EVOLUTION OF EDGE PEDESTAL TRANSPORT BETWEEN ELMS IN DIII-D

A Dissertation
Presented to
The Academic Faculty

by

John-Patrick Floyd II

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Nuclear & Radiological Engineering Program

Georgia Institute of Technology
December 2014

EVOLUTION OF EDGE PEDESTAL TRANSPORT BETWEEN ELMS IN DIII-D

Approved by:

Dr. W. M. Stacey, Advisor
Nuclear & Radiological Engineering
Program, School of Mechanical
Engineering
Georgia Institute of Technology

Dr. A. Erickson
Nuclear & Radiological Engineering
Program, School of Mechanical
Engineering
Georgia Institute of Technology

Dr. R. J. Groebner
DIII-D National Fusion Facility
General Atomics

Dr. R. T. McGrath
GTRI; School of Materials Science and
Engineering
Georgia Institute of Technology

Dr. T. T. Utschig
CETL; Nuclear & Radiological
Engineering Program, School of
Mechanical Engineering
Georgia Institute of Technology

Date Approved: October 7, 2014

To my parents, family, and friends, whose support and guidance made this possible.
To my brother, whose incredible accomplishments have helped me drive myself further.

To God be the glory.
“The fear of the Lord is the beginning of wisdom” Prov. 9:10a

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to a number of people who have been instrumental in helping me earn my doctorate:

Dr. W. M. Stacey has provided unparalleled guidance while I conducted my research. His patience, technical mastery of fusion and fission, and immense experience, combine to provide a welcoming research environment conducive to excellence.

Dr. R. J. Groebner and the DIII-D research teams at General Atomics have provided me with extensive assistance and guidance as I learned how to use their systems and databases to conduct my research. Dr. Groebner has generously shared his expertise in the DIII-D diagnostics systems with patience and clarity, and the hospitality he and General Atomics extended to me during my visit has been invaluable in this endeavor

Dr. N. E. Hertel has been a great friend and mentor throughout my time at Georgia Tech, and I appreciate his door always being open.

Dr. M. D. Salomone has been a strong mentor and introduced me to a number of exciting fields, while providing support and elite instruction in course development.

Dr. Z. W. Friis assisted my introduction into fusion research as an incoming graduate student, and I value his friendship and guidance.

Dr. J. K. Lincoln, through excellent courses, helped spark my interest in other fields, and further assisted and guided me in broadening my academic experience

Those distinguished figures who have served on my thesis committees.

The US DOE and the Nunn School at Georgia Tech for their financial support.

The top-quality undergraduates who have assisted in my research, Shubhang Tandon and Steve Mellard, and also my lab mates through the years.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Evolution of measured profiles of densities, temperatures and velocities in the edge pedestal region between successive ELM (edge-localized mode) events are analyzed and interpreted in terms of the constraints imposed by particle, momentum and energy balance in order to gain insights regarding the underlying evolution of transport processes in the edge pedestal between ELMs in a series of DIII-D discharges. The data from successive inter-ELM periods during an otherwise steady-state phase of the discharges were combined into a composite inter-ELM period for the purpose of increasing the number of data points in the analysis. These composite periods were partitioned into sequential intervals to examine inter-ELM transport evolution. The GTEDGE integrated modeling code was used to calculate and interpret plasma transport and properties during each interval using particle, momentum, and energy balance. Variation of diffusive and non-diffusive (pinch) particle, momentum, and energy transport over the inter-ELM period are examined for discharges with plasma currents from 0.5 to 1.5 MA and inter-ELM periods from 50 to 220 ms. Diffusive transport is dominant for $\rho < 0.925$, while non-diffusive and diffusive transport are very large and nearly balancing in the sharp gradient region $0.925 < \rho < 1.0$. Transport effects of ion orbit loss are significant for $\rho > 0.95$, and are taken into account. During the inter-ELM period, diffusive transport increases slightly more than non-diffusive transport, increasing total outward transport. Both diffusive and non-diffusive transport have a strong inverse correlation with plasma current. Weakening the electromagnetic pinch may increase outward particle transport, and enable control over the rebuilding of the edge pedestal between ELMs.

CHAPTER 1

A BRIEF HISTORY OF FUSION

1.1 An Introduction to Nuclear Reactions

While fusion and fission reactors have been longtime features of the natural world (stars and the uranium deposits in Oklo, Gabon[7]), it was only recently that the human race discovered how to harness nuclear reactions. This odyssey was begun in the early 20th century, sparked by Einstein's theories of mass-energy equivalency, and continued through important contributions from fellow luminaries such as Rutherford, Bohr, Meitner, and Hahn. Enabled by the work of these men and women, and driven by the horrors of war, scientific knowledge of nuclear reactions was gained at a rapid pace, and this knowledge was employed to achieve both terrible and beneficial ends.

At the heart of generating energy from nuclear reactions is the relationship between binding energy and quantity of nucleons in the nucleus of an atom. This relationship is shown in Figure 1, where the binding energy per nucleon (protons and neutrons, which comprise an atomic nucleus) is given as a function of the quantity of nucleons existing in the nuclei of common atoms. Binding energy is aptly named, as it can be thought of as the level of "adhesion" of the components of a nucleus. To understand the generation of energy through nuclear reactions, the concept of a deep, dry well is useful, where escaping the well means the nucleus is no longer bound together (nucleons can escape). If the binding energy for an entire nucleus is low, that means that a small amount of energy is necessary to break the nucleus apart, i.e. the nucleus is only partway down the dry well, and a small addition of energy could allow the nucleus to escape the well. However, if the binding energy in a nucleus is large, a large amount of energy is necessary to break the nucleus apart, i.e. the nucleus is near the bottom of the well, and a large amount of energy is necessary to bring it to the top.

Nuclear energy is generated by moving from a nuclear arrangement with a lower binding energy per nucleon, to a nuclear arrangement with a higher binding energy per nucleon. Using the well analogy, energy is generated by falling from higher in the well to lower in the well. When this occurs, the energy given up by the change in nuclear arrangement (falling further into the well) is eventually transformed into heat, and this heat can be gathered by meticulously designed environments (nuclear reactors), and then used for a variety of purposes. When fission occurs, a large atom with a low binding energy per nucleon is split into two atoms with higher binding energies per nucleon, generating heat that is collected or utilized by some type of nuclear reactor. Due to favorable binding energy and other characteristics, the atoms most often fissioned in reactors are the Uranium-235 atoms (seen on the right of Figure 1). These atoms are split to form two atoms, both located towards closer to the middle of the binding energy per nucleon curve. The energy released is mostly transferred to the two atoms, which proceed to heat whatever medium in which they are embedded.

Nuclear energy is generated by fusion through the same process of reconfiguring nucleons from lower to higher binding energies, but in the opposite direction of fission, and on the low mass side of the binding energy per nucleon curve. Fusion is achieved by smashing two light atoms (low number of nucleons) with low binding energies per nucleon together to create one larger atom with a higher binding energy per nucleon (again moving down in the well and generating energy). Stars generate energy by fusing two Hydrogen-1 atoms together, and as stars age, they join heavier and heavier atoms to move up the binding energy curve. This fusion is thought to be the only source of elements larger than hydrogen in the universe. Some stars are large and hot enough to fuse atoms all the way up to Iron-56, where the slope of the curve changes (implying that fusions involving Iron-56 or larger elements, and any other atoms, decrease the binding energy per nucleon, bringing the nuclear arrangement up the well, and absorbing energy). Once the fusion of Iron-56 begins, this subtraction of energy decreases the outward thermodynamic pressure which counteracts gravity to keep stars stable. Once this outward pressure is gone, stars above a certain size will succumb to gravity, and collapse into a black hole. However, it is also interesting to note that the fusion of large atoms during this collapse process is thought to be the only source of atoms larger than Iron-56, including gold, mercury, uranium, and all other heavy elements.

1.2 Fusion Reactors

Once scientists understood the implications of Figure 1, and fission reactions had been described and observed (splitting heavy atoms into lighter atoms, and moving up the right side of Figure 1), scientists realized that energy could also be released using the same phenomena by fusing lighter atoms to create heavier atoms (moving up the left side of Figure 1). However, they quickly realized that it was difficult to smash lighter atoms together, and although the larger elements had a convenient propensity to split when hit with an energetic neutron, there was no such trick for the lighter atoms. The electrostatic repulsion between ions dictates that they will almost surely deflect off one another when colliding, under most conditions, and in order to get atoms to fuse upon collision, they must be travelling at high speeds.

It is convenient to consider the idea of attempting to roll a bowling ball up the side of a volcano, and having it fall into the caldera. The energy of the ion must be sufficient to overcome the electrostatic Coulomb repulsion between it and the other ion (the volcano slope), and enter the range of the strong nuclear force (the caldera). Additionally, if the ball is not rolled directly up the side of the volcano, it will be diverted from its radial course, and roll down another side of the volcano. This is a good analogy for the nearly head-on impact that must occur between two ions for in order to overcome the Coulomb Barrier without being deflected. Needless to say, the chances of these two conditions (sufficient speed and direct angle of impact) being fulfilled at the same time are rather low. This is why large quantities of well-confined ions at high temperatures are necessary for fusion. If a large part of the ions have enough energy to overcome the Coulomb barrier and fuse, there are enough ions to have many ion-ion interactions, and the ions cannot easily escape the plasma, fusion will occur.

Stars use gravity to both thermodynamically heat (increase the speed of) atoms to a level where fusion is likely to occur upon a collision, and confine a large number of atoms to a limited volume, so that the atoms have many opportunities to fuse. On Earth, humans are currently unable to use gravity to serve either of these purposes, but two major confinement schemes have been devised to allow particles to be heated and confined in a small space. The most promising confinement scheme for fusion energy generation was proposed and developed in the 1950s by the Soviet physicists Tamm, Sakharov, and Lavrentiev. The scheme was improved to practicality by Artsimovich, and called a “tokamak”, a Russian acronym that is widely used today to describe the dominant type of magnetic confinement fusion reactors. It consists of a toroidal (doughnut-shaped) plasma chamber ringed by magnetic coils that induce a magnetic field running through the center of the device. A simplified schematic of a tokamak reactor and an actual cross section are shown in Figure 2 along with an illustration of the toroidal coordinate system, which will be used extensively in this document.

Gaseous hydrogen fuel is injected into the reactor, and a toroidal current is induced in the plasma through transformer action using the central solenoid magnet stack. This heats the fuel resistively (like an electric stove) to become a plasma. Once the fuel is in a plasma state, the constituent atoms have been ionized (positively charged), the electrons have been separated from the plasma ions, and both particle types follow magnetic field lines (although in opposite directions). A large toroidal magnetic field is then induced in the tokamak by the toroidal field coils that ring the plasma chamber. This field also causes the electrons and ions in the plasma to move in opposite toroidal directions, inducing additional toroidal current in the plasma. This toroidal plasma current then induces a poloidal magnetic field, which adds to the toroidal magnetic field generated by the toroidal field coils to produce an overall helical magnetic field (a helix is a spiral structure, like DNA). The charged particles stick to and follow this helical magnetic field, which (mostly) confines the plasma particles to the plasma chamber, and is resilient to the various electromagnetic forces and drifts that usually cause the plasma to escape confinement when using a poloidal or toroidal field alone.

This tokamak design is the basis for the leading fusion research reactors today, and is widely considered the leading contender for the design of an electricity generating reactor. Major improvements have been made in the design and operation of tokamak fusion reactors since they were proposed in the 1950s, but the two core requirements of fusion reactors, high plasma pressure ($p=nt$) and long energy confinement times (τ), remain difficult to achieve simultaneously. The sun has relatively low core temperatures (for the purposes of fusion), but it has excellent confinement and high density (around two orders of magnitude higher than that of water) in a very large volume. This ensures that each ion has a very low chance to escape, and a good probability to fuse with another of the many ions in the nearby area. Fusion reactors are much smaller, with much worse confinement, and much lower density plasmas. For comparison purposes, the density of the solar core is ~150 g/cm^3, liquid water density is 33.368 g/cm^3, STP air has a density of 0.025 g/cm^3, and high-performance fusion plasmas like those planned for ITER will have a density of ~3.3x10^{-10} g/cm^3. In the entire tokamak, the mass of the fuel at any given time will be several kilograms, but it will be outputting up to 500 MW thermal power by reacting just a small fraction of its mass. This small fuel mass has many safety advantages, and essentially prevents the possibility of damage from a disruption affecting anything outside the confinement configuration. A disruption is generally defined as any loss of confinement of the plasma, which allows it to impact the plasma-facing surfaces of the reactor.

Since fusion scientists have the advantage of being able to select the reaction they would like to facilitate for fusion energy generation, and they also have the ability to intensely heat the plasma, these tactics are both used to increase the achievable fusion yield in tokamak plasmas. In Figure 3, the fusion reactivities for several major fusion reactions are shown as a function of temperature. The Deuterium (Hydrogen-2, D) and Tritium (Hydrogen-3, T) reaction is envisioned for use in the first generation of fusion reactors due to it being the most reactive at the temperatures achievable using current technology.

$H^2 + H^3 \rightarrow He^4 + n^1 + 17.6$ MeV

The products of this reaction are a stable Helium nucleus (alpha particle), which carries away 3.5 MeV of the energy, and a neutron, which carries away 14.1 MeV of the energy. It is important to note that most operational research reactors use the less-favorable Deuterium-Deuterium reaction to mostly avoid the health, safety, and materials complications of dealing with neutron radiation and radioactive Tritium fuel. The reaction rates and other plasma properties from a D-D plasma can then be extrapolated to obtain a good estimation of what would be occurring in a D-T plasma. In the D-T reaction, the helium nucleus usually dissipates its energy in the plasma due to its highly charged state, while the neutron almost always leaves the plasma, and is used for heat generation, fuel creation (via interactions with Lithium or other atoms that produce another T atom, to allow a self-sustaining fuel source), or other purposes (such as transmuting spent nuclear waste, see the SABR concept[8]).

Confinement schemes like the tokamak have greatly increased the achievable energy confinement times and pressures in fusion reactors. Various heating schemes, such as radiofrequency heating (same idea as a microwave) or high-energy neutral beam heating (analogous to a blowtorch), have enabled scientists to increase the temperatures of particles to extremely high levels, and greatly increase the chance that they will fuse upon collision. In fact, the temperatures that will be achieved in fusion power generation reactors are at least an order of magnitude larger than those in the core of the sun[9]. These high temperatures, combined with advances in magnetic confinement, allow fusion reactions to occur in modern fusion reactors at a sufficiently high frequency to produce meaningful energy outputs. This is true even when using the D-D reaction, which is less productive than the D-T reaction that will be facilitated in electricity generating reactors.

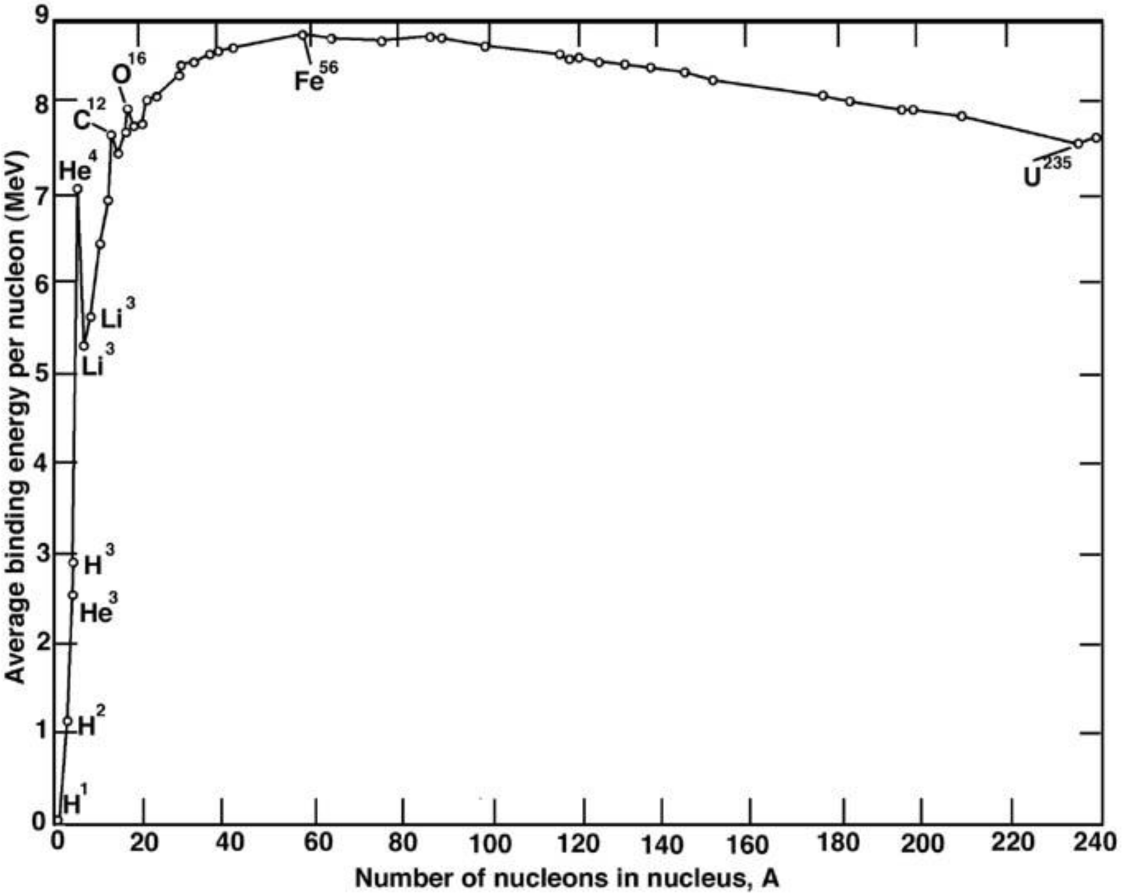


Figure 1: Binding Energy per Nucleon - Average Binding Energy per nucleon against the number of nucleons in atomic nuclei[3]. Uranium-235 is found in the upper right, Deuterium (Hydrogen-2) and Tritium (Hydrogen-3) are found in the lower left, and Iron-56 is located near the middle, at the point where the trend of the curve changes direction.

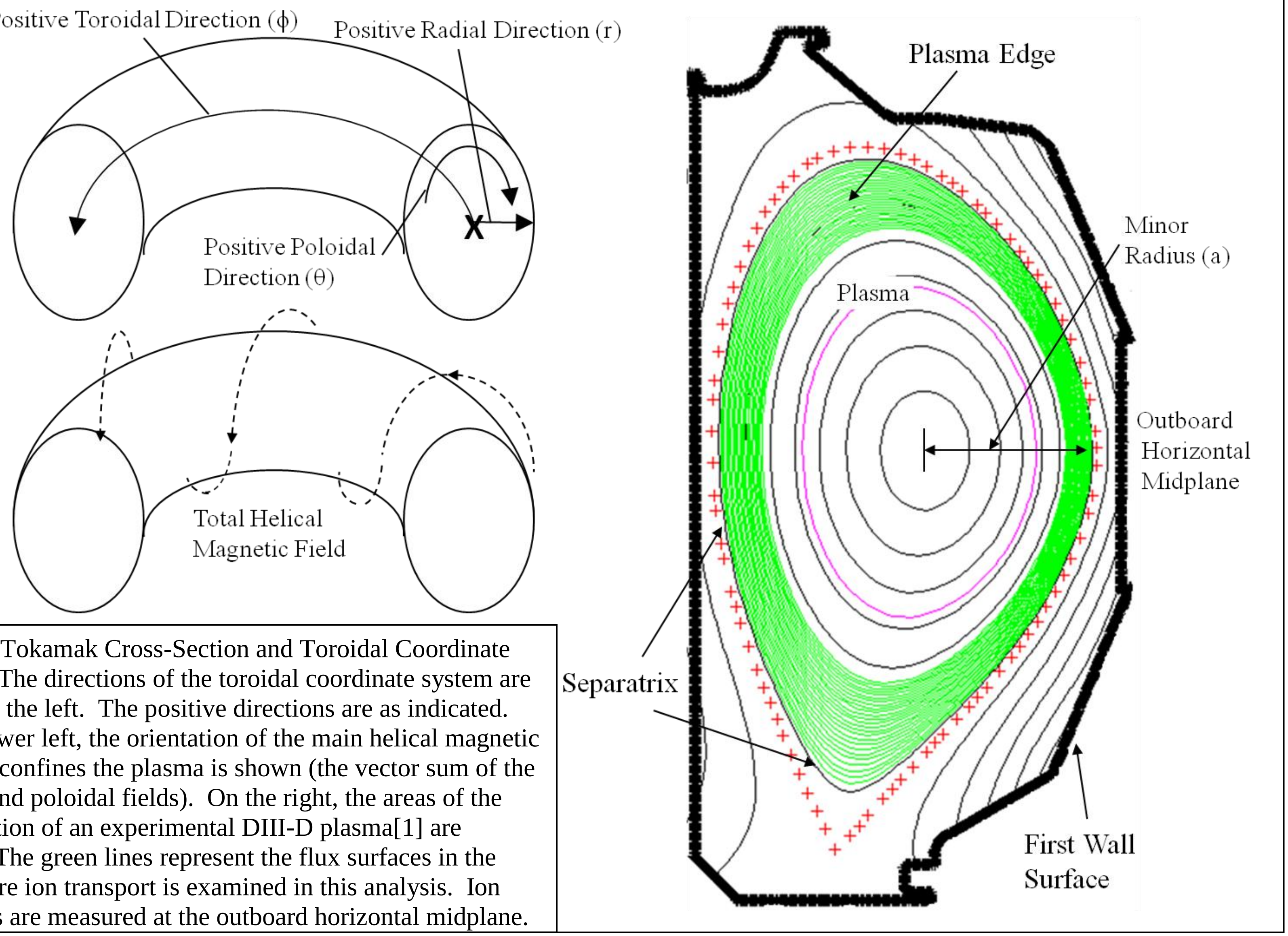


Figure 2: Tokamak Cross-Section and Toroidal Coordinate System - The directions of the toroidal coordinate system are shown on the left. The positive directions are as indicated. On the lower left, the orientation of the main helical magnetic field that confines the plasma is shown (the vector sum of the toroidal and poloidal fields). On the right, the areas of the cross-section of an experimental DIII-D plasma[1] are labeled. The green lines represent the flux surfaces in the edge where ion transport is examined in this analysis. Ion properties are measured at the outboard horizontal midplane.

