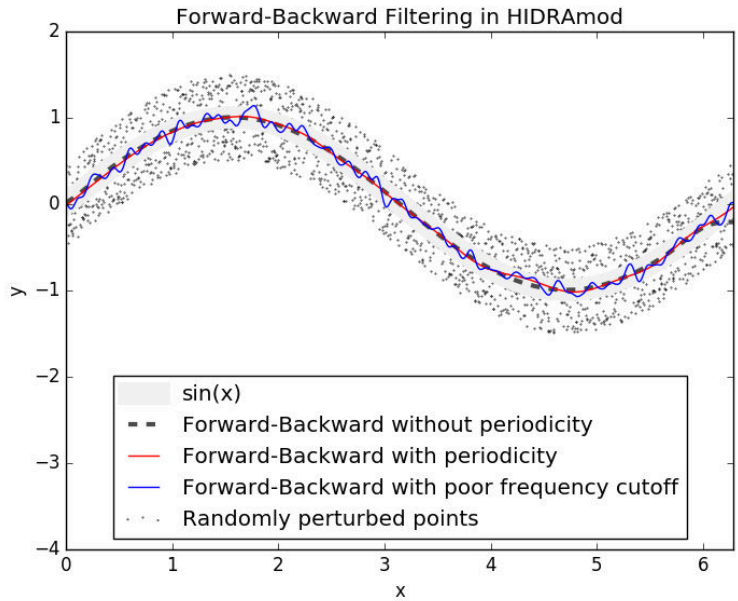


Figure 3.6: A demonstration of the Butterworth forward-backward filter used in `HIDRAmod`. Without significant effort, the Butterworth filter provides a good estimate of the flux surface location. Care must be taken to ensure that periodicity is enforced.

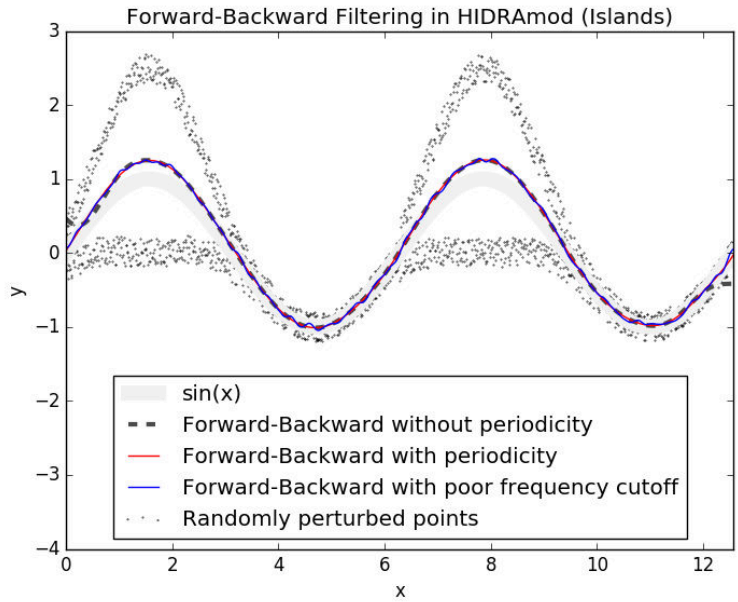


Figure 3.7: A demonstration of the Butterworth forward-backward filter used in `HIDRAmod`, with islands. Without any additional considerations from 3.6, the islands are well-fit.

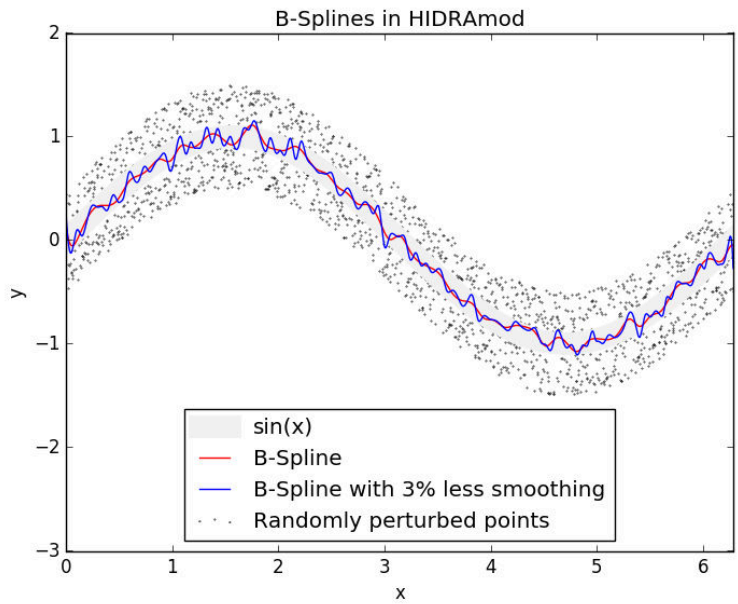


Figure 3.8: A demonstration of the splining used in `HIDRAmod` (scipy.interpolate.splrep()). The splining method is very sensitive to the smoothing factor which must be changed frequently, and has a phase response in the current implementation. Since the Butterworth filter works well without any changes from flux surface to flux surface, no effort was spent on improving the implementation.

is created with scipy.signal.butter(), and the coefficients are used in scipy.signal.filtfilt() for the forward-backward filtering. This forward-backward filter is very effective at removing the unwanted parts, and the presentation of a curve with significant deviation from a good value took an intentional effort as displayed in figure 3.6. Without any additional consideration, the forward-backward filter handles islands extremely well as in figure 3.7.

3.4.2 Splines

A spline is a smooth polynomial function used to evaluate potentially unknown functions between known values. An application of splines to the magnetic surfaces of a Poincaré set would ideally produce a smooth curve which closely approximates the shape of the surface due to an abundance of data points. To implement a spline, the spline representation of the curve of the magnetic surface must first be found, by a method such as linear least squares. Desired data may then be evaluated directly through the functional representation.

`HIDRAmod` nominally uses the scipy.interpolate.splrep() function based on the algorithm originally presented in [46], but can be trivially extended to any other methods in the scipy package as

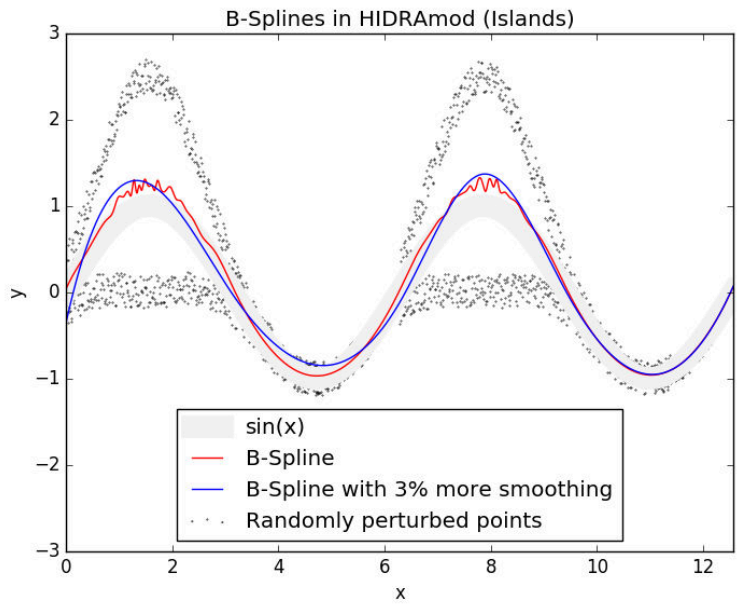


Figure 3.9: A demonstration of the B-spline used in `HIDRAmod`, with islands perturbed unevenly from the base points. The smoothing factor must be increased by approximately 400% to handle the switch to a profile with islands. The same smoothing factor would not work in the case with no islands, as shown in figure 3.8.

desired. For the simple problem posed by the splining of a periodic surface, the splrep() function does poorly without very precise smoothing, and it is difficult to get a consistent smoothing factor for a clean flux surface that works for the rest of the surfaces, as seen in 3.8 and 3.9.

3.4.3 Edge Recognition

EMC3-EIRENE has the ability to simulate a plasma in any unheated regions. In the case of HIDRA, as seen in the previous section, field lines diverge so rapidly outside of the last closed magnetic surface (LCMS) that maintaining a high degree of flux conservation is exceedingly difficult; while in principle it would be possible to simulate the unconfined region with single cells mapped to each other instead of RFLM blocks (reworded, an RFLM block that is only one cell long), the programming overhead and additional computational effort required for a questionable benefit make this undesirable. It would be necessary to implement such for non-limited discharges, but as a PMI device the case of a limited plasma is more interesting as the surface may be bombarded in a controlled fashion at higher energies.

In a limited plasma the Bohm radial boundary condition belongs to a surface which may be

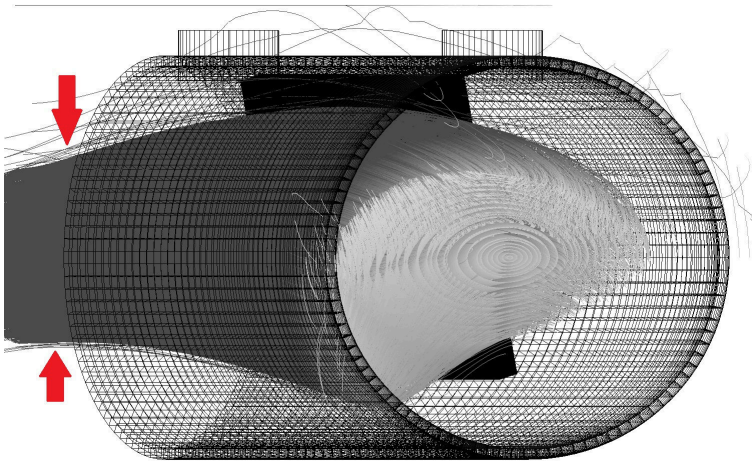


Figure 3.10: A depiction of one period of HIDRA field lines inside the vacuum vessel. Two ports and a test limiting surface are visible. There are many field lines outside of the last closed magnetic surface which still follow the plasma volume closely, and are most readily visualized on the top and bottom of the rear symmetric surface near the arrows. Separation of these lines from the confined lines automatically requires following the field lines for large number of turns.

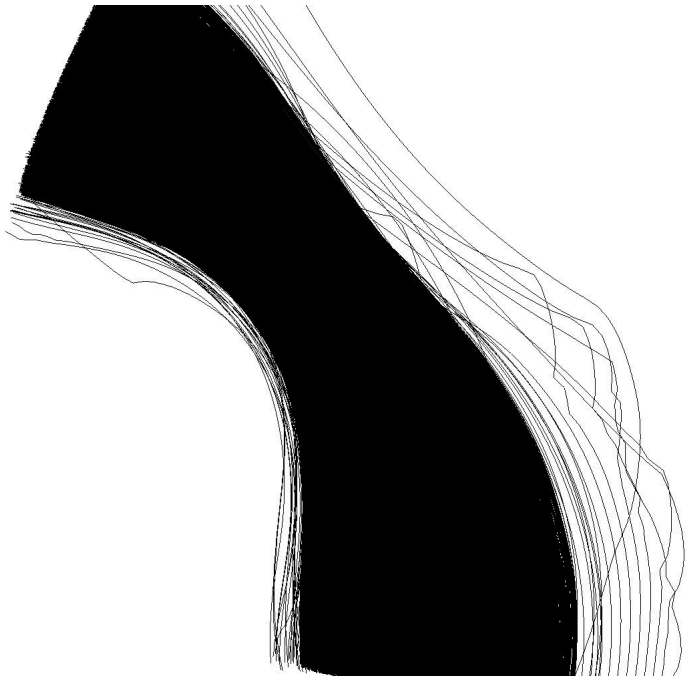


Figure 3.11: A top-view of the field lines in HIDRA, illustrating the stochasticity of field lines outside the LCMS. Some field lines on the inboard side of the plasma are very close to confinement, but an EMC3 grid which attempted to include them would likely violate the nonlinearity constraint without many RFLM surfaces.

moved inside the plasma volume. Relatively speaking the plasma which is not hitting the limiting surface and instead leaves the LCMS and diffuses to the walls is of little interest, except in the calculation of overall discharge efficiencies and return fluxes of neutrals and impurities. If the magnetic mesh is kept to the confined region inside the LCMS the requirements for a field-aligned and flux-conserving mesh become much easier to fulfill.

Defining the LCMS in the context of an EMC3 magnetic mesh requires reducing the smooth ergodic-stochastic transition to a single computational surface. For computation on tokamaks, the LCMS may be defined by identification of the separatrix which is an exact line (in real devices, the LCMS will be partially ergodic). In stellarators there is no such point relating to the confined plasma volume, but instead a gradual transition exists between the inner ergodic lines and outer stochastic lines over a small radial distance near the edge of the confined plasma. As visualized in figures 3.10 and 3.11, the ergodic-stochastic transition may be subtle.

`FIELDLINES` has a built-in method for finding the plasma edge in its '-full' flag (appendix A). The program first performs an edge search by placing many seed points along a line on the midplane and following them for many turns around the torus. Any field lines that diverge will end up moving very far from the magnetic axis or may even hit the coarse magnetic mesh boundary, in which case they are considered to be 'outside' the LCMS with all other lines inside. Then, resizing the seed points requested by the user into the interior of the plasma only, the field line trace is run again.

If such a method is used the points belonging to the outermost surface may grow to a large radial extent, especially if the lines are not followed for a suitable length, due to the increasingly stochastic nature of the magnetic fields at the transition. In this case, the field lines followed from seed points belonging to a computational surface filtered from this semi-stochastic region may be too stochastic to use in an EMC3 magnetic mesh. This transition, and the associated problem of assigning a specific value for the transition region, may be illustrated by the connection lengths as in figure 3.12.

Two options exist to remedy this; one may either elect to create an entirely new method to find the edge based on iterative optimization of the radial distance of the LCMS while the resulting filtered surface conforms to the EMC3 mesh nonlinearity requirements, or one may use a surface

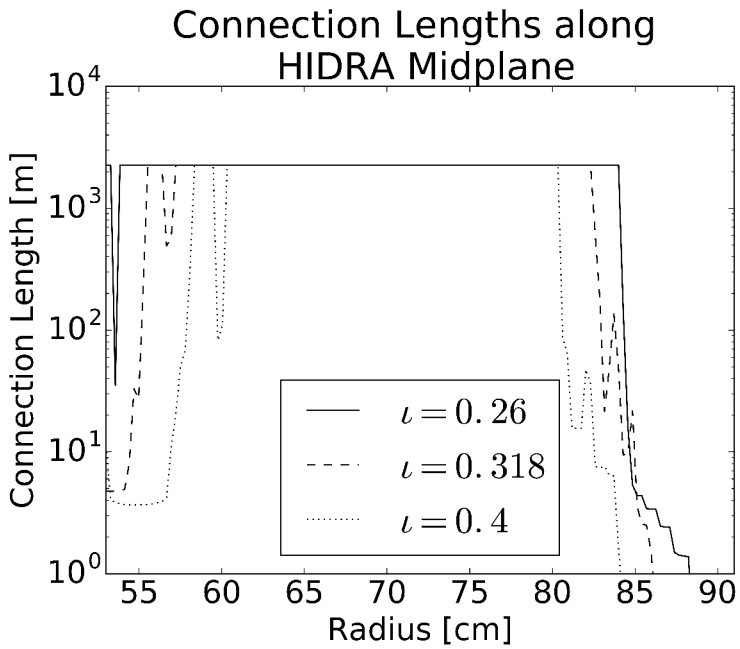


Figure 3.12: An illustration of the connection lengths for each ι considered, up to a maximum of 500 toroidal turns, over the midplane of deivce from wall to wall. The connection length belonging to a point in space is the distance along the traced field lines from that point to the vacuum vessel. As ι increases, the edge ergodic-stochastic transition becomes more broad

which is sufficiently inward of the LCMS to satisfy the requirements by iterating manually. For the current version of `HIDRAmod`, the latter was chosen as it requires no development time. As a full grid for HIDRA may be produced in a few minutes and tested in a similar time, iterating manually to find the edge is rapid enough to remove any excessive burden on the user. Future developments towards a fully-automatic mesh generation tool will be discussed in the conclusions in chapter 6.

In `HIDRAmod`, a similar implementation to `FIELDLINES` is included. A number of test seed points are introduced, followed for a large number of turns, selectively discarded based on divergence (specifically, a field line which leaves the vacuum vessel), and then resized for the computation of the final magnetic mesh. To accommodate for stochasticity in the ergodic-stochastic transition layer, a user-defined 'edge percent' may be chosen to restrict the value used for the LCMS radially inward into the confined region. In practice this ends up being a few percent of the distance between the magnetic axis and the found LCMS, which corresponds to a few millimeters of physical distance. In the highly diffusive edge region of a classical stellarator, the changes in plasma profiles are not currently sufficiently significant on this scale to warrant the additional development time.

3.5 Construction of 3D magnetic surfaces

When a Poincaré section is processed and traced through the vessel over an angular region, the resulting 3D topology is a roughly toroidal surface. If a collection of these surfaces have the same number of points in each toroidal plane and the points are roughly co-located around each surface, they may be connected to form a coherent magnetic mesh. A mesh made in this fashion will be an approximation of the true magnetic structure, following from the errors in processing the seat points, tracing the field lines through the ODE, and the usual computer-exclusive rounding and truncation. The errors aside from rounding and truncation are quantified in a self-consistent context by EMC3 itself in a number of checking routines for mesh nonlinearity, deviation, and flux conservation detailed in chapter 2. EMC3 does enforce rounding error indirectly in the creation of the magnetic mesh, as a mesh with fine detail but insufficient decimal precision will have poor flux conservation.

Depending on the class of the simulated machine the magnetic mesh has a different geometry and associated boundary conditions. Tokamaks may have toroidal blocks connected by periodicity and RFLM as in figure 2.3, while stellarators have up-down symmetric surfaces at periodic locations divided into many smaller regions by RFLM. Asymmetric or aperiodic devices must have the entire vessel simulated, which may lead to significant computational time increases from the large mesh required.

For a general stellarator, the usual method to creating a magnetic mesh (for a symmetric, periodic magnetic configuration) is to recognize that each period of the device has three up-down symmetric surfaces, one at the end of each period and one midway between each. The magnetic mesh is constructed toroidally from one of the ends of the period to the mid-period with each bounding surface an up-down symmetric surface. It may be further subdivided toroidally as accuracy requires for preserving mesh nonlinearity and flux conservation and for reducing misalignment on the RFLM surfaces.

3.5.1 Misaligned RFLM Surfaces

Due to the ergodicity of the magnetic field lines, it is not possible to create a one-block mesh for a stellarator. Each up-down symmetric surface would no longer be exactly up-down symmetric

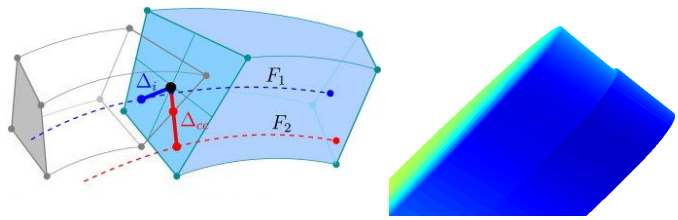


Figure 3.13: An illustration of mismatched RFLM surfaces caused by field line ergodicity. Left, a general per-physical-cell illustration of RFLM mismatch from [31] that shows the offset vectors. Right, a view of the two toroidal surfaces adjacent to the RFLM surface in a HIDRA mesh.

after being traced to the next location. Instead at least two blocks must be created, one from each up-down symmetric surface, and traced towards each other to a surface at the same toroidal angle for each block. The surfaces may then be joined by the RFLM technique. With cumulative errors from the ergodicity of the field lines, the ODE solvers, and numerical precision, the nested flux surfaces as traced from each up-down symmetric surface will not match up perfectly.

In the case of HIDRA, a relatively small device with potentially high shear, significant misalignment may occur on the order of a plasma cell or larger as illustrated in figure 3.13. When an entire flux tube terminates at the RFLM joining surfaces, it acts much like a short-circuit to ground in an electrical circuit; with nowhere to continue, particles traveling through flux tube at the RFLM surface exhaust into whichever non-simulated region is beyond it, losing the information it contains. To counteract this, a ghost cell with a larger radial extent than a typical cell may be introduced on the interior of the plasma. With a larger radial extent the majority of the cell overlaps with its neighbor across the boundary surface instead of overhangs, and the majority of the energy is retained instead of being lost. In full runs, the lost energy through the mapping surfaces is on the order of one percent with ghost cells, but can be on the order of ten percent or more without.

3.5.2 HIDRA Magnetic Meshes

A magnetic mesh with $\iota = 0.318$ with varying radial, poloidal, and toroidal discretizations was created for the grid refinement study presented in chapter 4 and and plasma profile exploration in chapter 5. Two versions were created, a high- and low-field version , which are identical and differ only by their axial field strengths of $B_0 = 0.5$ T, $B_0 = 87.5$ mT respectively. Similarly, the meshes with differing cell counts have the same designated EMC3 and EIRENE regions, and only

the number of discretizations internal to each region changes.

Of special consideration for EMC3-EIRENE, beyond the field-aligned mesh, is the necessary division between EMC3-EIRENE regions and EIRENE-only regions. EMC3 cannot be used in a region with heat deposition, and its boundary conditions most easily extend to surfaces with uniform properties, so the EMC3 region is typically set to notable structures like the LCFS in tokamaks. In HIDRA the inner boundary was set at 75% of the magnetic axis - LCMS distance, and was sufficient to allow for the movement of impurities around the torus unaffected by the core boundary.

3.6 Limiter Occlusion: Determining the Intersection Volume Between Two Meshes

In addition to the generation of a mesh for the plasma volume, the boundary conditions for `HIDRAmod` must also be set semi-automatically. While the bulk boundary conditions that mark the six edges of the domain are easily accessed through single values in the EMC3 input files, the Bohm velocity boundary conditions which are set at the surface are not necessarily in line with the physical cells of the EMC3 mesh. The system built into EMC3 to handle this case is simple in theory: the user is required to produce a list of cells which are not able to be accessed by the plasma, and EMC3 will automatically handle the problem by internally excluding those cells in the same fashion as it does the cells which may only be used by EIRENE. Although EMC3 has some shorthand methods to set large numbers of physical cells in this list at once as long as they are continuous blocks in some direction, when the surface does not perfectly align with these methods the resulting file grows to an unmanageable level. To rectify this, an automatic method is sought to write this input file for an arbitrary limiting surface shape.

3.6.1 Limiter Occlusion via 2D-Constrained Ray Tracing

The ultimate goal of the limiter occlusion tool developed in the course of this work is to identify the physical cells which are outside the plasma. To do so, we must first know what is inside the plasma, and then find what is not inside. The core of the plasma, specifically the magnetic axis,

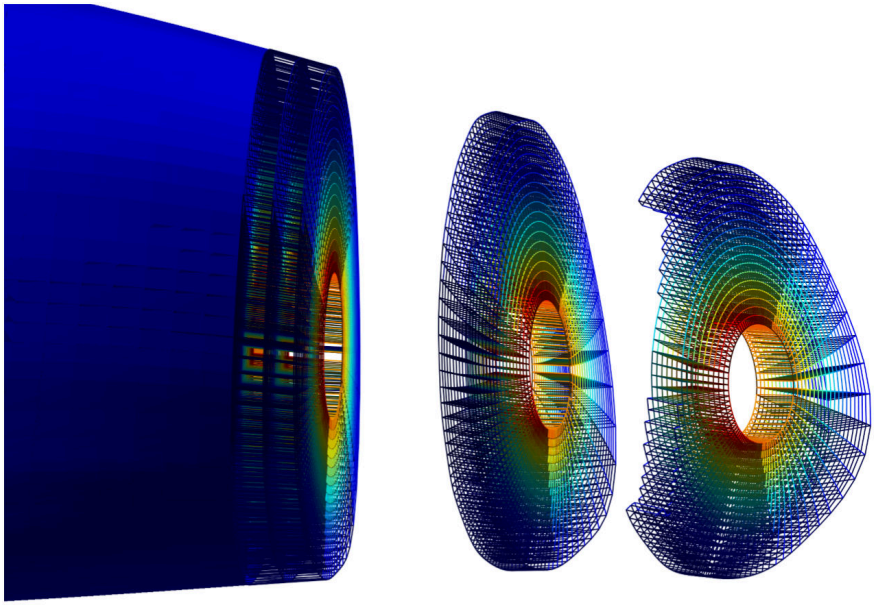


Figure 3.14: An example of the HIDRA plasma after the limiter cells have been removed. To prevent image clutter and highlight the limiter boundary, additional full toroidal cell blocks have been removed.

must be inside the plasma. If the surface is properly constructed and is either an enclosed polygon or creates an enclosed polygon via an intersection with the vacuum vessel, then the act of tracing a ray between the center of a physical cell which is inside the limiting surface to the magnetic axis must produce an odd number of intersections. Similarly, the act of tracing a ray between a physical cell outside the limiting surface must produce an even number of intersections, with the most common result being zero intersections.

3D ray-tracing is complex and very computationally intensive, but the act of finding the intersection between a line and a surface is trivial in comparison. Even so, it may be noted that the majority of plasma limiting surfaces are more or less parallel or perpendicular to the plasma and contain very simple shapes on the order of the EMC3 mesh size. That is, there are no limiters which have features on the millimeter or micrometer scale which are important to the the calculation of the fluid moments in EMC3, and in any case the details of such interactions would be washed out by the relative coarseness of the mesh. Since the surfaces are simple, the problem may be constrained to a 2D problem in a constant-θ plane, and the problem drops in complexity from the intersection of a line with a plane to the intersection of two line segments. The resulting problem, sketched in a cartoon in figure 3.15, is trivial in complexity and has an execution time of

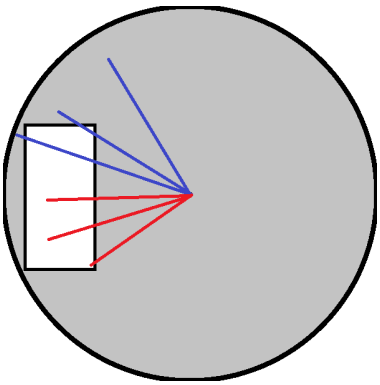


Figure 3.15: A cartoon of the limiter occlusion utility for completing the mesh generation and boundary condition generation. The gray volume represents unoccluded cells, and the white volume occluded cells. All of the blue lines (top three) come from the unoccluded volume and will cross the boundary an even number of times, while all of the red lines (bottom three) come from the occluded volume and will cross the boundary an odd number of times. This holds even in complex limiter geometries where a ray-trace to the magnetic axis would cause the ray from an occluded cell to leave, then reenter, then leave the occluded volume, or would cause a ray-trace from an unoccluded cell to enter, then leave the occluded volume.

under a second assuming proper coding and a reasonably size grid (under a 10^6 physical cells). An example of the resulting mesh occlusion is given in figure 3.14.

The user must also decide from where on each physical cell the ray should be started. EMC3 does not enforce any positioning of the occluded cells relative to the limiter surface and is fine using corners, face centers, or cell centers. However, the boundary conditions that EMC3 passes to EIRENE include a list of near-surface cells with the particle or heat flux onto that region; EIRENE then uses these cells to start the neutral particles. The neutral particles must start from a surface, but they are given in terms of a volume by EMC3, so EIRENE attempts to find a surface inside the volume to start a given particle on. Thus the volume of every cell must significantly overlap with the limiting surface, or the particles will not be able to be generated. Having a small fraction of cells that do not sufficiently overlap is allowed by EIRENE, but too many will crash the simulation. Since face-based or vertex-based occlusion may result in small intersection areas inside the cell volume, it has proven most robust to use geometric-center-based occlusion.

3.7 Summary of Software Produced

Two tools have been created in the course of this work: TORMESH, the component of `HIDRAmod` which constructs a field-aligned mesh from magnetic coil and current information, and LIM_OCC, a limiter occlusion utility in `HIDRAmod` which takes a magnetic mesh from TORMESH and a file containing limiting surface geometry and returns a list of all of the physical cells in the mesh outside of the plasma. TORMESH is written in Python 2.7, with the scipy package used for filtering and splining and `FIELDLINES` called from within the program for field line tracing. LIM_OCC is written in Julia 0.3.10 and uses only default packages. Descriptions of the codes are restricted to appendix A.

Chapter 4

Addition of Bohm Diffusivity to EMC3

Transport coefficients are of paramount importance to a transport solver. In EMC3, the transport coefficients may be set either to a constant value or to a per-plasma-cell value by a user function or by reading a formatted input file. Traditionally a constant value is used, as in the case of simulations done on LHD [28], NSTX [47], and others, with a value that may vary per plasma zone to differentiate the transport conditions in the edge from the divertor region or from outside the scrape-off layer. Some simulations may also use a fixed per-plasma-cell diffusivity from experimental data. In the case of HIDRA, diffusivity data is not available from WEGA experimental campaigns or from experiments on the present HIDRA setup, necessitating a user-defined option. Without data the value would need to be calculated in any case, and moving the calculation inside EMC3 offers an attractive solution increasing code portability between HIDRA discharges and user-friendliness as well.