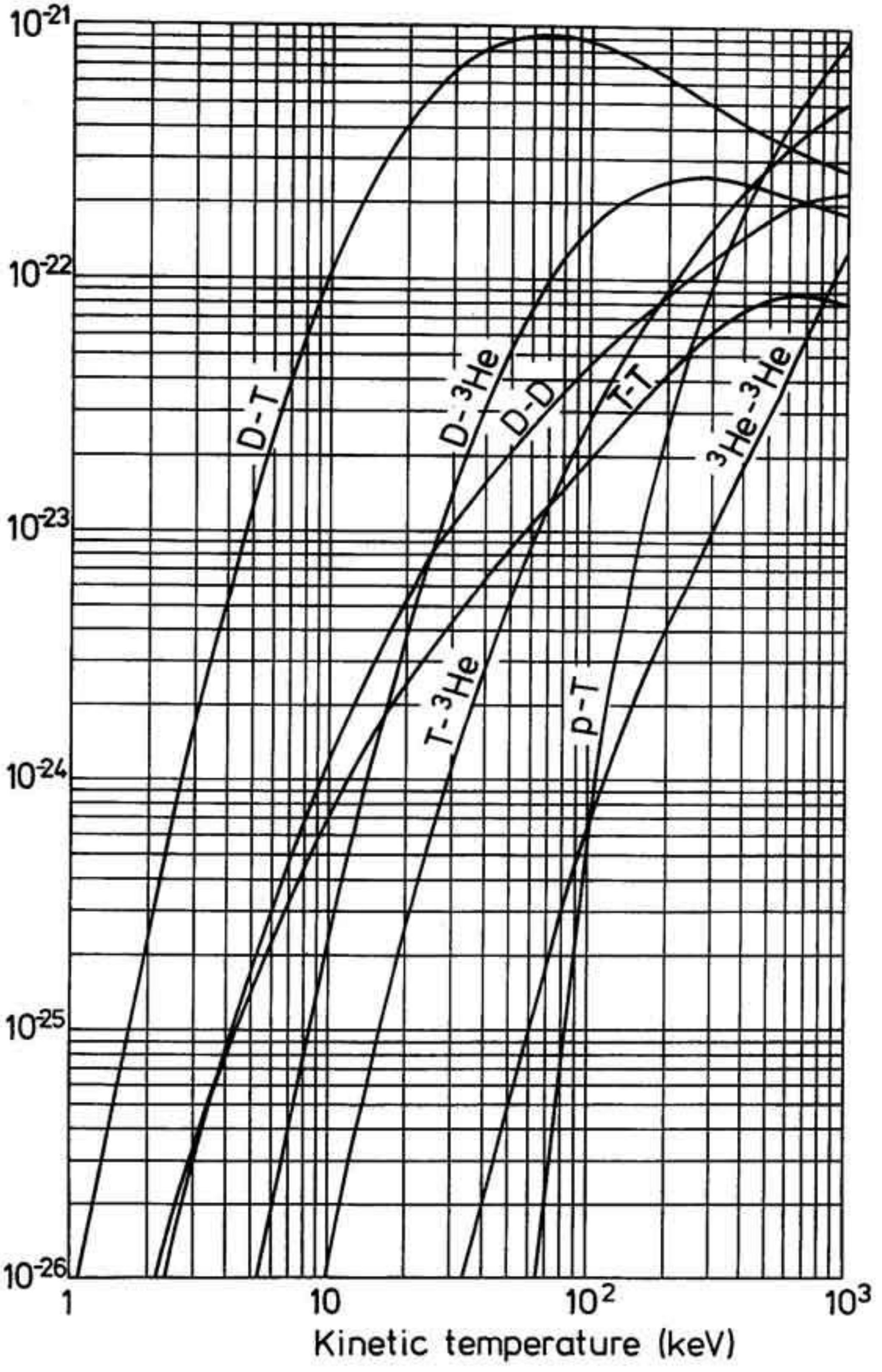


Figure 3: Fusion Reaction Reactivities - Reactivities for selected fusion reactions as a function of temperature[2]. The D-T reaction will be used in ITER and future reactors, while the D-D reaction is used in most research reactors.

CHAPTER 2

H-MODE, THE ITER PROJECT, AND THE STATUS OF FUSION ENERGY

2.1 The Discovery of H-mode tokamak operation

At the German ASDEX tokamak in the early 1980s, an important discovery was made that permanently changed the outlook of fusion energy. Upon application of a high level of neutral beam heating during an experiment, the plasma confinement doubled, and various other plasma characteristics changed dramatically. This marked the discovery of high-confinement mode [10-12] or H-mode operation in plasmas, which is characterized by high plasma confinement, better plasma performance, and sharp gradients in plasma parameter profiles in the edge of the plasma. The differences among the old operating regime, which mainly used Ohmic heating (now known as low-confinement mode or L-mode), and the new operating regime, are many, but one major aspect is shown in Figure 4. This plot shows the density profile in the edge of a shot before the low-to-high confinement mode transition, and after, illustrating the difference in the profiles that results from better plasma confinement (and other processes that will be examined in this document). The cliff-like structure near the separatrix in the H-mode density profile is an important feature of H-mode operation, and is called the edge pedestal.

This operational mode, now the benchmark for operating regimes, has played an important role in the design of tokamaks since its discovery, and has greatly enhanced the economic case for fusion. However, there are several issues that complicate the future use of H-mode in generating reactors, and one remains especially relevant – the presence of edge-localized modes (ELMs). These instabilities are localized in the edge region of the plasma. They destabilize the plasma, break local confinement, and send a surge of hot plasma ions radially outwards past the separatrix. Some of the ions are caught in the magnetic field lines and plasma flow going to the divertor, and impact there, but others impact the wall, delivering significant heat pulses. In research reactors, the ELMs occurring in deuterium plasmas rarely have enough energy to seriously damage the components of the plasma chamber which enclose the tokamak (these are called plasma facing components or PFCs). In sharp contrast, the heat pulses expected from these ELMs in a large, high performance reactor like ITER are extremely energetic, and would be impossible to withstand regularly without sustaining serious equipment damage. This obstacle is especially daunting for the development of economically feasible generating fusion reactors, which would be expected to run reliably for years without requiring major overhauls or repairs. Thus, it is apparent why the reliable control and/or mitigation of these ELMs continues to be a high priority for the fusion community.

Other challenges that the fusion community must solve to achieve a viable fusion generating reactor include several physics issues and the problems associated with finding or developing the materials capable of withstanding the extreme conditions required of fusion reactor PFCs. The ongoing generation of tritium to maintain a sufficient level of D-T fuel is another difficult engineering task, and the lack of data for radiation damage from such high energy neutrons as the fusion D-T neutrons complicates radiation protection efforts. All of these areas are the focus of intense research efforts, and most fusion scientists believe they can and will be solved by the time solutions are required.

2.2 The ITER Project

In 1973, Richard Nixon and Leonid Brezhnev met to discuss the creation of an international working group to develop a large experimental fusion reactor. The result of this effort was to be called the INTOR reactor, and several major nations, such as Japan, the United States, the Soviet Union, and a number of European nations, joined together to achieve this goal. The collaboration was named the INTOR workshop, and groups of scientists and engineers around the world were formed to design the reactor, and identify challenges that needed to be addressed.

Although the INTOR reactor was never built, the international institutions and norms regarding multilateral fusion development had been emplaced. The concepts, skills, and international relationships cultivated through this effort were used to infuse a “reboot” of the INTOR project with experience and purpose. The ITER project was the beneficiary of the INTOR Workshop, and was founded 1987, becoming the flagship international fusion development program.

Many major economies are partners in the ITER project, including the United States, Japan, the European Union, China, Russia, South Korea, and others. The ITER fusion development program is inherently complex and bureaucratic due to items such as in-kind contributions, an evolving design, and varying international commitment and funding levels reliant on domestic politics. Despite these challenges, the reactor site and support buildings are currently under construction in Cadarache, France, and construction and assembly of the tokamak itself is scheduled to begin in 2015. It incorporates some of the most advanced superconducting magnet, materials, and electronics technology available, and is expected to run for at least a decade and demonstrate the ability to produce ten times more power than it consumes. Along the way, the ITER reactor will serve as a test bed for prospective PFC materials, tritium generation equipment, advanced physics experiments, and radiation damage testing with 14.1 MeV neutron bombardment.

One of the biggest improvements incorporated in the ITER design is its large plasma volume of 840 m^3, which is much larger[9] than the largest plasma volumes achieved to date of 100 m^3. From a physics standpoint, the large size of the reactor is expected to provide sufficient volume for a high-performance plasma to operate without losing an inordinate amount of energy through the plasma surfaces. The characteristics of this plasma, which can be optimized for high-performance power generation, will provide a testing environment unique to ITER, where advanced plasma physics scenarios relevant to generating power reactors can be run, theories tested, and experiments completed. The tokamak is scheduled to begin operations in 2020, and its multi-decade expected lifetime will be invaluable to refining and completing development of those technologies, materials, controls, and diagnostics necessary to design an economically-attractive generating fusion reactor.

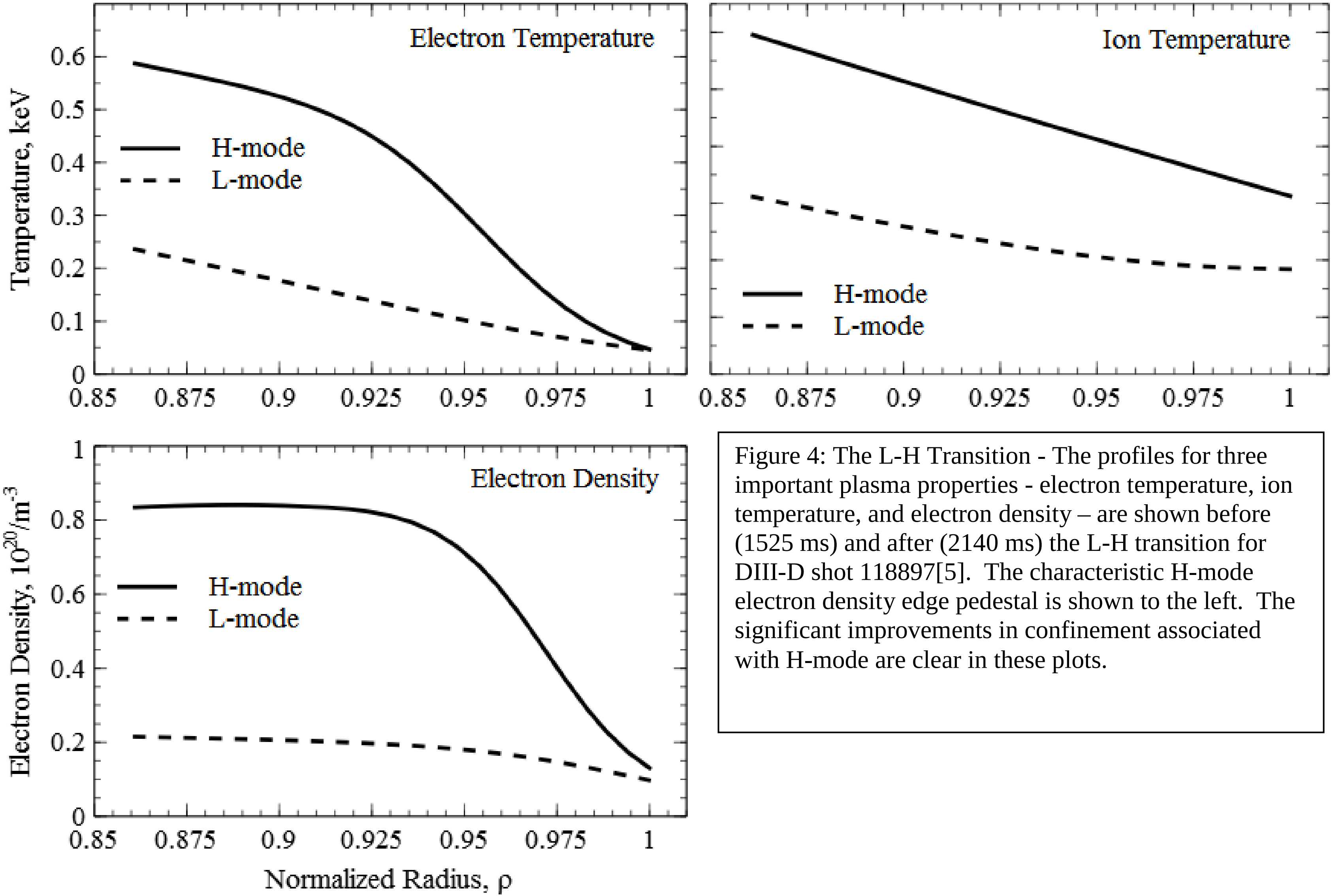


Figure 4: The L-H Transition - The profiles for three important plasma properties - electron temperature, ion temperature, and electron density – are shown before (1525 ms) and after (2140 ms) the L-H transition for DIII-D shot 118897[5]. The characteristic H-mode electron density edge pedestal is shown to the left. The significant improvements in confinement associated with H-mode are clear in these plots.

CHAPTER 3

THE EDGE AND EDGE-LOCALIZED MODES (ELMS)

3.1 The H-mode edge pedestal

The edge pedestal refers to the region of an H-mode plasma close to the separatrix, where the sharp gradients in density and other variables (which are characteristic of H-mode operation) are found. It is an important region of the plasma because there is experimental[13] and theoretical[14, 15] evidence that overall plasma performance depends on the edge pedestal parameters. This performance link is a result of the relationship between the value of the plasma temperature at the top of the edge pedestal (temperature pedestal height), and the maximum temperature in the center of the plasma. As shown in Figure 3, a higher core temperature generally increases the fusion rate and power generation, alongside other effects.

Another aspect of the formation of the H-mode edge pedestal is the sharp gradients that are formed between the top of the pedestal and the separatrix. These gradients are thought to be closely related to the increased confinement and performance of the H-mode regime[16, 17], as well as other important, complicated plasma phenomena such as bootstrap current. The close linkages between increased overall plasma performance, plasma confinement, and core temperatures, and the properties of the edge pedestal, ensure that the development of a more comprehensive understanding of the H-mode edge pedestal is a priority in the fusion community.

Both diffusive and non-diffusive transport mechanisms determine edge pedestal structure and transport, and they are subjects of ongoing research. Diffusive transport is usually outward, and relates to the plasma pressure and other thermodynamic forces, while non-diffusive transport mechanisms are related to electromagnetic forces such as the radial electric field, or VxB forces. The ultimate goal of these investigations into the edge pedestal region is to develop a predictive capability for edge pedestal structure and transport processes.

3.2 Edge-Localized Modes and mitigation techniques

The ELMs that have the potential to be most damaging are large, low frequency ELMs that eject large quantities of particles from the edge. These ejections eventually impact either PFCs or the divertor, causing large heat pulses to these components. This type of large ELM is generally known as a Type I ELM[18]. In the past decade, a theory explaining ELMs as a combination of peeling and ballooning modes[6] has been put forward. It is widely believed to provide a good description of the processes that directly cause ELMs, and also the ELM event itself, for Type I ELMs and other, smaller ELMs. The cyclical nature of ELMs in H-mode is demonstrated by the closed loop that plasma properties are believed to follow during the ELM cycle according to MHD peeling-ballooning mode theory. This loop is shown in Figure 5. The large loop is believed to describe the evolution and destabilization of Type 1 ELMs, while the smaller loops may describe smaller, Type III ELMs.

Cyclical ELMs[18] in normal H-mode operation will cause critical damage to PFCs, divertors, diagnostics, and other reactor components if allowed to occur in high-performance tokamaks like ITER and future devices. Efforts to retain the high confinement and good performance of H-mode operation, while eliminating or mitigating the ELMs, have taken several different forms in recent years. One of the most promising techniques has been the introduction of a special type of magnetic field coils (I-coils) into the first wall that induce a 3-D resonant magnetic perturbation fields in the plasma edge[19]. This additional field allows plasmas to operate in a state where few if any ELMs occur (across several machines and plasma configurations), and which maintains plasma performance levels in the neighborhood of H-mode plasmas. Other methods, such as injecting fuel pellets into the plasma edge[20], enable control of the ELMs by artificially inflating the local density to a level beyond the stability limit, thereby triggering ELMs. These induced ELMs are smaller than typical Type I ELMs, and they are frequently triggered, to ensure that large ELMs do not occur. High-performance operating regimes that do not feature ELMs or active ELM mitigation have been developed[21, 22], and further research is being done to learn more about the characteristics of these regimes.

All of these methods to mitigate or prevent ELMs have been studied extensively, and a theoretical understanding of the cause of ELMs has been provided by MHD peeling-ballooning theory[6]. However, most of the ELM-free operational modes and ELM mitigation techniques come with a plasma performance penalty, and many have other complications. During ELMing H-mode, the plasma edge leaves the stable parameter space, and an ELM occurs, decreasing gradients/currents and densities in the edge (as shown in Figure 5). Once the ELM event is over, the plasma edge traverses the stable parameter space, becoming unstable again at some point in time. This “traverse” is the time when edge pedestal transport is rebuilding the edge pedestal, which culminates in the plasma edge approaching the edge of the stable space and eventually leaving it, restarting the cycle. The transport that occurs in the edge pedestal as the gradients are rebuilt between ELMs is not well understood. However, by improving this understanding, it may be possible to provide important insight into the transport forces that occur in H-mode (the highest-performing plasma mode) and which rebuild the pedestal gradients, setting the stage for another ELM. Alteration or control of this edge transport between ELMs may prevent the plasma edge from again reaching an unstable area where an ELM occurs.

3.3 The edge, ELMs, and ITER and future devices

During the operation of ITER and future devices, potentially damaging events like ELMs must not be allowed to occur. Even in operating regimes that are mostly ELM free, occasionally large ELM-like modes can occur without warning, which could be very harmful in ITER. In order to obtain a better understanding of the triggers and likelihoods of these events, and ultimately, in order to develop a predictive capability, more details about the edge region, its transport processes, and its forces, must be gathered. Given the linkage between the edge pedestal, overall plasma performance, and confinement, a better understanding of transport in the edge pedestal may also be helpful in the optimization of plasma performance. The importance of the edge pedestal in tokamak operation, and of the instabilities that arise in that region, encourages the development of a predictive capability for edge structure, properties and transport dynamics. This capability is most useful if it encompasses a broad range of operating regimes, plasma events, and reactor designs.

The task is complicated by the fact that the edge region of the plasma is difficult to quantify, and less well understood than other parts of the plasma. The sharp gradients in the region change rapidly over a small area, making good spatial resolution in the area where the sharp gradients are located difficult to attain. As the plasma moves, the absolute radial location of the measurements changes, and although the data processing system tries to eliminate this behavior, this induces uncertainty in the final radial value for a given measurement. When ELMs occur, the magnetic field lines are broken, and then reconnect in a different configuration, further complicating the task of assigning a radial value to a measurement. Regarding qualitative understanding of edge transport, the extremely large forces, magnetic fields, neutral recycling, large gradients, electric fields, and various other plasma physics considerations, ensure that transport in the edge region is very difficult to understand or explain fully. This is all further complicated by the geometric assumptions (Figure 12) used in modeling to simplify the complex geometry of the tokamak plasma chamber, and the disproportionate effect they have on the outermost flux surfaces in the plasma.