4.1 Motivation

Calculation of the diffusivity inside of EMC3 has several advantages:

1. The diffusivity may vary over successive iterations between EMC3 and EIRENE, bringing a self-consistency to the transport parameters.

2. If calculated on a per-plasma-cell basis, the diffusivity can respond to changing parameters on a spatial scale much smaller than a per-zone diffusivity, e.g., near surfaces and around edges.

3. Reproduction of a simulation requires only the knowledge of a few parameters in the code which affect the diffusivity over the whole domain. Depending on the number of plasma zones,

this may result in fewer free parameters for a simulation than a corresponding constant per-zone diffusivity.

A varying diffusivity in EMC3 heightens the existing non-linearity in the energy balance equations for electrons and ions by introducing an additional factor of T_e to the second and third terms and is not guaranteed to converge. All existing convergence studies on the code have been done with a set diffusivity and correspondingly a weaker dependence on T_e and thus the code must be retested with the improvements in place.

Traditionally EMC3-EIRENE convergence takes on the order of twenty iterations for attached divertor and limiter plasmas and up to two hundred iterations for detached divertors, with convergence less certain in the latter case. In HIDRA, a situation like a detached divertor cannot presently be simulated in EMC3-EIRENE; the magnetic field lines in the stochastic region of a classical stellarator diverge too rapidly for a flux-aligned grid to be constructed, forcing the EMC3 region to terminate near the last closed magnetic surface. With only limited plasmas under consideration convergence trends (or divergence) should be evident in the first twenty iterations, with a potential refinement of such timing depending on the exact strength of the non-linearity and the implementation of the Bohm-like diffusivity in the EMC3-EIRENE computational cycle.

4.1.1 The Bohm Diffusivity of a Magnetized Plasma

In early magnetic isotope separation experiments, small differences between the radii of the arcs drawn out by fast-moving ions of slightly different masses allow for a preferential selection of isotopes. Bohm proposed a semi-empirical plasma diffusion coefficient

$$D_B = A\frac{kT_e}{B}, \quad A = \frac{1}{16} \tag{4.1}$$

to estimate the confinement times of these discharges. Bohm also noted that the factor A could vary by a multiple of 2-3 to account for differences in plasma conditions. The Bohm time, an indication of particle confinement given by estimating the decay of a magnetized plasma, is given by

$$\tau_B = \frac{N}{\frac{dN}{dt}} = \frac{R_a}{2D_B}. \tag{4.2}$$

Unlike the classical cross-field diffusion coefficient which scales as $D_\perp \propto B^{-2}$, Bohm diffusion scales as B^{-1}. A device which exhibits Bohm-like diffusivity is thus expected to have poor confinement properties, a regime which was noted in operations of early stellarators before computational magnetic field optimization was practical.

4.1.2 Application of a Bohm-like Diffusivity to classical Stellarators

In early stellarators, such as the Model C and B-3 stellarators, preliminary fusion experiments were carried out to determine if fusion had any potential as an energy source. Early US devices were racetrack-shaped, with current filaments wrapped helically around the length of the device to produce a rotational transform. Modest temperatures and densities of up to 400 eV and 10^{14} cm^{-3} respectively in the Model C stellarator were reached with 4MW of ICRH at 25 MHz, but confinement times remained low.

By approximating the discharges as a magnetized plasma cylinder and assuming a density profile, the Bohm diffusivity for magnetized plasmas can be evaluated to give a global particle confinement time. When the Model C [48] and B-3 [49] teams did so for their respective devices, they found a Bohm-like diffusivity matched the results from the density decay in the device with a constant of approximately 1/23 for the Model C and 1/5 for the B-3 stellarator [50]. The match persisted through different magnetic configurations and power levels, as shown in figure 4.1.

The coil geometry of HIDRA matches that of the early racetrack stellarators in the bends where helical windings were present, illustrated in figure 4.2, and the magnetic structures of each are unoptimized. HIDRA is expected to operate primarily as a partially-ionized device, similar to both the early stellarators and the original derivation of Bohm diffusivity. Additionally, electron thermal diffusivity studies on WEGA have found [51] good agreement with the Bohm value. Thus, it is prudent to examine a Bohm-like diffusivity for inclusion into `HIDRAmod`.

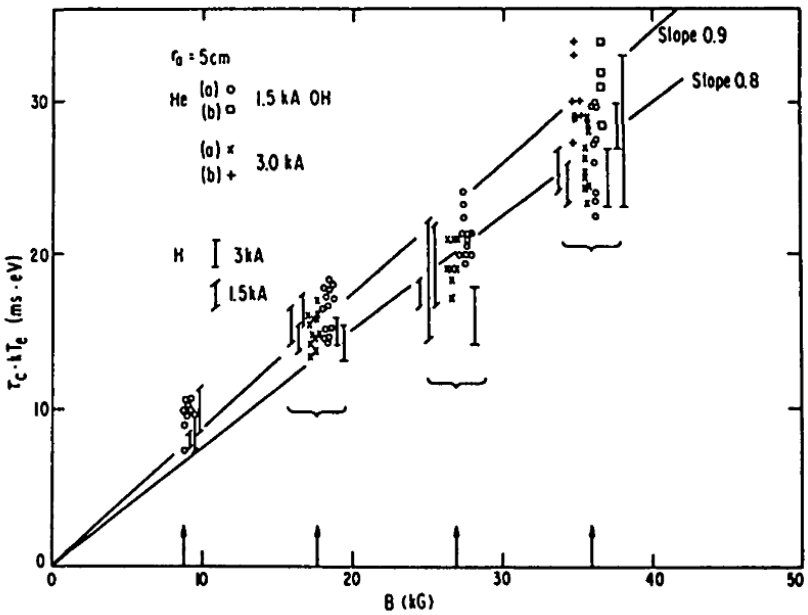
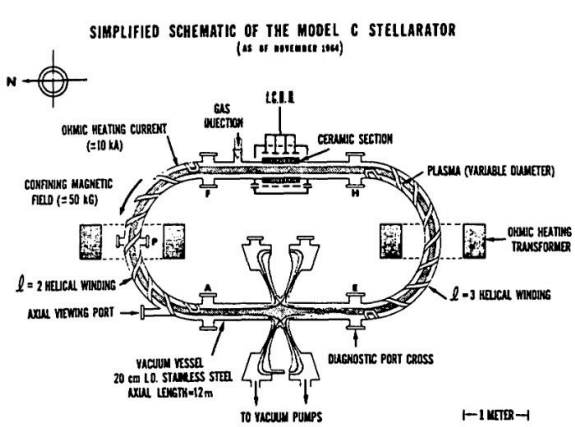
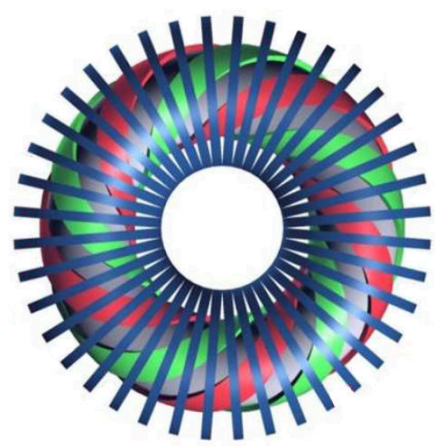


Figure 4.1: Bohm-like diffusivity in the model-C stellarator [50]. A slope of 1 corresponds to equation 4.1 with $A = 1/23$.



(a) A model-C stellarator schematic [50].



(b) A HIDRA schematic [22].

Figure 4.2: A comparison of the magnetic coil structure of HIDRA and the Model-C stellarator, on which Bohm-like diffusivity was observed. Where the helical windings exist on the model-C stellarator, the coil geometry is identical.

4.2 Model Equations: A Relaxed Bohm-like Local Diffusivity for EMC3-EIRENE

For simulating HIDRA discharges without access to discharge diagnostics, and in the anticipation of a simplified user experience for performing simulations with `HIDRAmod` and EMC3-EIRENE, a Bohm-like diffusivity was added to EMC3 as

$$D_{B,i} = R \cdot A \frac{1}{16} \frac{T_e}{B} + (1 - R) \cdot D_{B,i-1}. \tag{4.3}$$

The diffusivity per-iteration is evaluated as a percentage of the old diffusivity plus a corresponding percentage of the new diffusivity, local to each plasma cell. To ease the non-linearity of the scheme and promote assured convergence for ranges of operating conditions, a relaxation coefficient R was added to the traditional Bohm diffusivity. This prevents large changes in the diffusivity as a response to variations in the electron temperature between iterations, especially those resulting from the initial iteration which moves the temperature profile from a user-defined input condition which may be uniform or otherwise inaccurate to a spatially-varying profile calculated by EMC3. Both the electron temperature and the magnetic field strength are accessed from per-plasma-cell arrays and have a resolution in the magnetic mesh corresponding to the domain discretization in the radial and poloidal directions at the default plasma cell allocation. The relaxation factor varies between zero and unity, representing the permitted variance of diffusivity profile for the next iteration from the profile on the current iteration. A relaxation factor of zero prevents the profile from changing, while a relaxation factor of unity disables relaxation. The factor A was left as unity corresponding to the reported WEGA diffusivity scaling [51], with the electron temperature and magnetic field strength evaluated on a per-plasma-cell basis.

4.2.1 Implementation

The diffusivity has been implemented as a file read at the initial call of the program from the command line and a file write after each call to the energy module in EMC3, as shown in appendix B. A few factors informed this choice:

- EMC3 already handles file reading and writing at these locations. In the case of file reads, a

read into the diffusivity array is already present. For writing, a single for loop may be added in the style of other temporally adjacent file writes. The number of lines added to the code handling the entire diffusivity I/O is very small.

- Implementation of a diffusivity module in the style of the streaming, energy, or impurity modules would be time-prohibitive in development. Functionally a separate module would allow for finer control over the diffusion coefficient from the input files as well as allowing the diffusivity calculation only be called when necessary. Having control over when the diffusivity is changed may speed convergence, but was not seen to be either necessary or worth the trade-off in increased development time, as the diffusivity calculation is computationally light.

- The diffusivity is written at each call to the energy module but is only read at each call of the entire program. Were the diffusivity to be read at each call to the energy module or at some other frequent time, smaller oscillations in the plasma parameters that are solved by sub-iterating the streaming and energy modules together may be destabilized without an even stronger relaxation. Potentially an implementation in this fashion would diverge while the chosen method would converge, or convergence times in general may be harmed.

As the diffusivity is a simple algebraic manipulation of two arrays and some constant multiplicative factors which occur only a few tens of times per EMC3-EIRENE call, the direct computational cost of the implementation is negligible. The total number of calls is the number of sub-sub-iterations specified for the energy module times the number of sub-iterations between various modules in the EMC3-EIRENE call times the number of total program iterations (EMC3-EIRENE calls). The major computational cost of the Bohm-like diffusivity instead manifests in an increase in the required number of iterations for convergence. Since the number of iterations required is dependent on the relaxation coefficient, one wishes to pick the largest relaxation coefficient corresponding to the smallest amount of relaxation that converges reliably.

4.2.2 Considerations on the Calculation of the Magnetic Field Strength

In an EMC3-EIRENE run, the magnetic field strength is stored on the grid points and used for the RFLM technique and various mesh checking routines. To use the magnetic field for the calculation

of a local diffusivity in EMC3, it must first be converted to the same format as all other values; namely, each plasma cell must have its own associated magnetic field. Each plasma cell is comprised of multiple physical cells, and the physical cell to plasma cell assignment must be done at each call to EMC3-EIRENE. The assignment statement is an opportune place to perform a coalescence of the magnetic field information onto the plasma cells, given the proximity of the grid and magnetic field information to the cell assignment routine.

$$B_{cell} = \frac{1}{N_{cell}} \sum_{j=0}^{N_{cell}} \frac{\sum_{i=0}^{8} B_i}{8} \tag{4.4}$$

$$N_{cell} = \text{total number of physical cells} \tag{4.5}$$

In the current implementation, an additional array is created to store the magnetic field information on the plasma cells. The magnetic field is evaluated as a simple average between the eight verticies on each physical cell averaged between each plasma cell in the domain, shown in equation 4.4. Minor variations in the field strength lost are negligible considering the magnitude of the potential magnetic fields. By embedding the magnetic field evaluation on the plasma cells into the EMC3 code, portability between discharges is maintained and the user is not required to generate or manage additional input files.

4.3 Numerical Convergence

4.3.1 Definition of the L-2 Norm

To effectively analyze the convergence of a 3D volume, some method must be employed to reduce the variation in the data to one dimension. The usual L-2 norm, given by equation 4.6, provides an indication of the magnitude of a vector with respect to its components.

$$|\boldsymbol{x}| = \sqrt{\sum |x_i|^2} \tag{4.6}$$

The L-2 norm of given toroidal plane can be computed by considering each plasma cell on the plane as one component of a vector representative of the general magnitude of each plasma profile.

To account for the different cell areas, each component of the vector may be multiplied by the 2D area of the plasma cell it corresponds to on the toroidal plane, and the total magnitude divided by the total area as in equation 4.7.

$$|\boldsymbol{x}| = \frac{\sqrt{\sum |x_i A_i|^2}}{A_{Total}} \tag{4.7}$$

By taking the percent change of this area-normalized L-2 norm, an indication of convergence may be presented.

4.3.2 Convergence Tests

HIDRA has many potential operating regimes; it may operate in higher or lower neutral pressures, with varying power levels and magnetic field strengths, or with different rotational transforms or plasma-facing surface locations. For analyzing the convergence, a low-field discharge was chosen to present the worst-case scenario for expected diffusivity ranges over a single simulation. The rotational transform was kept constant around the minimum value for island formation, providing a sanity check on the boundary conditions set in the EMC3-EIRENE run. Neutral gas inlets were set such that the neutral density outside the LCMS was approximately 10^{10} cm^{-3}, and were turned off for a comparison between pure EMC3 convergence with a Bohm-like diffusivity and full EMC3-EIRENE convergence with the same.

To examine the regimes of convergence for a Bohm-like diffusivity, convergence tests were run to 60 EMC3-EIRENE iterations varying one of the particle count, the number of plasma cell discretizations in the radial and poloidal directions, and the relaxation coefficient on the Bohm diffusivity. The number of radial-poloidal-toroidal discretizations for the three sizes of grid, referred to as small, medium, and large, are 36-500-72, 56-500-72, and 96-700-72. Large grids bloat the file size of finished simulations and do not offer an appreciable increase in accuracy over the medium grids, while grids with a small number of radial or poloidal discretizations can severely limit the accuracy of flux deposition calculations on surfaces which are nearly parallel to the magnetic field.

With a sufficiently small relaxation coefficient, our implementation of a Bohm-like diffusivity in EMC3 was seen to be convergent even in the presence of impurities and significant neutral densities for the critical case of a low-field discharge. Figures 4.3, 4.4, 4.5, and 4.6 show convergence

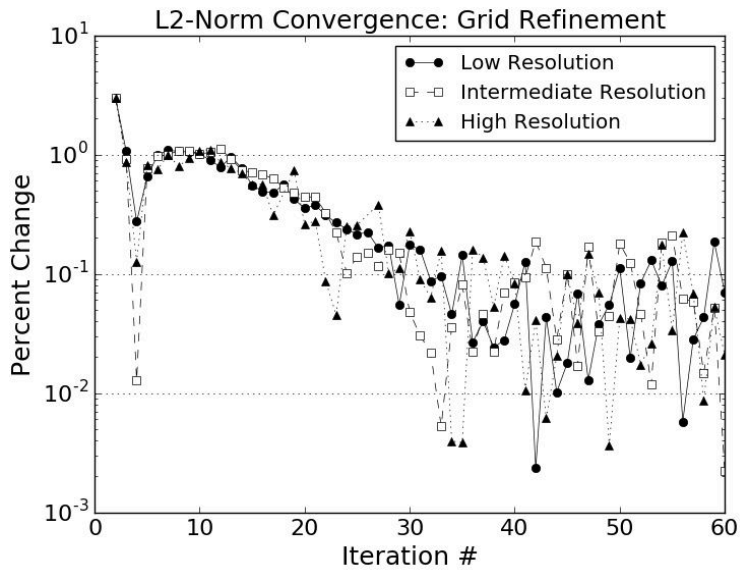


Figure 4.3: Convergence for the grid size refinement at a relaxation coefficient of $R = 0.15$ and a particle count of $8 \cdot 10^5$. No significant dependence is observed.

in these situations. When the field strength is low, small variations in the electron temperature from iteration to iteration may produce very large swings in the resulting diffusivity, but was not observed to be a problem past the initial iterations. The high-field mode of HIDRA, which one would like to use the same relaxation factor for to facilitate comparisons between the two conditions at similar stages of convergence, has smaller diffusivity swings resulting from both the higher magnetic field strengths and the electron temperature which was seen to increase more slowly than the magnetic field strengths when compared to the low-field mode. As a result, the same convergence is applicable to high-field discharges as long as a sufficient number of iterations are included. From figures 4.3-4.5, it can be seen that 20-25 iterations is sufficient for an L-2 variation of less than 0.5% (just before the onset of statistical noise), and from figure 4.6 roughly 40 iterations are required for statistical equivalence to the static diffusivity case.

Convergence occurs rapidly in the first few iterations as the diffusion coefficient profile shifts from the constant value the simulation was initialized at to one which varies locally. For the results presented in this chapter, the diffusivity was initialized to a constant $10m^2/s$. Given a discharge for which the equilibrium density is further from the initial value, convergence may take longer. In practice, the diffusivity can be easily estimated to within an order of magnitude based on knowledge of the magnetic equilibrium of the device, and convergence from extreme initial diffusivities is not

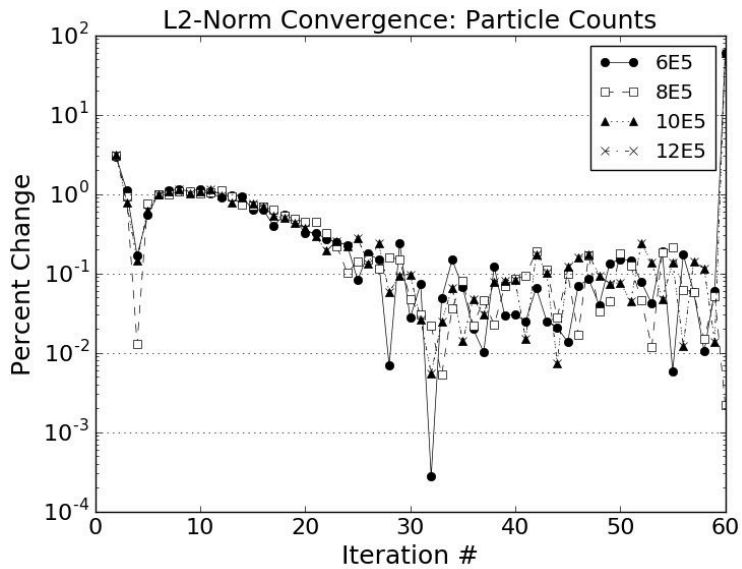


Figure 4.4: Convergence for the particle count refinement at a relaxation coefficient of $R = 0.15$ and a medium grid size. No significant dependence is observed.

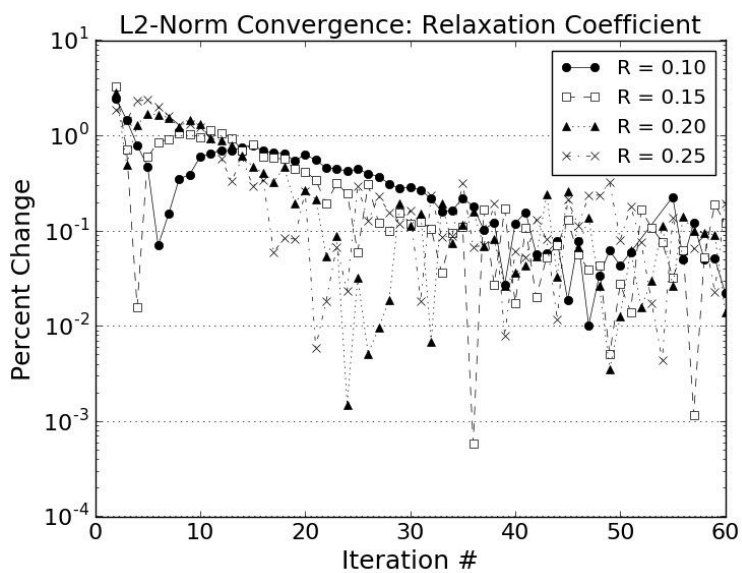


Figure 4.5: Convergence for the relaxation coefficient refinement at a particle count of $8 \cdot 10^5$ and a medium grid size. Lower relaxation coefficients take longer to converge as expected, and using an excessively low relaxation may prevent convergence in reasonable time-frames (longer than a few days).

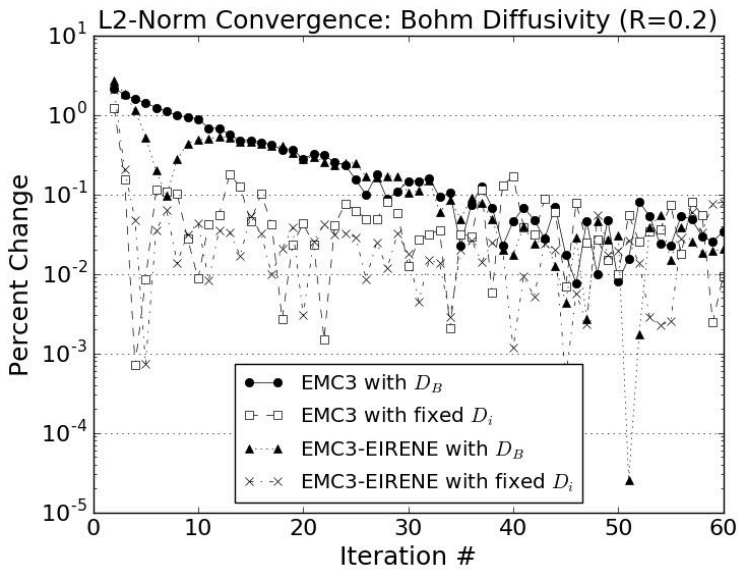


Figure 4.6: Convergence for EMC3 with and without EIRENE, and with and without a Bohm-like diffusivity. EIRENE has no impact on convergence, but the change in the neutral profile from a constant background as initialized to a spatially varying profile causes a false reduction in the L-2 norm in the first few iterations.

of concern.

A second convergence regime is seen after the convergence from the initial conditions and shows the relaxation of the diffusivity profile into its equilibrium position. This convergence regime is best illustrated by examining the island structures as the iterations progress, which originate closer to the inner boundary surface before relaxing outward slightly. Island equilibrium positions match within 1 mm with Poincaré set data showing the location of islands at the examined rotational transform as in figures 4.7 and 4.8.

A single EMC3-EIRENE iteration on a well-conditioned magnetic mesh for HIDRA, due to the small size of the simulation domain, can be completed in under two hours. Due to the relatively low cost of adding extra iterations to a simulation and the large range of the diffusivity, a conservative estimate of R = 0.2 was used for the relaxation factor for the simulations presented in chapter 5.

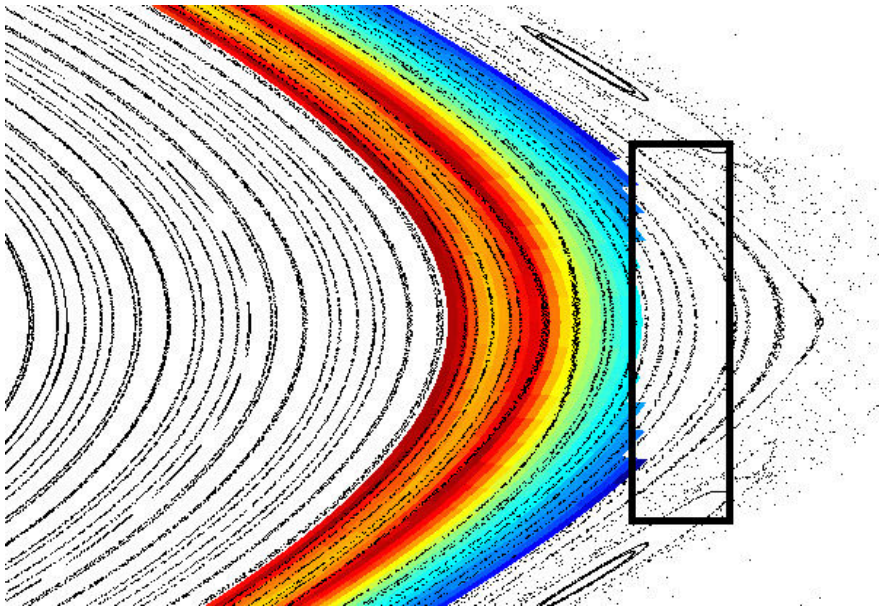


Figure 4.7: A Poincaré section overlaid onto the diffusivity profile. Islands form in the diffusivity at the location expected from the Poincaré section. The approximate position of the limiter is indicated by a black box. The island structures are seeded by the high diffusivity region in the center from which they radiate outwards before settling in the islands. The movement of the profiles follows the flow of information in the system as the Monte Carlo particles in EMC3 are born at the inner radial surface and traced outwards.

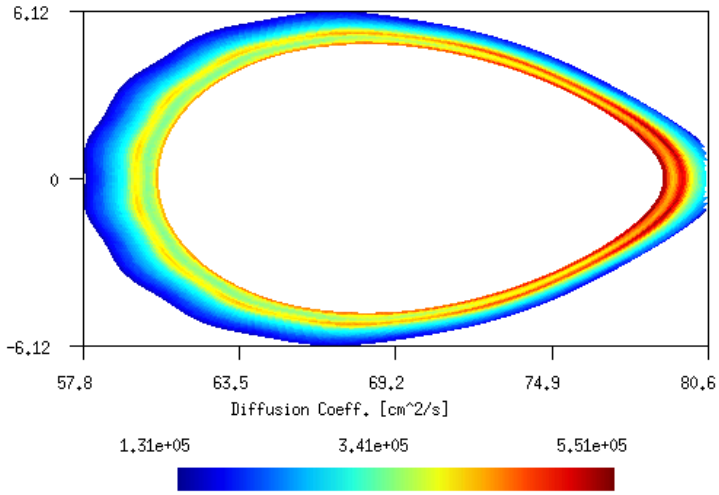


Figure 4.8: An equilibrium diffusivity profile for a 125kW discharge at 87.5 mT. The diffusivity profile dynamically forms islands at the correct locations, as evidenced by figure 4.7.

Chapter 5

Numerical Characterization of the Plasma Transport Conditions in HIDRA

Herein, the edge plasma of an expected typical HIDRA discharge is simulated for varying power levels and limiters of various materials and at various positions in a given magnetic configuration at differing axial magnetic field strengths. Similarities to and differences from WEGA experimental results will be noted when possible, mainly in the context of plasma parameters for which data is available. An overview of the conditions experienced at each limiter position as well as the efficacy of simulating each position will be be presented.

As mentioned in the first chapter, HIDRA as of the date of publication of this thesis has 26 kW of 2.45 GHz heating split between a 20 kW magnetron and a 6 kW magnetron. The goal of the present analysis is to calibrate the new `HIDRAmod` code on HIDRA's edge plasma in the context of a limiter discharge and identify those combinations of limiter placement, magnetic field intensity and surface material which seem most promising for the acquisition of PMI-relevant heat fluxes in the device. This analysis is extended to higher power levels (up to 250kW) as a further investigation for future device extensions.

Overview of Simulated Conditions

A magnetic mesh (figure 5.1) was made with $\iota = 0.318$ at two axial field strengths of 87.5 mT and 0.5 T. The edge plasma mesh is generated from $0.8a$ to $0.99a$, where a is the minor radius of the plasma volume, and the total mesh is generated from $0.05a$ to the vacuum vessel wall for use in neutral calculations. The plasma mesh covers 36° of the device, one-half of a period, with the two outer toroidal surfaces handled by an up-down symmetry boundary condition and the two inner toroidal surfaces handled by the RFLM technique.

Three limiter placements are simulated in the present work, as in figure 5.1. An inboard and outboard limiter make up the bulk of the comparisons, with a trench limiter presented for

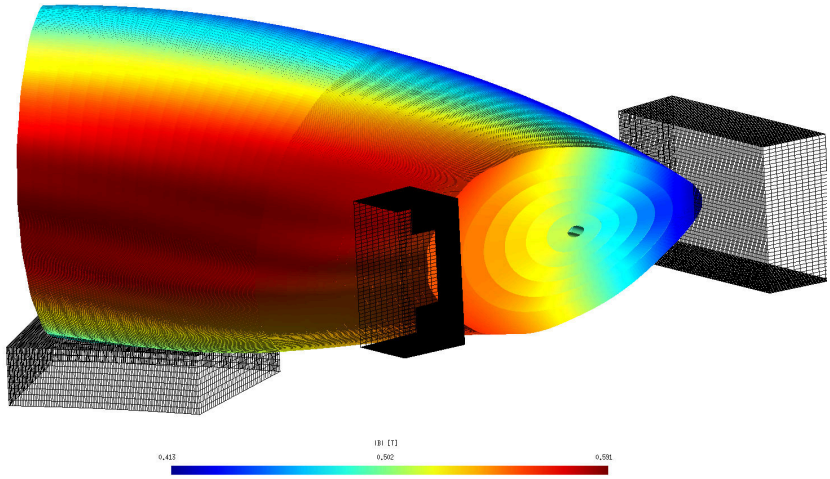


Figure 5.1: Simulated limiter positions in HIDRA, with the HIDRA mesh (the outer EIRENE region is occluded for clarity). The distinct shading difference between the two halves shows the plasma zone division seen by EMC3. Only one limiter position is used in each discharge. Limiter positions are named, from left to right, 'tench', 'inboard', and 'outboard'. Coloration is provided by the magnetic field magnitude at the mesh location linearly from 0.413 T to 0.591 T, with an axial magnetic field of 0.5 T.

completion. All limiters are placed on up-down symmetric surfaces; the inboard and outboard are located on the midplane on one surface, while the trench limiter is located a half-period away on the bottom of the other surface. Three materials were tested at these limiters: iron, molybdenum, and lithium.

Plasma and surface deposition profiles are presented at power levels of 26 kW, 125 kW, and 250 kW, with an assumed RF deposition of 100%. Decay lengths for the particle, energy, and momentum were set to 3 cm, 3 cm, and 2×10^4 cm respectively. Initial density profiles were uniform at 5×10^{12} cm^{-3}, and initial electron and ion temperatures set to 20 eV and 1 eV respectively. From an initial uniform diffusivity of 10 m^2 s^{-1}, the profiles were free to evolve under a Bohm-like diffusivity.

Hydrogen was used as the working gas, with trace impurities of the limiting surface materials. A source of neutral particles was added at a toroidal angle of 18° on the bottom of the vacuum chamber, at a major radius of 72 cm and a height of -19 cm, emitting isotropically. The particle density leaving the core was set at 3 10^{12} cm^{-3}.

Each simulation was run for twenty iterations, of which each iteration consisted of an EIRENE

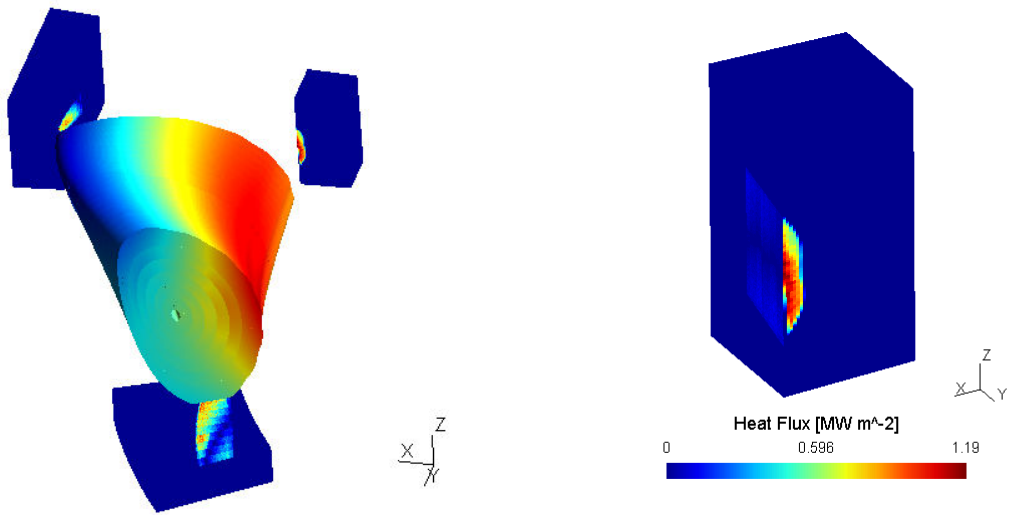


Figure 5.2: Left: Limiter positions and corresponding deposition patterns are shown here with a representation of the location of the plasma in HIDRA, to illustrate the angle of incidence and physical orientation of each surface map shown in section 2. The left and right limiters on the top of the illustration are the outboard and inboard limiters, while the bottom limiter is the trench limiter.
Right: The inboard limiter profile is shown in perspective to illustrate the two regions of power flux deposition. Particle flux deposition follows similar trends. Due to the low magnitude of the profile on the side facing the plasma, these have been excluded from analysis in section 3.

Table 5.1: Simulated Limiter Dimensions

Limiter	width (x)	Length (y)	height (z)
Inboard	5 cm	5 cm	10 cm
Outboard	5 cm	16 cm	5 cm
Trench	17.7	18.7	4 cm

run followed by ten iterations between the streaming (density) and energy (temperature) modules. At the conclusion of each set of iterations, a post-processing run was done to create the limiter heat and particle deposition maps.

Limiter Geometries

Figure 5.2 shows the orientation of each limiting surface with respect to the plasma. Dimensions for the surfaces are given in table 5.1, measured along the corresponding axis.

5.1 Available Data on WEGA Operational Conditions

WEGA Accessible Rotational Transforms

As mentioned in [23], the maximum rotational transform for high-field operation is $\iota = 0.4$. As all rotational transforms accessible by the plasma in the high-field operation are also accessible in the low-field operation, a slightly smaller rotational transform of $\iota = 0.318$ was chosen to show a representative discharge while still exercising the grid generation routines in `HIDRAmod`. The higher the rotational transform, the more difficult it is for `HIDRAmod` to create a linear flux-conserving mesh with a substantial amount of the edge included.

Historical WEGA Heating Schemes

WEGA electron temperatures vary significantly depending on the heating mode. In 2006, WEGA was equipped [52] with a 10 kW 28 GHz EBW source which has since been removed. The aforementioned 26 kW of 2.45 GHz heating is still present on the device, as is the transformer for ohmic heating. In the early years when WEGA was used as a tokamak for RF heating studies, it also had a 100 kW 500 MHz RF source [8] [9].

Electron Temperature and Electron Density: 26 + 10 kW

With both the 2.45 GHz and 28 GHz sources, WEGA temperatures and central densities were reported [53] in argon and helium discharges at 10s of eVs and near 1.3×10^{19}, above the OXB cutoff density due the additional EBW source, after which the discharge could be sustained by the 10 kW EBW alone. A small number of supra-thermal particles on the order of 10 keV have also been reported [54] in these discharges.

Electron Temperature and Electron Density: 6 kW

With the 6kW 2.45 GHz source, [55] reports a relatively uniform density of 2.5×10^{18} m^{-3} of argon, helium, and hydrogen across the plasma profile for different ι, with a slow-electron temperature peaked at the LCFS at 5-6 eV and a fast-electron temperature in the range of 100-200 eV, at off-resonance magnetic axis field strengths of 61.2 mT, 70 mT, and 78.7 mT. In a discharge of 52.5

mT also with the 6kW 2.45 GHz source, [56] reports LCFS-peaked slow-electron temperatures of 8-10 eV, fast-electron temperatures of 50-150 eV, and an axial-peaked density of 1.4 $\times 10^{18}$ m^{-3} of argon.

Neutral Gas Pressures

A neutral pressure of 1.6 $\times 10^{-3}$ Pa (argon) is reported by [56], corresponding to a neutral density of 4 $\times 10^{17}$ m^{-3}, and roughly an ionization fraction of 30%. Neutral pressures in higher-power discharges [51] [57] are reported at 2 $\times 10^{-4}$ mbar (helium) and 2 $\times 10^{-4}$ Pa (helium) respectively, corresponding to 5 $\times 10^{18}$ m^{-3} and 4 $\times 10^{16}$ m^{-3}.

Diffusivity

The electron thermal diffusivity is reported in [51] as 1.9 m^2 s^{-1} for a 9 kW 28 GHz helium discharge, and finds good agreement with the Bohm value in the 0.5 T axial field strength case.

5.2 Plasma Profiles (n,T,D)

Electron Temperature T_e

Electron temperatures are presented for a 100% efficient 26 kW, 125 kW, and 250 kW discharge at 87.5 mT and 0.5 T axial field strengths in figure 5.3.

For a deposited RF power of 26 kW, 125 kW, and 250 kW, electron temperatures of up to 40 eV, 100 eV, and 152 eV are calculated in the edge plasma. The deposited RF power refers to the power actually coupled to the plasma crossing the core-edge boundary, and not the full device power; non-ideal absorption and radiative losses in the core reduce the heating efficiency. Expected increases in temperature with higher input powers and stronger axial field strengths are observed.

In figure 5.4, the 26 kW case for all three limiter positions is shown at low and high fields to illustrate the lack of effect on the electron temperature of the limiting surface position. The high Bohm-like diffusivity means that the majority of plasma losses are to walls and not the limiting surface, which in turn limits the effect the limiting surface may have on the bulk plasma profiles.

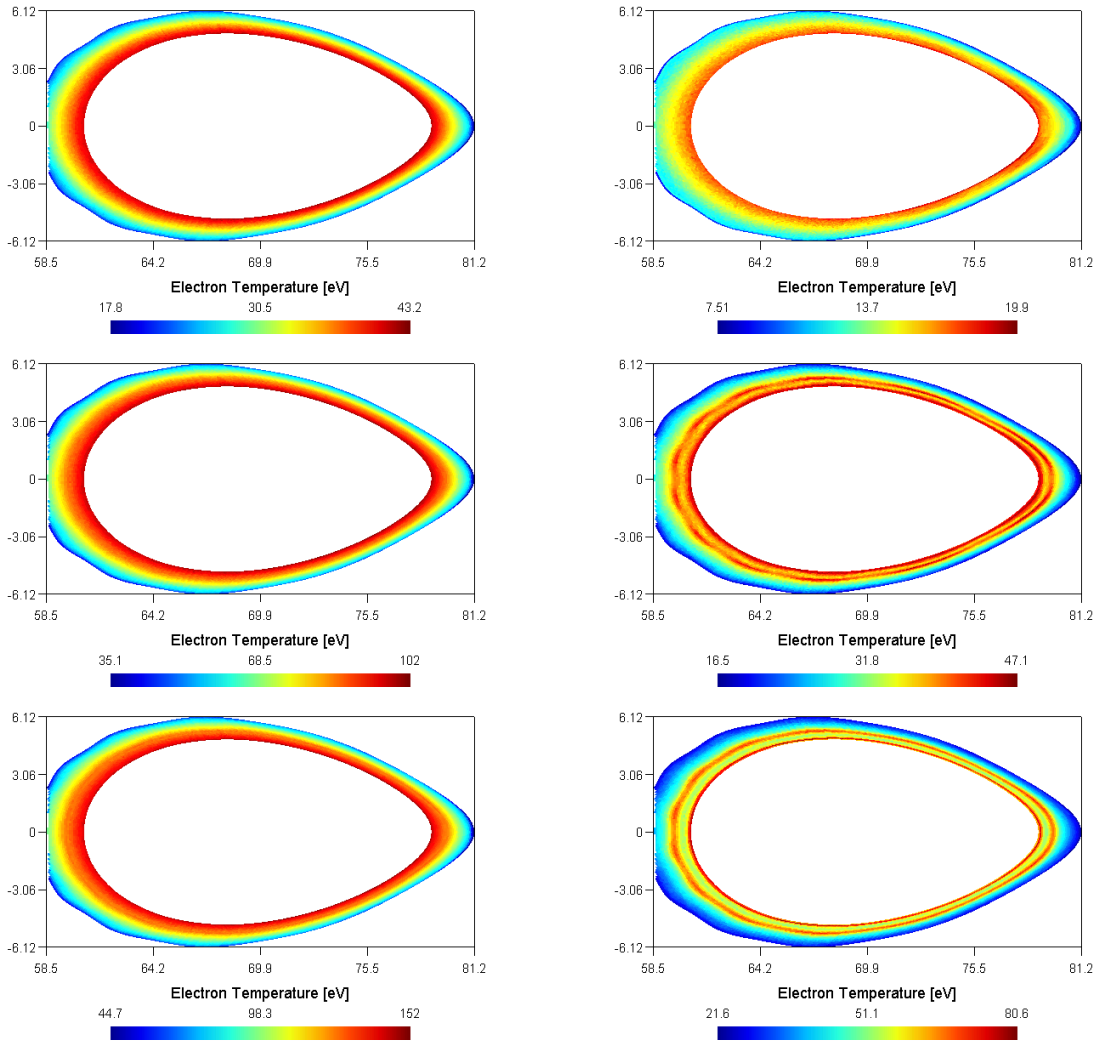


Figure 5.3: Simulated T_e comparisons for varying axial field strength and power levels (no sputtering). The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. In the low-field case, an island chain appears naturally corresponding to their location in the Poincaré set for this ι. Notably, the high T_e values compared to representative WEGA conditions listed in the first section may be attributed to the 100% RF input efficiency assumption, which is relaxed in section 6 of this chapter.

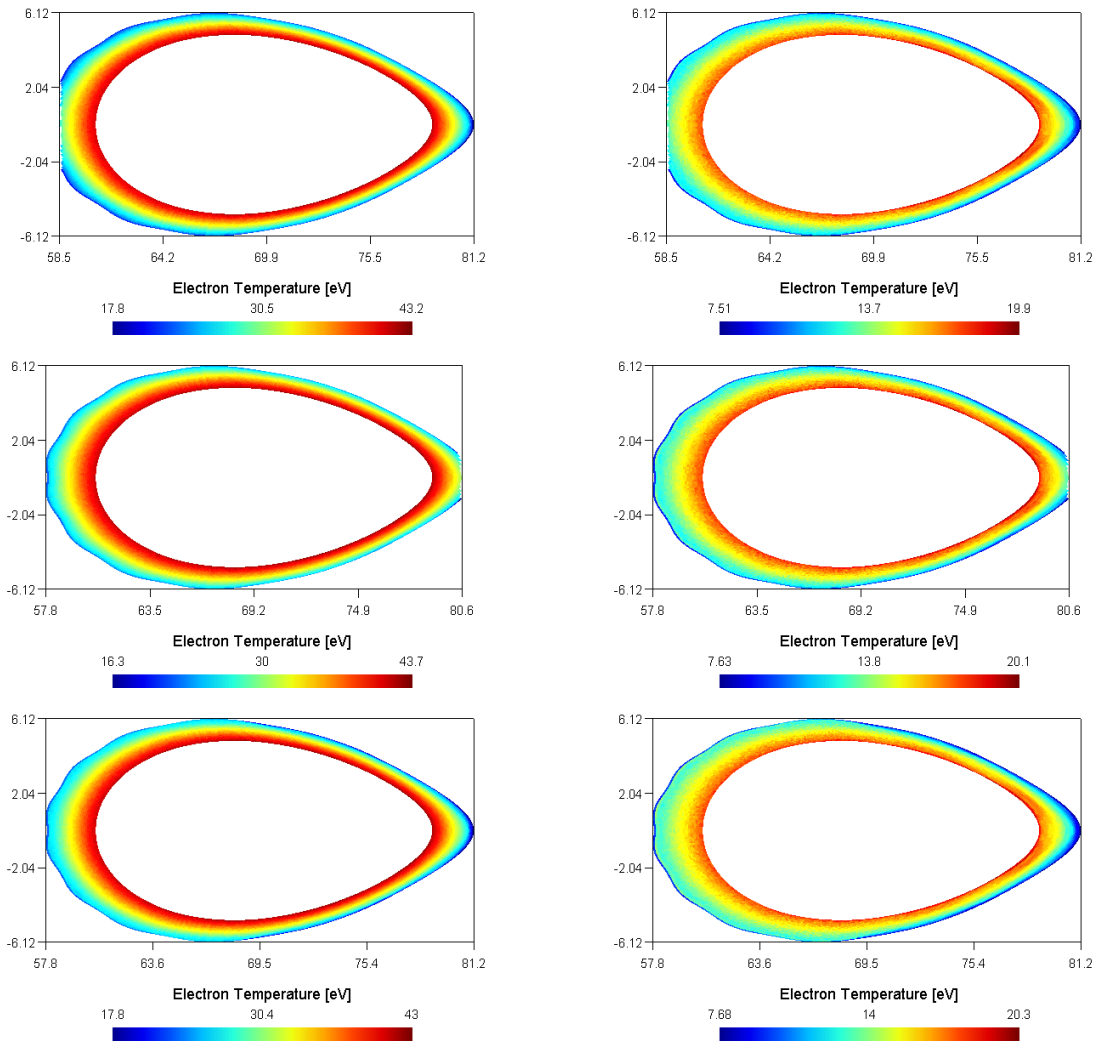


Figure 5.4: Simulated T_e comparisons for varying axial field strength and limiter position. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for an inboard, outboard, and trench limiter position. For comparison, the trench limiter profiles are plotted at the same toroidal location as the other limiters. The device power is set to 26 kW. Analysis shows no significant differences between profiles, which may be attributed to a high Bohm-like diffusivity. The majority of power losses are to the walls and not the limiting surface; for surfaces barely intruding into the plasma like these, the power deposition efficiency, or the ratio of the surface-integrated limiter heat flux to the full device power, is on the order of 1-5%.

Ion Temperature T_i

For a similar comparison to the electron temperatures, the ion temperatures are presented in figure 5.5 for all power levels and axial field strengths and in figure 5.6 for the various limiter positions, showing trends similar to the electron temperatures. As the 2.45 GHz microwaves are not strongly absorbed by the ions, the majority of the power is deposited into the electrons. With a low density and poor confinement inherent in Bohm-like plasmas, there is no time for the electrons to pass power to the ions before leaving the volume, and as a result the equilibration is strongly suppressed. On the inboard side where electrons are better confined, there is a stronger transfer of power resulting in higher ion temperatures.

Electron Density n_e

Electron densities in simulated HIDRA discharge have a strong r^{-1} drop-off along the minor radius as expected. As all simulations were started with identical particle transport boundary conditions, the resulting profiles are similar in magnitude in both the axial field and power scan in figure 5.7 and the field and limiter position scan in figure 5.8. A power scan shows some strengthening of the density profile in the island chains in relevant profiles.

Bohm-like Diffusivity D_B

Added in the course of this work, the variable per-plasma-cell Bohm-like diffusivity allows for the dynamic formation of expected island chains in the temperature and density profiles. Since the Bohm-like diffusivity is coupled directly to the electron temperature profile, the two are always self-consistent, and thus the Bohm-like diffusivity follows an expected convolution of the major-radial falloff of the magnetic field and the minor-radial falloff of the electron temperature, as illustrated in figure 5.9.

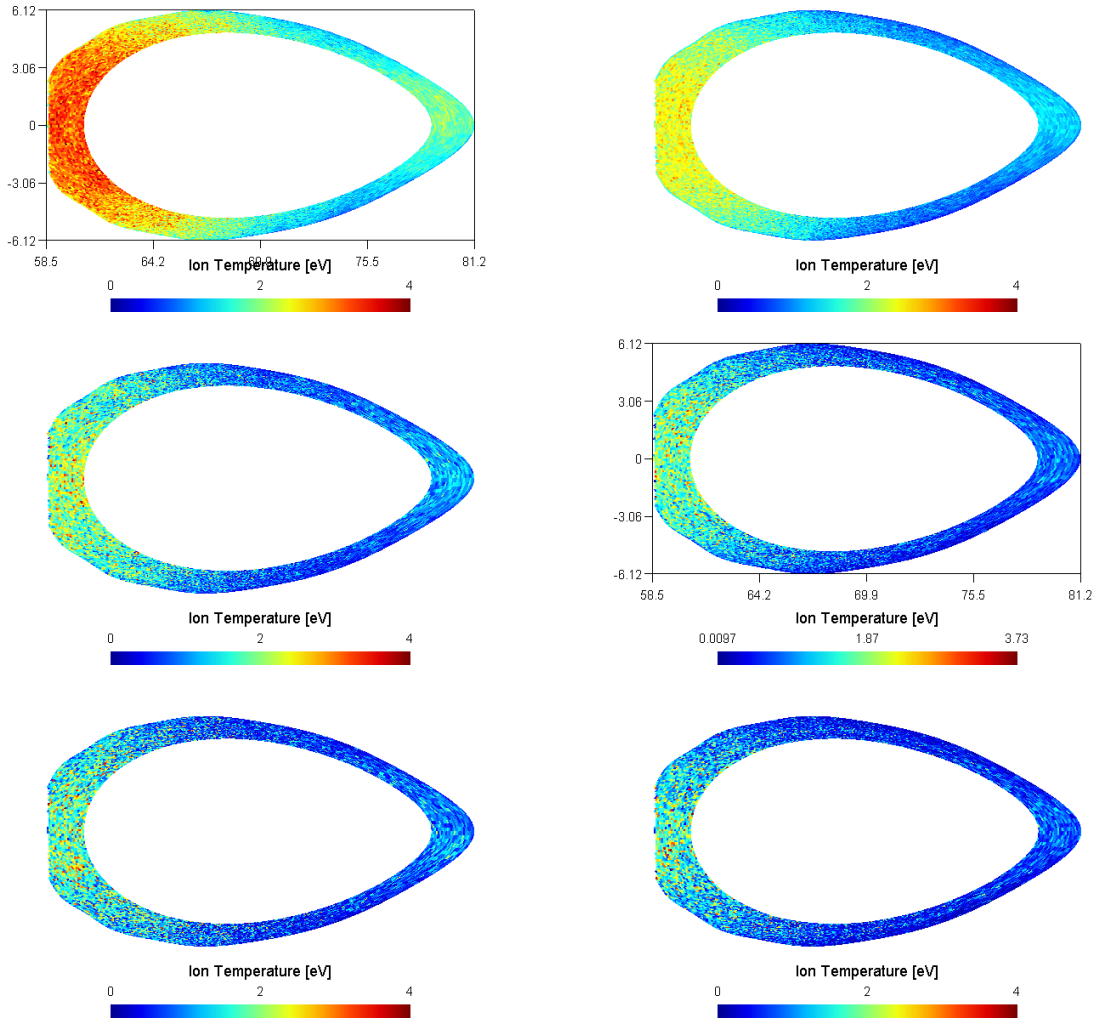


Figure 5.5: Simulated T_i comparisons for varying axial field strength and power levels (no sputtering). The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The ion temperatures, as they are not efficiently heated by the RF source, are low with poor statistics. The inboard side with a higher magnetic field enables the transfer of more energy from the electrons to the ions before the former are lost. Bohm-like diffusivities (equation 4.1) are based on the electron temperature; as the electron temperature increases, particles are more poorly confined by the discharge, and there is less time for power transfer between the electrons and ions.

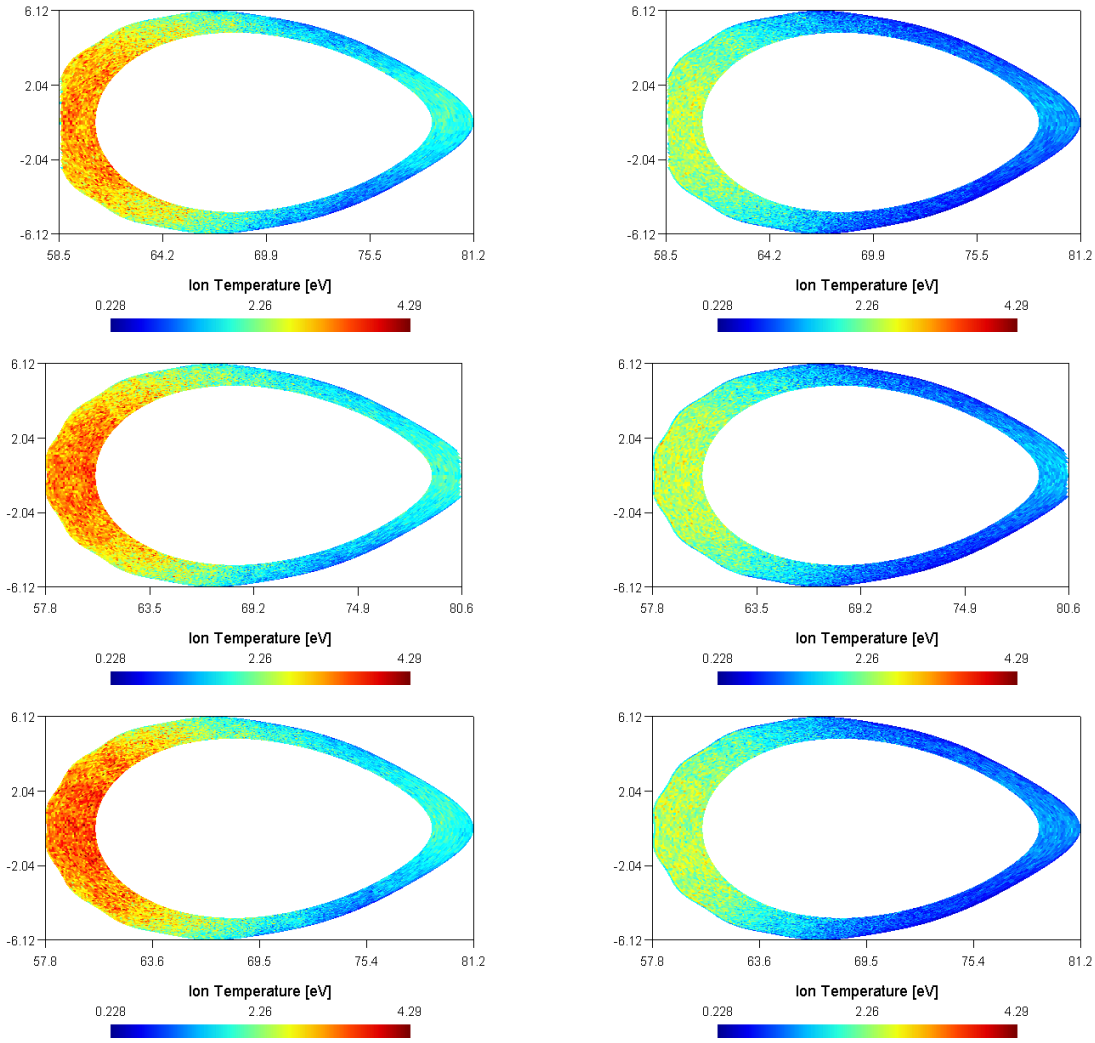


Figure 5.6: Simulated T_i comparisons for varying axial field strength and limiter position. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for an inboard, outboard, and trench limiter position. For comparison, the trench limiter profiles are plotted at the same toroidal location as the other limiters. The device power is set to 26 kW. Profiles compared between limiters show identical features and temperature magnitudes.

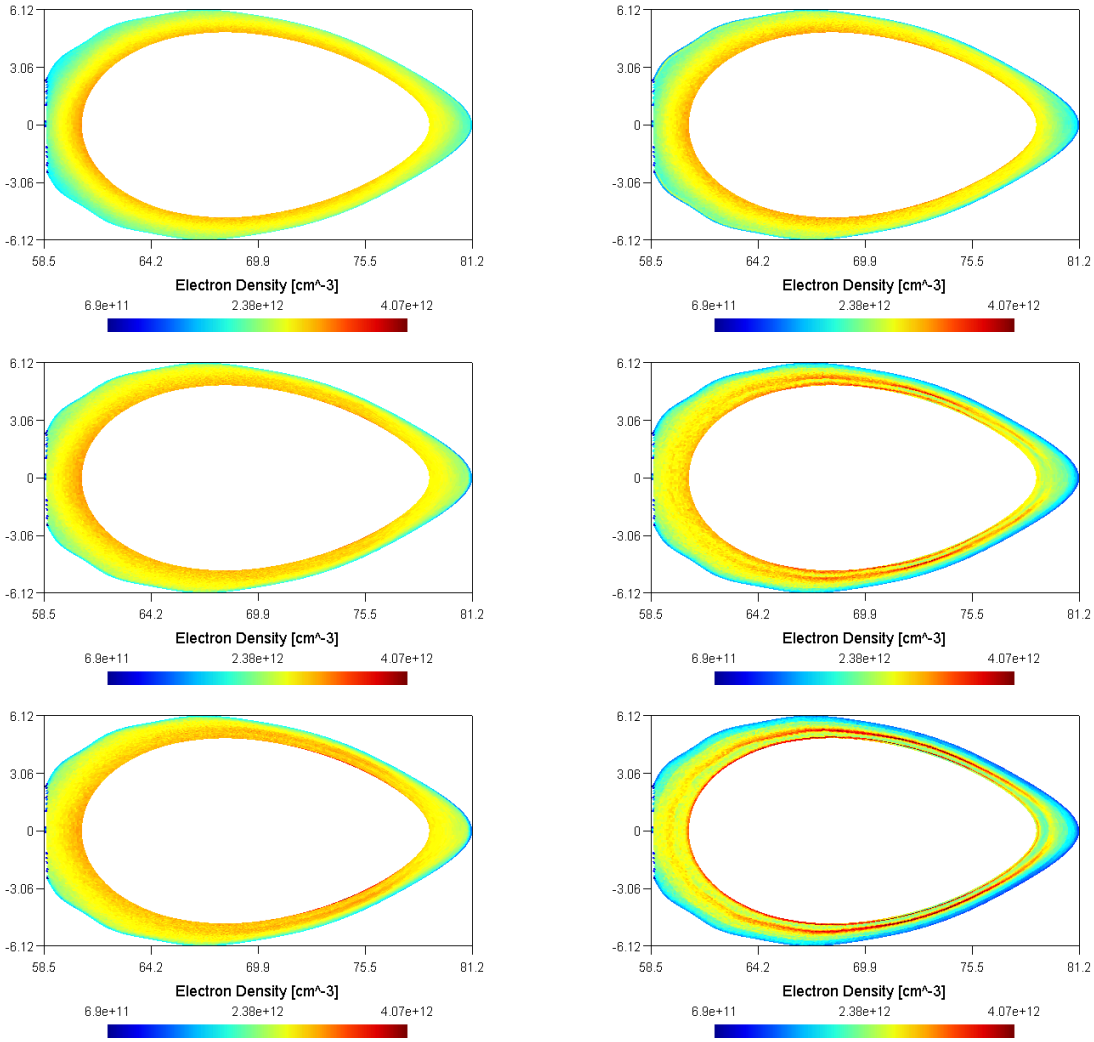


Figure 5.7: Simulated n_e comparisons for varying axial field strength and power levels (no sputtering). The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The density is set and calculated separate from all energy boundary conditions with the exception of the coupling through Braginskii's equations, leading to similar profiles. In the higher powers and lower field strengths, the island chain becomes pronounced simultaneously with the electron temperature and diffusivity profiles.