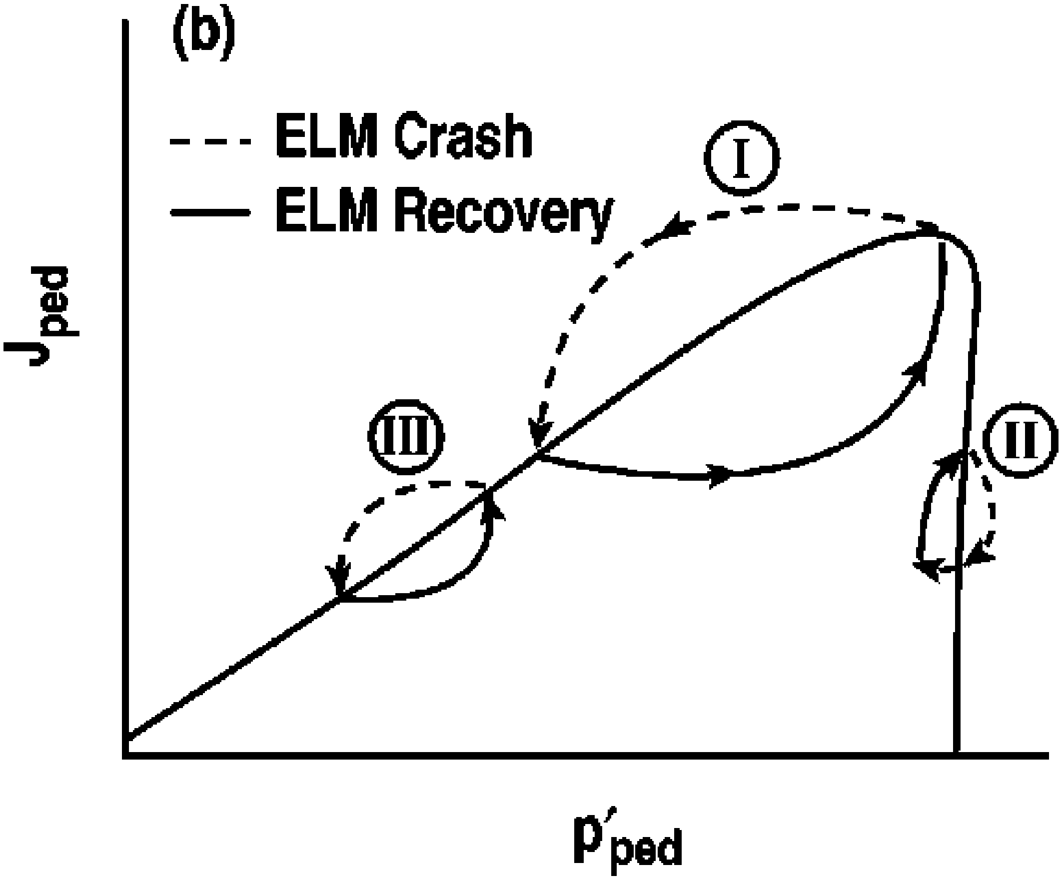


Figure 5: MHD Peeling-Ballooning Mode ELM Theory - A diagram of the ELM cycle predicted by MHD peeling-ballooning mode theory[6]. Current in the edge pedestal is on the y-axis, and pressure in the pedestal is on the x-axis. The plasma phase space stable to peeling or ballooning instabilities is enclosed by the triangular shape. Large, Type I ELMs are widely believed to cyclically follow the path marked I, while smaller ELMs may follow the similar, but shorter, circuits labeled II and III.

CHAPTER 4

EXPERIMENTAL DATA COLLECTION AT DIII-D

The world-class diagnostic systems in use at DIII-D are an important part of its research success, and are a feature that allows maximum utilization of the time and resources necessary to conduct an experimental discharge on the tokamak. General Atomics maintains a research team including a substantial number of diagnosticians and other experimental physicists who continually work to improve existing diagnostic capabilities, develop new ways to use the measurements, and help collaborating scientists use the data accurately in their own work. Data is collected from the diagnostics systems, processed to varying degrees, and written to the DIII-D MDSPlus experimental database, where it can be accessed by members of the extended DIII-D research effort.

The diagnostics most important to this research are those two which measure the properties of plasma electrons and impurity ions in the edge. A brief, general overview and description of these systems and their relevant functions will be given in this chapter. The charge exchange recombination system (CER, also known as CXR or CXRS) measures carbon (an impurity from the PFCs) temperature, density, and rotation velocity profiles, while the Thomson scattering system (TS) measures electron temperature and density in DIII-D. Measurements of the main deuterium ion properties in the plasma were not available for the shots examined here. However, the capability to acquire that data exists at DIII-D, and is in the process of being expanded to provide main ion measurements for more shots in the future.

4.1 The DIII-D CER System

Charge exchange recombination systems use photon spectroscopy to determine several important properties of the ions in the plasma[4]. Recent research[23] has shown that main ion (deuterium) data can be obtained using the DIII-D CER system, but for the shots analyzed in this work, the CER system is measuring the properties of the impurity carbon. During normal DIII-D operation, carbon from the plasma facing components (PFCs) is the dominant impurity in the edge.

The CER system relies on the presence of an injected beam of neutral deuterium atoms. At DIII-D there are four neutral beam systems where deuterium ions are brought to high energies (much hotter than the plasma ion temperatures), neutralized with minimum energy loss, and injected into the plasma for the purposes of heating, fueling, generating current, and driving rotation. Once in the plasma, the deuterium atoms in the beam interact with the fully ionized carbon atoms through charge exchange, as the energetic deuterium atom donates an electron to the fully ionized carbon atom, becomes ionized in the process, and joins the plasma. The carbon is now partially ionized and excited, and it radiates energy through the decay of its new electron from an excited to a ground state. This electron de-excitation releases a photon whose energy is characteristic of the electron orbits in carbon and also the deuterium energy. After being collected and analyzed, any deviations between the measured photon energies and the expected energies can be used to determine the properties of the originating carbon atoms. A schematic showing the geometry of the DIII-D neutral beam system can be found in Figure 6.

Optical viewing chords are positioned to intersect the neutral beam in the plasma and detect these characteristic photons generated along its edges from the charge-exchange and de-excitation process. To best study the edge pedestal region, which generally extends across about 10-15% on average of the (vertical) plasma cross-section radius and contains very large radial gradients in plasma properties, it is important to use diagnostics systems with high spatial resolution. The DIII-D CER system was designed to provide a high spatial resolution in the plasma edge (as well as good coverage throughout the plasma) and it usually can provide data about every 5 mm in the edge at the outboard midplane (about every 1% of the minor radius).

The DIII-D CER system has two sets of edge viewing chords that intersect the neutral beam lines, and are situated vertically and horizontally in the plasma. This allows measurement of the plasma from two different directions, and greater flexibility in using the collected data set to explore the plasma. (Sets of viewing chords for the core and other areas exist, and are shown in Figure 7 and Figure 8, but the edge CER data is most relevant to this work.) Recent analysis by the DIII-D diagnostics team has determined that the data from the tangential chords is slightly more reliable in the edge in most shots due to certain geometric advantages[24]. As a result of these findings, in this work the tangential CER data was used whenever possible, with vertical data being added when necessary. Most often this occurred when the data from the tangential chords was relatively sparse, and vertical chord data was needed to supplement the tangential measurements, and obtain enough data for profile fitting. The geometries of the tangential, vertical, and other CER chords, and the relevant neutral beams in DIII-D, are shown in Figure 7 using a horizontal cross-section of the tokamak, and in Figure 8 using a vertical cross-section of the tokamak.

From the known energy of a deuterium atom injected into the plasma as part of a beam, and the properties of a characteristic photon emitted as a result of a charge-exchange interaction between that beam atom and a plasma carbon atom, it is possible to calculate properties of the plasma carbon atom[4]. The expected location and structure of spikes in characteristic photon intensity on the energy scale resulting from this charge-exchange interaction can be compared with the actual properties of the intensity spike. The CER viewing chords record the observed characteristic photon intensity and energy data generated from the charge exchange interactions between the neutral beam deuterons and the carbon ions in the plasma. Figure 9 shows the expected photon emission intensities for representative transitions in oxygen and helium impurity ions at 0 and 400 eV. The observed peak, its width, and its location on the energy spectrum can be used to determine oxygen density, and oxygen poloidal and toroidal rotation velocities, and the same procedures are used with carbon in DIII-D to measure its properties..

After taking geometric data into account, this data can be used in four important, interrelated ways: to measure (i) carbon density, (ii) temperature, and (iii) toroidal and (iv) poloidal rotation velocity. If the approximate cross section for the charge exchange reaction is known from atomic physics, and the density of the beam is known, then (i) the intensity of the spike in detected photons at the characteristic energy would allow determination of the quantity of carbon atoms undergoing charge exchange. This could be used to calculate the density of carbon atoms at the corresponding intersection point between the CER viewing chord and the neutral beam on which the optics are focused. A related value, the (ii) width of the intensity peak centered on the characteristic energy, would provide a measure of the thermal velocity of the carbon atoms. This would be detected through the Doppler Effect, as the velocity of the photon-emitting carbon ion would cause them to have a small, symmetric variation in energy centered around the energy of the characteristic photon. If the particle were approaching the detector, the photon would be “blueshifted”, while it would be “redshifted” if it were receding from the viewing chord. The width in energy of the intensity peak would correspond to the magnitude of the thermal velocity of the particle.

Any difference between the location of the center of the intensity peak, and its expected location on the energy scale, would also be due to the Doppler Effect. A uniform shift up or down the energy spectrum would be attributable to a bulk particle velocity component towards or away from the viewing port. In this manner, (iii) the carbon toroidal rotation velocities can be determined from the tangential chords, and (iv) the carbon poloidal rotation velocities can be determined from the vertical chords. In addition, imperfections exist in orientation of the tangential and vertical chords, and the neutral beams, due to space constraints and other considerations. Through the geometric decomposition of the detected photon signals, additional carbon toroidal and poloidal rotation velocity data can be gathered from this misalignment.

Once the carbon (i) density, (ii) temperature, and (iii) toroidal and (iv) poloidal rotation velocities are found, they can be used in the equilibrium force balance equation:

$$E_r^{exp} = \frac{1}{-n_k^{meas} e_k} \frac{\partial p_k^{meas}}{\partial r} + B_\theta V_{\phi k}^{meas} - B_\phi V_{\theta k}^{meas} \qquad (1)$$

to calculate the radial electric field profile[25], which is a very important parameter in edge transport. In this equation, the radial electric field is signified by E, density by n, the charge of an electron by e, pressure by p, magnetic field by B, and velocity by V. The subscripts θ and φ refer to the poloidal and toroidal directions, respectively, while r refers to the radial direction, and k refers to the carbon species. Measurements of multiple impurity properties can be used to calculate multiple radial electric field profiles, which would be would be identical in theory, and independent of the impurity used in the calculation. In practice this is not the case, and any differences seen among electric field profiles calculated with the properties of various plasma impurities are useful in calibrating the CER system and for plasma analysis. Although carbon is assumed to be the dominant impurity in the edge of the plasma, other impurities do exist at much lower levels and could be used as described here.

4.2 The DIII-D Thomson system

Thomson scattering is an elastic process during which a photon from a laser with known energy is scattered by a plasma electron, and the Thomson scattering (TS) system gathers information about the scattered photon. The mechanisms by which the electron density and temperature are determined are similar to those used by the CER system to identify impurity density and temperature. The intensity of the scattered light is proportional to the electron density, while Doppler broadening in the expected intensity peak of the scattered light indicates the thermal velocity of the electrons, allowing a determination of their temperature. While the CER system uses the neutral atoms from the neutral beam and viewing chords to gather this information, the TS system uses lasers and a photon collection system.

The Doppler Effect allows data to be extracted from the Thomson scattered photon in a similar way to the CER system. The lasers used are highly-coherent to ensure that the incident photon properties are well-known, and the product of the Thomson scattering interactions are collected and analyzed to determine the density and temperature of plasma electrons across the radius of the plasma. The TS system uses three lasers to generate these measurements, a core laser, a tangential laser, and a divertor laser, and three collection systems to gather the scattered photons. The three lasers and their collection systems allow data gathering at 54 points in the plasma, and schematics of the system are shown in Figure 10 and Figure 11.

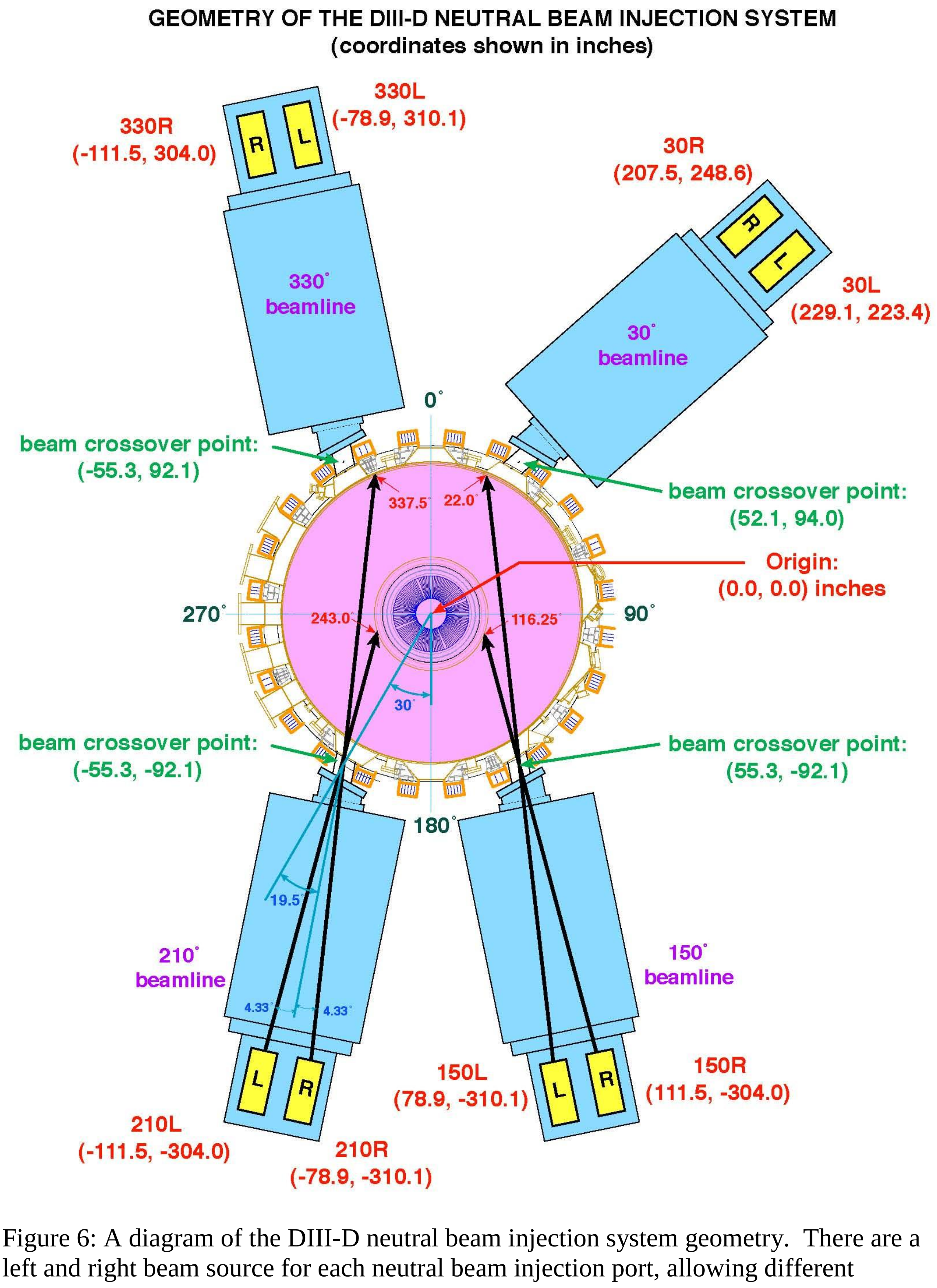


Figure 6: A diagram of the DIII-D neutral beam injection system geometry. There are a left and right beam source for each neutral beam injection port, allowing different injection locations. In addition, the 210 degree beamline port is situated to be able to deliver a beam in the opposite toroidal direction from the other three, increasing the experimental flexibility of DIII-D.

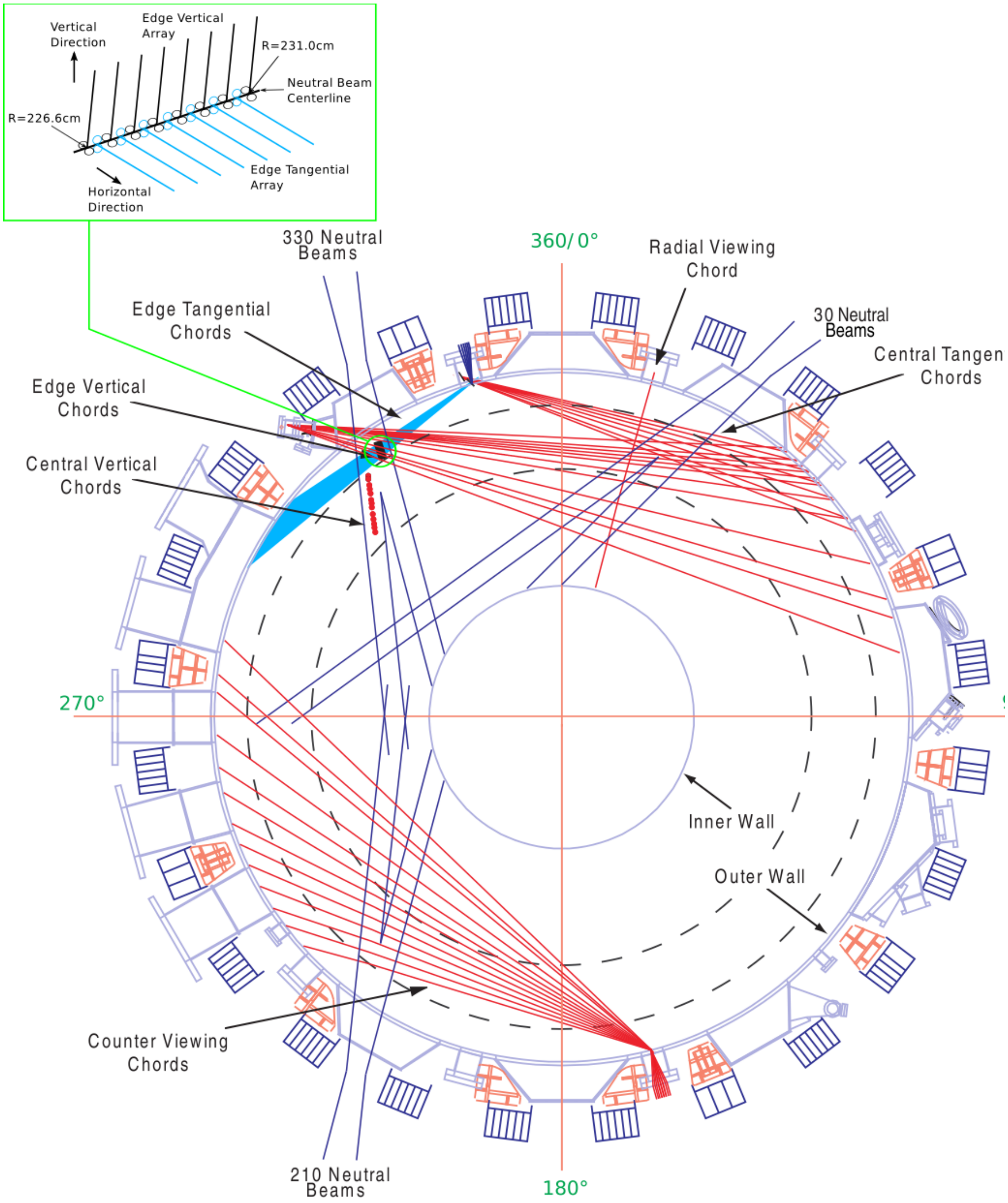


Figure 7: A diagram of the DIII-D CER system. The red lines are viewing chords, and they observe the interaction between the plasma and the neutral beams at the locations where the viewing chords and the neutral beams intersect. The 150 degree port shown in Figure 6 is not associated with any viewing ports. In the top left, there is a portrayal of the intersection between the vertical and tangential edge chord arrays and the 330 degree port beams from a different perspective, showing the vertical edge chord geometry.

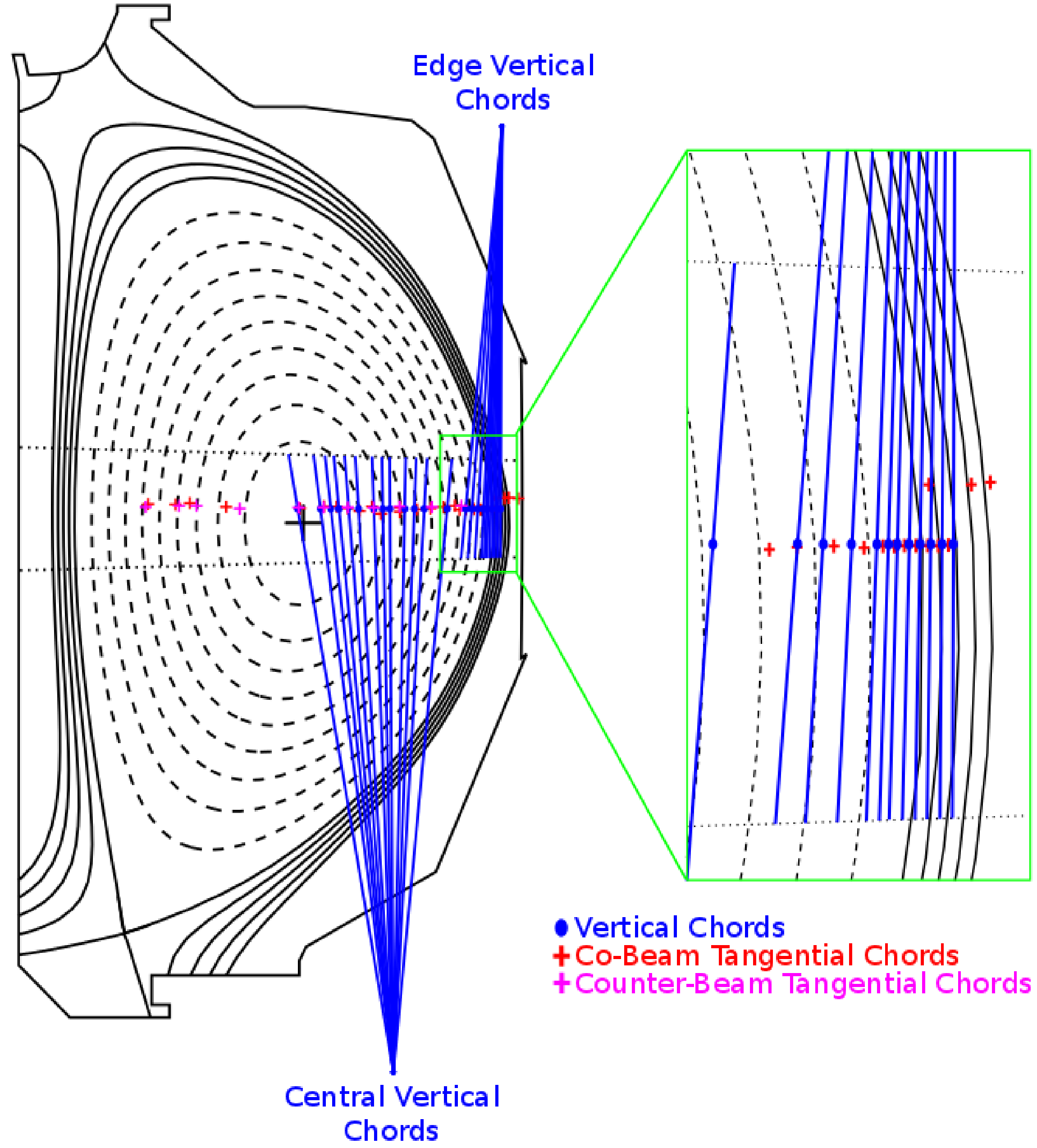


Figure 8: This is a schematic of the vertical viewing chords of the CER system. The locations of the horizontally-oriented chords in the edge shown in Figure 7 are indicated by red and purple pluses in this figure.

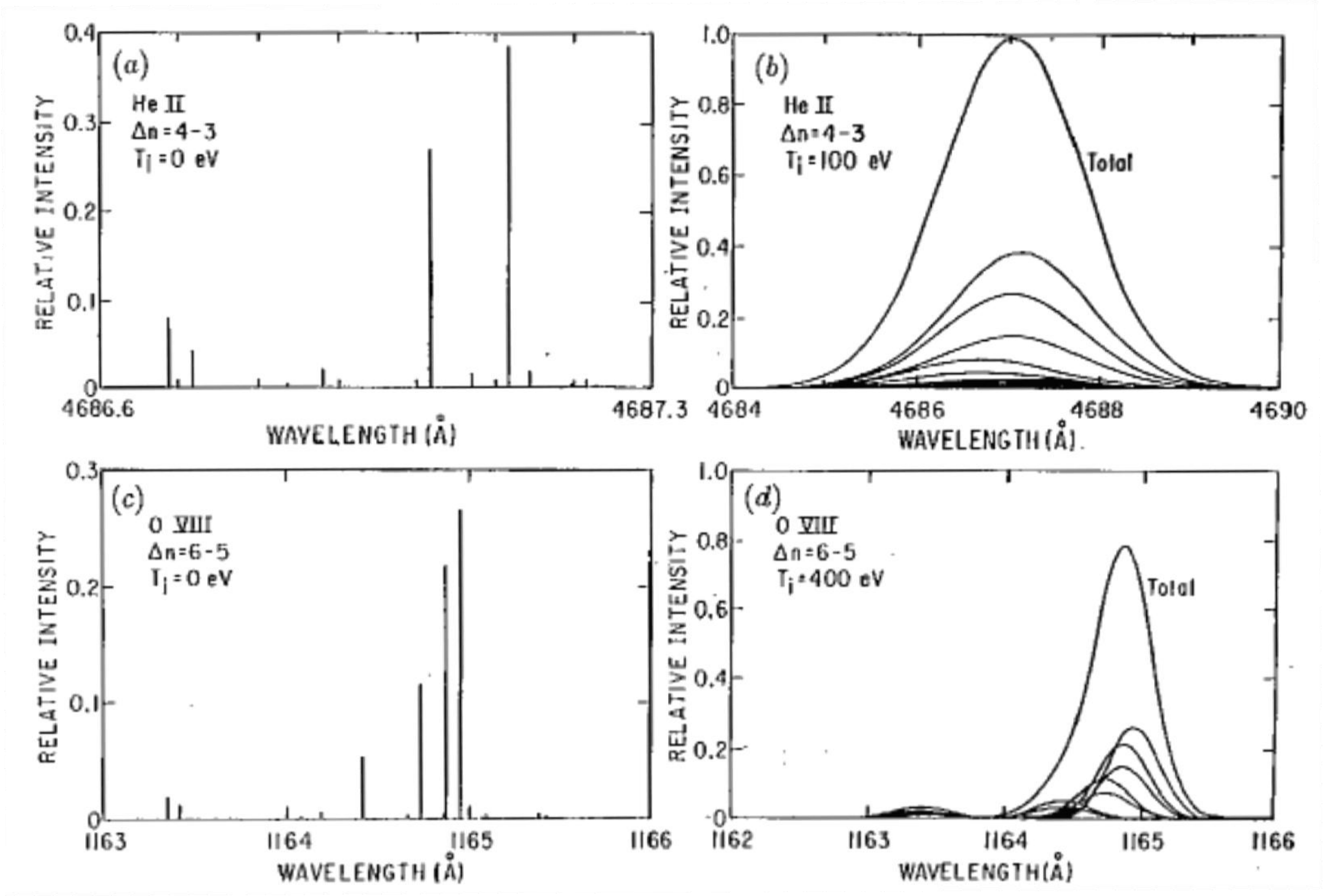


Figure 9: Doppler Broadening of Emission Lines – Doppler broadening of expected emission lines from representative transitions in Helium and Oxygen due to an increase in temperature from 0 to 400 eV[4].

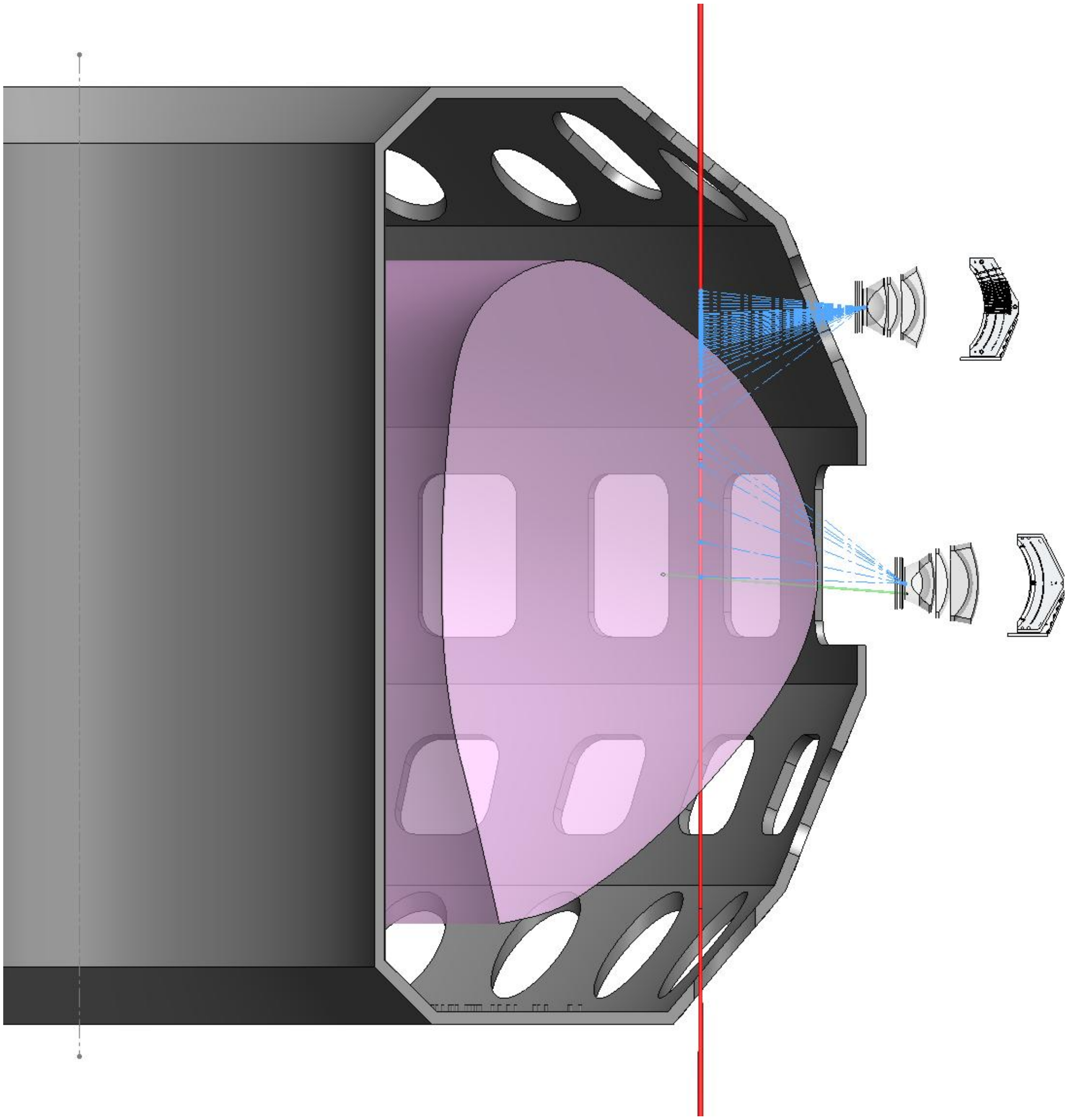


Figure 10: TS System Schematic - This schematic shows the core viewing chords of the DIII-D TS system that monitor several locations on the core laser, and also the path of the core laser.

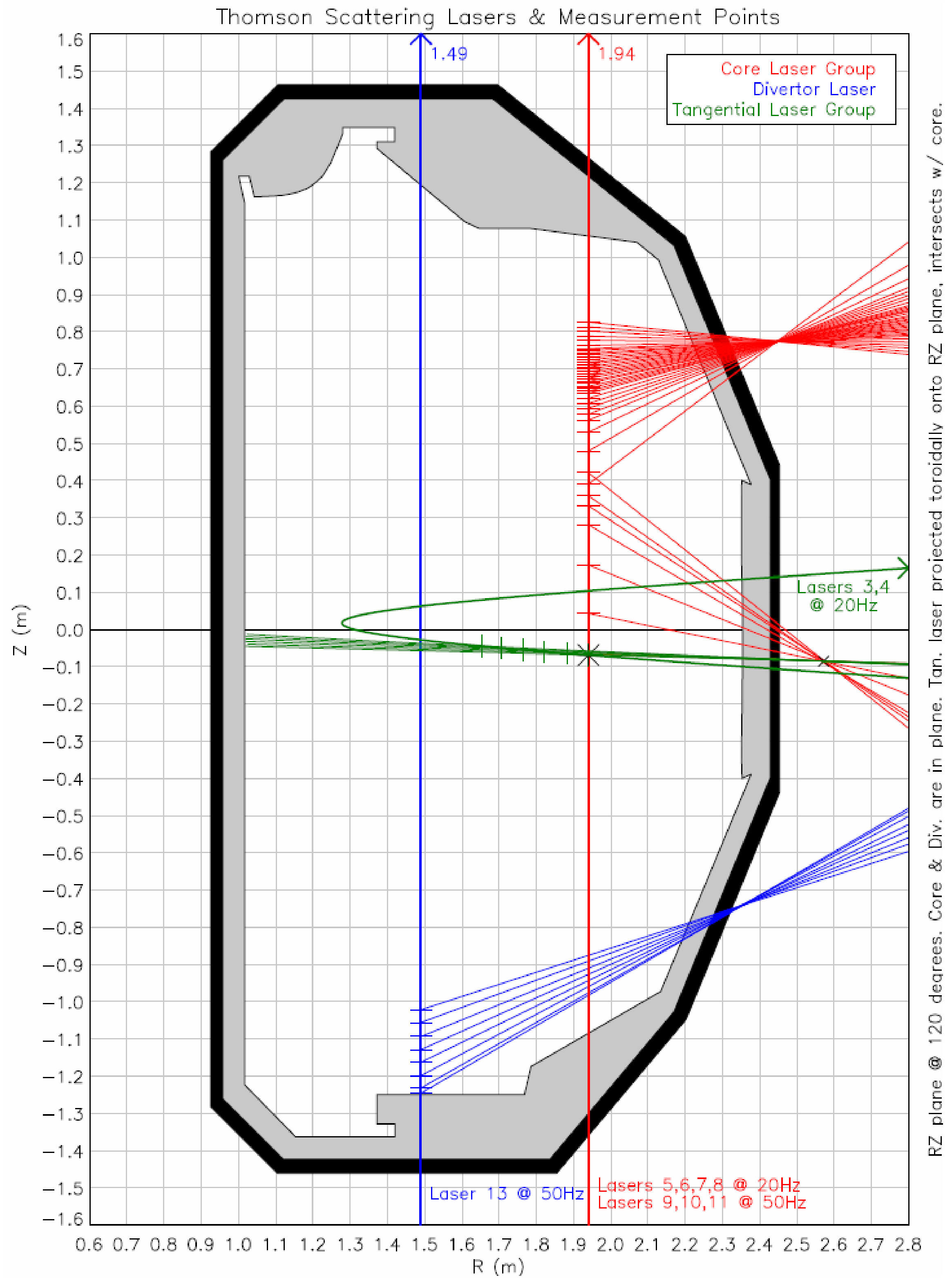


Figure 11: TS System Schematic - This schematic of the DIII-D TS system shows the core, tangential, and divertor laser systems used to detect plasma electron properties across the plasma. The viewing chords and the lasers are labeled and grouped by color.

CHAPTER 5

RESEARCH OVERVIEW AND INTRODUCTION

5.1 Research Overview

The purpose of the present research is to enhance the understanding of the edge pedestal, contribute to efforts to solve the ELM problem, and contribute to the development of a predictive capability for both. A better understanding of edge pedestal transport and its main drivers during the inter-ELM rebuilding process, which reconstructs the sharp edge gradients of H-mode after every ELM, could contribute towards ELM mitigation or elimination efforts that focus on preventing the plasma from reaching a state where an ELM is likely to occur. To this end, the measured evolution of plasma parameters between ELM events is interpreted in terms of the underlying evolution of diffusive and non-diffusive transport.

This study is a broadly expanded follow-up to the one described in Ref. [26]. The ELMs linked to this research are successive, Type I (large, relatively low frequency, heating power dependent[18]) ELMs in H-mode on DIII-D, and the analysis is performed for four DIII-D shots with different plasma currents. Interpretation of this data is performed in the context of particle, momentum, and energy balance constraints on as fine a time scale as possible in order to facilitate the identification of inter-ELM evolution patterns. The measured plasma profile evolutions, and variations thereof with current, are presented in Chapter 8, where the procedures for reducing the data to a form amenable to analysis are also described. Chapter 9 discusses the interpretive methodology, and the interpretation of diffusive and non-diffusive transport and heat transfer in these discharges is presented in Chapters 10 and 11. Conclusions are summarized in Chapter 12.

5.2 Geometric considerations

The cross-section of an actual tokamak plasma is frequently a very complex shape (shown in Figure 2) that somewhat resembles an asymmetric ellipse. Since reproducing this shape mathematically would introduce excessive complexity into plasma physics calculations, fusion scientists often represent the true plasma shape with a circular cross section that preserves certain essential features. This approximation is carefully carried out, and constraints are often imposed on the transformation to ensure that the approximation replicates those properties of the actual plasma which are most relevant to the research being conducted. The experimental data in this research is measured just above the outboard horizontal midplane (location shown in Figure 2), and the Miller equilibrium model[3] was used to map the data poloidally. The data was then averaged poloidally over each flux surface, and these values were plotted as a function of the normalized minor radius for comparison to calculation. The radius of the effective circular model is $\bar{r} = r\sqrt{0.5\left(1+\kappa^2\right)}$, where κ is the actual plasma elongation, and the normalized radius which is used in the plots for this paper is defined as $\rho = \bar{r}/\bar{a}$, where a is the effective plasma minor radius. This transformation is shown in Figure 12. After the data is recast into an effective circular model, the data set is sampled at twenty-five points in the edge, each with a separation of $\bar{r} = 0.005\,\mathrm{m}$

Another assumption that is often made when examining the edge of tokamak plasmas is that the edge pedestal region of the plasma can be approximated as an infinite slab. This is usually a good assumption near the separatrix because the width of the edge is very small compared to the other dimensions of the edge region (especially at the outboard midplane, where the data for this research is measured). However, further away from the separatrix, this assumption becomes less accurate due to geometric attenuation, causing area-dependent quantities such as fluxes or densities to be underestimated in the inner edge. Fortunately, when examining inter-ELM edge transport evolution, the inner edge of the plasma is one of the least active regions in most of the shots.

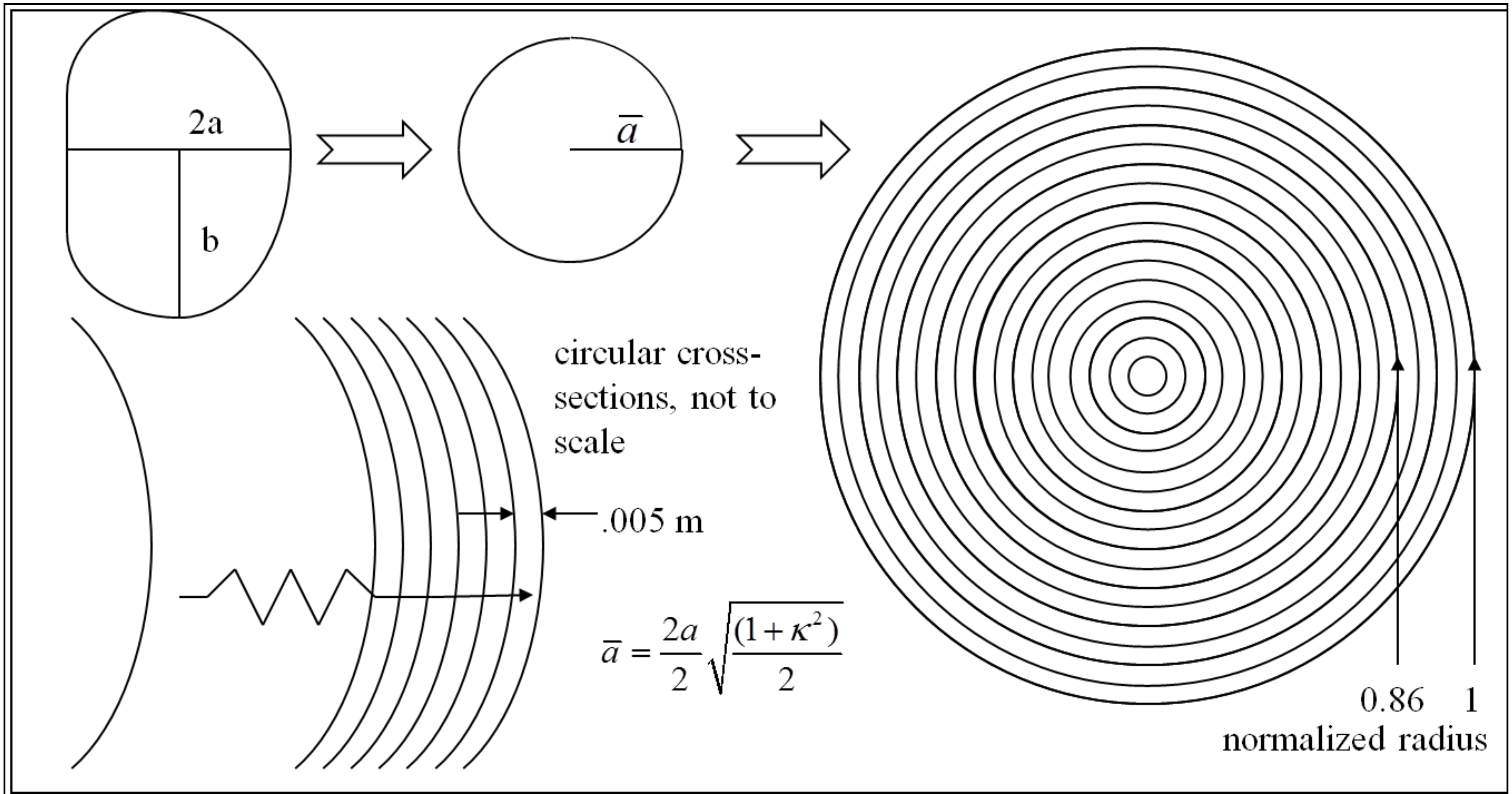


Figure 12: Geometric Transformation for Analysis - The geometric transformation from complex geometry to a circular cross-section for analysis, while preserving important flux surface area and other important quantities of the original geometry.