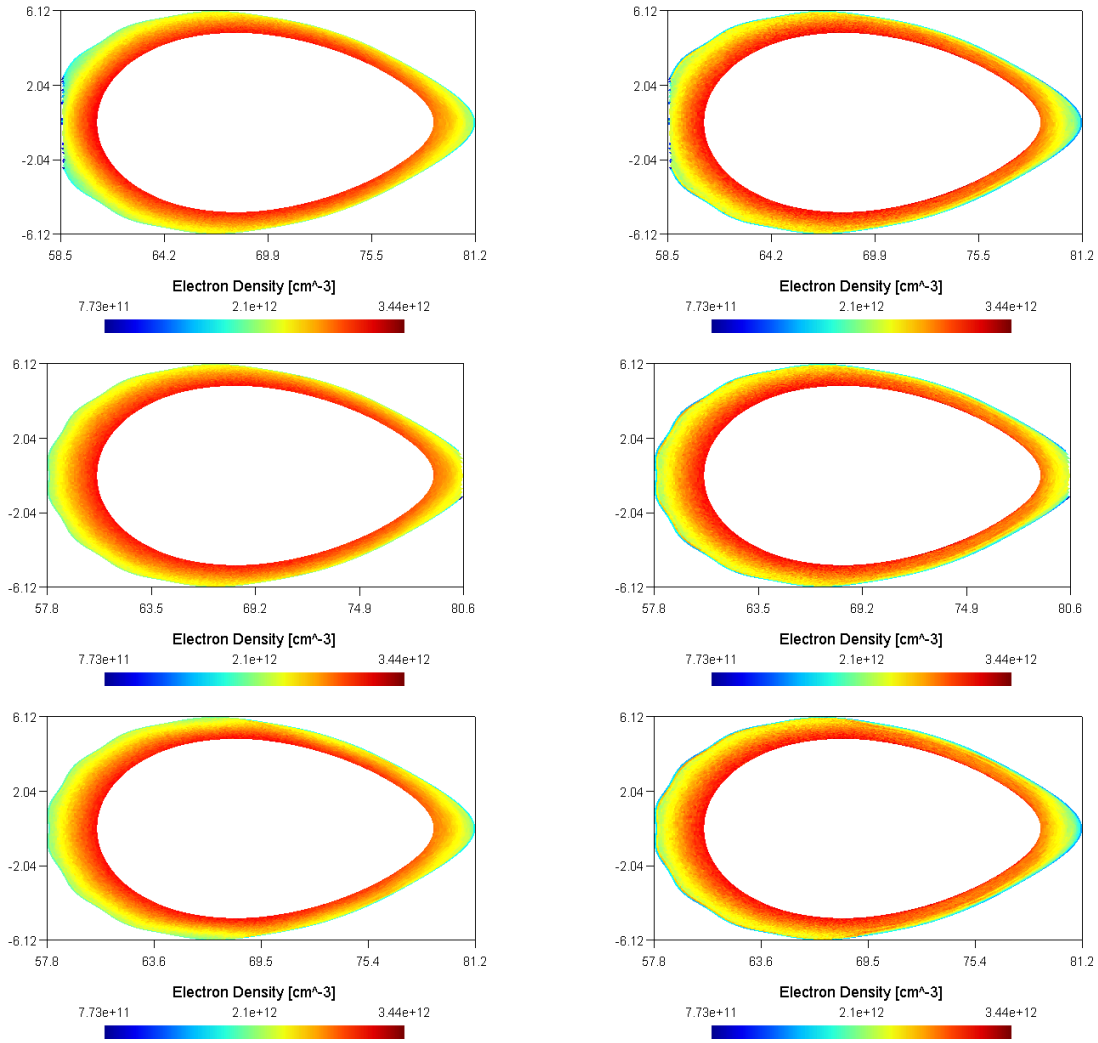


Figure 5.8: Simulated n_e comparisons for varying axial field strength and limiter position. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for an inboard, outboard, and trench limiter position. For comparison, the trench limiter profiles are plotted at the same toroidal location as the other limiters. The device power is set to 26 kW. Electron density profiles show no significant difference between limiter positions, resulting from a highly diffusive plasma.

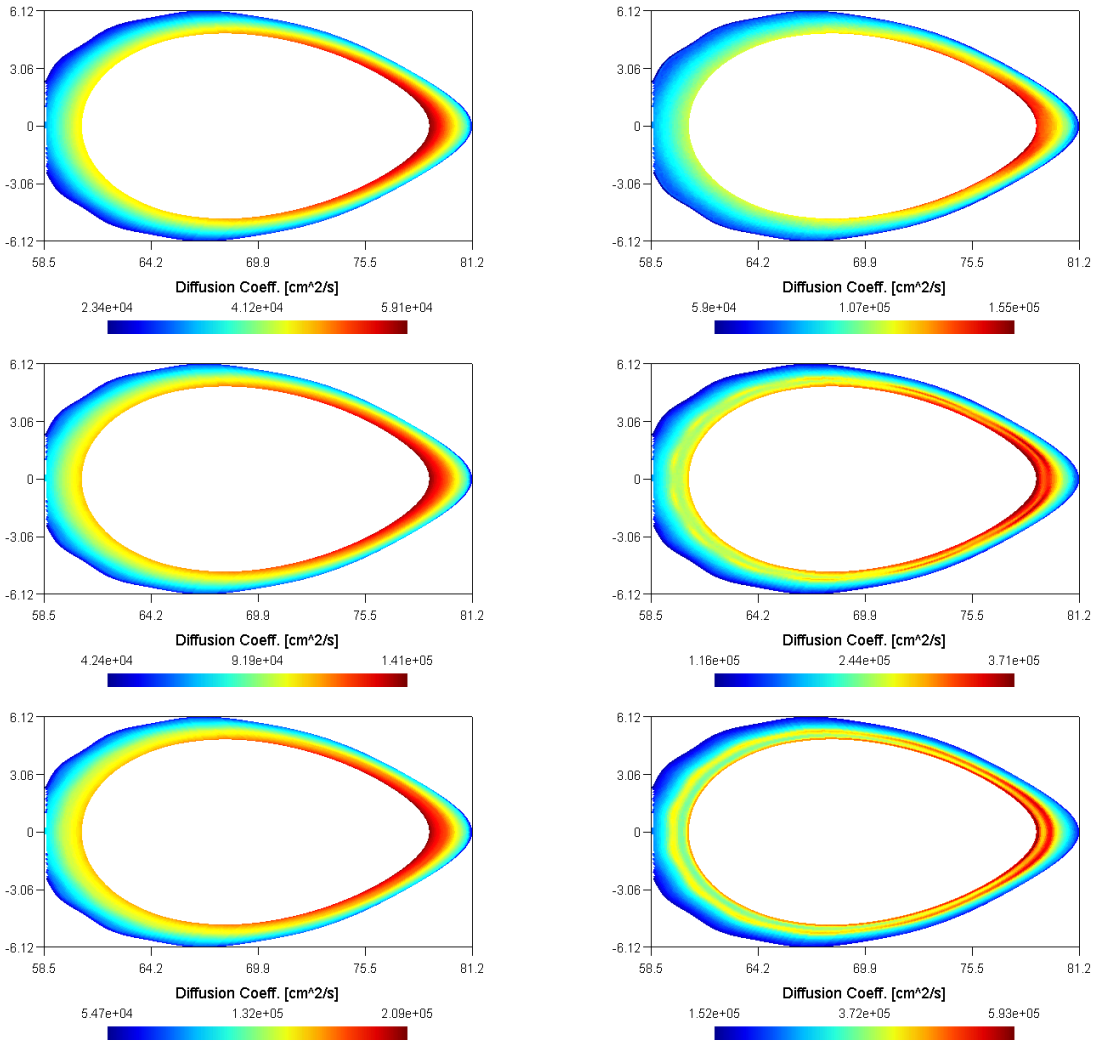


Figure 5.9: Simulated D_B comparisons for varying axial field strength and power levels (no sputtering). The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. A Bohm-like diffusivity, free to vary over the entire profile, allows the dynamic formation of stable island structures with enhanced diffusivity as expected from Poincaré set predictions. The magnitude of the diffusivity is typically high, in excess of 10 m^2 s^{-1}, and results from the 100% efficient RF deposition assumption.

5.3 Particle and Power Fluxes at different limiters

Particle Fluxes to the Limiting Surface

Simulated peak particle fluxes range between 0.07 and 0.2 A cm^{-2} for the outboard and trench limiter positions, and between 0.7 and 1.4 A cm^{-2} for the inboard position, with the higher flux caused by the smaller wetted area. In the outboard limiter, a negligible amount of deposition takes place on the side facing radially inwards to the plasma volume, and is neglected from figure 5.10. In figures 5.11 and 5.12, hot spots on the surface deposition map follow the contours of the impinging flux surfaces.

All limiting surfaces are located on up-down symmetric boundary surfaces, and so the profiles on the surface will be mirrored about the division. This operates as expected on the inboard and outboard limiters, but raises a problem for the trench limiter. The corresponding up-down symmetric trench limiter would be on the top of the device, but is absent in simulations. As a result, some extra particles which otherwise not hit the limiting surface will be registered as coming from the other direction and falling into the surface, creating a second smaller profile near the up-down symmetric boundary. Due to the high diffusivity this has a minor effect on both the plasma profiles and the diffusivity map, and differences between the surface collection efficiency for the trench and outboard configurations are non-existent.

On the outboard limiter configuration, the comparison at high energies in figure 5.11 shows a prominent concentrating effect of the magnetic field on the surface deposition pattern. In the low-field case the first hot spots are very close to the outer boundary of the wetted area, while in the high-field case they are located further inward.

All outboard and trench limiter profiles are stripped in the toroidal direction at locations corresponding to the plasma cell divisions in the toroidal direction. The post-processing module in EMC3 takes the values of temperature and density in the cells right above the limiting surface and traces field lines through them onto the surface to create higher-resolution images. When the values from cell to cell change more abruptly than the interpolation of the magnetic fields would allow, the stripped profiles result.

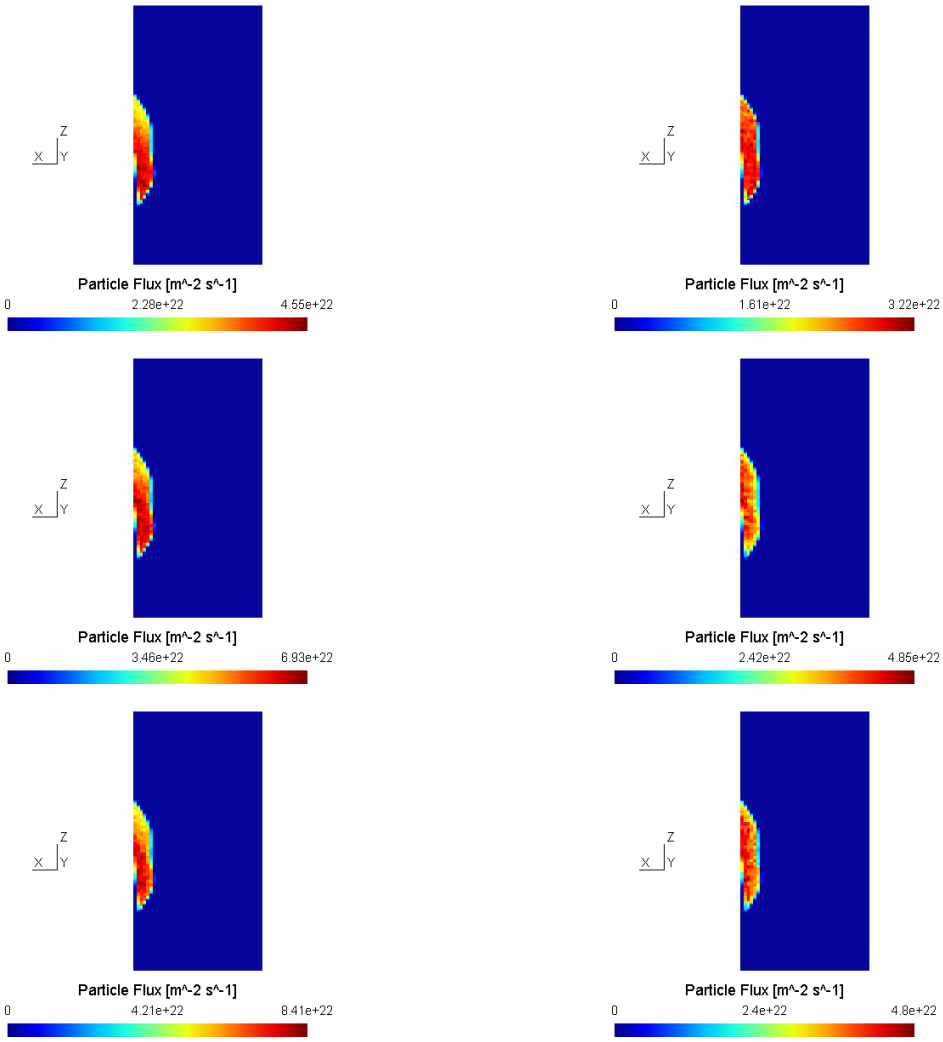


Figure 5.10: Simulated limiting surface particle flux comparisons for varying axial field strength and power levels. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The limiter is in the inboard position and is made of a heavy element assumed to have little sputtering. Profiles are viewed from the direction of the predominant magnetic field to view the regions of high flux.

Particle fluxes on the inboard limiter position are relatively high in comparison to the other geometries, but the wetted area is low and the proximity of the LCMS to the vacuum vessel makes these geometries difficult to realize. Some deposition is seen on the sides facing radially inward (in this instance radially inward corresponds to the x-direction), but is negligible as the field lines are mostly parallel to the surface and the radially-inward surface is shielded by the concavity of the plasma volume on the inboard side.

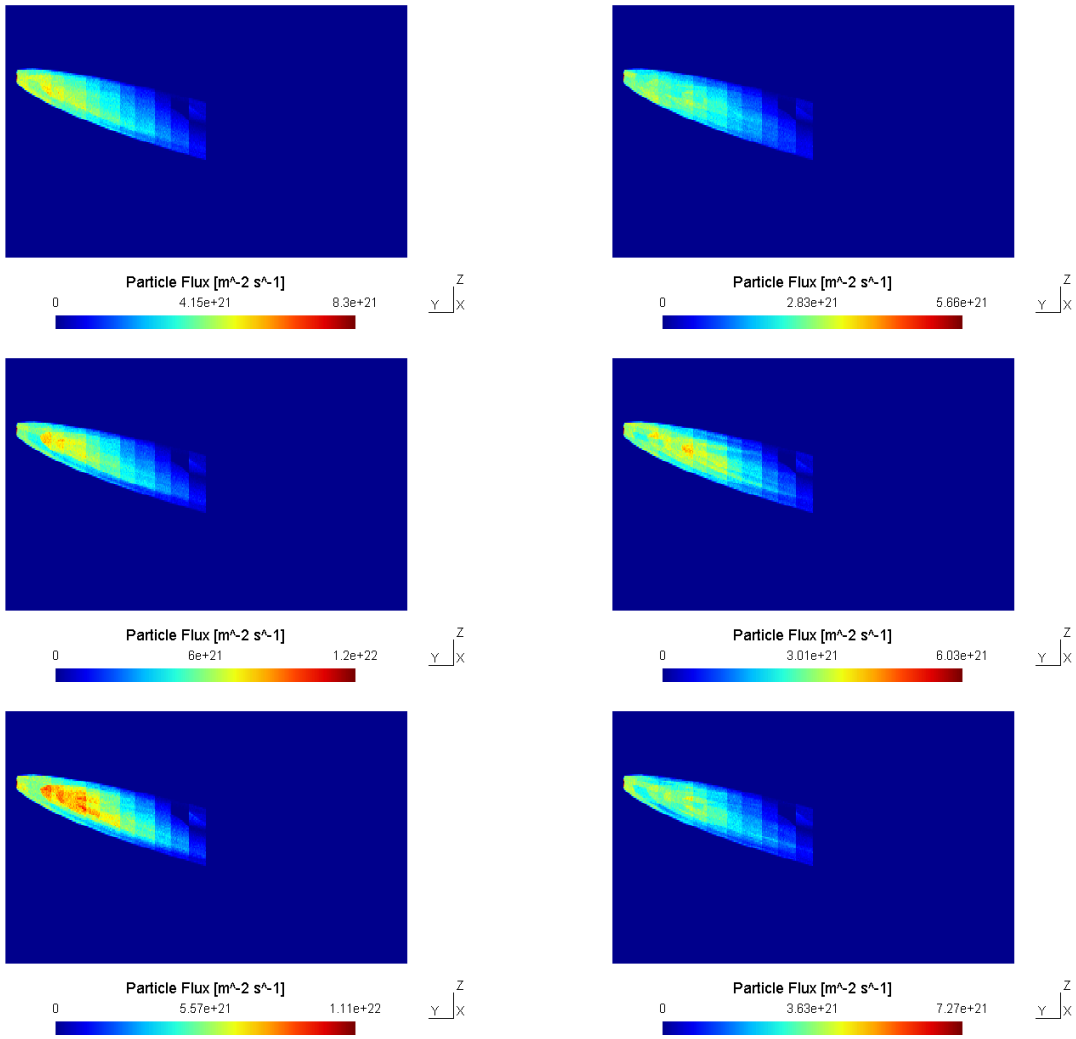


Figure 5.11: Simulated limiting surface particle flux comparisons for varying axial field strength and power levels. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The limiter is in the outboard position and is made of a heavy element assumed to have little sputtering. Profiles are viewed as if standing inside the plasma looking outward.

Modest particle fluxes are observed compared to figure 5.10, but with a larger wetted area and more accessible geometry.

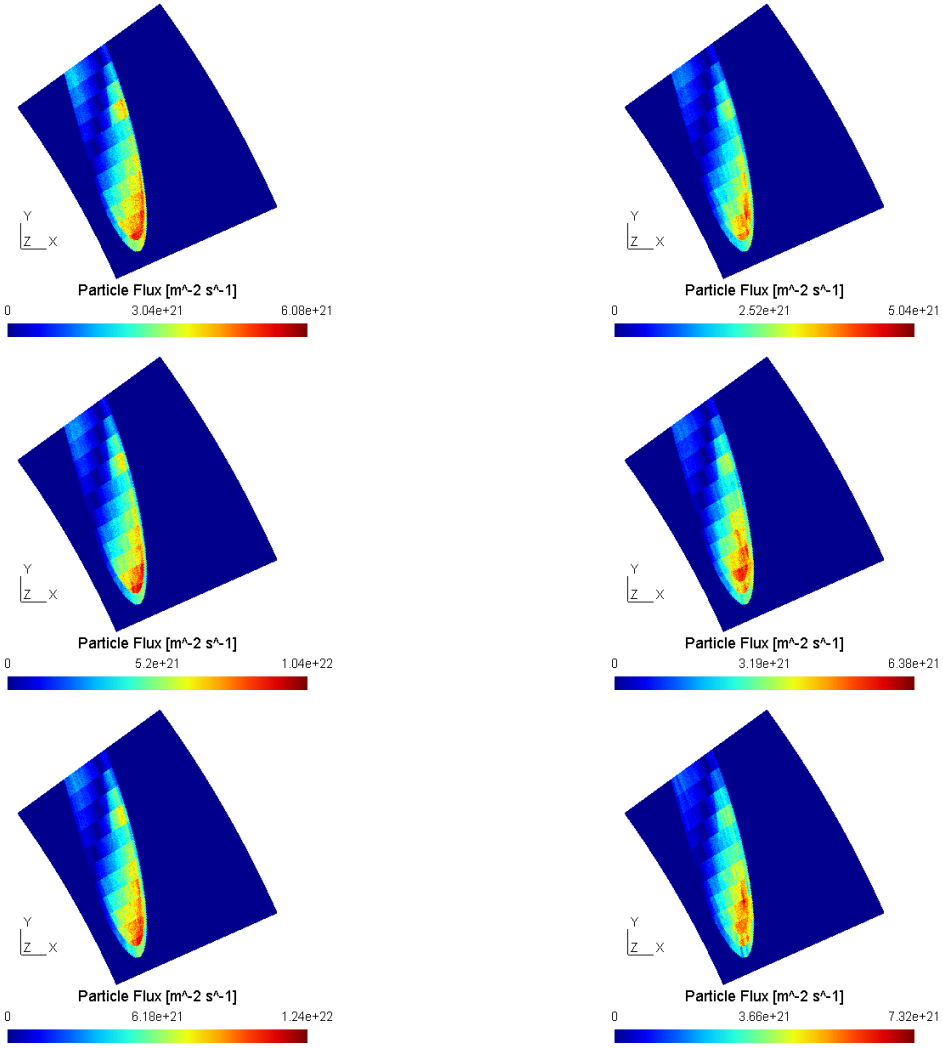


Figure 5.12: Simulated limiting surface particle flux comparisons for varying axial field strength and power levels. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The limiter is in the trench position and is made of a heavy element assumed to have little sputtering. Profiles are viewed as if looking down from the top of the device. Modest particle fluxes similar to figure 5.11 are observed. The trench would be symmetric about the upper edge as viewed here in a full device. Due to the lack of a companion surface on the top of the device, the magnitude of the trailing edge of the mirrored profile visible on the upper side is higher than is physical. The magnitudes of the hot spots are not affected by this, as they result primarily from the normal interaction of the simulated plasma on the lower edge of the surface.

Heat Fluxes to the Limiting Surface

Heat fluxes to the limiting surface as illustrated by figures 5.13, 5.14, and 5.15 for the simulated input power and axial field conditions follow trends identical to the particle deposition maps, and with identical reasoning. In these discharges which assume 100% efficient RF deposition, the resultant power flux to the limiter is on the order of 1-5% of the input power. Peak power fluxes on the order of 1-10 MW m^{-2} were calculated on the inboard limiter, and on the order of 0.1-1 MW m^{-2} on the outboard and trench limiters.

5.4 Impurity release

The plasma impurity density in EMC3 is managed by two boundary conditions similar to those used for the main ion species (a core exit flux and a decay length) and its source is the plasma-wall interface. In EMC3, the return flux of ions is given by a single user-set float which gives the return flux of ions as this factor times the incident particle flux. This method prevents any evolving influence from the profiles, and thus is inaccurate except when set to match experimental data for a given discharge.

As the electron temperatures at the limiter in the 26 kW discharges and in those expected in HIDRA are low, the impurity density is negligible for the heavy elements. Lithium is present, but at less than 1% to 10% of the main ion density, and only in a localized region above the limiter. Consequently, the power lost through impurity radiation is negligible.

In figure 5.16, impurity profiles for Lithium for a 26 kW, 0.5T discharge and a 250 kW, 87.5 mT discharge are presented. The latter case is shown for an illustration to the effects of the island chains on the impurity density and radiation profiles, when they are present. Expected operating conditions for HIDRA suggest the former case is more representative.

5.5 Modification of RF Input Power Efficiency to Approximate WEGA Conditions

As reported in section 1, WEGA electron temperatures with 36 kW of RF power were in the 10s of eVs and with 6 kW were in the 5-6 eV range. The profiles are relatively uniform across the

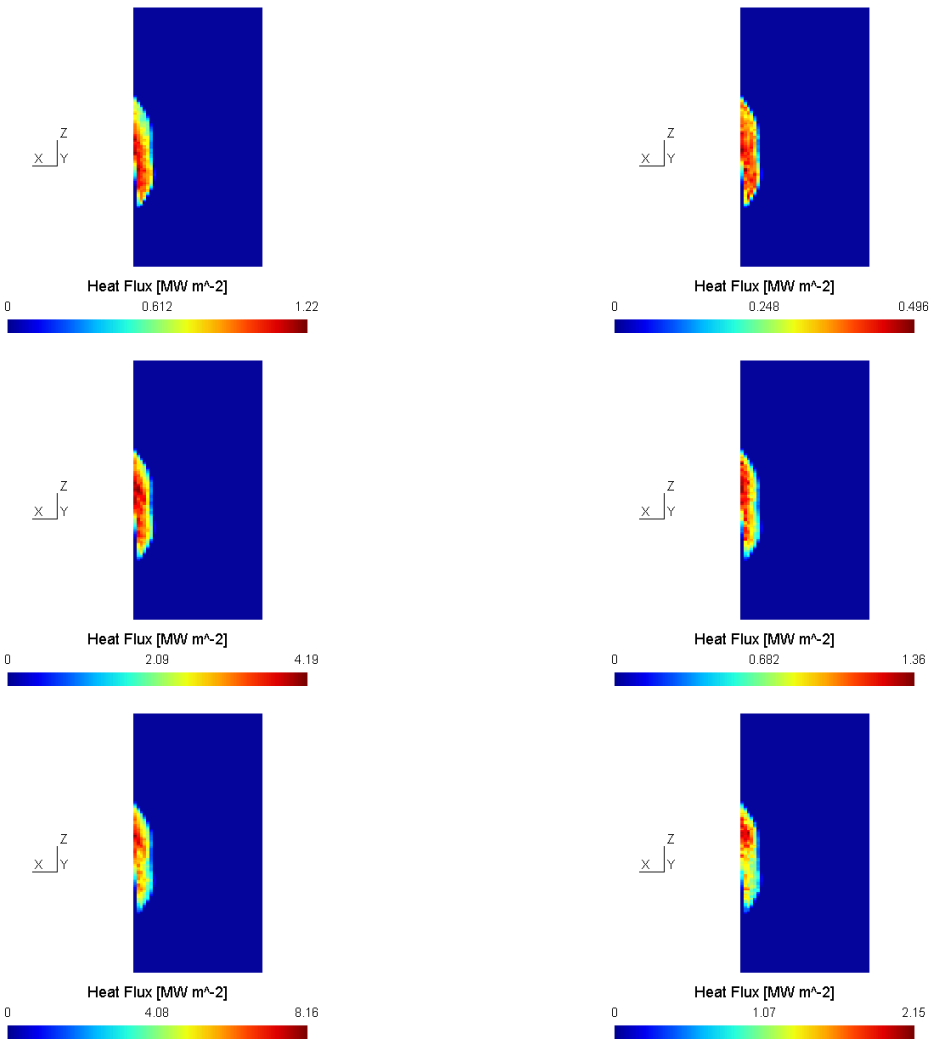


Figure 5.13: Simulated limiting surface heat flux comparisons for varying axial field strength and power levels. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The limiter is in the inboard position and is made of a heavy element assumed to have little sputtering. Profiles are viewed from the direction of the predominant magnetic field to view the regions of high flux.

Peak power fluxes are strongly dependent on both axial field strength and power input. Simulated values may reach up to 10 MW m^{-2} with a high axial field strength and 250 kW of input power, or up to 1 MW m^{-2} with a high axial field strength and 26 kW of input power. Notably, this is with an assumption of 100% efficient RF input power.