CHAPTER 6

EXPERIMENTAL DIII-D DATA: PROCEDURES FOR SELECTION, AGGREGATION, PROCESSING, AND PREPARATION FOR USE

6.1 Data Selection

The four shots selected for analysis are all from DIII-D H-mode plasmas run during the same set of experiments (DIII-D shots 144981, 144974, 144977, and 144987). They had plasma currents of 1.5, 1.2, 1.0, and 0.5 MA, respectively (through the rest of the document the shots will be referred to by their plasma currents), and the 1.5, 1.0, and 0.5 MA discharges were part of a current scan, being otherwise similar. In a parameter scan, such as a current scan, the parameter is changed, but the operators attempt to hold all other variables the same. The 1.2 MA shot was a reference discharge with significantly different plasma geometry. It had a slightly larger volume, and the x-point (lower extent) of the plasma was located on the plasma floor. A representative vertical cross-section of the plasma for the four shots is shown in Figure 13, illustrating the differences between the current scan shots and the reference shot.

As shown in Figure 2, in a right hand system, the positive toroidal direction is counterclockwise when the tokamak is viewed from above, and the positive poloidal direction is downward at the outboard midplane. For these shots, the toroidal plasma current was positive, the toroidal magnetic field negative, and the neutral beam was in a co-current configuration.

Several plasma properties varied slightly from shot to shot, and others were impacted by the ELM cycle, but most of the important properties of the current scan shots were approximately equal. They ran with a toroidal magnetic field of $B_\phi = -2.11$ T, a major radius of $R = 1.76$ m, a minor radius of $a = 0.59$ m, an elongation of $\kappa = 1.73$, a beam power P_{beam} = 4.2 MW, and a lower single null divertor configuration. The differences in geometry between the current scan shots and the reference shot (seen in Figure 13) are reflected in some of its geometric properties, such as a larger elongation of κ = 1.83, different upper and lower triangularities, a larger volume, and the x-point located directly on the floor. Other differences, such as a smaller toroidal field strength of B_ϕ = -2.06 T, were slight, but the sum of all the small differences in geometry and plasma properties combine to cause the reference shot (1.2 MA) data to be somewhat different from the current scan shots (I_p = 0.5, 1, and 1.5 MA).

This analysis used the established practice[27] of constructing a composite inter-ELM period from a set of successive inter-ELM periods, all occurring during the quasi-steady-state portion of an ELMing H-mode DIII-D discharge. The method of constructing composite inter-ELM periods is extensively detailed in Appendix A. The quasi-steady-state intervals used to construct the composite period for each of the four shots are shown in Figure 14.

The fourth plot in Figure 14 clearly shows that the 0.5 MA shot included many ELMs exhibiting “dithering” behavior [18]. This is characterized by the presence of subsequent, smaller peaks occurring immediately after the large, initial peak in the divertor D_α signal (which marks the leading ELM, starting the inter-ELM period). The dithering behavior was not present in the higher-current shots (due to better confinement, among other things), and in order to maximize the similarity among the four composite inter-ELM periods being examined, it was decided to exclude from analysis those inter-ELM periods exhibiting this “dithering” behavior for the 0.5 MA shot.

All of the inter-ELM periods in the quasi-steady-state period for the 0.5 MA shot were considered for inclusion in the analysis, but it is important to note that several inter-ELM periods appear to be “clean” in the 0.5 MA shot, when in fact they are dithering inter-ELM periods for which the dithering has come to an end. Care was taken to exclude these pseudo “clean” inter-ELM periods from the analysis, and often, other parameters were examined alongside the D_α signal to determine whether a large, leading ELM had occurred to start a new inter-ELM period, or whether the D_α spike was simply part of an existing inter-ELM period. These included the “prmtan_neped” (approximate pedestal electron density) and “density” (average plasma electron density) values discussed in Appendix A, as well as the estimated pedestal electron temperature “prmtan_teped” and the MHD power of the plasma, measured at a rapid frequency “wmhdf”. All of these variables were very helpful in excluding pseudo “clean” inter-ELM periods from the analysis for the 0.5 MA shot. The inter-ELM periods in the quasi-steady-state period of the 0.5 MA shot that were included in the analysis are denoted by a “+” in Figure 14.

6.2 Aggregating the Data and Choosing Intervals

The ion data utilized in this research was gathered using the DIII-D CER/CXRS diagnostic system [28], and the electron data was gathered using the DIII-D Thomson laser system [29], both of which were detailed in Chapter 5. The measured data from each inter-ELM period within the time selection (black rectangles in Figure 14) were combined into a composite inter-ELM period for each shot, and these composite inter-ELM periods were then partitioned into chronologically-ordered fractions of the whole, hereafter referred to as intervals. Accounting for minimum data quantity requirements and a desire for narrow time intervals (for high time resolution), a minimum interval width was found for each shot. The minimum interval fractional width (of the composite inter-ELM period) for a shot was dependent on the number and length of the inter-ELM periods in the time selection for that shot. This process is detailed further in Appendix A.

The ELM event disrupts edge transport, breaks field lines, and transports large quantities of particles and heat to the chamber wall. This study examines the rebuilding of the edge pedestal and other parameters between these events, and excludes the phenomena that occur during the ELM. In order to ensure the exclusion of transport directly associated with the ELM event, the divertor D_α signal (shown in Figure 14) was closely monitored for the large spikes in this measurement that are generally understood to mark the occurrence of an ELM. The duration of each "leading" ELM event that began an inter-ELM period was measured for each inter-ELM period from the quasi-steady-state period for each shot (except in the 0.5 MA shot, where only the leading ELMs of the selected inter-ELM periods were examined). Then the durations of the leading ELM events were normalized to the length of the subsequent inter-ELM, and the largest normalized duration was recorded. Finally, the first composite inter-ELM interval was begun after this recorded percent, preventing the overlap of any ELM event in the shot and any composite inter-ELM interval. The division of a sample inter-ELM, and the exclusion of the ELM event, is shown in Figure 15.

The inter-ELM period parameters are summarized in Table for each shot, including the minimum interval width and the longest normalized leading ELM event duration. The minimum interval width (best time resolution) for a shot is directly related to the total length of time elapsing between ELMs in the time selection for that shot. Measurements show that plasma parameters change more rapidly early in the inter-ELM period, and the best time resolution available was used early in the composite inter-ELM period. However, as each interval required a significant amount of effort to prepare, and due to the slower rate of evolution of the plasma profiles late in the inter-ELM period, coarser time resolutions were used for later in the composite inter-ELM period. Due to the slowed rate of change at this point, it is unlikely that significant transport evolution details were obscured by the coarser time resolution. Thus for the chronologically early intervals in the composite inter-ELM period, the minimum interval widths are used, while coarser time resolutions are used in the later intervals (usually a resolution of 20%, starting with the 40-60% interval, as shown in Figure 15).

Table 1: Characteristics of the selected ELMing H-mode DIII-D quasi-steady-state period from each shot. Percentages are the ratio of the interval width to the total composite inter-ELM period of the shot, and "best time resolution" refers to the highest time resolution (minimum interval width) available for the composite inter-ELM period of each shot. The last column notes the end of the widest D_α spike of any leading ELM event in the selected inter-ELM periods for each shot.

plasma current I_p	average inter-ELM period	number of inter-ELM periods	minimum interval widths	longest leading ELM event duration
0.5	54 ms	13 selected	17.5%	7.5%
1.0	156 ms	13	7.5%	7.5%
1.2	125 ms	4	10%	6.5%
1.5	220 ms	6	5%	5%

For the 1 MA shot, a large number of inter-ELM periods (13), and their relatively long average length of around 156 ms provided enough data to have a minimum interval width (maximum time resolution) of 7.5% of the composite inter-ELM period, enabling good profile evolution tracking for this shot. In contrast, the 0.5 MA shot had the same number of dither-free inter-ELM periods (13) during quasi-steady-state operation, but a relatively short average inter-ELM period length of around 54 ms. This resulted in less data being available for analysis, and forced the minimum interval width to be a relatively long 17.5% of the composite inter-ELM period length. Consequently, it is more difficult to track the evolution of edge transport for the 0.5 MA shot early in the inter-ELM period. Measurements show that plasma parameters change more rapidly early in the inter-ELM period, and in the first half of the composite inter-ELM period for each shot, the respective minimum interval width was used. This was done to maximize the ability to track profile evolution when it was occurring most rapidly. For efficiency, in the second half of the inter-ELM period when the evolution slowed down, larger interval lengths were used. However, due to the slow evolution, this seems to have had a minimal effect on the ability to observe the characteristics of the profile evolutions. The interval widths in the first half of the composite inter-ELM period are usually the minimum interval width for the shot, while in the second half of the composite inter-

ELM period, an interval width of 20% is often used. An illustration identifying all the interval widths used for the analysis of all four shots is shown in Figure 16.

The intervals of the composite inter-ELM period for each of the shots are shown in Figure 16. The relationship between the fractional width (of the composite inter-ELM period for each shot) of an interval, and the length of the same interval in absolute time, is displayed graphically. The locations of the intervals from each shot and their durations, with respect to absolute time, in relation to those same properties of the intervals of the other shots are also displayed. The 1.5 MA shot, in the uppermost position, has the longest average inter-ELM time of 220 ms of any shot, and therefore, its intervals collectively span the largest space. Although all of the intervals shown in Figure 16 were processed, only five intervals are shown for each of the four shots in most plots. This is to avoid unnecessarily complex graphs, and to clearly portray the trends.

The short average inter-ELM period length (~55 ms) for the 0.5 MA shot limited the amount of data available for analysis of the shot. To mitigate these limitations, and collect enough data to enable a reasonable time resolution for the analysis of this shot, the location of the end of the longest leading ELM event of all the selected inter-ELM periods for the 0.5 MA shot was determined in a more aggressive manner (i.e. the ELMs events in this shot were deemed to have ended once the peak D_α signal returned to 125% of the baseline inter-ELM signal, rather than 110%). Consequently, for the first interval (7-25%) of the 0.5 MA shot, there is a higher chance of the measured data being altered by ELM transport processes than for the first intervals of the other shots.

6.3 Processing (Fitting) and Using the Data

Once the intervals were created for each shot, data analysis scripts (mainly the profiles.py script) were used to facilitate the process of fitting profiles to the measured data contained in each interval. The script automatically fitted the electron data with “tanh” fits, which generally approximated the plentiful TS electron data trends well, and the ion data was fitted with spline fits. Due to a limited quantity of measured CER data available for these shots, the automatically-generated spline profiles fitted to the ion data were less accurate, and had to be manually revised. This process is detailed in Appendix A. These fitted, continuous profiles were sampled at twenty-five points in the plasma edge, starting in the flattop region at $\rho = 0.86$ and ending at the separatrix ($\rho = 1$) (shown at the outer midplane in green in Figure 2). Other data necessary to simulate the experimental plasma was retrieved from the DIII-D MDSPlus database, and all the gathered data was compiled for input to the GTEDGE code[30-32], which was used to interpret edge transport processes. An emphasis was placed on maximizing the use of experimental data or values traditionally inferred from experiment (such as the radial electric field, etc.). The values of other quantities needed for the analysis were calculated or interpreted from experimental data using GTEDGE, and the methods of doing so are discussed in Chapter 9.

Once the plasma was simulated using GTEDGE, several model parameters were tuned to ensure the best possible match between the simulated plasma and the experimental plasma. The experimental parameters matched were the energy confinement time, line-averaged density, central density, central temperature, and plasma density and temperature at the top of the pedestal. These parameters were matched by adjusting corresponding quantities in the simulation, including the parabola-to-a-power shape of the density and temperature profiles, and the height of the temperature pedestal, then updating the simulation. This exercise was mostly done using an automated script developed for this purpose, but given the often nonlinear relationships between the pairs of quantities, the tuning process sometimes required complementary manual operation to be fully completed. The custom scripts developed to automate the process of collecting fitted profile data, retrieving and processing measured data from the GA MDSPlus database, and tuning the model parameters, are fully detailed in Appendix B.

6.4 Simplifying Terminology

Beginning in Chapter 8, some terminology will be simplified to make explanations less cumbersome. References to “inter-ELM periods” will describe the composite inter-ELM periods for each shot, except when this convention is explicitly negated. References to “intervals” will be understood to refer to intervals of the composite inter-ELM period. Any data analyzed for a shot will be understood to come from the composite inter-ELM period created from the selected inter-ELM periods occurring during the selected quasi-steady-state operation of the shot. References to “profiles” will describe the profiles fitted to the measured plasma parameter values using spline or tanh fits. The term “evolution”, and its related words, will refer to the profile evolution that occurs in the edge during an inter-ELM period, between the leading ELM event, and the subsequent ELM. The “first interval” will refer to the interval immediately following the time excluded due to the occurrence of an ELM, the boundaries of which vary, but can be found in Figure 16. The “last/pre-ELM/asymptotic” profile refers to the last interval in the inter-ELM period found immediately prior to the closing ELM, which is defined as the 80-99% interval in each shot.

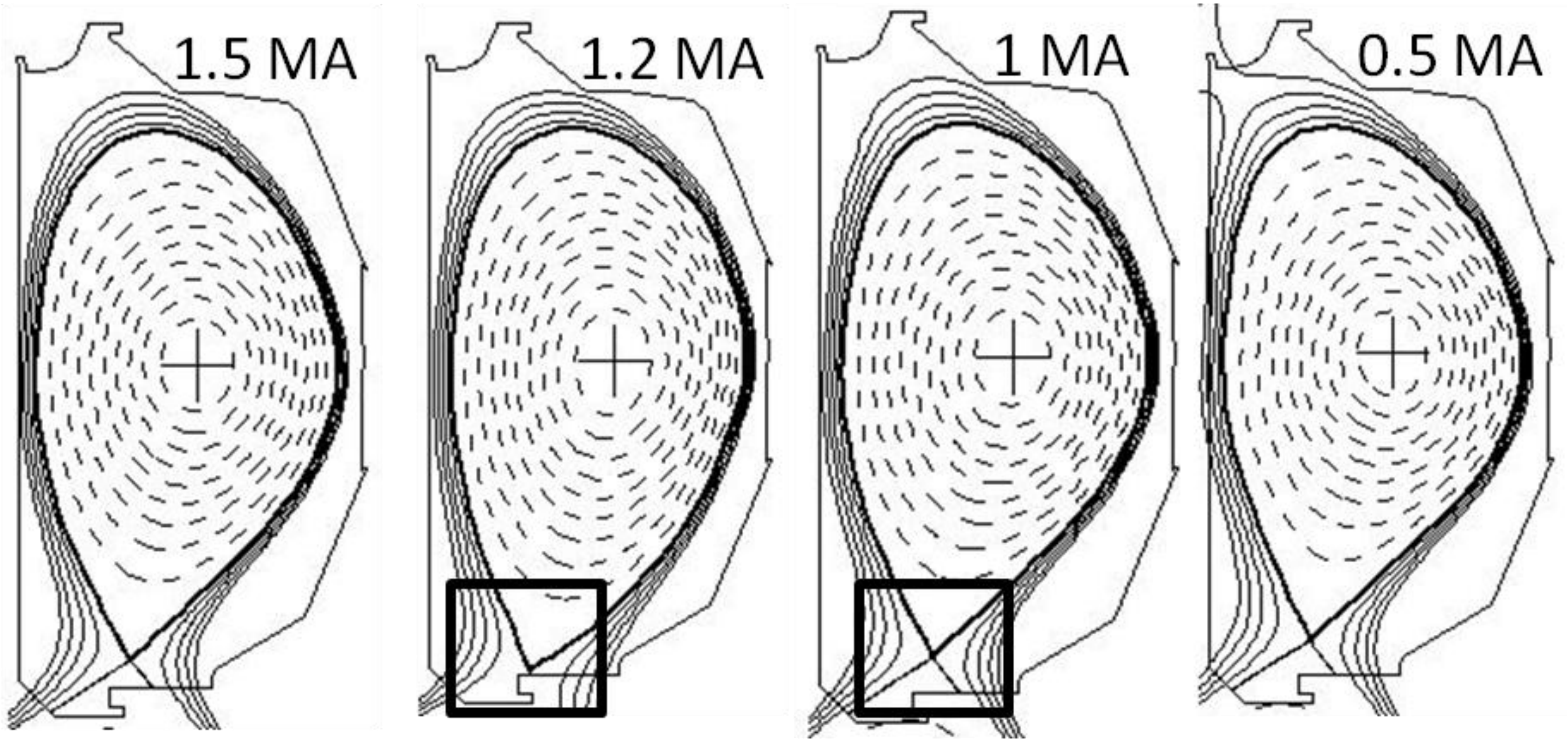


Figure 13: Plasma Shape for Each Shot – The flux surfaces for each shot at a representative time are shown. The different geometry of the 1.2 MA reference shot is highlighted with a box, showing the different location of the x-point. The other shots have geometries similar to each other.

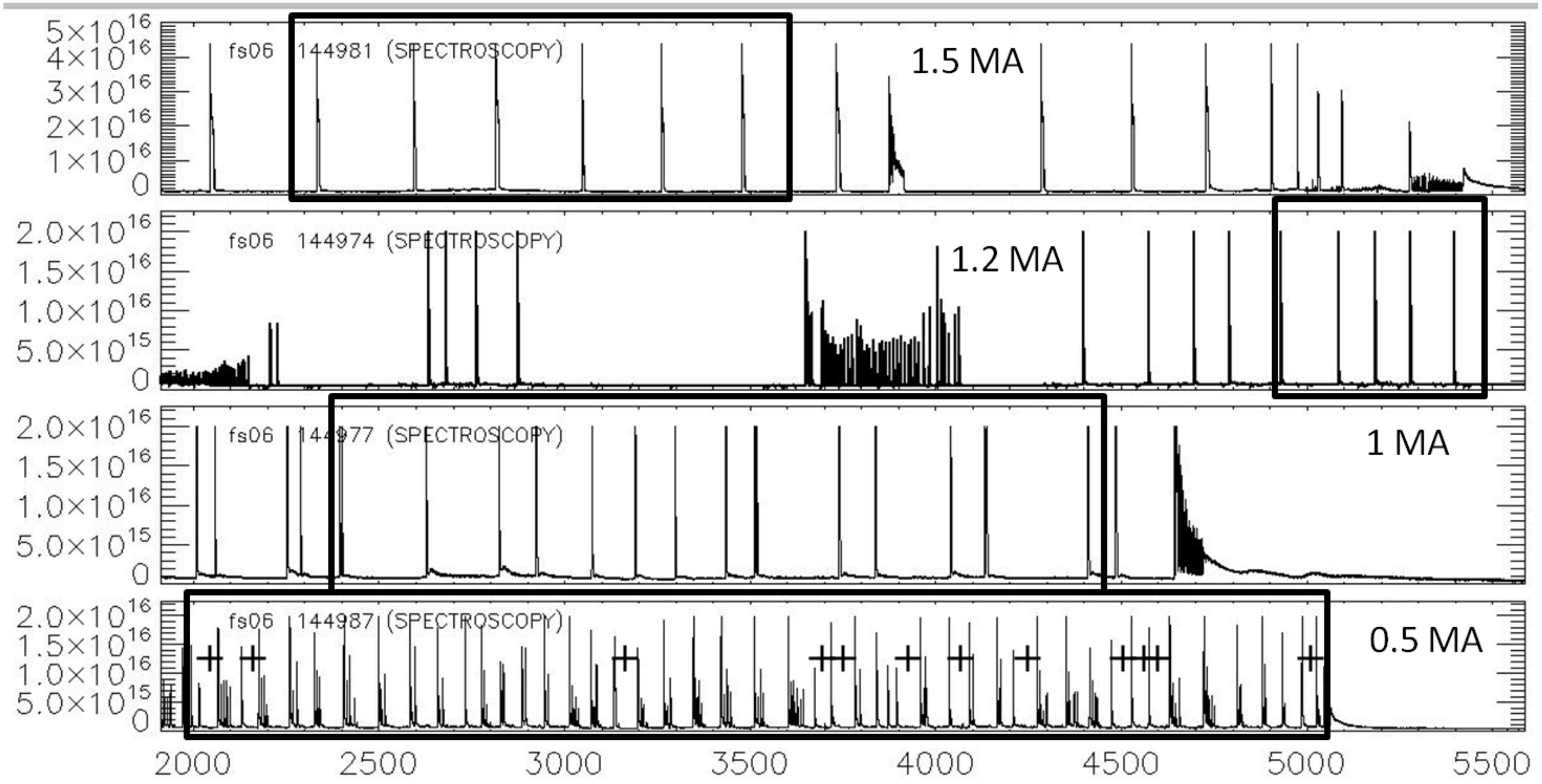


Figure 14: ELMs in the Four Shots - The divertor D_α signal (photons/s) used to identify ELMs for the four shots between 2 and 5.5 s. The plots are arranged by current in decreasing order. The areas outside the time selection (denoted with a black rectangle) did not exhibit quasi-steady-state operation during the shot. The inter-ELM periods selected for analysis from the 0.5 MA shot quasi-steady-state period are denoted with a "+", and in the other shots, the entire set of consecutive, whole inter-ELM periods lying completely within the boxes were analyzed.

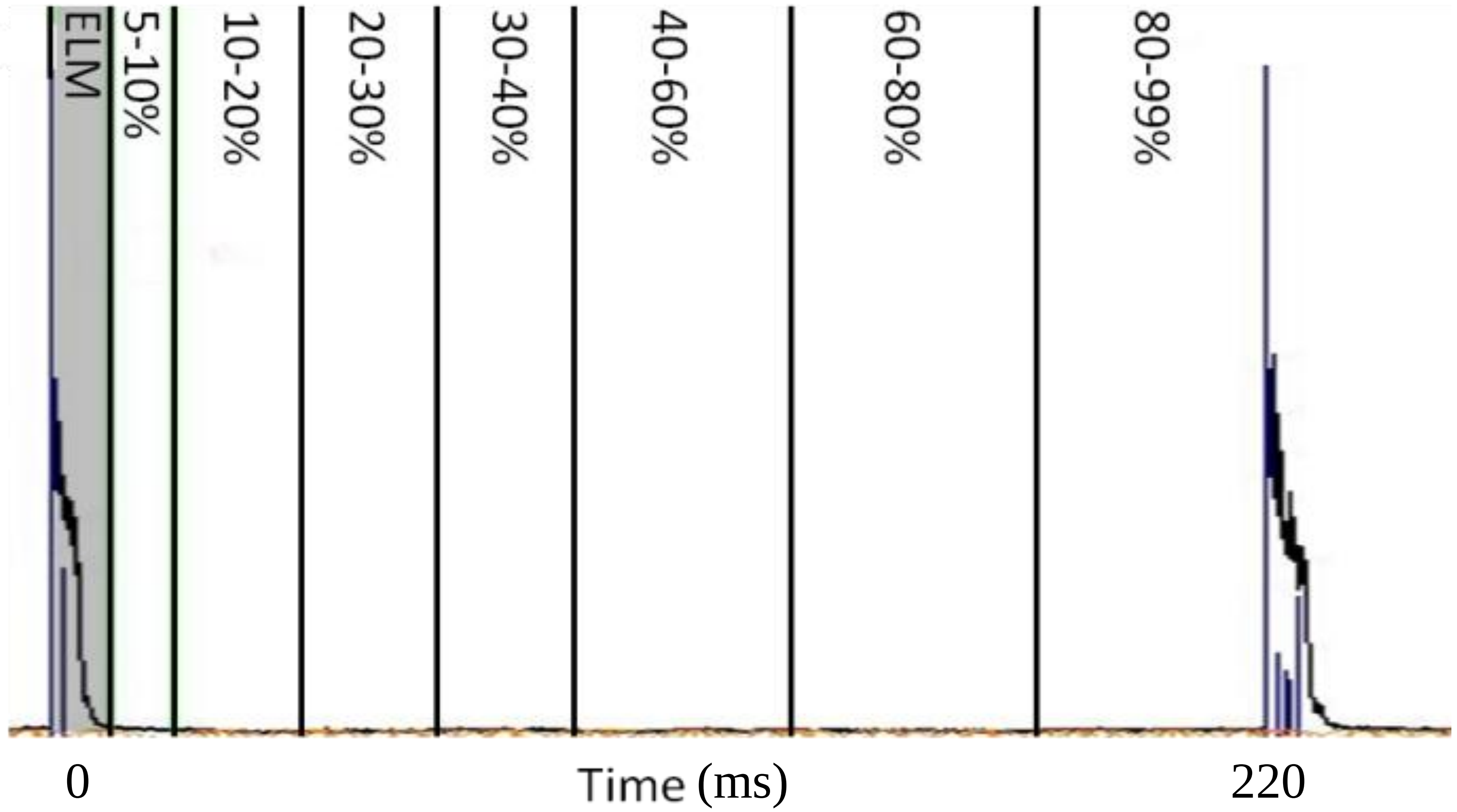


Figure 15: Sample Inter-ELM Period - A sample inter-ELM period for the 1.5 MA shot. The shaded and excluded area at the beginning of the shot is considered to be part of the ELM event (0-5% of the inter-ELM period). This illustrates how the intervals begin for every inter-ELM period after the duration of the longest leading ELM event of the shot. For this inter-ELM period, the ELM event seems to end before the excluded area ends. The first interval to the right of the ELM event has the minimum interval width for this shot (5%), and spans from 5% to 10% of the inter-ELM period. The subsequent intervals represent 10-20%, 20-30%, 30-40%, 40-60%, 60-80%, and 80-99% of the inter-ELM period. This division is repeated for each inter-ELM period selected for analysis, and the intervals are then combined into composite inter-ELM periods for the shots.

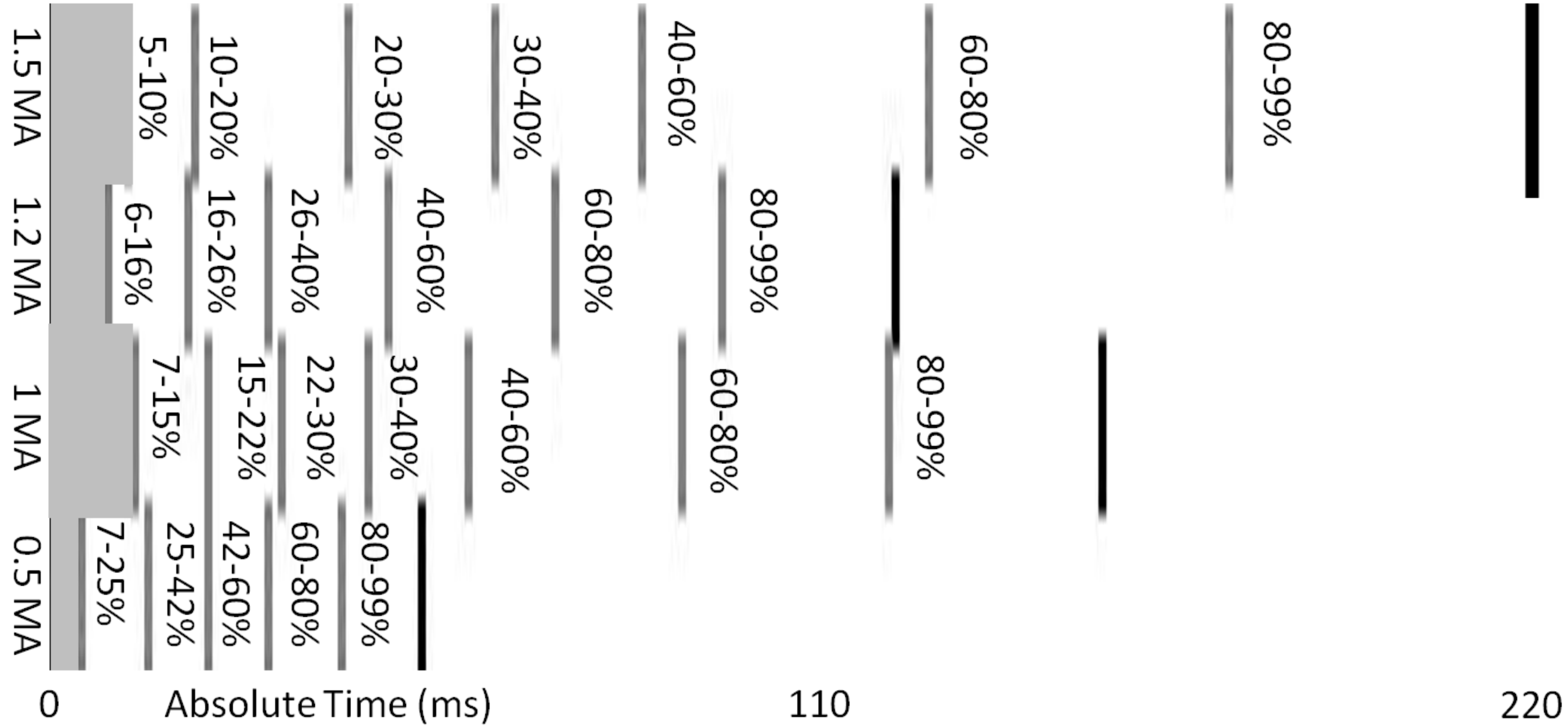


Figure 16: Time Interval Comparison - The intervals of the composite inter-ELM period for each shot, shown in absolute time. The shaded area represents the length of the widest leading ELM event of the inter-ELM periods in the shot, while the black lines represent the end of the composite inter-ELM period for each shot. It is notable that several of the intervals almost fully overlap in absolute time for all four shots, such as the 5-10%, 6-16%, 7-15%, and 25-42% intervals of the 1.5, 1.2, 1, and 0.5 MA shots, respectively. This feature could be useful for future analyses.

CHAPTER 7

EXPERIMENTAL DIII-D DATA: MEASURED AND EXPERIMENTALLY INFERRED VALUES

The ion quantities measured with the CER system on DIII-D include the ion temperatures, densities and poloidal and toroidal rotation velocities [25, 28], while the Thomson laser system on DIII-D measures electron densities and temperatures [29]. These diagnostics systems are reviewed in Chapter 5. The experimentally derived plasma parameter profiles will be shown in this chapter for the four shots.

The electron density profiles are shown in Figure 18, and display the familiar "edge pedestal" structure associated with H-mode operation. This plot is important to understanding the different regions of the plasma edge that will be referenced throughout the paper. The structures of the electron density profiles that exist in the edge will be labeled, and used to refer to the corresponding area of the edge region when discussing various parameters. This is somewhat complicated by the fact that these regions shift in the edge both throughout the inter-ELM period and across different shots. However, the labeled regions of the shots are present in every density profile taken during H-mode operation, despite their variation in radial location and width. These labels are displayed for a single electron density profile in Figure 20. The "flattop region" is the inner part of the edge, where the density is relatively constant ($\rho < 0.9125$ in Figure 20). The "top of the pedestal" is the area where the densities begin to decrease ($0.9125 < \rho < 0.945$ in Figure 20), and the "sharp gradient region"/"pedestal" is the area between the flattop region and the separatrix ($\rho > 0.945$ in Figure 20, where $\rho = 1.0$ is the separatrix). It is important to note that for many of the interpreted quantities, no data is available at $\rho = 1.0$ due to the numerical methods used to calculate these quantities, and the profiles end at ρ = 0.994. However, the measured quantities have profile fits that extend throughout the edge.

The electron density appears to increase monotonically in the flattop region as the pedestal is rebuilt in Figure 18. However, it is important to note that the time differences between the profiles in each shot are not constant, and in reality, the electron density pedestal height, and the corresponding flattop region value, change nonlinearly with time. The inter-ELM evolution is fastest immediately after an ELM (the first interval), and as the inter-ELM period continues, the evolution gradually slows as it approaches an asymptotic value (the value before the subsequent ELM, represented by the 80-99%, or last, interval profile). Although the inter-ELM evolution increases the pedestal and flattop region height, and the width of the pedestal in the electron density profile, different profile structures present in the other profiles will be altered by the evolution in different ways. However, the chronologically-nonlinear property of inter-ELM profile evolution is widely observed in the evolutions of the other plasma profiles.

For the edge electron density profiles, the maximum density seen in each interval is found in the flattop region, and this value increases during the inter-ELM period, reaching a global maximum at the end of the inter-ELM period in the last, asymptotic, 80-99% interval profile. The electron density profiles also increase as current increases, reflecting the improvement in plasma confinement associated with higher currents.

Early in the inter-ELM period, while the density increases in the flattop region, it is actually decreasing in the sharp gradient region, a trend that is most apparent at lower currents. A very small decrease with evolution is found in the 1.5 MA electron density sharp gradient region during the first 30% of the inter-ELM period, but in the 0.5 MA shot, a decrease of nearly a third is found (at ρ ~ 0.96) in the same time period. The trend is also apparent in the deuterium density profiles (shown in Figure 19), and together, these trends strongly imply inward transport for both particles in the early part of the inter-ELM period.

As the inter-ELM period continues, the density value at the top of the pedestal increases in value and moves inward as it is rebuilt. This raises the flattop region plateau, sharpens the edge gradients, and widens the pedestal.

The impurity fraction (% of ions) as a function of radius is shown in Figure 17. For this analysis, it is assumed that the edge impurity ions are all carbon, which simplifies the interpretation of transport variables. This is a reliable assumption for the DIII-D tokamak, as the great majority of the plasma-facing components (PFCs), and the divertor, are coated in graphite[33]. The impurity fraction appears to increase slightly during the inter-ELM period. The more striking trend is the degradation of the gradient in the sharp gradient region with decreasing current (and decreasing confinement). The lower current, lower confinement, and increased transport allow the carbon impurities to be more easily exhausted from the plasma.

The deuterium density profiles shown in Figure 19 are calculated from the electron density, and the impurity fraction profile, derived from the fitting discussed in Appendix A, and shown in Figure 17. They appear similar to the electron density profiles, but have a lower magnitude. This is due to the fact that impurities in the plasma have multiple electrons, while the deuterium main ions only have one. The deuterium density profiles evolve in time and with current in similar ways to the electron density profiles, and also show the aforementioned density decrease in the edge pedestal region.

The electron temperature profiles in Figure 21 also show a non-linear growth rate that is largest early in the inter-ELM period. The profile structures and evolutions are similar to those of the electron density, but no corresponding decrease in temperature in the sharp gradient region is found. Also, no clear correlation between maximum temperatures and plasma current is apparent in the electron temperatures.

The ion temperature profiles are shown in Figure 22. These profiles are the measured carbon temperatures, and the deuterium ions are assumed to have the same temperature profiles. The cliff-like pedestal shape of the electron density profiles is not found in the ion temperature profiles, where the “edge pedestal” is much less defined. Instead, the profile gradually decreases throughout the edge, with a slight change in slope indicating the presence of an ion temperature “pedestal” (e.g. $\rho \sim 0.91$ in the 1 MA shot). The values of the ion temperature generally increase across the edge during the inter-ELM period, but the profile evolutions vary with current across a large part of the edge. In the 1.5 MA shot, temperatures near the separatrix rise faster through the inter-ELM period than those at smaller radii. However, in the 0.5 MA shot, the temperatures near the separatrix actually decrease in the first part of the shot, finally rising in the latter half of the inter-ELM period. In the 1 MA shot, the midpoint of the current scan shots, the profiles evolution behavior appears to fall in between the high and low current shots. This shows that the temperature profile rebuilding process between ELMs is sensitive to the magnitude of the current. This variation in inter-ELM evolution is the main effect on the temperature profiles of a changing current magnitude, as no other clear trends can be identified.

The carbon poloidal rotation velocities for the four shots are shown in Figure 23. As shown in Figure 2, a positive value in Figure 23 implies a downward velocity at the outboard midplane. The carbon poloidal rotation velocity profiles form a negative well in the sharp (density) gradient region, and the well slightly decreases in width throughout the inter-ELM period. The characteristics of this well structure change substantially across the current scan. In the 1.5 MA shot, the minimum of the relatively shallow velocity well is around -7.5 km/s, increasing sharply to positive values near the separatrix. In the 0.5 MA shot, the well is much deeper and narrower, reaching a well minimum of -20 km/s in the 0.5 MA shot. The major changes with current seem to be localized on the inside wall of the negative well, where the gradient steepens with higher current, forming a more defined well. The “well wall” near the separatrix, remains steep throughout the current scan, but its maximum value at the separatrix decreases somewhat with decreasing current.

These velocity evolutions exhibit a slight “overshoot” behavior, in which extreme profile values are reached midway between ELMs. Rather than approaching the “asymptotic”, 80-99% profile as the inter-ELM period progresses, when the overshoot behavior is seen, the earlier profiles surpass the asymptotic profile, then relax back to it as the inter-ELM period ends. This feature is well illustrated in all the plots in Figure 23, as the “asymptotic” pre-ELM (80-99%) interval profile is rarely at an extreme. Similar overshoot behavior is found in many other plasma parameters, and is most often localized in the sharp gradient region.

The carbon toroidal rotation velocities, shown in Figure 24, decrease monotonically with radius throughout the edge. Between ELMs, the profiles generally increase, and no clear trends in magnitude with current are found.

The CER-measured carbon density, temperature, and toroidal and poloidal rotation velocities are used (along with magnetic field strengths) to calculate the radial electric field [25] from the equilibrium carbon radial force balance equation.

$$E_r^{exp} = \frac{1}{-n_k^{meas} e_k} \frac{\partial p_k^{meas}}{\partial r} + B_\theta V_{\phi k}^{meas} - B_\phi V_{\theta k}^{meas} \qquad (2)$$

The radial electric field profiles for the four shots are shown in Figure 25.

All three components of Eq. (2) have a strong effect on the radial electric field. The carbon toroidal rotation velocity causes the electric field to be positive in the inner edge. However, this velocity declines with radius, and in the sharp gradient region, the negative combination of the pressure gradient and the carbon poloidal rotation velocity well structure overcome it to create a well structure in the electric field profile. Near the separatrix, the carbon poloidal rotation velocity attains more positive values, which is reflected in the local increase of the radial electric field. However, the pressure gradient term remains large and negative near the separatrix (due in part to the carbon density in the denominator), and limits the return of the electric field to more positive values. Generally, the negative well structure in the electric field profile deepens throughout the inter-ELM period, and also exhibits overshoot behavior. The location of the bottom of the negative well shifts inward through the inter-ELM period, a trend most clearly seen in the 1 MA shot, and one which is largely driven by the simultaneous increase in density pedestal width. The radial electric field becomes more negative across the edge with decreasing current, a trend correlated with a larger, negative pressure gradient term (due to smaller densities and a wider pedestal) and decreasing negative poloidal rotation velocity values.

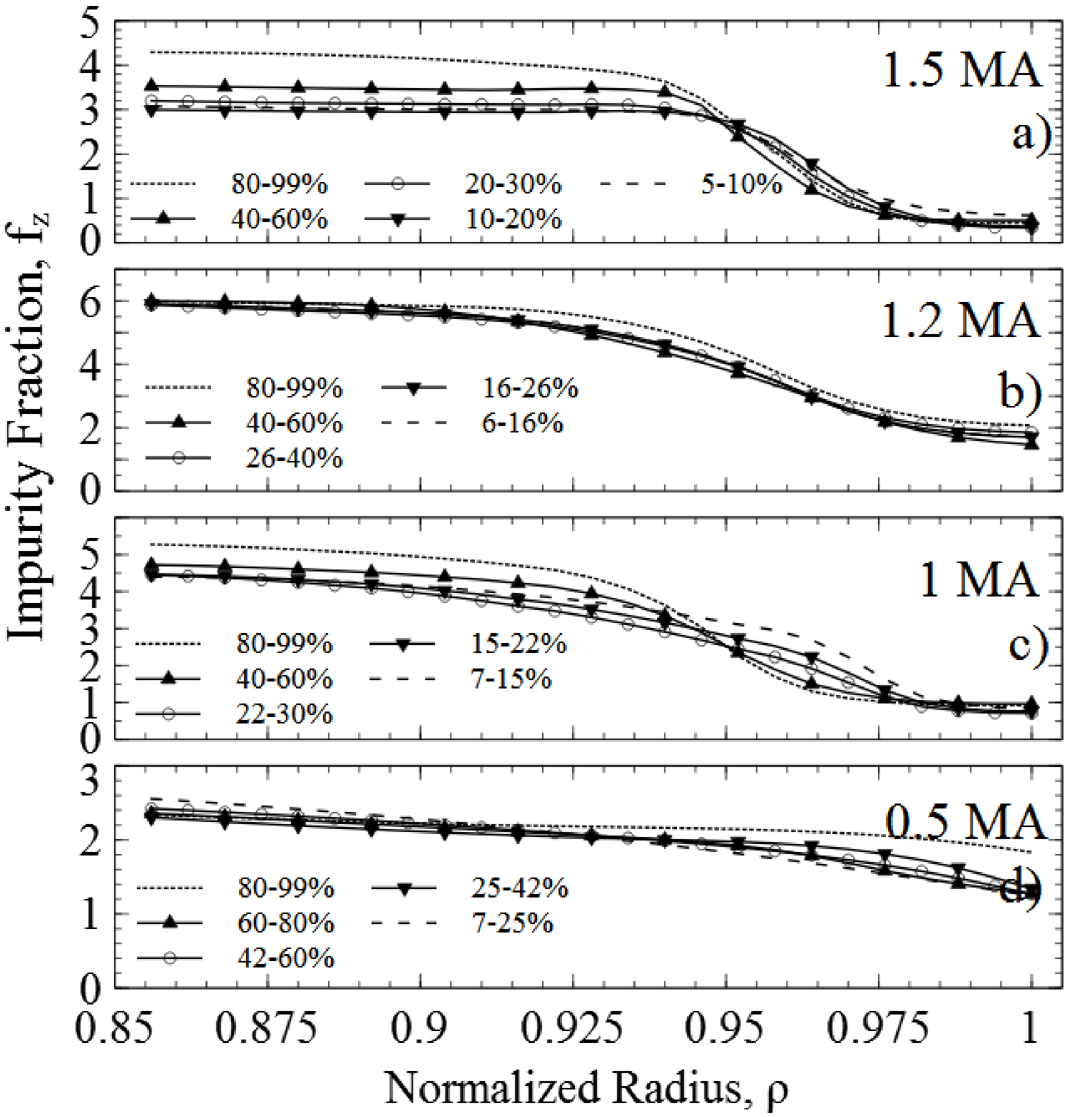


Figure 17: Impurity Fraction - The fraction of the plasma ions that are impurities.