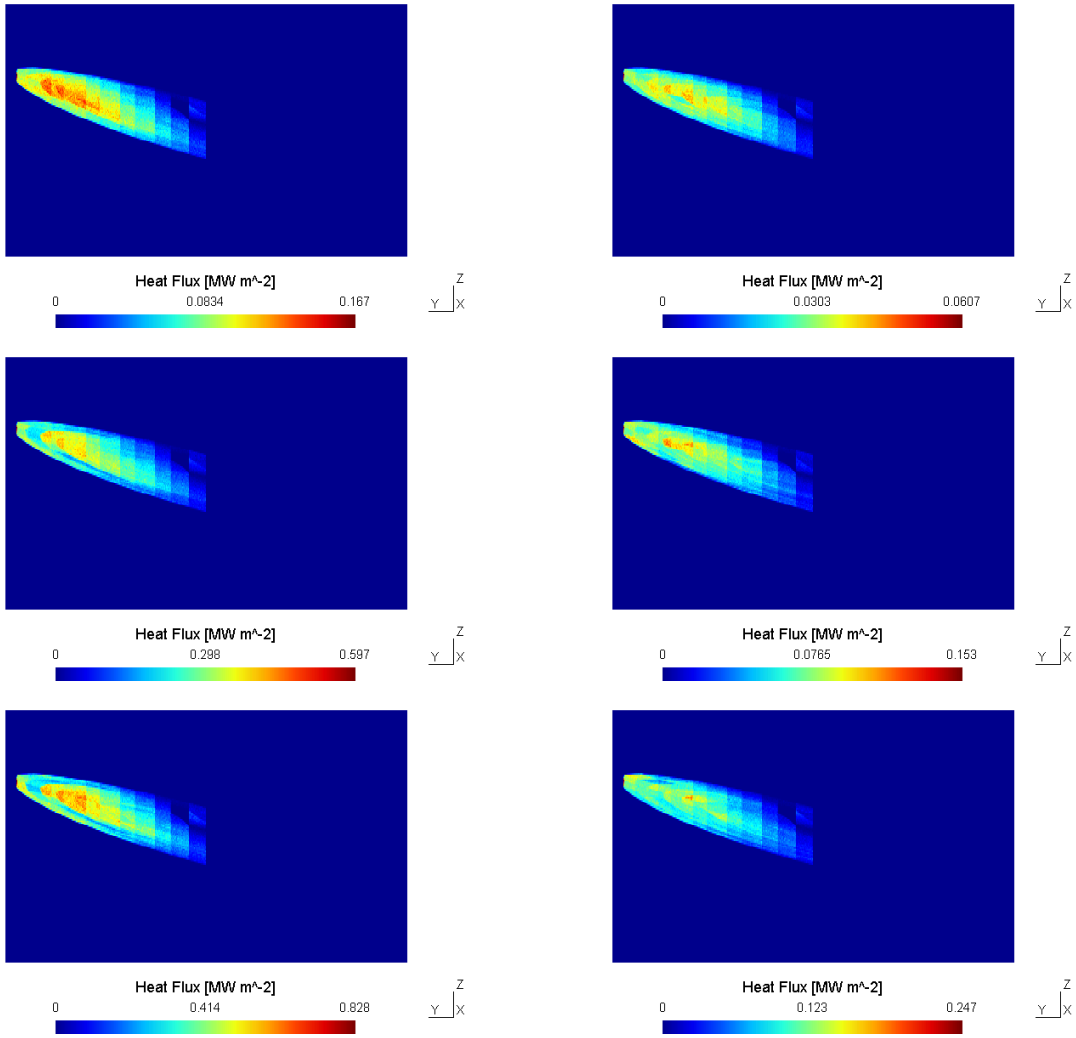


Figure 5.14: Simulated limiting surface heat flux comparisons for varying axial field strength and power levels. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The limiter is in the outboard position and is made of a heavy element assumed to have little sputtering. Profiles are viewed as if standing inside the plasma looking outward. Simulated power fluxes show a smaller magnitude compared to figure 5.13, but are realized over a larger area.

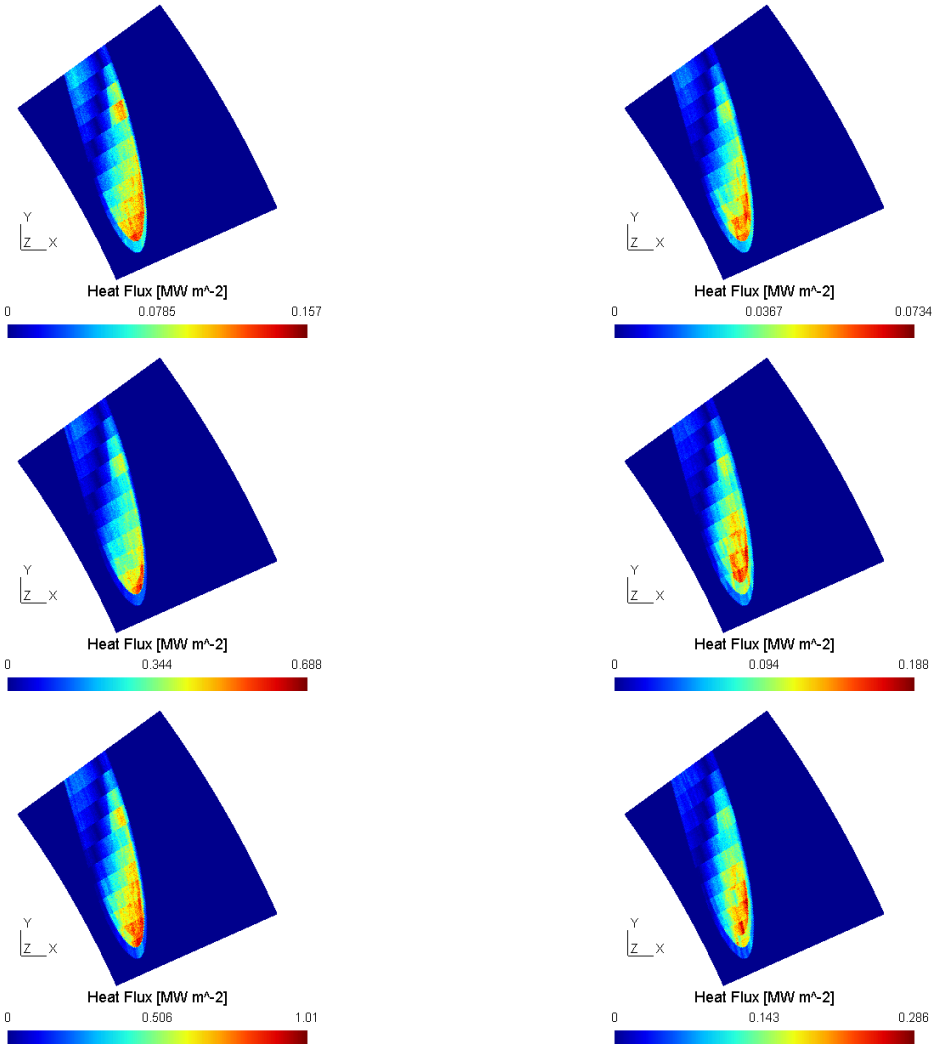


Figure 5.15: Simulated limiting surface heat flux comparisons for varying axial field strength and power levels. The left column is at an axial field strength of 0.5 T, and the right column at 87.5 mT. Rows from the top are for a 26 kW, 125 kW, and 250 kW device power discharge with 100% RF input efficiency. The limiter is in the trench position and is made of a heavy element assumed to have little sputtering. Profiles are viewed as if looking down from the top of the device.
Power fluxes in the limiter position are approximately 20% higher than in the outboard position. Qualities of the distribution described in figure 5.12 and in the previous section are present in the heat flux surface maps as well.

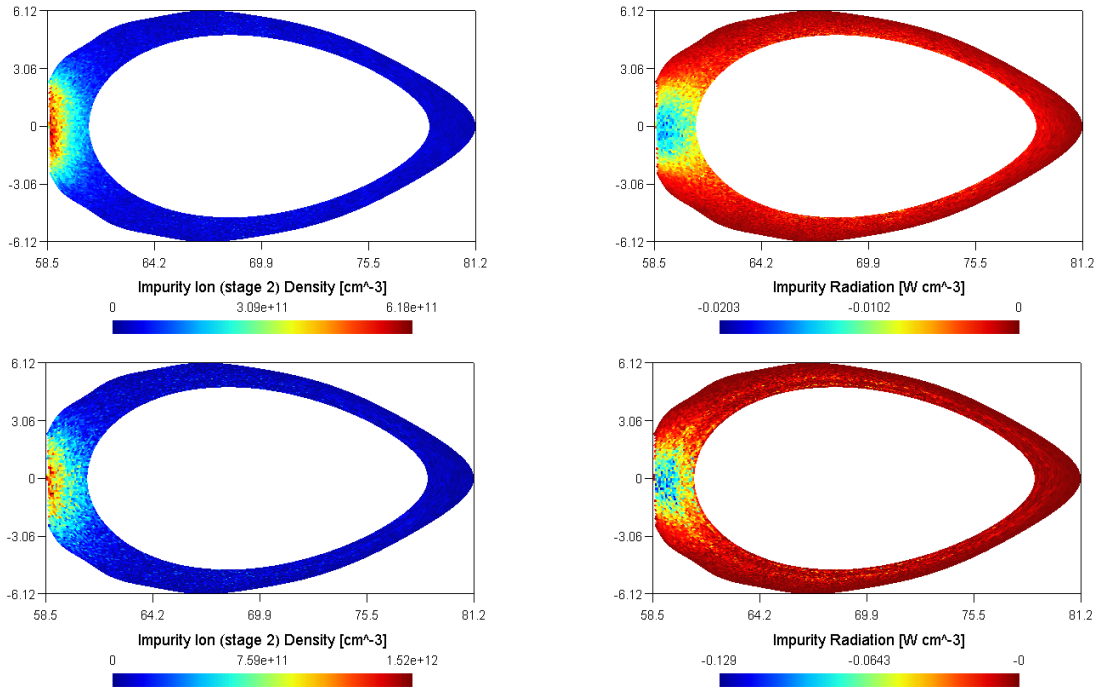


Figure 5.16: Simulated impurity density and radiation profiles for a high-field low-power (top) and a low-field high-power (bottom) discharge with a lithium limiter. The impurity radiation profile is given as a generic plasma energy source in the EMC3-EIRENE equations; since the impurity radiation is a sink of energy to the plasma, it is output with a negative sign.
The sputtered lithium is quickly lost due to the high diffusivity and forms only a small cloud over the limiter surface, but when island chains are present the enhanced diffusivity can allow noticeable impurity densities to move around to the outboard side of the plasma.

plasma in the edge region [57] [56]. Simulations performed for the present work suggested electron temperatures in the range of 20-40 eV in the edge region alone with a 26 kW discharge, as well as a strong falloff. Simultaneously, the assumption of 100% input power efficiency must be relaxed on thermodynamics grounds.

To attempt to match these values, the input power flux across the inner EMC3 boundary in a 26 kW discharge was set to 2.6 kW (10%), 7.8 kW (30%), and 13 kW (50%). Figure 5.17 shows the resulting T_e and D_B profiles, and figure 5.18 shows the effect of this power reduction on the particle and heat fluxes reaching the surface.

In addition, [51] provides an estimate of the diffusivity, which was seen to be Bohm-like, as $\approx$ 2 m^2 s^{-1}. This singular value is given to be representative of an entire profile, and is matched at varying points across the profiles in figure 5.17 depending on the input power, but remains an order of magnitude estimate.

5.6 Discussion

RF Efficiency and Plasma Profiles

Electron temperatures in WEGA with heating schemes similar to those in HIDRA are 10-20 eV higher than should be expected. By lowering the input power on the 26 kW discharge to represent RF deposition inefficiency resulting from core radiation and incomplete absorption, it is possible to bring the electron temperatures to approximate WEGA experimental data. The radial falloff on electron temperatures is seen to be higher than WEGA results in these simulated HIDRA discharges, but at the limiting surface the magnitude is within a few eV of the bulk temperature reported in WEGA results. These less-efficient discharge simulations may then serve as a lower bound to the fluxes which may be extracted from the limiting surface.

Bounding the Particle and Heat Fluxes

In many cases, data from WEGA is lacking similarities to the expected operating conditions of HIDRA. There is little data on a discharge of 26 kW only, and all reported data is on non-limited discharges. In the absence of these detail, which provide crucial boundary condition data

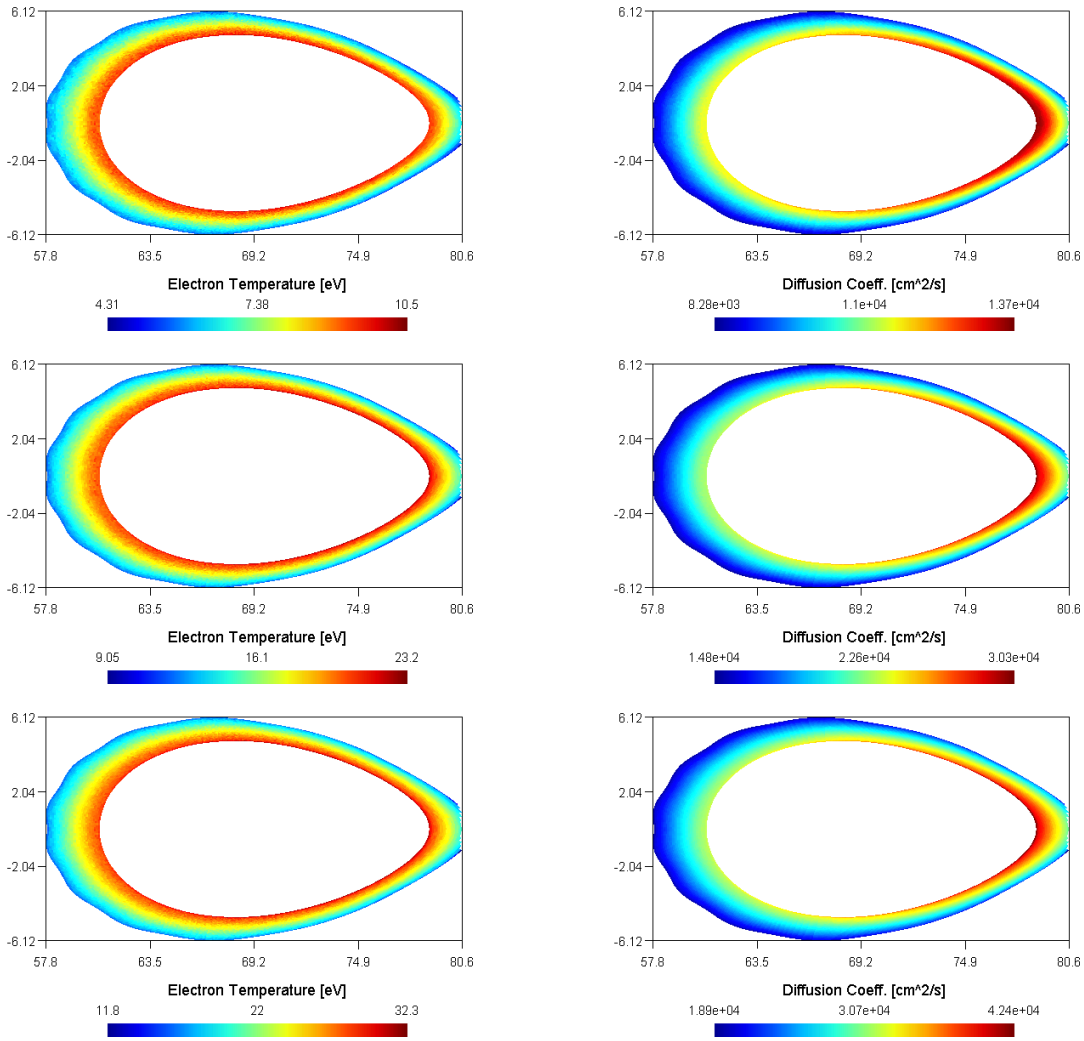


Figure 5.17: T_e and D_B for 10% (top), 30% (middle), and 50% (bottom) RF efficiency in a 26 kW discharge with an axial field strength of 0.5 T. The energy flux boundary condition is set to the respective percent in order to emulate RF deposition inefficiencies and energy lost from the core via radiation. The 10% case has a stronger temperature drop-off than can be observed in WEGA results, but the magnitude of the temperature in the edge region ($\approx$ 5-10 eV) matches experimental results.

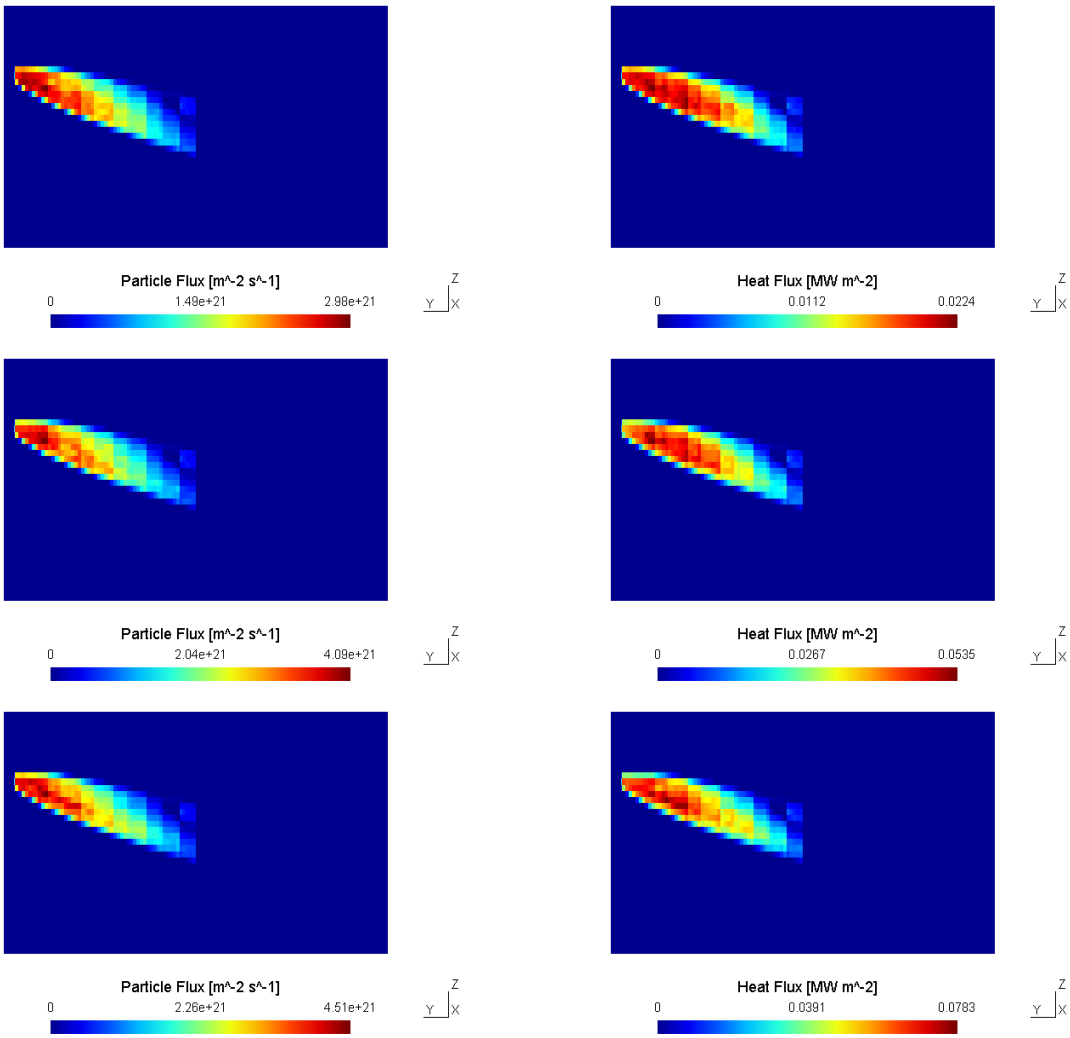


Figure 5.18: Particle and heat fluxes to an outboard limiting surface made of a heavy material for 10% (top), 30% (middle), and 50% (bottom) RF efficiency in a 26 kW discharge with an axial field strength of 0.5 T. The energy flux boundary condition is set to the respective percent in order to emulate RF deposition inefficiencies and energy lost from the core via radiation.

to run EMC3-EIRENE on, and in the lack of a fully self-consistent `HIDRAmod` which would include calculations on RF deposition efficiency and core power losses, it is possible to instead simulate for minimum and maximum values that may be expected from a discharge and provide reasonable bounds of confidence on the expected results.

For the case of the 26 kW of power that HIDRA will operate with, these bounds may be set for a 0.5 T discharge by the combination of figures 5.11, 5.14, and 5.18. Reasonably, HIDRA may expect peak heat fluxes between 0.02 MW m^{-2} and 0.2 MW m^{-2} for the readily accessible outboard limiter position. Similarly, for the inboard and trench limiters, HIDRA may expect peak heat fluxes between 0.1 and 1 MW m^{-2} and between 0.015 and 0.15 MW m^{-2}. In terms of particle fluxes, HIDRA may expect up to 0.7 A cm^{-2} for the inboard limiter, 0.08 A cm^{-2} for the outboard limiter, and 0.09 A cm^{-2} for the trench limiter.

Optimization Strategies for Extracted Fluxes

These HIDRA simulations have room for improvement with an intention of simulating a fully representative discharge, and then potentially optimizing device characteristics to increase these calculated heat and particle fluxes. Currently, the profiles are periodically symmetric with periodically symmetric limiter placements. However, it is known that WEGA in its latest incarnation has aberrations in the magnetic field which create significant asymmetry [58] [52]. By creating a magnetic mesh over the entire torus including this asymmetry (or potentially any intentionally induced symmetries), it may be possible to find a more opportune location for particle and heat fluxes. Using extra magnetic coils to perturb the plasma may allow for a higher density of flux surface intersections on the limiting surface, increasing the resulting particle and heat fluxes on a surface. Potentially, removing the limiter symmetry alone may increase the resulting fluxes.

5.6.1 Scaling Laws for Peak T_e, T_i, D_B, and Surface Fluxes

From the analyzed data, fits may be done to extract scaling laws for quantities of interest. Figures 5.19 to 5.23 illustrate these scaling laws in terms of the applied RF power for T_e, T_i, D_B, and surface fluxes, with tables 5.2 and 5.3 documenting their statistics. In general, the T_e, D_B, and Surface flux profiles were seen to obey approximately a power law with the particle fluxes saturating at

Table 5.2: High-field peak profile fit statistics

Profile	Fit Type	Coefficients (95% confidence)	R-square
T_e	$f(x) = ax^b$	$a = 7.395$, $b = 0.5466$	0.999
T_i	$f(x) = (ax + b)/(x + c)$	$a = 2.222$, $b = 152.5$, $c = 20.11$	0.9878
D_B	$f(x) = ax^b$	$a = 9609$, $b = 0.5575$	0.9996
Heat Flux	$f(x) = ax^b$	$a = 1.478$e-2, $b = 0.7456$	0.9976
Particle Flux	$f(x) = ax^b + cx$	$a = 2.672$e-2, $b = 0.5212$, $c = -1.116$e-3	0.9842

Table 5.3: Low-field peak profile fit statistics

Profile	Fit Type	Coefficients (95% confidence)	R-square
T_e	$f(x) = ax^b$	$a = 2.704$, $b = 0.6098$	0.9938
T_i	$f(x) = (ax + b)/(x + c)$	$a = 1.593$, $b = 105.8$, $c = 24.68$	0.9723
D_B	$f(x) = ax^b$	$a = 2.138$e4, $b = 0.5995$	0.9986
Heat Flux	$f(x) = ax^b$	$a = 10.23$, $b = 0.5369$	0.9714
Particle Flux	$f(x) = ax^b + cx$	$a = 2.336$e-2, $b = 0.3719$, $c = -3.96$e-4	0.9774

higher powers, while T_i follows a rational scaling law with numerator and denominator of degree 1.

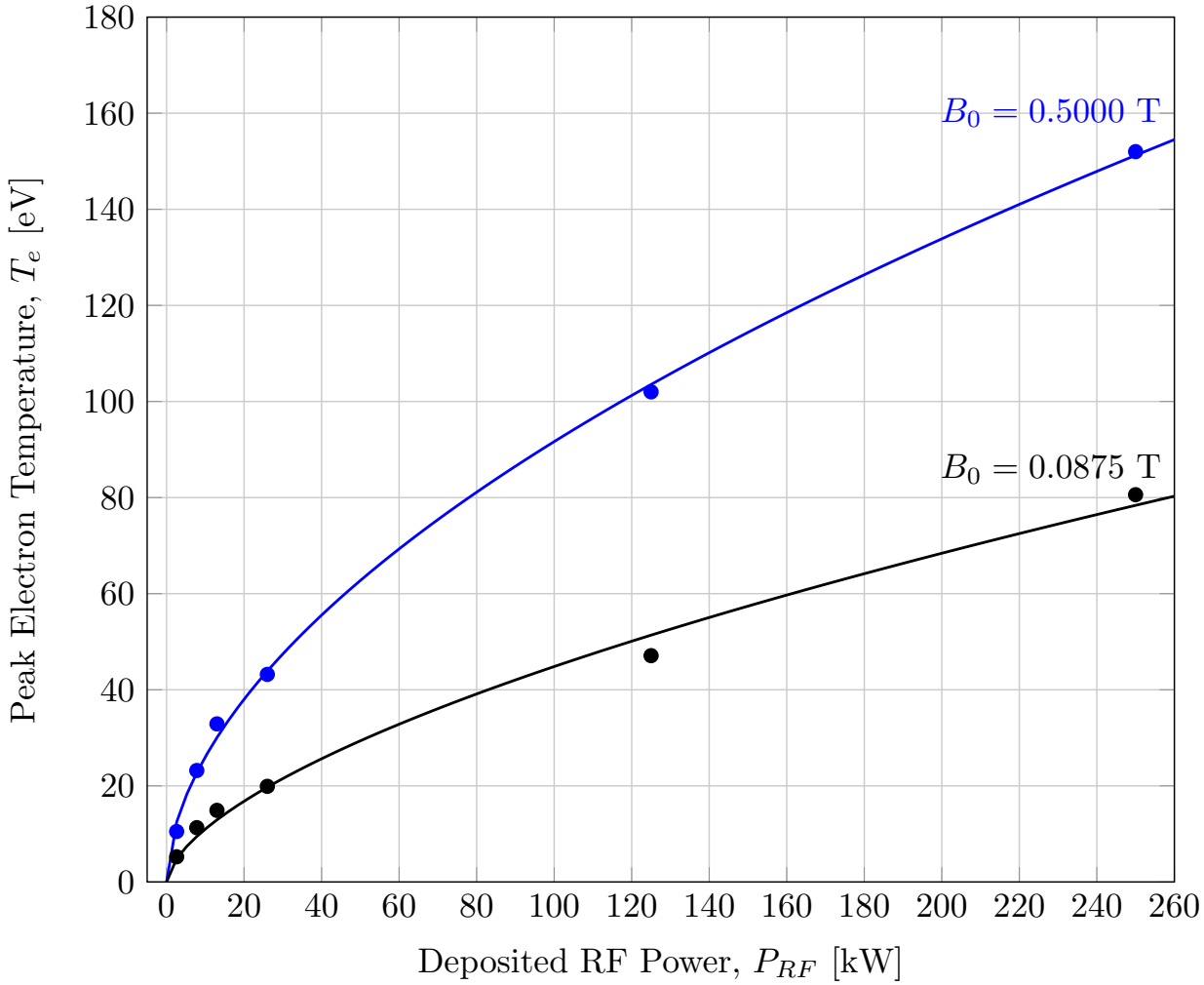


Figure 5.19: Peak profile fits for the electron temperature versus RF power in high and low axial field strength discharges. Clear power-law trends are observed in the peak electron temperature, with an exponent in the range $b \approx 0.55 - 0.60$.

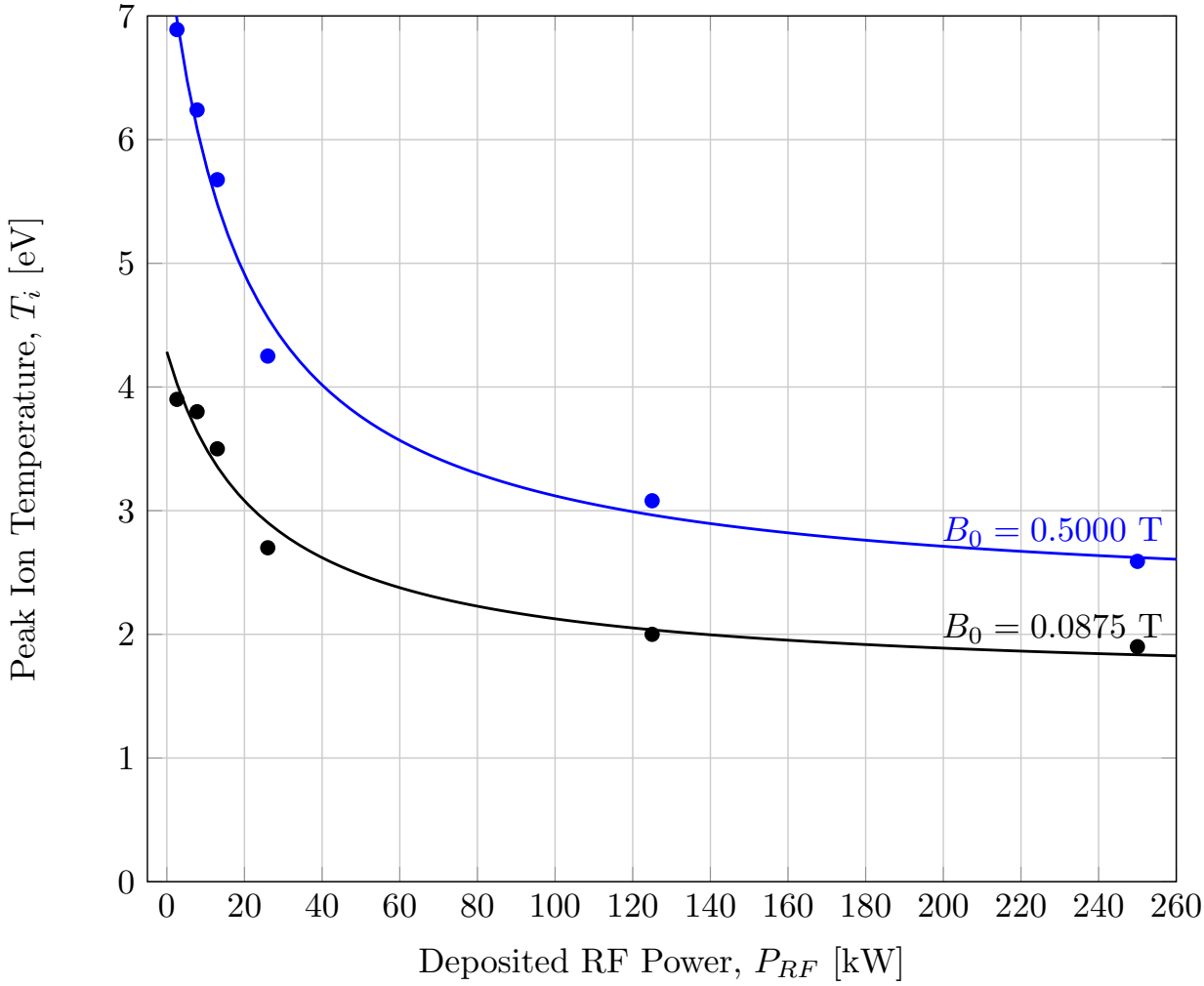


Figure 5.20: Peak profile fits for ion temperatures versus RF power in high and low axial field strength discharges. The ion temperatures follow a rational law with numerator and denominator degree 1, approaching the values $T_i = 2.2$ eV and $T_i = 1.6$ eV in high- and low-field discharges as power increases. With little time to equilibrate before particles are lost to the high diffusivity in higher-power (and higher T_e) discharges, the ion temperature is maximized by lower-power discharges.

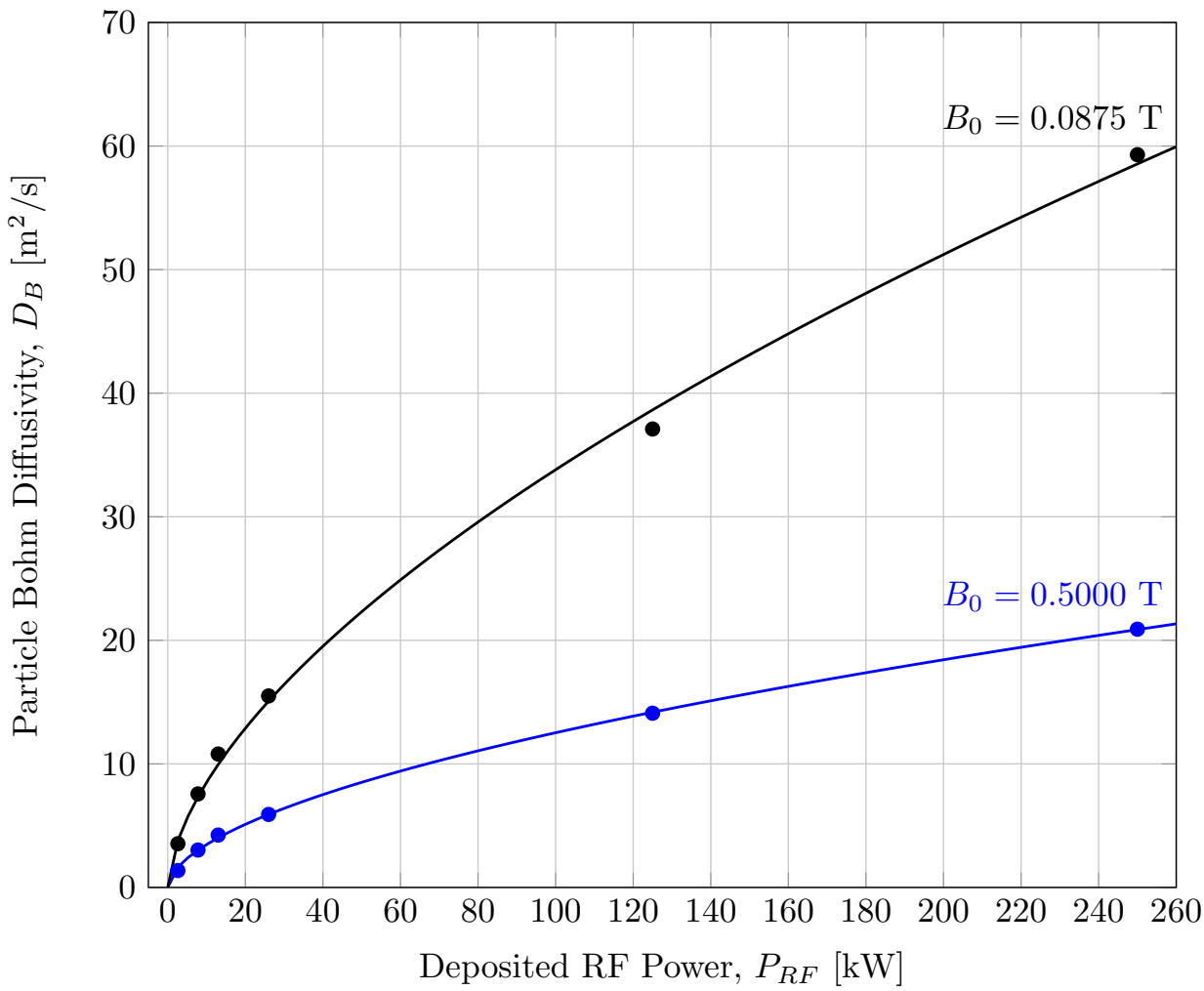


Figure 5.21: Peak profile fits for particle diffusivity versus RF power in high and low axial field strength discharges. Clear power-law trends are observed in the peak diffusivity, with an exponent in the range $b \approx 0.55 - 0.60$, similar to the electron temperature in figure 5.19.

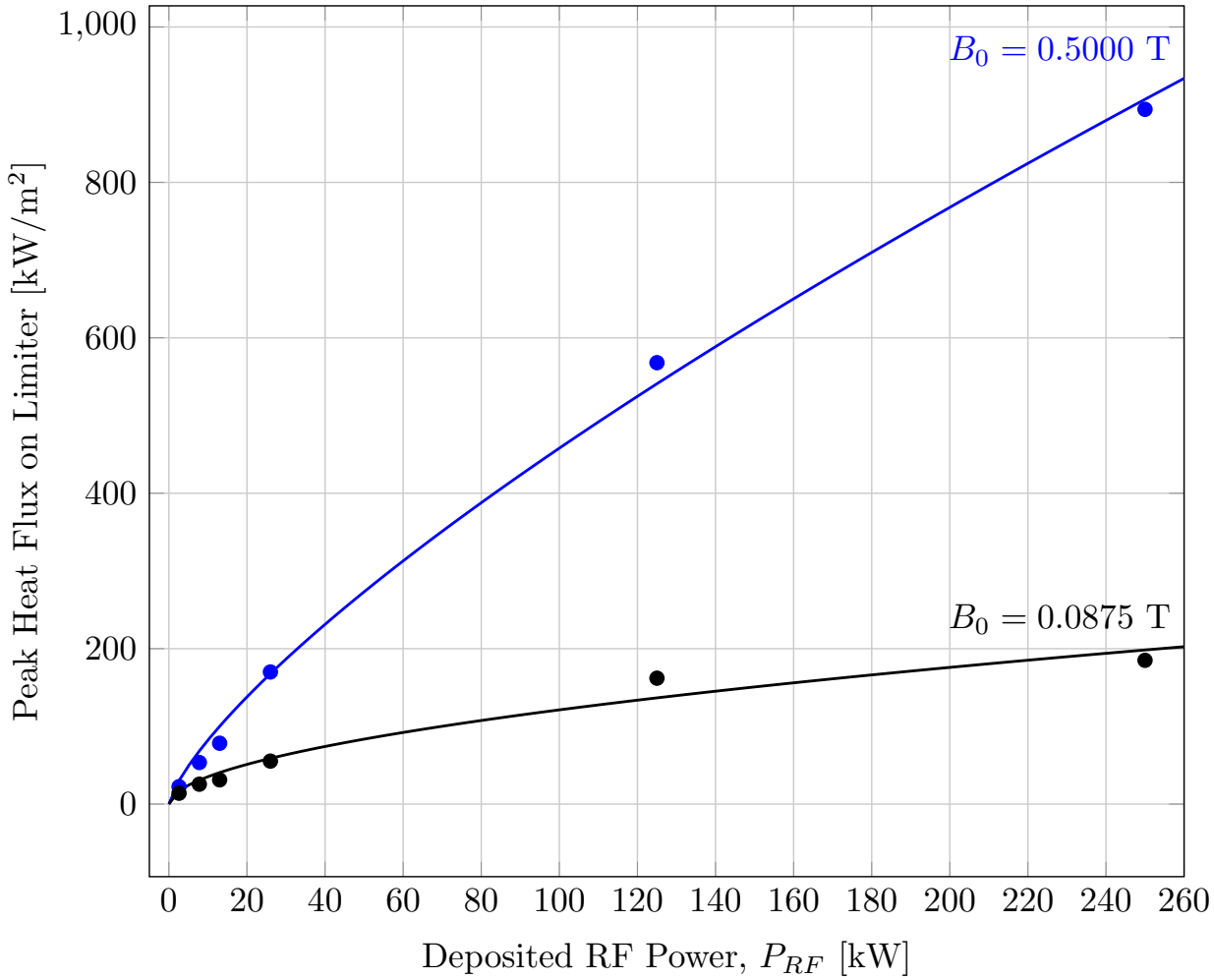


Figure 5.22: Peak profile fits for heat flux (outboard limiter) versus RF power in high and low axial field strength discharges. Peak power fluxes in the high-field case are more linear than T_e and D_B, with $b \approx 0.75$, while the low-field case has an exponent $b \approx 0.54$ similar to the electron temperature in figure 5.19.

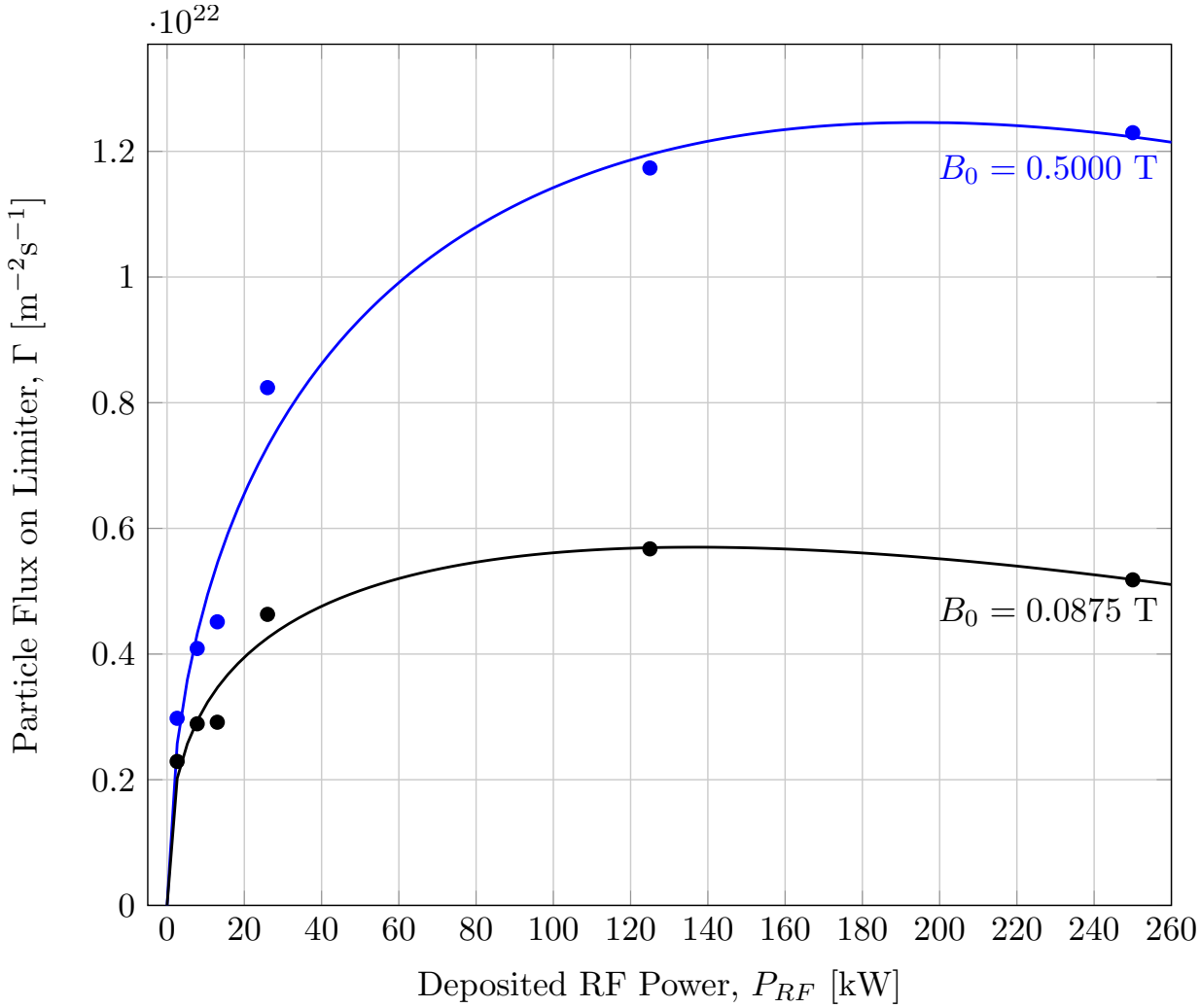


Figure 5.23: Peak profile fits for particle flux (outboard limiter) versus RF power in high and low axial field strength discharges. Particle fluxes were seen to saturate at high input powers in both the high- and low-field discharges.

Chapter 6

Conclusions

6.1 Summary of Outcomes

As part of this thesis the initial framework of `HIDRAmod`, a 3D edge plasma simulation environment for the HIDRA stellarator at Illinois, has been established. This work has achieved the following objectives:

1. The creation of a 3D flux-conserving field-aligned linear polyhedral mesh generator, named TORMESH;
2. The integration of the 3D plasma and neutral transport solver EMC3-EIRENE into `HIDRAmod`;
3. The extension of the EMC3-EIRENE algorithm to include a local Bohm-type diffusivity, and the characterization of its numerical convergence;
4. A numerical characterization of the edge transport conditions in HIDRA, with calibration against available data from the previous WEGA facility;
5. The identification of scaling laws of the main relevant plasma parameters: electron temperature, particle diffusivity, heat flux, and particle flux on an outboard limiter (i.e., at the most practical location for PMI experiments).

Electron and ion temperature, density, and diffusivity profiles in the plasma were investigated for axial magnetic field strengths of 0.5 T and 87.5 mT, and surface heat and particle deposition maps created in post-processing with these quantities. An inboard limiter design was seen to provide the highest surface fluxes. Geometrical considerations suggest an outboard to be the most practical, with a trench limiter to be roughly equivalent in peak flux magnitudes and wetted area.

For the electron temperature, Bohm-like diffusivity, and heat and particle fluxes on the limiter surface, scaling laws were obtained as a function of power. Peak electron temperatures, particle diffusivity, and heat fluxes at the outboard limiter were seen to follow approximately a power-law of type $f(P_{RF}) \propto aP_{RF}^{b}$, with typical exponents in the range $b \sim 0.55-0.60$. Higher magnetic fields have the tendency to "linearize" the heat flux dependence upon the RF power, with exponents in the range of $b \sim 0.75$. Particle fluxes on the outboard limiter are seen to saturate first, and then slightly decline for RF powers in excess of 120 kW in the low-field case and 180 kW in the high-field case.

6.2 Future Work

`HIDRAmod` would benefit substantially from three additional components: a full-wave solver to calculate RF power deposition efficiencies, a radiative core model to solve for the power loss in the core, and an improved PMI interface to allow predictive use of the impurity species. Each of these components would allow for a more self-consistent simulation, allowing better comparisons to ongoing experimental activities with fewer user-set variables. In particular, the first two would allow for a predictive estimation of heat and power fluxes on the limiting surfaces beyond the setting of an interval of confidence.

The quality of simulations in an analytic sense, rather than predictive, could be be increased significantly by the presence of operational data on the current device setup. Without any extra components in `HIDRAmod`, a short amount of manual iteration between boundary power and particle flux conditions on the part of an individual would allow them to match the discharge profiles in the physical device, and hence obtain the deposition profiles to a much narrower interval of confidence.

The magnetic field in WEGA was known [58] [52] to have a symmetry-breaking error field. If the field is still present on the HIDRA device, it must be taken into account by a full-torus magnetic mesh. At the same time, with more toroidal zones the boundary conditions may be updated to better represent the locality of the power deposition, as the relevant boundary conditions may decrease at further toroidal angles from the RF sources. Neutral sources could be updated to provide a more realistic coverage with one input location versus the currently simulated five. A

full-torus mesh would also allow the simulation of fully arbitrary limiting surfaces that may be too large for the current half-period mesh, or would allow for the simulation of limiting surfaces with periodicities of one.

Taking these into account, an optimization may then be then be done to find operating conditions, limiter placements, and potentially magnetic field perturbations which maximize the particle and energy fluxes to create a full-fledged stellarator for PMI investigation.

Appendix A

A Description of `TORMESH` and `LIM_OCC`

A.1 An Overview of TORMESH

TORMESH is a multi-purpose code which may be used to interface with FIELDLINES for purposes other than the generation of a field-aligned magnetic mesh, such as the plotting of Poincaré sections or the writing of FIELDLINES input files for general use. To ensure this, it is necessary for TORMESH to contain a description of the FIELDLINES input file that may be passed almost verbatim.

TORMESH is structured around three calls to FIELDLINES. The first call is on many points which sample the domain, testing the connection lengths and using the result to rescale the domain based on the desired EMC3-EIRENE mesh sizes. The second call uses the rescaled domain as initial value points (seed points) and traces out the Poincaré section. Internally, TORMESH then filters or splines the surfaces (enforcing up-down symmetry if desired) and interpolates the surfaces for the radial and poloidal starting positions of the final mesh. TORMESH then calls FIELDLINES a final time on these points to trace out the resulting mesh.

The source code for TORMESH is divided into three files: `classes.py`, which has a listing of the variables and class structure used by TORMESH; `inputs.py`, which contains all of the inputs for a given run of TORMESH; and `tormesh.py`, the main file. The input file will be described in this appendix, as it pertains to the running of TORMESH as well as FIELDLINES.

A.2 TORMESH - inputs.py

In the following demonstration of a TORMESH input file, each block of code will be followed by its accompanying description.

```
#!/usr/bin/env python

import numpy as np
import classes

def userdata():
```

The code begins with the usual import commands. The function `userdata()` is called at the beginning of a TORMESH run to overwrite the default values in the `inputStructure` class. The `inputStructure` class is used to store all relevant variables for the code.

```
# Degree to radians
d2r = np.pi/180.0

# Initialize input structure
data = classes.inputStructure()

#-----------------------------------------------------------
#
#     CURRENT COILS
#
#-----------------------------------------------------------

# Device name
data.name = 'HIDRA'

# Device periodicity
data.periods = 5
```

The input structure is initialized, and the device name and periodicity are set. Periodicity is necessary for the coil information input file for `FIELDLINES`.

```
# ===========================================================
# ---- HIDRA Helical Field Coils ------------------------
# ===========================================================

# 'Coil Type' groups coils together
# in DEVICE.coil for easy changing currents
# on different current systems
CoilType = 1
CoilName = 'Helical'
data.coil_types += 1  # increase coil type counter

# Coil parameters
Ns    = 150          # Number of points in coil
rs    = 0.19+0.05    # Minor radius of coils [m]
Rs    = 0.72         # Major radius of coils [m]
m     = 2.5          # Coil periodicity [decimal], expressed
                     # as (poloidal / toroidal) revolutions

Handedness = -1.0    # (+-1) Multiplies the Z-component during
                     # coil generation

# Poloidal offset [rad]
# (0.0 deg offset is outboard at Z = 0)
AngOffset = np.concatenate((np.linspace( -25.0*d2r, 25.0*d2r, 7 ),
                            np.linspace((-25.0+90.0)*d2r,   \
                            (25.0+90.0)*d2r,7)))
# Current [A]
curMag  =  -2000.
```

```
Current = np.concatenate( (np.linspace( curMag,  curMag, 7) ,
                           np.linspace(-curMag, -curMag, 7) ) )
data.EXTCUR.append(Current[0])

# Coil Generation Loop
for i in range(len(AngOffset)):
   # append new coil
   data.coils.append(classes.coil([Ns, CoilType]))
   # increase coil reference index
   data.coil_index += 1
   data.coils[data.coil_index].name = CoilName
   data.coils[data.coil_index].setCoilHelical( [rs, Rs, m, \
                                                AngOffset[i], \
                                                Handedness, \
                                                Current[i]] )
```

This block generates the helical field coils for HIDRA, using the `setCoilHelical()` function. Care was taken to ensure that most variables have sensible names, and that most lines in the input and most sections in the rest of the code have comments instructing the potential user in their usage.

The line referring to `EXTCUR` is a `FIELDLINES`-only variable. The `EXTCUR` field allows one to overwrite the current information in `FIELDLINES`, which is nominally stored in the `HIDRA.coils` file (or any other device name) with the coil information.

```
# ==================================================
# ---- HIDRA Toroidal Field Coils ----------------
# ==================================================

CoilType = 2
CoilName = 'Toroidal'
data.coil_types += 1

Ns        = 50              # Number of points in coil
R0_coils  = 0.72            # Major radius of the center of the coil[m]
Ra_coils  = 0.19 + 0.1558   # Coil radius [m]
N_coils   = 40              # Number of coils
Current   = 500.*13.          # Coil current [A]
# N_windings = 13           # Number of windings per coil

data.EXTCUR.append(Current)
phi = np.linspace( 0.0, 2.0*np.pi, N_coils+1)+(2.0*np.pi/N_coils)/2

# Coil Generation Loop
for iCoil in range( N_coils ):
   Xc = R0_coils * np.cos( phi[iCoil] )
   # Xc, center of the i-esim coil
```

```
    Yc = R0_coils * np.sin( phi[iCoil] )
    # Yc, center of the i-esim coil
    Zc = 0.0                          # Zc, center of the i-esim coil
    nx = -np.sin( phi[iCoil] )        # nx, Unit Normal to the coil
    ny =  np.cos( phi[iCoil] )        # ny, Unit Normal to the coil
    nz =  0.0                         # nz, Unit Normal to the coil
    data.coils.append(classes.coil([Ns, CoilType]) )
    data.coil_index += 1
    data.coils[data.coil_index].name = CoilName
    # make a loop
    data.coils[data.coil_index].setCoilLoop([ Ra_coils, Current ])
    # rotate to normal n
    data.coils[data.coil_index].rotateCoil([nx, ny, nz])
    # translate to Xc
    data.coils[data.coil_index].translateCoil([[Xc], [Yc], [Zc]])
```

Toroidal coil generation for HIDRA illustrates the usage of the `setCoilLoop()` function, as well as its companion functions `rotateCoil()` and `translateCoil()`.

```
# ==================================================
# ---- HIDRA Vertical Field Coils ----------------
# ==================================================


#--------------------------------------------------
#
#     VACUUM VESSEL
#
#--------------------------------------------------

# Major radius of the vacuum vessel [m]
data.VacuumVessel_R0  =  0.72

# Minor radius of the vacuum vessel [m]
data.VacuumVessel_ra  =  0.19
```

The vertical coils on HIDRA were not simulated in the present work. The proper generation and positioning of them would look similar to the toroidal coils and use the same functions, but would be simpler as they share a common normal.

```
#--------------------------------------------------
#
#     MAGNETIC MESH
#
#--------------------------------------------------

# Number of periods to output
# (must be less than or equal to device periodicitiy)
data.num_periods_output =  1
```

```
# force up/down symmetry on seat point plane
data.force_up_down_sym = 1

# Location of first EMC3 Plane, integer
data.PHI_START_surf_tracing_index = 0
# location of first EMC3 Plane, actual angle (deg)
# -- User is responsible for ensuring the index above matches
data.PHI_START_surf_tracing = 0.
data.PHI_END_surf_tracing = data.PHI_START_surf_tracing + \
                            int( 360. * data.num_periods_output /
                                 data.periods )
```

Information for generating the magnetic mesh is written in this section. This first section covers general information for the start and end planes of the resultant mesh. It is illustrating to go variable by variable:

- `num_periods_output` controls how many periods the resulting mesh will cover. For a stellarator, this value should either be the device periodicity or one.
- `force_up_down_sym` forces up-down symmetry on the beginning surface by creating another surface which is mirrored about the x-y plane, and averaging both surfaces.
- `PHI_START_surf_tracing_index` sets the toroidal angle which the profiles are started from for EMC3. This option may be used to generate grids with multiple zones.
- `PHI_START_surf_tracing` sets the toroidal angle which the profiles are started from for `TORMESH` and `FIELDLINES`.

```
# Effective radial location of surfaces
#   Options for eff_emc3_outer :
#     -- 'vac_vessel' -- outer boundary of EIRENE
#                        is vacuum vessel radius (torus only)
#     -- % of edge (in decimal - 100% = 1.0)
#   Options for everything else
#     -- % of edge (in decimal - 100% = 1.0)
data.eff_eirene_outer = 'vac_vessel'
data.eff_eirene_inner = 0.01
data.eff_emc3_inner = 0.75
data.eff_emc3_outer = 0.95
# Trace surfaces or linearly interpolate for final grid?
# (Trace has trouble with islands)
# 'trace' or 'interpolate'
data.emc3_region_filltype = 'interpolate'
```

```
data.eirene_region_filltype = 'interpolate'
```

In this section, information on the surface spacing and edge location are input. For this section, the variables are in terms of a percent of the distance from the magnetic axis to the plasma edge. Since the edge is determined automatically, leaving these values as percents allows for portability across differing ι.