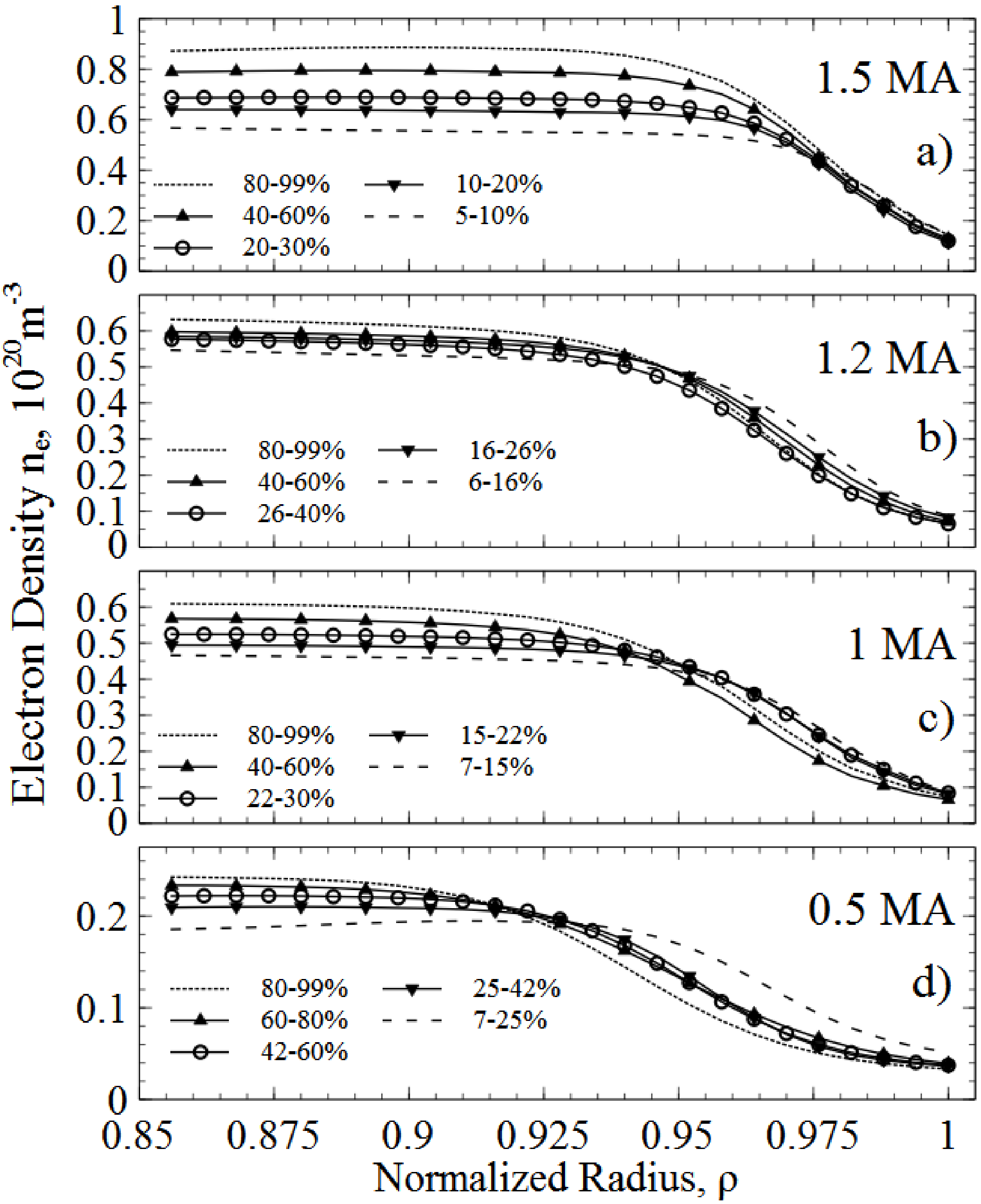


Figure 18: Electron Density - The measured electron density profiles. For visual clarity, the **Y AXES MAY DIFFER IN SCALE THROUGHOUT THE DOCUMENT**.

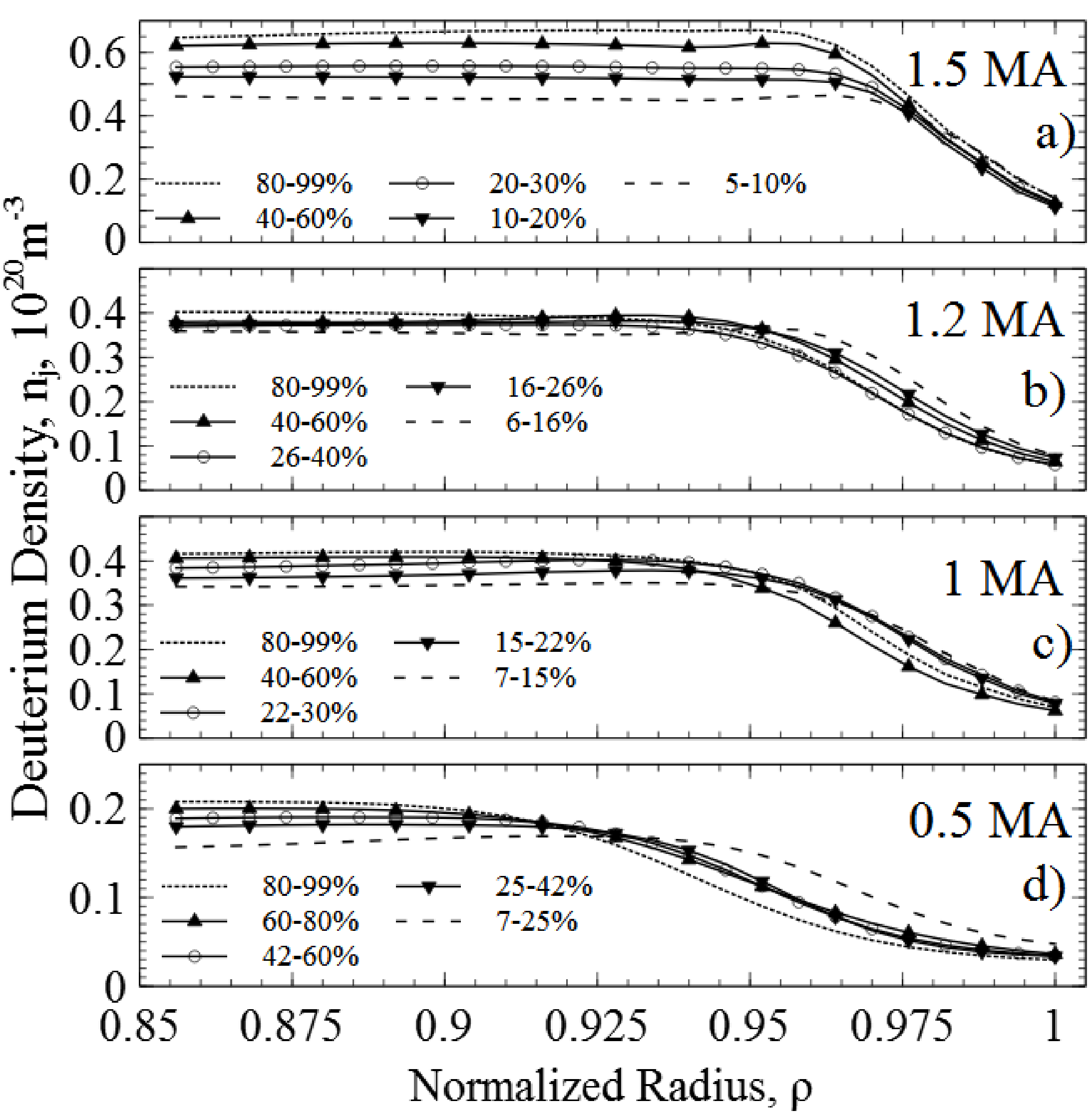


Figure 19: Deuterium Density - The deuterium density profiles. For visual clarity, the <u>**Y AXES MAY DIFFER IN SCALE THROUGHOUT THE DOCUMENT**</u>.

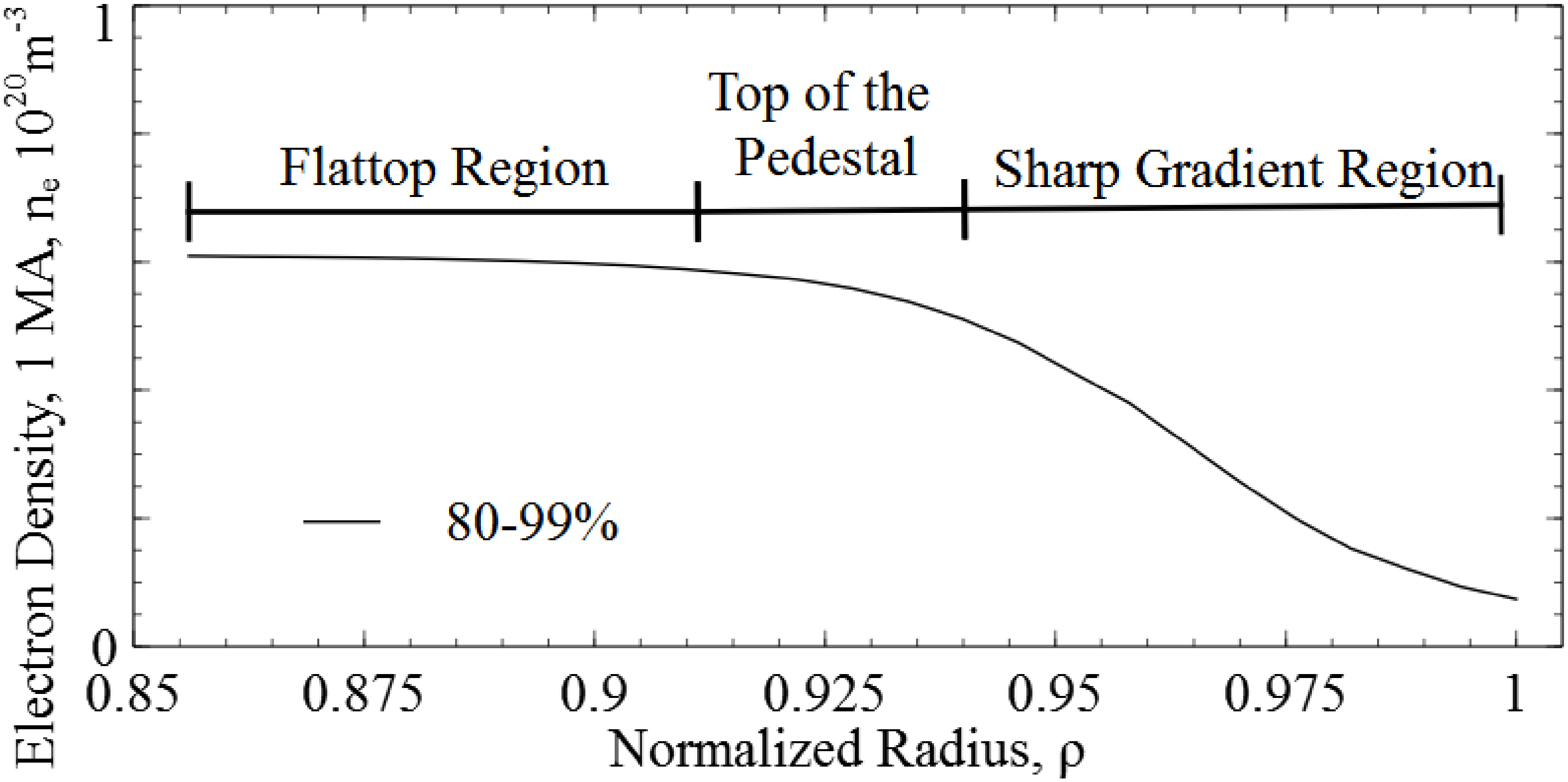


Figure 20: Regions of the Edge - The general locations in the edge identified by structures in the electron density profile. The profile shown in this figure is the 80-99% interval of the 1 MA shot. The other profiles for the shots have similar structures, but the radial location of these structures vary depending on shot current, chronological location in the inter-ELM period of the interval, and other parameters.

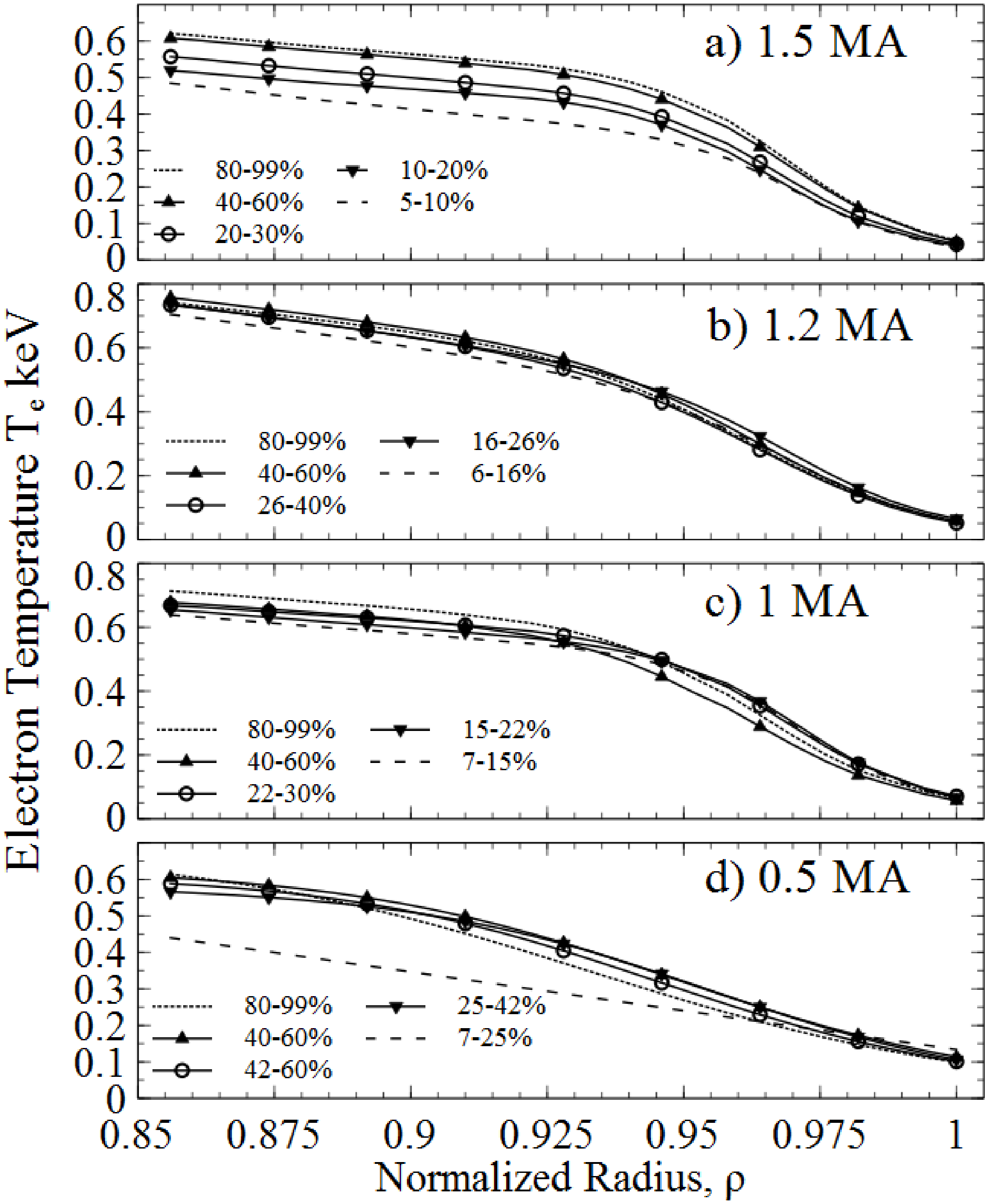


Figure 21: Electron Temperature - The measured electron temperature profile evolution.

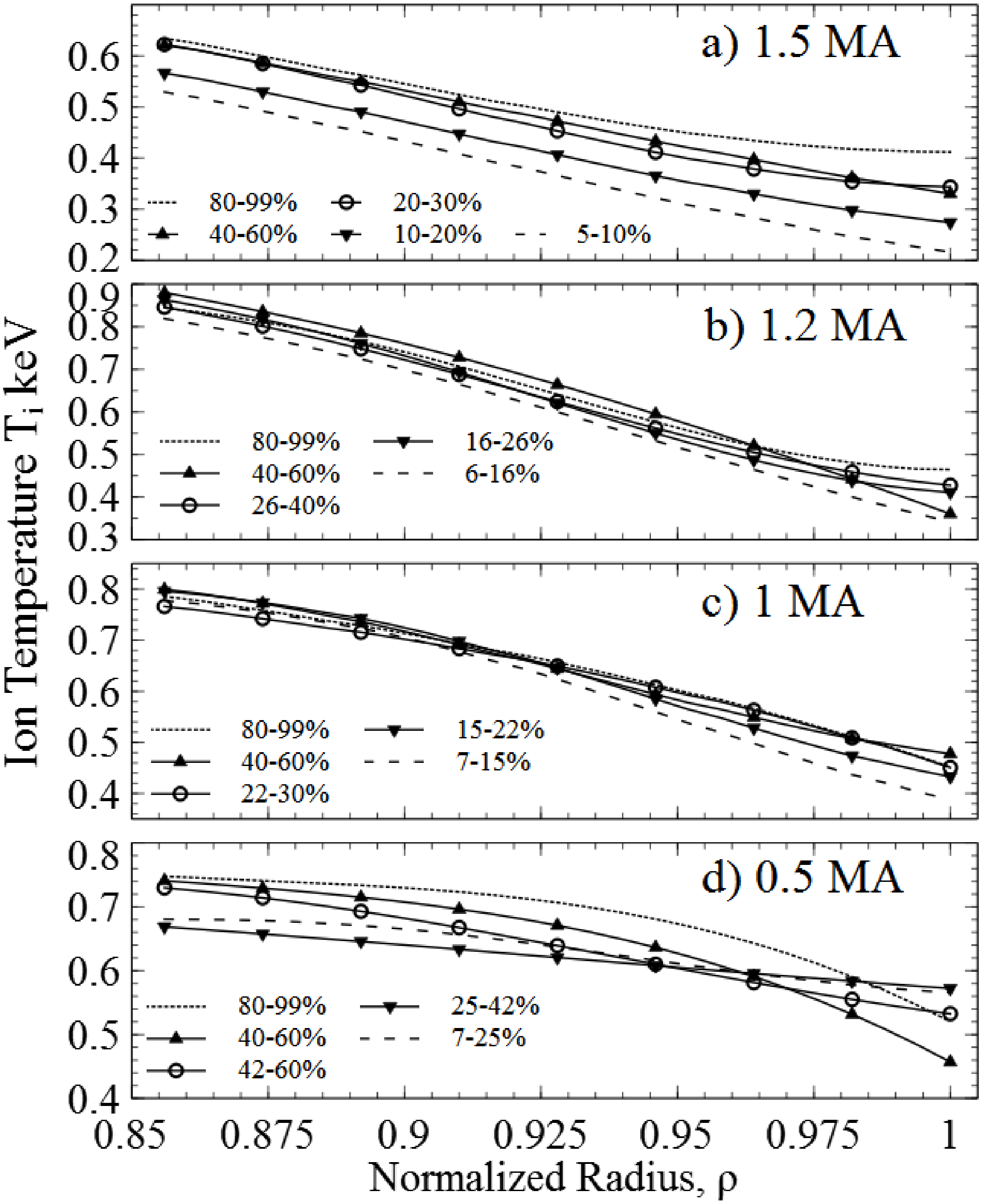


Figure 22: Ion Temperature - The measured ion temperature profile evolution.

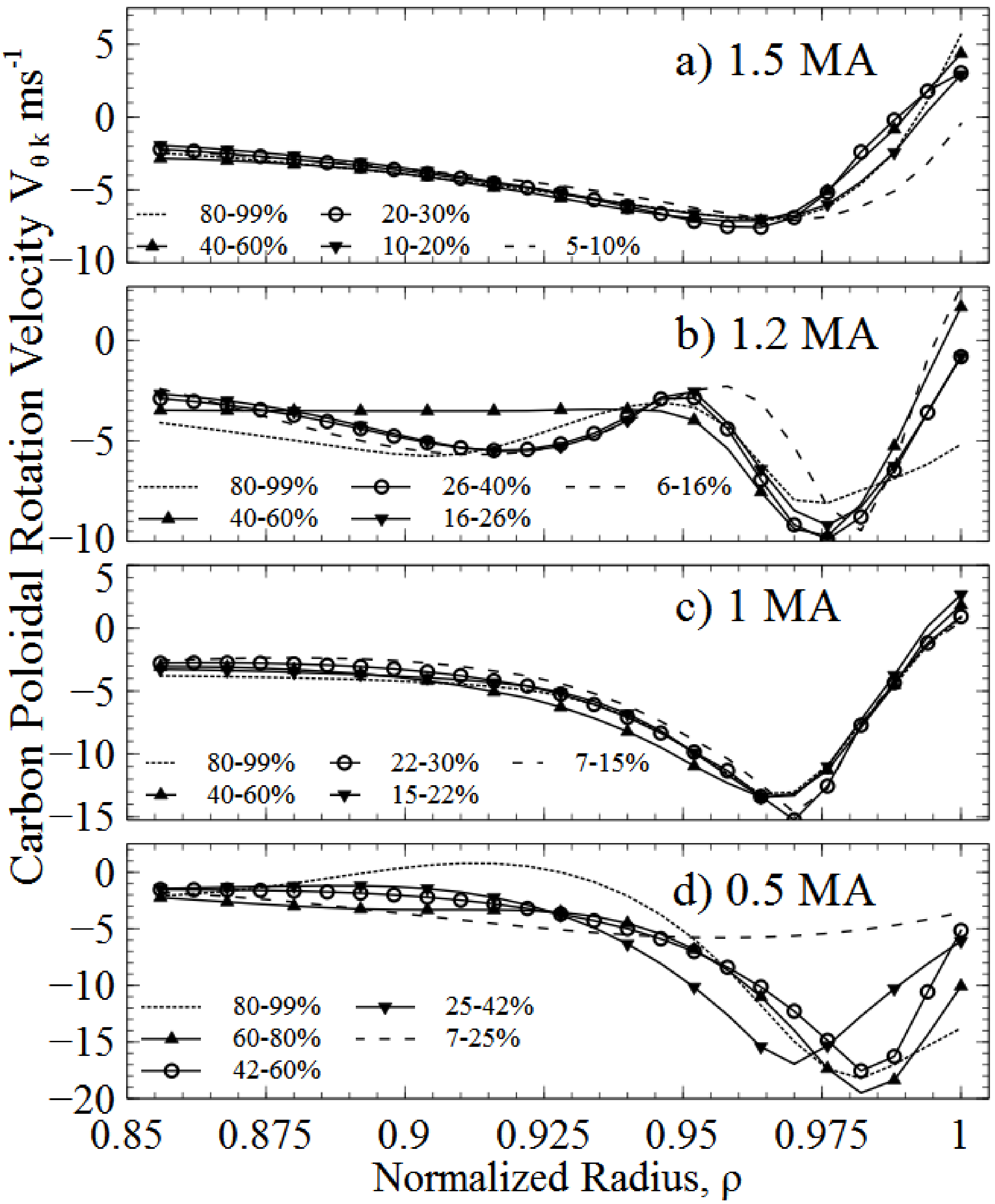


Figure 23: Carbon Poloidal Rotation Velocity - The measured carbon poloidal rotation velocity profiles.