- `eff_eiren_outer` is the effective EIRENE region outer boundary. Currently, only 'vac˙vessel' is implemented, which indicates that the domain is to be extended to the vacuum vessel radius.
- `eff_eirene_inner` is the percent of the magnetic axis to LCMS distance on which the EIRENE region will start.
- `eff_emc3_inner` is the percent of the magnetic axis to LCMS distance on which the EMC3 region will start.
- `eff_emc3_outer` is the percent of the magnetic axis to LCMS distance on which the EMC3 region will end.
- `emc3_region_filltype` and `eirene_region_filltype` allows one to either directly use the surfaces output by `FIELDLINES` and splined or filtered by `TORMESH`, or to interpolate new surfaces between the surfaces which were defined as the inner and outer surfaces for a given region in the previous four variables.

```
# Number of Mesh points in radial (R),
# poloidal (TH), toroidal (PHI) direction
# number of points in EMC3 region (number of cells -1)
data.N_R_EMC3_POINTS  = 31
# number of points in inner eirene region (number of cells -1)
data.N_R_INNER_POINTS = 6
# number of points in outer eirene region (number of cells)
data.N_R_OUTER_POINTS = 0
data.N_TH_POINTS  = 100
data.N_PHI_POINTS = 72+1
# zone info
data.TOROIDAL_ZONES = 1
# needs at least 1 extra value
# (value ignored, keeps list range consistent with variable setting)
data.N_PHI_POINTS_perzone = [37, 0, 0]
```

Values for the final EMC3 grid are found in this section.

```
data.output_flux_surf_and_quit = True
# list of surface numbers to output (radially outwards, zero-indexed)
data.flux_surface_output_vector = ([ 0, 1,2,3,4,5,6,7,8,9,10,
                                     11,12,13,14,15,16,17,18,19,20,
                                     21,22,23,24,25,26,27,28,29,30,
                                     31,32,33,34,35 ])
# list of angles in degrees to output
data.flux_surface_output_angles = [0., 36.]
```

This section allows one to use TORMESH to instead output files containing all the points in a splined and filtered flux surface.

```
# Don't touch this - syncs surface numbers for various parts of code.
data.N_R_POINTS = data.N_R_OUTER_POINTS + data.N_R_EMC3_POINTS + \
                  data.N_R_INNER_POINTS - \
                  ( bool(data.N_R_INNER_POINTS) )
```

It is necessary to ensure that surfaces are not counted twice.

```
#-----------------------------------------------------------
#
#     FIELDLINES INPUT
#
#-----------------------------------------------------------
```

In the FIELDLINES section, with the exception of the next two blocks, all variables share a name with FIELDLINES input file variables unless otherwise noted. Thus, this section may serve as a reference for FIELDLINES as well.

```
# Flag to run fieldlines -- defaults to true
# Set to false for quicker debugging
data.run_fieldlines = 1
```

This block is useful only for developers, to prevent FIELDLINES from rerunning. It allows for rapid testing of portions of TORMESH which are computationally light but dependent on FIELDLINES data.

```
# Flag to output poincare data --- defaults to False
# (Slow!)
data.output_poincare = False
data.poincare_per_period = 5
```

An option to output Poincaré sections is included.

```
# Free Boundary Parameters (needed only for runs from mgrid file)
data.LFREEB = True
```

```
data.MGRID_FILE = 'mgrid_wega_with_VFCoils'
data.NZETA = 72
data.NVACSKIP = 0
```

These options are only for FIELDLINES runs with mgrid, which was not tested in this work.

```
# B-grid used by FIELDLINES for fieldline interpolation
# Number of radial   gridpoints, overridden if using mgrid
data.NR      = 101
# Number of toroidal gridpoints, overridden if using mgrid
data.NZ      = 101
# Number of vertical gridpoints, overridden if using mgrid
data.NPHI    = 36
# Minimum extent of radial grid, overridden if using mgrid
data.RMIN    = 0.46
# Maximum extent of radial grid, overridden if using mgrid
data.RMAX    = 0.97
# Minimum extent of vertical grid, overridden if using mgrid
data.ZMIN    = -0.25
# Maximum extent of vertical grid, overridden if using mgrid
data.ZMAX    = 0.25
# Minimum extent of toroidal grid, overridden if using mgrid
data.PHIMIN = 0.0
# Maximum extent of toroidal grid, overrideen if using mgrid
data.PHIMAX = 1.25664

# Fieldline diffusion
data.MU = 0.0        # Fieldline diffusion (mu=sqrt(D*tau*2))
```

The FIELDLINES input variables for the coarse magnetic mesh are listed here. FIELDLINES calculates the field on a coarse mesh and interpolates between the nodes listed in this section in order to more efficiently calculate magnetic fields, as per chapter 3.

```
# Coordinates of starting locations of fieldlines
data.start_points = 46
# Use this or the following lines for R_ and Z_START
data.generateRZInitial_LINE()
# Radial starting locations of fieldlines
data.R_START    = np.linspace( 0.46, 0.97, 46)
# Vertical starting locations of fieldlines
data.Z_START    = np.zeros( (46, 1) )
# Toroidal starting locations of fieldlines (deg)
data.PHI_START = 0.0
# Maximum distance in toroidal direction to follow fieldlines (deg)
data.PHI_END    = 72.0
```

FIELDLINES starting locations may be set here. For the purposes of TORMESH, the lines are only used for the initial seed points for finding the edge. After the edge is found, the domain is

rescaled automatically as per `eff_eiren_outer`, etc.

```
# Poincare periodicity
# Number of toroidal points per-period to output on
# the field line trajectory
# (Poincare plots, choosing phi_start_surf_tracing
#  on one of phis from NPOINC)
data.NPOINC          = int(360./data.periods)
# toroidal points per-period for surf tracing step
data.NPOINC_surf_tracing = data.N_PHI_POINTS
```

The information required for `NPOINC` is automatically input based on previous entries.

```
# FIELDLINES Numerical Integrator
# Fieldline integration method (NAG, RKH68, LSODE)
data.INT_TYPE  = 'LSODE'
data.INT_TYPE_full_poincare = 'LSODE'
# Fieldline following tolerance
data.FOLLOW_TOL = 1e-9
# Virtual casing tolerance (if using plasma field from equilibria)
data.VC_ADAPT_TOL = 1e-7
data.FOLLOW_TOL_full_poincare = 1.0E-9
data.VC_ADAPT_TOL_full_poincare = 1.0E-7
```

Here the numerical integrators of `FIELDLINES` may be set for different execution stages of the `TORMESH` code independently.

```
# Magnetic Island Filtering Method
#
#      ISLAND_FILTERING_METHOD can be
#
#      'ForwardBackward',   'UnivariateSpline',   'SplRep'
#      'ForwardBackwardCubic' (slow!),
#
# 'ForwardBackward'  :
# Generate 1st-order Butterworth digital and analog filter
# http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/
#        scipy.signal.butter.html
# 'SplRep' scipy.interpolate 'splrep'
# http://docs.scipy.org/doc/scipy/reference/tutorial/
#        interpolate.html
# 'UnivariateSpline' : scipy.interpolate "UnivariateSpline"
# http://docs.scipy.org/doc/scipy-0.16.0/reference/generated/
#        scipy.interpolate.UnivariateSpline.html

data.ISLAND_FILTERING_METHOD =  'ForwardBackward'

#-------------------------------------------------------------
#
#    OUTPUTS  (decide what to plot)
#
```

```
    #-----------------------------------------------------------
    data.filename_h5 = 'fieldlines_'+data.name+'.h5'

    return data
```

The final line is necessary for saving information in TORMESH from FIELDLINES intermediate runs.

A.3 A Note on the Usage of FIELDLINES

A representative FIELDLINES input file is reproduced below, with the explanations as given in the previous section for similarly named variables. This input file would be named input.HIDRA.

```
&INDATA
!----- Free Boundary Parameters -----
  LFREEB = T
  MGRID_FILE = 'mgrid_wega_with_VFCoils'
  NZETA = 0072
  EXTCUR(001) = -2000.0
  EXTCUR(002) = 6500.0
  NVACSKIP = 0000
/
&FIELDLINES_INPUT
  NR = 101
  NZ = 101
  NPHI = 36
  RMIN = 0.51
  RMAX = 0.93
  ZMIN = -0.21
  ZMAX = 0.21
  PHIMIN = 0.0
  PHIMAX = 1.25664
  MU = 0.0
  R_START = 7.0285917599e-01      6.8229912817e-01
            6.6173908034e-01      6.4117903252e-01
            6.2061898469e-01      6.0005893687e-01
  Z_START = 7.1434825021e-08      7.1434825018e-08
            7.1434825016e-08      7.1434825013e-08
            7.1434825011e-08      7.1434825008e-08
  PHI_START = 6*0.0
  PHI_END = 6*6283.18530718
  NPOINC     = 72
  INT_TYPE   = 'LSODE'
  FOLLOW_TOL = 1.0e-09
  VC_ADAPT_TOL = 1.0e-07
/
```

A companion `coils.HIDRA` file, which contains the current information, is also reproduced below. The majority of the points have not been copied to facilitate readability.

```
periods 5
begin filament
mirror NIL
0.709544 -0.122258 0.23976 -5.00000E+03
0.62819 -0.00851261 0.221509 -5.00000E+03
0.544803 0.0787375 0.169536 -5.00000E+03
0.476137 0.147603 0.0917522 -5.00000E+03
0.709544 -0.122258 0.23976 0.000e+00 1 Helix
-0.408132 0.985318 1e-12 -1.75000E+04
-0.407086 0.982793 0.043428 -1.75000E+04
-0.403966 0.97526 0.086171 -1.75000E+04
-0.39882 0.962837 0.127555 -1.75000E+04
-0.408132 0.985318 1.0002e-12 0.000e+00 2 toroidal_field
1.5  0 -0.82  0.00000E+00
1.49698  0.0951359 -0.82  0.00000E+00
1.48793  0.189889 -0.82  0.00000E+00
1.47289  0.283877 -0.82  0.00000E+00
1.45192  0.376722 -0.82  0.00000E+00
1.5 -3.67382e-16 -0.82  0.000e+01 3   Vertical_Field_Coil
end
```

To perform a `FIELDLINES` run on the files `input.HIDRA` and `coils.HIDRA`, the program should be called via the example syntax

```
mpirun -np 4 xfieldlines -vmec HIDRA -coil coils.HIDRA -raw -vac
```

If one wishes to use the built-in edge location and domain scaling of `FIELDLINES`, one would append to this command the additional flag `-full`. This `-full` flag of `FIELDLINES` works in the same fashion as `HIDRAmod` does: the domain is tested, rescaled based on connection lengths, and run again to create the output data set.

The reader is referred to the VMECwiki [59] for a thorough description of code operation and output file formats.

A.4 LIM_OCC

The Julia function which occludes the limiting surface is reproduced below, with more detailed comments below each segment. Notably, Julia allows the usage of non-Unicode characters like θ and π as variable names. For the purposes of this reproduction, these two symbols are replaced by the words 'THETA' and 'PI' respectively when in the code blocks.

```
function occlusionTest( LIMHEADER, LPLANE, LPAIR, HEADER, CELL )
  println("Beginning Occlusion Test")
  #Cells which lie beyond min and max THETA are not occluded
  ( minAngle, maxAngle ) = ( 360.0, 0.0 )
  for i in range(1,LIMHEADER.N)
    if LPAIR[i].Angle > maxAngle
      maxAngle = LPAIR[i].Angle
    end
    if LPAIR[i].Angle < minAngle
      minAngle = LPAIR[i].Angle
    end
  end
```

The code performs multiple occlusion passes to reduce the size of the problem. The first is the most important, removing any cells from consideration whose centers lie outside of the toroidal angle range of the limiter. This section sets the min and max angles the limiter surface is defined on.

```
  println(minAngle,"   ", maxAngle)
  # use first node (innermost) as target for node occlusion
  (XAvg, YAvg) = (CELL[1].CenterPol.X, CELL[1].CenterPol.Y)
  # (XAvg, YAvg) = (70.44, 0.0)
```

Two options were used to set the limiting surface: the innermost plasma cell, and a hard-coded magnetic axis. Both options work equally well, although the former is more general as long as that particular cell is not occluded by the limiter.

```
  #first occlusion pass
  println("First Occlusion Pass: Toroidal Angle Neglect")
  CELLTEST = int( zeros( length(CELL), 1 ) + 1 )
  for i in range( 1, length(CELL) )
#     println(CELL[i].CenterPol.THETA)
    for j in range(1,8)
      tempr = sqrt((CELL[i].Vertex[j].X)^2 +
                   (CELL[i].Vertex[j].Y)^2)
      tempTHETA = atan2(CELL[i].Vertex[j].Y,
                        CELL[i].Vertex[j].X)*180.0/pi
      if tempTHETA < 0.0
        tempTHETA += 360.0
      end
      if tempTHETA < minAngle || tempTHETA > maxAngle
#         println("neglecting ",CELL[i].CenterPol.THETA)
        CELLTEST[i] = 0
      end
    end
  end
```

This section neglects cells with centers which lie beyond the limiter surface.

```
( p1, p2, p3, p4 ) = ( Verticies(), Verticies(),
                       Verticies(), Verticies() )
( P1, P2, P3, P4 ) = ( VerticiesPol(), VerticiesPol(),
                       VerticiesPol(), VerticiesPol() )
```

The relevant vectors used in the computation are initialized. A `Verticies` has an `X`, `Y`, and `z` field, and a `VerticiesPol` has an `X`, `Y`, and `THETA` field. In the latter case, `X` and `Y` refer to the constant-θ plane. Note the distinction between capital and lowercase variables.

```
totalIntercepts = 0
println("Main Occlusion Begin")
for i in range(1, length(CELLTEST) )
  if CELLTEST[i] == 1
    intercepts = 0
    for k in range(1,8)
      tempr = sqrt((CELL[i].Vertex[k].X)^2 +
                   (CELL[i].Vertex[k].Y)^2)
      tempTHETA = atan2(CELL[i].Vertex[k].Y,
                        CELL[i].Vertex[k].X)*180.0/pi
      if tempTHETA < 0.0
        tempTHETA += 360.0
      end
      for j in range( 1, length(LPLANE) )
        # occludes unnecessary planes
        if tempTHETA >= LPLANE[j].MinTHETA &&
                        tempTHETA <= LPLANE[j].MaxTHETA
#         println("begin intercept")
          # find interpolation percent from THETA of node
          percent = (tempTHETA - LPLANE[j].MinTHETA)/
                    (LPLANE[j].MaxTHETA - LPLANE[j].MinTHETA)
          # find points: P1 on upper (mid verticies 1+4),
          #      P2 on lower (mid verticies 2+3)
          p1 = LPLANE[j].Vertex[1] +
               (LPLANE[j].Vertex[4] - LPLANE[j].Vertex[1])*percent
          p2 = LPLANE[j].Vertex[2] +
               (LPLANE[j].Vertex[3] - LPLANE[j].Vertex[2])*percent
          # find points: p3 at node center and p4 at
          #      r,z approximate midpoint
          p3 = Verticies(CELL[i].Vertex[k].X,
                         CELL[i].Vertex[k].Y,
                         CELL[i].Vertex[k].Z)
          p4 = Verticies(XAvg * cos(CELL[i].CenterPol.THETA),
                         XAvg * sin(CELL[i].CenterPol.THETA),
                         YAvg)
```

The line `if CELLTEST[i] == 1` permits only non-occluded cells. Then, the second occlusion pass instead omits planes from consideration. Any planes which do not have a component in the

constant-θ plane are omitted.

After this omission, the points which will be used for the interpolation are found. When the limiting surface is defined much more coarsely than the magnetic mesh, a single limiter surface may be required for many cells in the toroidal direction. In this instance, linear interpolation to find the intersection of the limiter surface with the constant-θ plane is required to preserve accuracy.

```
            # collapse points to 2D plane
            P1 = cartToPol(p1)
            P2 = cartToPol(p2)
            P3 = cartToPol(p3)
            P4 = cartToPol(p4)
            # test intersection
            test::Bool = testInterception( P1, P2, P3, P4 )
            if test == true
              intercepts +=1
            end
```

`testInterception()` is the function which implements the interception test expression, and is reproduced after the conclusion of the description of this function.

```
          end
        end
      end
      # final occlusion
      # -- if number of intercepts is even (0 is even),
      #    node is not occluded
      if intercepts%2=0
        totalIntercepts += 1
        CELLTEST[i] = 0
      end
    end
  end
  println("Main Occlusion Fin")
  return CELLTEST
end
```

The end of the function is self-explanatory.

`testInterception()`

```
function testInterception(P1::VerticiesPol, P2::VerticiesPol,
                          P3::VerticiesPol, P4::VerticiesPol)
  denom = (P4.Y-P3.Y)*(P2.X-P1.X) - (P4.X-P3.X)*(P2.Y-P1.Y)
  if denom == 0
    return false
```

```
  end
  u1  = ((P4.X-P3.X)*(P1.Y-P3.Y) - (P4.Y-P3.Y)*(P1.X-P3.X))/denom
  u2 = ((P2.X-P1.X)*(P1.Y-P3.Y) - (P2.Y-P1.Y)*(P1.X-P3.X))/denom
  if (0 < u1) && (u1 <= 1) && (0 < u2) && (u2 <= 1)
    return true
  else
    return false
  end
end
```

This solves for the parameterized distance along line segment that the intersection lies as per [60]. Values less than zero or greater than unity correspond to lines which would intersect but segments which do not, and so are not counted.

Appendix B

A Description of the Implementation of a Bohm-Like Diffusivity in EMC3-EIRENE

The Bohm-like diffusivity has been added to EMC3-EIRENE in two source files. In each case, the impact on the functioning of the code has been eliminated except for memory usage increases.

B.1 energy.f

A single block is added with the rest of the file writing in the form below to calculate and write out the Bohm diffusivity.

```
      DO 10 IC=1,NC_PL
c       write(*,*)TEMPDIFF(IC)
         DIFFUS(IC) = R_B*(TEMP0(IC,0) / B_CEL_AVG(IC)
     .*                    0.0625 * 100. * 100.)
     .+                (R_B-1)*TEMPDIFF(IC)
c       write(*,*)DIFFUS(IC)
         CHIS(IC)    = DIFFUS(IC) * 3.
10    CONTINUE
```

B.2 CELL_GEO.f

To calculate the quantity `B_CEL_AVG`, it is necessary to compute the average magnetic field over the cell. In lieu of a module implementation, the most opportune location to calculate a value for each physical cell is in the cell array calculation, where the areas must be evaluated anyways (to check the mesh quality).

In the inner-most loop in subroutine `SETUP_CELL_GEO()`, one line calculates the total magnetic field strength per plasma cell:

```
        BF_TOTAL = BF_TOTAL
     .+            0.25*(BFSTREN(I1) + BFSTREN(I2)
     .+            BFSTREN(I3) + BFSTREN(I4))
```

At the end of the loop, another line saves the magnetic field information to the required arrays:

```
        B_CEL_AVG(IC) = BF_TOTAL/NUM_CELLS
```

Appendix C

A Description of the EMC3-EIRENE Formatted Input Files

Aside from the mesh files, nominally GRID_3D_DATA and CELL_GEO, EMC3 requires a number of plasma files and boundary condition files for which the documentation is lacking. Such is the subject of this chapter. The singular EIRENE input file contains hundreds of interdependent variables, and the prospective user must be directed to the EIRENE user manual [29]. As a general note, the input files read by EMC3 (but not by EIRENE) are not fixed-width format, but use a custom function `scrape()` to parse the input files. In this function, an asterisk (*) in the first column denotes an entire commented line, and an exclamation mark (!) denotes a partial comment at the end of a line. Otherwise, the files are not whitespace sensitive.

C.1 input.ctr

This file controls the iteration structure of the code. The energy transport, continuity, impurity calculation, neutral transport (EIRENE), and various post-processing utilities may be called here in a modular form.

In the version of EMC3-EIRENE used for this work, the iteration between EMC3 and EIRENE is bugged, and the neutral transport module may not be called more than once per EMC3-EIRENE iteration. As a result, the code must be run for one EMC3-EIRENE iteration, and then recalled from the command line or from a script for the desired number of iterations. This has since been fixed in later versions of the code.

The Main EMC3-EIRENE Iteration

```
* Program control
  1      3410                    ! Global iteration, RANSET
  1        1                     ! iterations, calculations included
NEUTRAL
0    10000     1      11         ! IND, PARTICLES, CPU_LIM
250.   0.5     0.     0.         ! Rela. fact, write out
END
* Program control
  1      3410                    ! Global iteration, RANSET
  10     2                       ! iterations, calculations included
STREAMING
0    100000     0       0        ! IND,PARTICLES,para. viscus,
250.   0.5     0.     1.E-6
ENERGY
0    100000     1       0        ! IND, PARTICLES, CPU_LIM -1:kie=0
```

```
250.   0.5     4.E-7 4.0E-8   ! Rela. fact , write out
END
FIN
```

An EMC3-EIRENE Iteration is presented here. Of these values, which may be referenced in the source code as required (many of which are defunct), the important quantities for the STREAMING and ENERGY modules which calculate density and temperature respectively are the second number in the first line, here listed as 100000, and the last number in the second line. These are values for the number of Monte Carlo histories to store, and the timestep of the Monte Carlo scheme.

It is important to note the dual nesting of loops, the innermost of which is terminated by the keyword END and the outermost by FIN. The program ends when the latter is called, but if for example it is desired to iterate between density and temperature profiles to reach a stable solution to then run neutrals on, it is possible to set up sub-iterations by only terminating the inner loop as illustrated here. The presented code block computes the neutral profile, then iterates between density and temperature profiles ten times before exiting.

It is also possible to calculate the impurity concentration by using the IMPURITY module, and the connection length, the limiter fluxes, and other potentially user-defined functions with the POST_PROCESSING module, using the first value in the first line to determine the desired function.

C.2 input.geo

The input file input.geo contains the information for EMC3 on how to read the mesh file GRID_3D_DATA and on how to assign boundary conditions for the various surfaces. It also contains an option to adjust the physical cell to plasma cell map.

Zone Number and Zonal Mesh Dimensions

```
2
37  401  19
37  401  19
```

The number of zones, followed by a line with the number of radial points, poloidal points, and toroidal points in each zonal mesh, one-indexed. In the case of HIDRA, the grids for both currently used zones are the same size. This listing is per-vertex.

Non-Default Surfaces

```
* radial
0
* poloidal
* Type of surface=1:  periodic
*                 2:  up/down symmetric
*                 3:  Mapping
4
0 0 1
0 35 0 17
400 0 1
0 35 0 17
0 1 1
0 35 0 17
400 1 1
0 35 0 17
* toroidal
4
0 0 2
0 35 0 399
18 0 3
0 35 0 399
0 1 3
0 35 0 399
18 1 2
0 35 0 399
```

Non-default surfaces are the traditional boundary conditions of periodicity and up-down symmetry, as well as the mapping boundary condition enumerated in chapters 2 and 3. These boundary conditions are used to join adjacent surfaces together where appropriate. It is important to note that while the original listing of the mesh sizes in the first section is one-indexed, all following mesh size references are zero-indexed. Each boundary condition has a 2-line form:

```
0 0 1
0 35 0 17
```

```
400 0 1
0 35 0 17
```

For the first line, the first number denotes the constant-radial, -poloidal, or -toroidal surface on which the boundary condition is defined, the second the zone on which it is defined, and the third the type of boundary condition as mentioned in the comments. For the second line, the numbers form two pairs covering the whole domain in the other two directions. As an example, the two

separate code blocks denote the constant-poloidal seam in the toroidal cross-section which must be joined by a periodicity boundary condition for continuity. The poloidal surfaces to be joined are obviously the first and last, in the first zone, and over the entire radial and toroidal range. As a further complication, the ranges in the second line are per-cell instead of per-vertex, resulting in a total subtraction of two indicies from the original mesh values.

Non-Transparent Boundary Conditions

```
* radial
4
4 0 1
0 399 0 17
34 0 -1
0 399 0 17
4 1 1
0 399 0 17
34 1 -1
0 399 0 17
* POLOIDAL
0
* TOROIDAL
0
```

The non-transparent boundary conditions are used to restrict a region of the mesh to EMC3 only, instead of EMC3-EIRENE. Formatting is otherwise identical to the previous section, with boundary condition type assumed, and the values of -1 and 1 used to denote the surface normal. In this example it may be seen that the first four physical cells and the last physical cell radially outwards from the center are excluded from EMC3. Even if the cells are excluded here, they must be included in the boundary condition ranges in the previous section.

Plate Surface

```
* radial
-1
* POLOIDAL
-1
* TOROIDAL
-1
```

The plate surface may either be described in terms of the underlying magnetic configuration, restricting it to field-aligned geometry, or may be described by a triangle file. Because the triangle

file is perfectly generic, it was used exclusively (and displayed here).

Physical to Plasma Cell Map Options and Mesh-Checking

```
1   3
T
4
```

In the first line, the first value may be positive, negative, or 0. In the negative case, the user is expected to provide the file `CELL_GEO`; in the positive case, the code calculates its own `CELL_GEO` and saves it. The second parameter is used in the positive case to determine how many toroidal divisions there are in the default plasma cell map along each flux tube. The value of 3 used here implies that in each zone of 18 toroidal cell divisions, each plasma cell is 6 cells in the toroidal direction. Finer discretizations allow for more toroidal variance in the plasma profiles at the cost of a heightened computation time and storage requirements.

The final two options are a T/F flag to check the quality of the mesh as according to chapter 2, and a parameter which may be used by the code if required. It may be safely ignored.

C.3 `input.par`

The input file `input.par` contains plasma species information, radial boundary conditions for particles, energy, and momentum for each plasma species, as well as domain-wide diffusion coefficients and sources. The file is listed herein in sequential blocks, so that the concatenation of all in the presented order would give a functional input file.

Ion and impurity species information

```
2
H       1       1.              !CHARGE MASS
Fe      26      55.845          !CHARGE MASS
```

In this example, there are two species in the plasma: the main species, in this instance hydrogen, must be first, with each impurity atom listed on its own line. The total number of species is given alone on the first line.

Transport Parameters

```
-667                                  !Cross-B  for ION 1 (D cm^2/sec)
-667                                  !Cross-B  for ION 2 (D cm^2/sec)
-668  -668                          !Cross-B shi_e  shi_i(\chi_e,i =3*D)
```

The first and second lines contain the information for the perpendicular diffusion coefficient D_i for each species i, and the final line contains the electron and ion thermal diffusivity $\chi_{i,e}$. Each species receives its own diffusion coefficient, but all species share the same χ_i. The values may be given either as scalars over the entire domain in exponential form, or as a file name as demonstrated here (with a minus sign directly proceeding).

Particle Transport Boundary Conditions

```
41        3.E12                       !0=recyc.FLUX(A),1=ne_up
4                                      !surface number
1    1     1        0.                !R,P,T_SF NR.(2.2), ID_SF, DATA
1    2    -2      3.E+0               !R,P,T_SF NR.(2.2), ID_SF, DATA
1    3     1        0.                !R,P,T_SF NR.(2.2), ID_SF, DATA
1    4    -2      3.E+0               !R,P,T_SF NR.(2.2), ID_SF, DATA
0       3.E08                       !0=recyc.FLUX(A),1=ne_up
4                                      !surface number
1    1     1        0.                !R,P,T_SF NR.(2.2), ID_SF, DATA
1    2    -2      3.E+0               !R,P,T_SF NR.(2.2), ID_SF, DATA
1    3     1        0.                !R,P,T_SF NR.(2.2), ID_SF, DATA
1    4    -2      3.E+0               !R,P,T_SF NR.(2.2), ID_SF, DATA
```

Radial boundary conditions for transport are given in two blocks, the first for the main plasma and the second for the impurities. The second block is repeated for each species. For each block the pair of boundary conditions must be repeated for each zone the first for the inner radial surface, and the second for the outer radial surface. The surfaces are ordered per-zone in the direction they are defined in in the geometry input file `input.geo`. In this instance, the plasma is composed of two species and two zones.

Energy Transport Boundary Conditions

```
4                                      !surface number
1    1     1      1.30E+3  0.00E+0    !R,P,T_SF NR.(2.2), ID_SF, DATA
1    2    -2      3.E+0    3.E+0      !R,P,T_SF NR.(2.2), ID_SF, DATA
1    3     1      1.30E+3  0.00E+0    !R,P,T_SF NR.(2.2), ID_SF, DATA
1    4    -2      3.E+0    3.E+0      !R,P,T_SF NR.(2.2), ID_SF, DATA
```

Similar to the particle transport, the energy transport has input powers and decay lengths for electrons and ions respectively in watts and centimeters. The input power is separated between the two depending on the amount of thermalization in the core; in the case of devices like HIDRA with lower densities, the lack of thermalization implies that that majority of power should be going going into the electrons, which would then leave the core at a much higher energy than the ions. The energy of the device must be split by both the number of periods and the number of toroidal zones. In the provided example, the device is experiencing a 26 kW discharge with a lossless fully-absorbing core. To simulate radiative losses and imperfect RF absorption these power inputs must be lowered (compared to the input power of the RF system) by some percentage which either comes from experimental data or from a full-wave and core solver.

Momentum Transport Boundary Conditions

```
4                                          !surface number
1    1    1     0.    !absorbing   !R,P,T_SF NR.(2.2), ID_SF, DATA
1    2   -2     2.E4  !absorbing   !R,P,T_SF NR.(2.2), ID_SF, DATA
1    3    1     0.    !absorbing   !R,P,T_SF NR.(2.2), ID_SF, DATA
1    4   -2     2.E4  !absorbing   !R,P,T_SF NR.(2.2), ID_SF, DATA
```

The block for momentum transport is identical to that of the particle transport, with the exception of the edge-core boundary flux definition in the first line.

Initial Conditions

```
30    0    0          !30          !Te,Ti_00(eV) (input file unit
31    0               !31          !ni_00,n_M(ire.),De_const
33                    !33          !Mach0 mach1
```

The initial conditions for an EMC3-EIRENE simulation must be set on the first iteration, in order to provide a plasma background for EIRENE. After the first iteration, the initial conditions must be commented out and replaced with input file numbers, as seen here. This example is for the file numbers; an initial iteration file (perhaps called `input.par0`) would have the first column removed and the zeros replaced with appropriate guesses for the initial conditions. The values of 0 are necessary to preserve the format of the input files.

Volumetric Sources

```
40 !41    !V_s_part_ion(main ion and impurities
42 0 0 !44    !V_s_ener_e  (from n0,Z
43 0 0 !45    !V_s_ener_i  (from n0,Z
46 0 0 !47    !M source (transport and CX with n0
```

Volumetric sources should be left alone, except in the case where EMC3 is being run without EIRENE. Combined with the previous block, these file number definitions control the iteration passing from EMC3 to EIRENE. The values of 0 are necessary to preserve the format of the input files.

C.4 `input.IMP`

One of these files must be generated per impurity species which contains all information relevant to the sources and sinks of impurities.

Impurity Model

```
1
```

The first entry toggles between impurity models. In the version of the code used for the present work, only the trace impurity fluid model is implemented.

Atomic Data

```
STRAHL
*ADAS          ! Call ADAS database
*3             ! 3 total processes
*/home/USER/data/ADAS/acd96_li.dat
*/home/USER/data/ADAS/scd96_li.dat
*/home/USER/data/ADAS/plt96_li.dat
```

The user is offered a choice between two atomic models, STRAHL and ADAS. Which database is used depends most prominently on the impurity atom and whether its information exists in either of the databases in a usable form. In the case of STRAHL, the entire database must be linked to EMC3 in an expected default Fortran file name. For ADAS, the data files must be linked in this file for each process individually.

Intrinsic Impurity Neutral Sources

```
0.0     0.0001   0.3   ! physical, chemical, self sputtering
0.13   1     1          ! E0, Alpha, Beta
```

The first line lists the intrinsic neutral sources as physical sputtering, chemical sputtering, and self-sputtering. Physical sputtering has not been implemented in the present version, and self-sputtering is a computational tool used to ensure that sputtered impurities are tracked through multiple re-ionizations. Otherwise the domain would be inadequately covered by impurity species, especially in larger devices. In effect, a self-sputtering coefficient of 1 would cause the impurity module of the code to run forever, and a self-sputtering coefficient of 0.3 to 0.5 is adequate in most cases for domain coverage.

Chemical sputtering is implemented as a return flux of incident particles. Because a number must be put in this input file instead of being calculated in a self-consistent manner in the code, the material model can be seen to be sorely lacking.

The second line covers the sputtered particle distribution with the energy given as E_0, according to

$$F(\theta) = \sin(\theta)^{\alpha}\cos(\theta)^{\beta}. \tag{C.1}$$

External Sources

```
0        10     ! total points,beta
* E0(ev), Strength(A), R(cm),Z(cm),PHI(Deg), VR, VZ, VPHI(relative)
*5.        1.    75.   0.  7.e-3       0.     0.    0.
```

External sources may be given as according to this block. This option may be used to simulate gas puffs or other transient plasma pollution events.

Core Interface Boundary Conditions for Impurities

```
*    0: local zero-flux for all Z
*    1: surface-average zero-flux for all Z
*   >1: coupled with 1D core model (number of radii)
*    R_eff      Ne         TE         D        V
1
```

```
      5.     1.E11        50.        1.E5      0.
      35.    1.E11        50.        1.E5      0.
```

This block defines boundary conditions for the impurities at the inner and outer radial boundaries. A rough approximation to the plasma conditions is sufficient, as in this case.

C.5 input.n0g

The file `input.n0g` contains information on the domain coupling between EMC3 and EIRENE.

EIRENE Radial Boundary Conditions (Non-Transparent)

```
* radial
4
0 0 -3
0 399 0 17
36 0 -2
0 399 0 17
0 1 -3
0 399 0 17
36 1 -2
0 399 0 17
* POLOIDAL
0
* TOROIDAL
0
```

Analogous to the EMC3 boundary conditions, EIRENE must have its own radial boundary conditions set. Unlike the EMC3 radial boundary conditions which do not necessarily extend to the walls and inward to the inner-most radial surface, the EIRENE boundary conditions must. Instead of a 1 or -1, the surface type corresponds to the material index in the EIRENE input file.

Extra Plasma Information for EIRENE

```
10      70              ! NC_DEF, >0 Unit number for IDCELL
1  ! zone number
*   ZONE  R1     R2     DR     P1     P2     DP     T1     T2     DT
*         ne        Te        Ti        M
2     1
      0     0     1     1     0   400   400     0    18    18
      9.0E+12        20.       0.2       0.000
2     1
      0     1     2     1     0   400   400     0    18    18
```

```
     9.0E+12        20.      0.2      0.000
2    1
     0     2     3     1     0   400   400    0     18    18
     8.0E+12        20.      0.2      0.000
2    1
     0     3     4     1     0   400   400    0     18    18
     5.9E+12        20.      0.2      0.000
2    0
     0    0     36    36     0   400    1     0     18    1
     1.0E+07        0.1      0.1      0.000
```

EMC3 only operates on a sub-domain of the mesh, but EIRENE must operate over the entire mesh. EIRENE must have a plasma background set in this input file in the regions EMC3 is not active in. The core region of tokamaks and stellarators is not likely to be vary significant in either the poloidal or toroidal directions, and so is set as a range in that region. In this example, the principle variation is in the radial direction, so the cells are defined using a DR of 1 over the entire toroidal and poloidal ranges. The last point has a different definition type (0 instead of 1) and acts as a catch-all over the domain., including the edge region. If it is not set properly, the code will encounter a segmentation fault.

One of these blocks must be repeated for each zone, excluding the top line with `NC_DEF` in which the total number of definition blocks across all zones must equal `NC_DEF`. Unfortunately for partially-ionized machines, the number of neutrals reaching the core is not small, and so the solution to the neutral profiles and hence the power loss in the plasma may be sensitive to these values. As they are set in the input files and not calculated in the code, it is more difficult to use EMC3-EIRENE in a predictive sense without first tackling this issue.

Additional Surface Geometry Information for EIRENE

```
* N0S  NS_PLACE   NSSIDE
0          0          1
*** 4 Additional surface
ADD_SF_LIM
```

This block covers options related to the geometry of limiting surfaces or other non-field-aligned structures.

The first line has three variables. `N0S` allows for the reading of the sources for EIRENE from an input file if set to a negative value, or from EMC3 if set to 0. `NS_PLACE` informs the code as

to the location of the definition of the surface, and if it is 0, then NSSIDE determines which side of the surface the particles in EIRENE are traced from.

The second line, ADD_SF_LIM, is a file pointing to the file locations of the triangle files. In ADD_SF_LIM, there is another variable to change the surface orientation . Since the triangle file also by definition determines the surface normal, there are three negation opportunities to do so.

Note that the first line of ADD_SF_LIM must contain a line commented with an asterisk, or the code will encounter a segmentation fault.

Additional Neutral Diagnostic Information

For the purpose of this document, none of the remaining options were used. They may be safely turned off if desired, or used if necessary. Options exist to pull flux and energy spectra from a given radial surface, or from an additional surface.

Appendix D

A Description of the File Formats for EMC3-EIRENE

D.1 GRID_3D_DATA

GRID_3D_DATA contains the mesh information for EMC3-EIRENE. It is organized into zones which are filled by toroidal planes, which have a continuous list of (r, z) points which belong to that toroidal plane. Point information is written in the poloidal direction first, then the radial direction, then for each toroidal plane. Values for r and z are stored in separate successive blocks. When the surface is filled, the next surface starts with a new line and the toroidal angle for that surface. Each zone has a three-integer header with the number of radial points, poloidal points, and toroidal planes respectively. These numbers are given in terms of the points, so the number of physical cells is one less in each direction.

An example C routine to read one zone can be found below.

```
// Read GRID_3D_DATA
        int gridHeader[3];
        printf("OPENING GRID_3D_DATA\n");
        fid = fopen(argv[1],"r");
        fscanf(fid,"%i %i %i", &gridHeader[0],
           &gridHeader[1], &gridHeader[2]);

        printf("        ALLOCATING GRID ARRAY\n");
        grid = malloc( sizeof(*grid) * gridHeader[2] );
        for(i=0; i<gridHeader[2]; i++){
                grid[i].angle = 0.0;
                grid[i].x     = malloc( sizeof(double) *
                        gridHeader[0] * gridHeader[1] );
                grid[i].y     = malloc( sizeof(double) *
                        gridHeader[0] * gridHeader[1] );
                grid[i].z     = malloc( sizeof(double) *
                        gridHeader[0] * gridHeader[1] );
        }
        printf("        FILLING GRID ARRAY\n");
        for(i=0; i<gridHeader[2]; i++){
                fscanf(fid,"%lf", &grid[i].angle);
                for(j=0; j<gridHeader[0]*gridHeader[1]; j++){
                        fscanf(fid,"%lf", &tempDouble);
                        grid[i].x[j] = tempDouble *
                           cos(grid[i].angle * pi / 180.0);
                        grid[i].y[j] = tempDouble *
                           sin(grid[i].angle * pi / 180.0);
                }
                for(j=0; j<gridHeader[0]*gridHeader[1]; j++){
                        fscanf(fid,"%lf",&tempDouble);
                        grid[i].z[j] = tempDouble;
                }
        }
```

```
        fclose(fid);
        printf("        GRID_3D_DATA_READ\n");
```

D.2 CELL_GEO

CELL˙GEO holds the physical to plasma cell map. It has a three-integer header giving the total number of physical cells, the total number of plasma cells without EIRENE, and the total number of plasma cells with EIRENE. After the header the file is followed by an unbroken string of integers, each one corresponding to a physical cell in the order they are listed in `GRID_3D_DATA`. Every zone is listed sequentially with no delimiters.

A C routine to read `CELL_GEO` is thus trivial to write.

```
// Read CELL_GEO
        printf("OPENING CELL_GEO\n");
        fid = fopen(argv[2],"r");
        fscanf(fid,"%i %i %i", &geoHeader[0],
           &geoHeader[1], &geoHeader[2]);
        printf("        ALLOCATING CELL_ARRAY\n");
        geo = malloc(sizeof(int) * geoHeader[0]);
        printf("        FILLING CELL_ARRAY\n");
        for(i=0; i<geoHeader[0]; i++){
                fscanf(fid,"%i", &geo[i]);
        }
        fclose(fid);
        printf("        CELL_GEO_READ\n");
```

D.3 Plasma and Neutral Profile Output Files

Plasma and Neutral profile output files are also trivial to read; the only difference between the two is the usage of the total number of plasma cells without EIRENE for the former and with EIRENE for the latter.

There are two exceptions to this rule:

1. `TE_TI` holds both electron temperature and ion temperature in that order, with no delimiters. It is similar to two concatenated output files of other plasma profiles.

2. `DENSITY` holds the main species ion, as well as each ionized impurity species in order. An

index separates the two, indicating whether the following block is the main ion density (1) or an impurity density (2 to N+1, where N is the atomic number of the impurity.

Some examples of file reading in C are listed below. Note that `geoHeader` is as read in the previous section on `CELL_GEO`.

```
\\      Read a neutral atomic density profile
        DENSITY_A = malloc( sizeof(double) *
                        geoHeader[2] );
        for(i=0; i<geoHeader[2]; i++){
                fscanf(fid,"%lf", &DENSITY_A[i]);
        }
```

```
\\ Read Te and Ti profiles
        TE = malloc( sizeof(double) * geoHeader[1] );
        TI = malloc( sizeof(double) * geoHeader[1] );
        for(i=0; i<geoHeader[1]; i++){
                fscanf(fid,"%lf", &TE[i]);
        }
        for(i=0; i<geoHeader[1]; i++){
                fscanf(fid,"%lf", &TI[i]);
//              printf("%f\n", TI[i]);
        }
```

D.4 Limiter Geometry Files

The limiter surface is defined to EMC3-EIRENE by two files. One file must contain the limiter geometry in stripes of point pairs (r, z) along a constant-θ plane, with a header giving the number of blocks, the number of points per block, and the periodicity of the limiting surface. Two more optional parameters are available for shifting the entire limiter by some r or z.

As EMC3-EIRENE creates triangles with this data, there is an expectation of a minimum number of points (one should not drop below 3 toroidal planes or 3-4 points per plane). When EMC3 creates particle and heat deposition maps, the surface may be further subdivided, so mesh size is not a major issue. In fact, large meshes will make the limiter deposition profile calculation much more time-intensive.

```
* TEST
14   7   5   0.000000   0.000000
-5.316195
 85.948193  5.000000
 85.948193  5.000000
```

```
 85.948193  2.500000
 85.948193  0.000000
 85.948193  -2.500000
 85.948193  -5.000000
 85.948193  -5.000000
-5.316185
 85.948193  5.000000
 80.948193  5.000000
 80.948193  2.500000
 80.948193  0.000000
 80.948193  -2.500000
 80.948193  -5.000000
 85.948193  -5.000000
-4.353732
 85.833254  5.000000
 80.833254  5.000000
 80.833254  2.500000
 80.833254  0.000000
 80.833254  -2.500000
 80.833254  -5.000000
 85.833254  -5.000000
 ...
```

The other file required contains the listing of cells which are occluded by the plasma, in any order, in the following format per cell:

```
Z        R        P        2        T        T
```

Where `Z` in the zone number, `R`, `P`, and `T` are the radial, poloidal, and toroidal indicies, and 2 indicates the format. All indicies are zero-indexed. The file is able to handle other formats, but none give the per-cell fidelity and generality of this method.

D.5 Limiter Flux and Temperature Profiles

Limiter surface profiles, for "Tilmann's Wish", are formatted with successive r, z, and θ array blocks, which are then followed by successive surface profile blocks. An initial header contains the limiter index, the limiter name, integrated particle and power fluxes, and the numbers of points in each direction which comprise the surface. An example on how to read the surface profile files are provided below in C. Below that, an example on writing a surface profile to Gmsh mesh formats is presented.

```
\\ Read TARGET_PROFILES
                    filePos = fscanf(fid,"%i", &tempInt);
```

```
                filePos = fscanf(fid," %s ", tempStr);
                filePos = fscanf(fid," %lf %lf ",
                        &integParticle, &integPower);
                filePos = fscanf(fid," %s %s %s ",
                        trash1, trash2, trash3);
                filePos = fscanf(fid," %i %i ", &nPol, &nTor);
                rArray = malloc( sizeof(double) * (nPol+1)*(nTor+1) );
                zArray = malloc( sizeof(double) * (nPol+1)*(nTor+1) );
                thetaArray = malloc( sizeof(double)*(nPol+1)*(nTor+1));
                DEPO_P = malloc( sizeof(double) * (nPol)*(nTor) );
                DEPO_E = malloc( sizeof(double) * (nPol)*(nTor) );
                DEPO_DE = malloc( sizeof(double) * (nPol)*(nTor) );
                DEPO_TE = malloc( sizeof(double) * (nPol)*(nTor) );
                DEPO_TI = malloc( sizeof(double) * (nPol)*(nTor) );
                for(i=0; i<(nPol+1)*(nTor+1);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        rArray[i] = tempDouble;
                }
                for(i=0; i<(nPol+1)*(nTor+1);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        zArray[i] = tempDouble;
                }
                for(i=0; i<(nPol+1)*(nTor+1);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        thetaArray[i] = tempDouble;
                }
                for(i=0; i<(nPol*nTor);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        DEPO_P[i] = tempDouble;
                }
                for(i=0; i<(nPol*nTor);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        DEPO_E[i] = tempDouble;
                }
                for(i=0; i<(nPol*nTor);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        DEPO_DE[i] = tempDouble;
                }
                for(i=0; i<(nPol*nTor);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        DEPO_TE[i] = tempDouble;
                }
                for(i=0; i<(nPol*nTor);i++){
                        filePos = fscanf(fid,"%lf", &tempDouble);
                        DEPO_TI[i] = tempDouble;
                }
```

```
\\ Print Heat Flux Deposition Limiter map
        fprintf(fid,"View \"Heat Flux [MW m^-2]\" {");
        index = 0;
        for(j=0; j<nTor; j++){
```

```
                for(i=0; i<nPol; i++){
                        I1 = i+j*(nPol+1);
                        I2 = I1+1;
                        I3 = I2+(nPol+1);
                        I4 = I1+(nPol+1);
                        fprintf(fid, "SQ(");
                        fprintf(fid, "%f,%f,%f,",
                    rArray[I1]*cos(thetaArray[I1]),
                    rArray[I1]*sin(thetaArray[I1]),
                    zArray[I1]);
                        fprintf(fid, "%f,%f,%f,",
                    rArray[I2]*cos(thetaArray[I2]),
                    rArray[I2]*sin(thetaArray[I2]),
                    zArray[I2]);
                        fprintf(fid, "%f,%f,%f,",
                    rArray[I3]*cos(thetaArray[I3]),
                    rArray[I3]*sin(thetaArray[I3]),
                    zArray[I3]);
                        fprintf(fid, "%f,%f,%f){",
                    rArray[I4]*cos(thetaArray[I4]),
                    rArray[I4]*sin(thetaArray[I4]),
                    zArray[I4]);
                        fprintf(fid, "%f,%f,%f,%f};\n",
                    DEPO_E[index]/100.0,
                    DEPO_E[index]/100.0,
                    DEPO_E[index]/100.0,
                    DEPO_E[index]/100.0);
                        index++;
                }
        }
        fprintf(fid,"};");
```

References

[1] K. Ikeda, "Iter on the road to fusion energy," *Nuclear Fusion*, vol. 50, no. 1, p. 014002, 2010.

[2] F. Wagner, G. Becker, K. Behringer, D. Campbell, A. Eberhagen, W. Engelhardt, G. Fussmann, O. Gehre, J. Gernhardt, G. v. Gierke, G. Haas, M. Huang, F. Karger, M. Keilhacker, O. Klüber, M. Kornherr, K. Lackner, G. Lisitano, G. G. Lister, H. M. Mayer, D. Meisel, E. R. Müller, H. Murmann, H. Niedermeyer, W. Poschenrieder, H. Rapp, H. Röhr, F. Schneider, G. Siller, E. Speth, A. Stäbler, K. H. Steuer, G. Venus, O. Vollmer, and Z. Yü, "Regime of improved confinement and high beta in neutral-beam-heated divertor discharges of the asdex tokamak," *Phys. Rev. Lett.*, vol. 49, pp. 1408–1412, Nov 1982.

[3] A. Team, "The h-mode of asdex," *Nuclear Fusion*, vol. 29, no. 11, p. 1959, 1989.

[4] J. Stober, P. Lomas, G. Saibene, Y. Andrew, P. Belo, G. Conway, A. Herrmann, L. Horton, M. Kempenaars, H.-R. Koslowski, A. Loarte, G. Maddison, M. Maraschek, D. McDonald, A. Meigs, P. Monier-Garbet, D. Mossessian, M. Nave, N. Oyama, V. Parail, C. Perez, F. Rimini, R. Sartori, A. Sips, P. Thomas, C. to the EFDA-JET workprogramme, and the ASDEX Upgrade Team, "Small elm regimes with good confinement on jet and comparison to those on asdex upgrade, alcator c-mod and jt-60u," *Nuclear Fusion*, vol. 45, no. 11, p. 1213, 2005.

[5] H. Zohm, "The physics of edge localized modes (elms) and their role in power and particle exhaust," *Plasma Physics and Controlled Fusion*, vol. 38, no. 8, p. 1213, 1996.

[6] P. Lang, G. Conway, T. Eich, L. Fattorini, O. Gruber, S. Gnter, L. Horton, S. Kalvin, A. Kallenbach, M. Kaufmann, G. Kocsis, A. Lorenz, M. Manso, M. Maraschek, V. Mertens, J. Neuhauser, I. Nunes, W. Schneider, W. Suttrop, H. Urano, and the ASDEX Upgrade Team, "Elm pace making and mitigation by pellet injection in asdex upgrade," *Nuclear Fusion*, vol. 44, no. 5, p. 665, 2004.

[7] T. Evans, M. Fenstermacher, R. Moyer, T. Osborne, J. Watkins, P. Gohil, I. Joseph, M. Schaffer, L. Baylor, M. Bcoulet, J. Boedo, K. Burrell, J. deGrassie, K. Finken, T. Jernigan, M. Jakubowski, C. Lasnier, M. Lehnen, A. Leonard, J. Lonnroth, E. Nardon, V. Parail, O. Schmitz, B. Unterberg, and W. West, "Rmp elm suppression in diii-d plasmas with iter similar shapes and collisionalities," *Nuclear Fusion*, vol. 48, no. 2, p. 024002, 2008.

[8] C. Gormezano and D. Moreau, "Lower hybrid wave coupling in the wega tokamak," *Plasma Physics and Controlled Fusion*, vol. 26, no. 3, p. 553, 1984.

[9] P. Blanc, W. Hess, G. Ichtchenko, P. Lallia, and T. Nguyen, "Rf heating experiments in the wega tokamak," *urnal de Physique Colloques*, vol. 38, pp. C3–165–C3–169, 1977.

[10] J. Rapp, T. M. Biewer, T. Bigelow, J. B. O. Caughman, R. Duckworth, D. Giuliano, R. H. Goulding, D. L. Hillis, R. Howard, R. J. Ellis, T. Lessard, J. D. Lore, A. Lumsdaine, E. Martin, W. D. McGinnis, S. J. Meitner, L. W. Owen, H. Ray, G. Shaw, and V. Varma, "The material plasma exposure experiment mpex: Pre-design, development and testing of source concept," in *2015 IEEE 26th Symposium on Fusion Engineering (SOFE)*, pp. 1–8, May 2015.

[11] D. M. Goebel, G. Campbell, and R. W. Conn, "Plasma surface interaction experimental facility (PISCES) for materials and edge physics studies," *Journal of Nuclear Materials*, vol. 121, pp. 277–282, May 1984.

[12] PISCES-Team, *PISCES program plasma-surface interactions research: Summary of research, 1988–1989*. May 1989.

[13] S. Jung, D. Andruczyk, and D. N. Ruzic, "Laboratory investigation of vapor shielding for lithium-coated molybdenum in devex," in *2011 IEEE/NPSS 24th Symposium on Fusion Engineering*, pp. 1–5, June 2011.

[14] M. Tokitani, S. Masuzaki, H. Kasahara, Y. Yoshimura, R. Sakamoto, N. Yoshida, Y. Ueda, T. Mutoh, and S. Nagata, "Initial growth phase of w-fuzz formation in ultra-long pulse helium discharge in {LHD}," *Nuclear Materials and Energy*, pp. –, 2016.

[15] M. Baldwin and R. Doerner, "Helium induced nanoscopic morphology on tungsten under fusion relevant plasma conditions," *Nuclear Fusion*, vol. 48, no. 3, p. 035001, 2008.

[16] M. Miyamoto, D. Nishijima, Y. Ueda, R. Doerner, H. Kurishita, M. Baldwin, S. Morito, K. Ono, and J. Hanna, "Observations of suppressed retention and blistering for tungsten exposed to deuteriumhelium mixture plasmas," *Nuclear Fusion*, vol. 49, no. 6, p. 065035, 2009.

[17] G. M. Wright, D. Brunner, M. J. Baldwin, K. Bystrov, R. P. Doerner, B. Labombard, B. Lipschultz, G. De Temmerman, J. Terry, D. G. Whyte, and K. Woller, "Comparison of tungsten nano-tendrils grown in alcator c-mod and linear plasma devices," *Journal of Nuclear Materials*, vol. 438, Supplement, pp. S84 – S89, 2013. ¡ce:title id="tt005"¿Proceedings of the 20th International Conference on Plasma-Surface Interactions in Controlled Fusion Devices¡/ce:title¿.

[18] S. TAKAMURA, N. OHNO, D. NISHIJIMA, and S. KAJITA, "Formation of nanostructured tungsten with arborescent shape due to helium plasma irradiation," *Plasma and Fusion Research*, vol. 1, pp. 051–051, 2006.

[19] P. Fiflis, D. Curreli, and D. Ruzic, "Direct time-resolved observation of tungsten nanostructured growth due to helium plasma exposure," *Nuclear Fusion*, vol. 55, no. 3, p. 033020, 2015.

[20] H. Zerriffi, "Tritium: The envinronmental, health, budgetary, and strategic effects of the department of energy's decision to produce tritium," tech. rep., Institute for Energy and Environmental Research, January 1996.

[21] S. Mumm, "Npre gains major plasma/fusion facility from german institute," 2014 (accessed April 18th, 2017).

[22] E. Chlechowitz, "Plasma current and pressure measurements at the wega stellarator," Master's thesis, 2010.

[23] M. Preynas, H. P. Laqua, M. Otte, T. Stange, T.Wauters, and D. Aβmus, "Study of plasma start-up initiated by second harmonic electron cyclotron resonance heating on wega experiment," in *AIP Conference Proceedings*, vol. 1580, pp. 498–501, February 2014.

[24] S. Braginskii, "Transport processes in a plasma," *Review of Plasma Physics*, vol. 1, pp. 205–311, 1965.

[25] Y. Feng, F. Sardei, J. Kisslinger, and P. Grigull, "A 3d monte carlo code for plasma transport in island divertors," *Journal of Nuclear Materials*, vol. 241, pp. 930 – 934, 1997.

[26] Y. Feng, F. Sardei, and J. Kisslinger, "3d fluid modelling of the edge plasma by means of a monte carlo technique," *Journal of Nuclear Materials*, vol. 266269, pp. 812 – 818, 1999.

[27] Y. Feng, J. Kisslinger, and F. Sardei, "Formulation of a monte carlo model for edge plasma transport," in *27th EPS Conference on Controlled Fusion and Plasma Physics*, vol. 24B, pp. 1188–1191, 2000.

[28] S. Dai, M. Kobayashi, G. Kawamura, S. Morita, H. Zhang, T. Oishi, Y. Feng, D. Wang, Y. Suzuki, and the LHD Experimental Group, "Emc3-eirene modelling of edge impurity transport in the stochastic layer of the large helical device compared with extreme ultraviolet emission measurements," *Nuclear Fusion*, vol. 56, no. 6, p. 066005, 2016.

[29] D. Reiter, *The EIRENE Code User Manual*, April 2017.

[30] "The eirene and b2-eirene codes," *Fusion Science and Technology*, vol. 47, pp. 172–186, February 2005.

[31] H. Frerichs, D. Reiter, Y. Feng, and D. Harting, "Block-structured grids in lagrangian 3d edge plasma transport simulations," *Computer Physics Communications*, vol. 181, no. 1, pp. 61 – 70, 2010.

[32] Y. Feng, F. Sardei, J. Kisslinger, P. Grigull, K. McCormick, and D. Reiter, "3d edge modeling and island divertor physics," *Contributions to Plasma Physics*, vol. 44, no. 1-3, pp. 57–69, 2004.

[33] Y. Feng, F. Sardei, and J. Kisslinger, "A simple highly accurate field-line mapping technique for three-dimensional monte carlo modeling of plasma edge transport," *Physics of Plasmas*, vol. 12, no. 5, p. 052505, 2005.

[34] "What are the limitations of braginskii's fluid equations and hazeltine's drift kinetic equation?." March 2005.

[35] H. G. Frerichs, "Three-dimensional plasma transport in open chaotic magnetic fields: A computational assessment for tokamak edge layers," Master's thesis, February 2010.

[36] e. a. M.C. Zarnstorff, "Equilibrium and stability of high-beta plasmas in wendelstein 7-as," in *20th IAEA fusion energy conference*, November 2004.

[37] V. Shafranov, "Plasma equilibrium in a magnetic field," *Reviews of Plasma Physics*, vol. 2, p. 103, 1966.

[38] L. Hively, "Tf-ripple losses from a non-circular tokamak," *Nuclear Fusion*, vol. 24, no. 6, p. 779, 1984.

[39] J. P. Freidberg, *MHD macroscopic equilibrium*, p. 245295. Cambridge University Press, 2007.

[40] J. P. Freidberg, *Ideal MHD*. Cambridge University Press, 2014.

[41] K. Radhakrishnan and A. Hindmarsh, *Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations*, 1993.

[42] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1971.

[43] D. Sarafyan, "Improved sixth-order runge-kutta formulas and approximate continuous solution of ordinary differential equations," *Journal of Mathematical Analysis and Applications*, vol. 40, no. 2, pp. 436 – 445, 1972.

[44] T. E. Oliphant, "Python for scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.

[45] S. Butterworth, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, pp. 536–541, 1930.

[46] P. Dierckx, "An algorithm for smoothing, differentiation and integration of experimental data using spline functions," *Journal of Computational and Applied Mathematics*, vol. 1, no. 3, pp. 165 – 184, 1975.

[47] J. Lore, J. Canik, Y. Feng, J.-W. Ahn, R. Maingi, and V. Soukhanovskii, "Implementation of the 3d edge plasma code emc3-eirene on nstx," *Nuclear Fusion*, vol. 52, no. 5, p. 054012, 2012.

[48] K. M. Young, "The c-stellarator-a review of containment," *Plasma Physics*, vol. 16, no. 2, p. 119, 1974.

[49] T. H. Stix, "Highlights in early stellarator research at princeton," *Journal of Plasma Fusion Research SERIES*, vol. 1, pp. 3–8, 1998.

[50] S. Yoshikawa and T. Stix, "Experiments on the model c stellarator," *Nuclear Fusion*, vol. 25, no. 9, p. 1275, 1985.

[51] D. Zhang, M. Glaubitz, H. Laqua, S. Marsen, M. Otte, and T. Stange, "Determination of electron thermal diffusivity at the wega stellarator," *Nuclear Fusion*, vol. 52, no. 4, p. 043002, 2012.

[52] M. Otte, D. Andruczyk, E. Holzhauer, J. Howard, R. König, L. Krupnik, H. Laqua, O. Lischtschenko, S. Marsen, J. Schacht, J. Urban, Y. Podoba, J. Preinhalter, F. Wagnet, G. Warr, and A. Zhezhera, "The wega stellarator: Results and prospects," in *AIP Conference Proceedings*, 2008.

[53] M. Otte, H. P. Laqua, E. Chlechowitz, S. Marsen, J. Preinhaelter, T. Stange, A. Rodatos, J. Urban, , and D. Zhang, "Overview of recent results from the wega stellarator," *Nukleonika*, vol. 57, pp. 171–175, 2012.

[54] M. Otte, H. Laqua, E. Chlechowitz, P. Drewelow, M. Glaubitz, S. Marsen, E. Mller, Y. Podoba, J. Schacht, T. Stange, F. Steffen, F. Wagner, R. Wolf, and D. Zhang, "Overview of experimental results from the wega stellarator," *Contributions to Plasma Physics*, vol. 50, no. 8, pp. 780–784, 2010.

[55] K. Horvath, J. Lingertat, M. Laux, and F. Wagner, "Langmuir probe measurements in the wega stellarator," *Contributions to Plasma Physics*, vol. 44, no. 7-8, pp. 650–655, 2004.

[56] K. Horvath, J. Lingertat, M. Otte, and F. Wagner, "Investigations of the electron temperature profiles at the wega stellarator," *Plasma Physics and Controlled Fusion*, vol. 48, no. 2, p. 315, 2006.

[57] S. Marsen, M. Otte, and R. Wolf, "Impact of magnetic islands on turbulent transport in the wega stellarator," *Nuclear Fusion*, vol. 52, no. 5, p. 054014, 2012.

[58] "Operation of the wega stellarator with vertical field and compensation coils," in *30th EPS Conference on Controlled Fusion and Plasma Physics*, July 2003.

[59] S. Lazerson, "Vmecwiki," (accessed April 18th, 2017).

[60] P. Bourke, "Intersection point of two lines," (accessed April 18th, 2017).