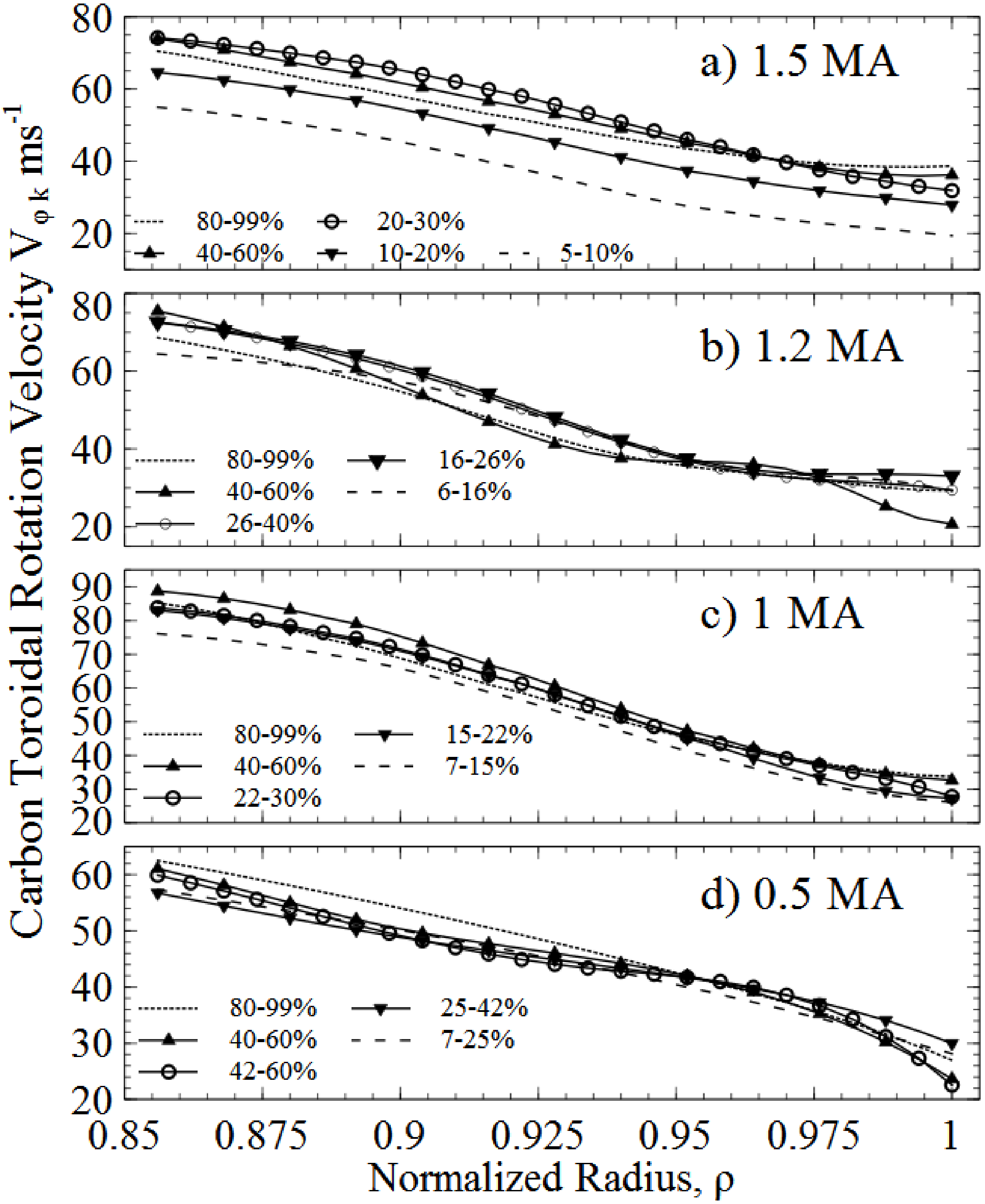


Figure 24: Carbon Toroidal Rotation Velocity - The measured carbon toroidal rotation velocities

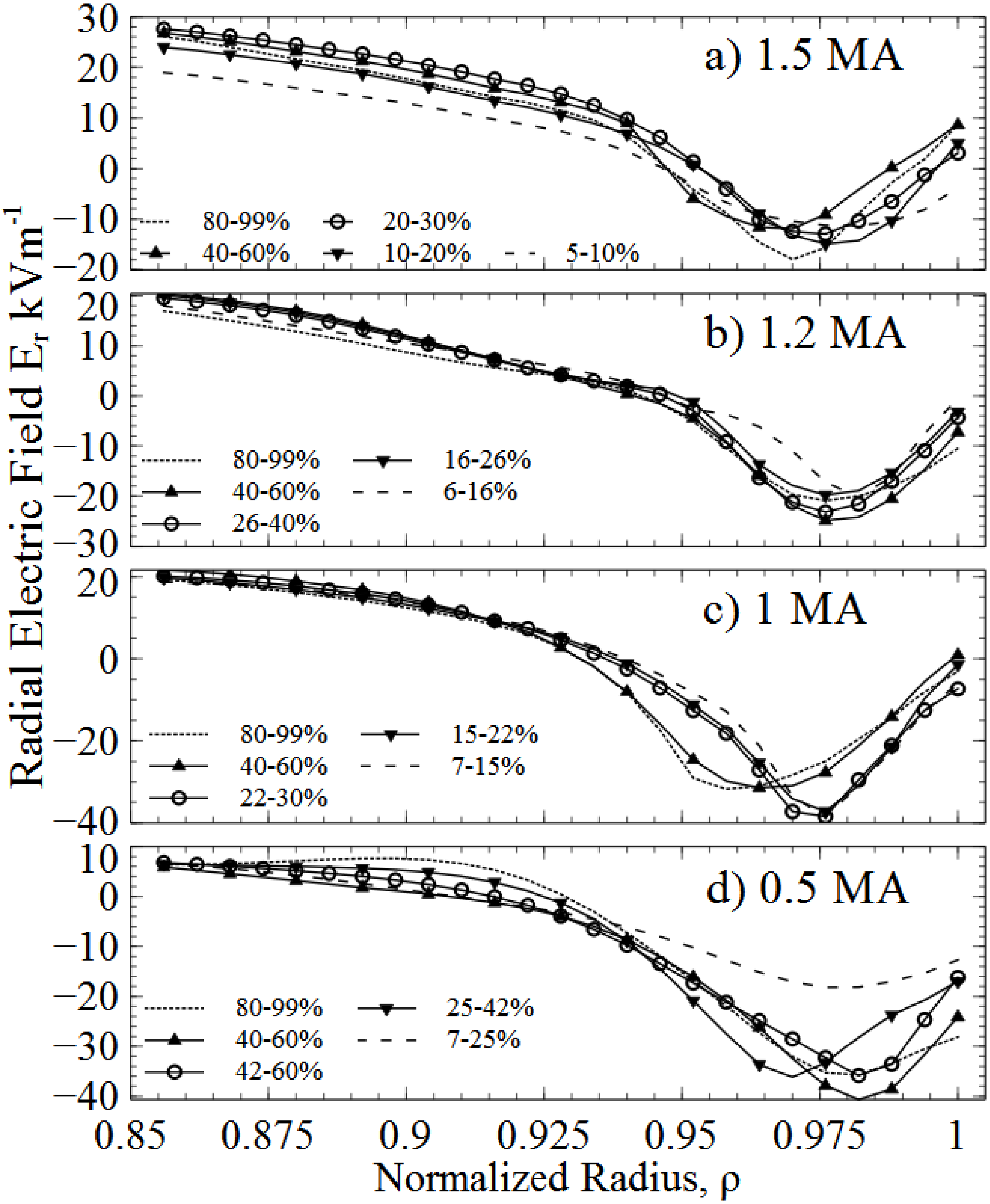


Figure 25: Experimental Radial Electric Field.

CHAPTER 8

THEORETICAL FRAMEWORK: PARTICLE, MOMENTUM, AND FORCE BALANCE, THE PINCH-DIFFUSION RELATION, AND ION ORBIT LOSS

The pinch-diffusion relation model[34] used in GTEDGE[30-32] is built on MHD fluid theory particle, momentum, energy, and force balance equations that are well known in plasma physics, and that govern the plasma. In this chapter, the toroidal and radial components of the momentum balance equation are combined to derive the pinch-diffusion equation for the radial ion flux and to prescribe the diffusion coefficient and electromagnetic pinch velocity that are required by momentum balance. It is important to note that in the edge plasma, some of the more energetic thermalized ions can access orbits that exit the plasma[35], and instantaneously be lost from it. To take these losses into account, we present an ion orbit loss correction[36, 37] to the radial ion flux flowing in the plasma. In the following section we will use the experimental data of the previous section to evaluate the particle diffusion theory and pinch velocities prescribed by momentum conservation, and calculate the diffusive and non-diffusive radial particle fluxes in the edge region.

8.1 Radial Ion Particle Transport

During the ELM event, there is a large flow of plasma in the edge directed radially outwards towards the first wall, much of which gets swept into the divertor. As the pedestal rebuilds immediately after the ELM event, oppositely directed inward flows of ions are necessary to replace the ions lost through the ELM and rebuild the pedestal structure. There are several fundamental transport mechanisms that are involved in this rebuilding of the edge pedestal throughout the inter-ELM period. For the ions, momentum conservation requires that the following forces remain in balance: (i) toroidal angular momentum exchange with other ions and neutral atoms via collisions, (ii) viscous and inertial torques exerted in the plasma due to ion bulk particle flows, (iii) electromagnetic (VxB, electric field) forces, (iv) thermodynamic forces (pressure gradient), and (v) external forces (e.g. momentum exchange with neutral beam particles). The loss of particles, energy, and momentum by ions that access unconfined trajectories, or loss orbits, and (nearly) instantaneously exit the plasma across the separatrix, must also be taken into account.

The collisional momentum exchange frequency between main deuterium plasma ions (denoted by subscript j) and the carbon impurity ions (subscript k) is represented by the collision frequency ν_{jk}, and can be calculated from measured data. Here, other types of toroidal momentum transfer in the radial direction, including charge exchange and the outward momentum flow due to viscous and inertial torques, are represented by the composite radial transport frequency of toroidal angular momentum, ν_{dj}. This composite frequency will be referred to as the drag frequency, and its magnitude will be interpreted from toroidal rotation measurements[38]. The electromagnetic VxB forces, the electrostatic E_r and E_ϕ forces, and the external beam momentum input $M_{\phi j}$ are represented explicitly in the ion momentum balance equation, the radial and toroidal components [39] of which are shown in Eqs. (3) and (4).

$$V_{\phi j} = \frac{1}{B_\theta}\left(E_r + V_{\theta j} B_\phi - \frac{1}{n_j e_j}\frac{\partial p_j}{\partial r} \right) \tag{3}$$

$$B_\theta e_j \Gamma_{rj} = n_j m_j \left(\nu_{dj} + \nu_{jk} \right) V_{\phi j} - n_j m_j \nu_{jk} V_{\phi k} - M_{\phi j} + n_j e_j E_\phi^A \tag{4}$$

In these equations, V refers to particle fluid velocity, electric fields are denoted with an E and magnetic fields with a B. Density and charge are represented by n and e, and the momentum exchange (ν) frequencies have already been defined. Similar equations can be obtained for the impurity ion species by interchanging the “k” and “j”.

The ion radial particle flux in the plasma exerts a torque that produces toroidal and poloidal flows of the plasma ions. For a two-species plasma model, the radial and toroidal momentum balance equation components (Equations (3) and (4)) can be solved [39] for the radial deuterium particle flux Γ_{rj}.

$$\Gamma_{rj} = -\frac{n_j D_j}{p_j}\frac{\partial p_j}{\partial r} + n_j V_{rj}^{pinch} = n_j (D_j L_{pj}^{-1} + V_{rj}^{pinch}); \qquad L_{pj}^{-1} = \frac{-dp_j / dr}{p_j} \tag{5}$$

The collection of terms D_j

$$D_j \equiv \frac{m_j T_j \nu_{jk}}{\left(e_j B_\theta\right)^2}\left(1 + \frac{\nu_{dj}}{\nu_{jk}} - \frac{e_j}{e_k}\right) \tag{6}$$

is a momentum-conserving definition of the diffusion coefficient. The second, non-diffusive term in Eq. (5) has a convective form, with the electromagnetic and external forces all collected in the “pinch velocity” V_{rj}^{pinch} .

$$V_{rj}^{pinch} = -\frac{M_{\phi j}}{n_j e_j B_\theta} - \frac{E_\phi^A}{B_\theta} + \frac{m_j\left(\nu_{dj} + \nu_{jk}\right)E_r}{e_j B_\theta^2} + \frac{m_j \nu_{jk} V_{\phi k}}{e_j B_\theta} + \frac{m_j\left(\nu_{dj} + \nu_{jk}\right)V_{\theta j} B_\phi}{e_j B_\theta^2} \tag{7}$$

The magnitude of the radial particle flux can be determined by solving the continuity equations (using the calculated recycling neutral source and neutral beam contributions).

$$\frac{\partial \Gamma_j}{\partial r} = -\frac{\partial n_j}{\partial t} + n_e n_0 \sigma \upsilon_{ion} + S_{nb} = -\frac{\partial n_j}{\partial t} + S_{nj} \tag{8}$$

The neutral beam source in this equation is denoted by S_{nb}, while the neutral ionization rate is given by the expression $n_e n_0 \langle\sigma v\rangle_{ion}$. The total source of deuterium ions is denoted with S_{nj}.

8.2 Ion Orbit Loss Effects

Another important mechanism in edge transport is the instantaneous escape of ions that access loss orbits and immediately leave the plasma. This ion orbit loss (IOL) reduces the particles, energy, and momentum in the plasma; a return current of ions from the scrape-off layer (SOL) is required to balance the charge loss [40] and maintain macroscopic plasma neutrality. Both the ion orbit loss and the return current are taken into account using a numerical model[35, 41] which calculates the minimum energy required for a particle at a given location and with a given velocity to access a possible loss orbit and escape confinement. Using the conservation relations in Ref. [41], minimum escape energies [$\varepsilon_{min}(\zeta_0) = 0.5mV_{0,min}^2(\zeta_0)/kT$] are calculated numerically, and expressions for the cumulative IOL-driven particle, momentum, and energy loss fractions in a given plasma at any radial location are evaluated.

$$F_{orb} = \frac{R_{loss}^{iol} \int_{-1}^{1} \Gamma\left(3/2, \varepsilon_{min(\zeta_0)}\right) d\zeta_0}{2\Gamma(3/2)} \tag{9}$$

$$M_{orb} = \frac{R_{loss}^{iol} \int_{-1}^{1} \zeta_0 \Gamma\left(2, \varepsilon_{min(\zeta_0)}\right) d\zeta_0}{2\Gamma(2)} \tag{10}$$

$$E_{orb} = \frac{R_{loss}^{iol} \int_{-1}^{1} \Gamma\left(5/2, \varepsilon_{min(\zeta_0)}\right) d\zeta_0}{2\Gamma(5/2)} \tag{11}$$

Here, $F_{orb}(\rho)$, $M_{orb}(\rho)$, and $E_{orb}(\rho)$ are the radially cumulative loss fractions for ions, momentum, and energy, respectively, that escape across the separatrix, calculated from conservation of energy, magnetic moment, and canonical angular momentum [41]. The functions Γ(n) and Γ(n,x) are the gamma and incomplete gamma functions, respectively, while ζ_0 denotes the directional cosine of the particle velocity with respect to the magnetic field.

R_{loss}^{iol} is the fraction of particles crossing the separatrix due to IOL that do not return into the plasma. The model for IOL used in this analysis does not model the trajectory of the ion on an exit orbit after it leaves the plasma, and some of the plasma ions may leave the plasma through IOL, then return without colliding with a PFC. Due to this, known model limitations, and to be conservative, the assumption was made in this research that one half of the particles calculated to leave the plasma do not return and are actually lost, $R_{loss}^{iol} = 0.5$.

With this taken into account, the cumulative fraction of total ions lost, F_{orbl}, at each radial location is shown in Figure 26. These calculations take into account the measured variations of the radial electric field over the inter-ELM period and their effects on the ion orbit loss. The cumulative ion orbit loss fraction is small except in the outer ~5% of the plasma radius, where it increases exponentially to reach a value of 30-40%, implying that 30-40% of the ions at that radius have escaped the plasma. The F_{orb} has a slight inverse correlation with current, and changes very little with time over the inter-ELM interval.

The cumulative energy loss fraction E_{orb} profile evolutions are shown in Figure 27. The profiles are similar in structure, evolution, and current sensitivity to the ion loss fraction profiles, but the magnitudes are slightly greater. This is due to the fact that those ions with higher energies can escape more easily through IOL, and are preferentially lost. However, this effect is slight.

The value for the radial ion flux derived from the continuity equation (Eq.(8)) must be reduced by the ions which escape confinement through IOL, and further reduced by the compensating inward current of ions necessary to maintain plasma neutrality [40].

$$\hat{\Gamma}_{rj}(r)=\left[1-2F_{orbl}(r)\right]\Gamma_{rj}(r) \tag{12}$$

This reduced flux is the actual flux found in the plasma which drives many important transport quantities. Variables directly modified to account for IOL effects, such as the reduced radial ion flux, will be denoted with a carat. The radial ion flux profiles used in this analysis are calculated directly from the continuity equation and corrected for ion orbit loss, as indicated in Eq. (12). Figure 28 compares the calculated radial ion flux profiles just before and just after an ELM event for different values of R_{loss}^{iol}.

Figure 26 shows that a large fraction F_{orb} of the ions flowing radially outward in the plasma become free-streaming ions on orbits that escape the plasma in the outer radial few percent of the plasma. (The loss fraction depends on the fraction R_{loss}^{iol} of the ions exiting across the separatrix which do not return inward across the separatrix to rejoin the ions flowing outward in the plasma, a parameter set to 0.5 for this work.) Since these free-streaming ion-orbit-loss ions are not being transported in the plasma, they should be subtracted from the total radial particle flux calculated from the continuity equation before that radial flux is used to interpret particle transport in the plasma edge. Similar corrections are needed for the outward flux of energy and momentum by ion-orbit-loss. This prevents interpreted transport variables from being required to account for particle that have already left the plasma, and therefore, being inaccurate. For instance, if IOL was not taken into account, a peak in the radial ion flux would be expected to be found near the separatrix in the 1 MA shot, a location where ion density is very low. This overestimation of transport properties if IOL is neglected would also occur in numerous other transport descriptors shown in this document. Note that as a result of the conservative estimate of R_{loss}^{iol}, if the estimation were too low, the effect would be an overestimation of the outward radial ion flux.

In the higher current shots, the IOL effect is concentrated near the separatrix, but as the current decreases, IOL becomes somewhat larger further inward in the sharp gradient region. This increased effect in the sharp gradient region as current decreases is shown in Figure 26, and is most prominent in the 0.5 MA shot. Taking the return current (the return flow of ions into the plasma to replace the positive charge lost via IOL, and keep the plasma macroscopically neutral) into account increases the influence of even small levels of ion orbit loss. The variations in the radial ion flux profiles with changing R_{loss}^{iol}, shown in Figure 28 for the 1 MA shot, are representative of the dependence on IOL shown by the radial ion fluxes of the other shots.

Because ions are lost due to IOL, and because counter-current ions constitute a disproportionate share of the losses[41], the momentum loss (Eq. (10)) due to IOL is preferentially counter-current. This induces an intrinsic co-current rotation in the edge plasma, being especially influential where IOL is high. An expression for the net parallel counter-current momentum loss rate due to IOL using the momentum loss rate (Eq. (10)) [41] can be used to calculate the intrinsic co-rotation caused by IOL.

$$V_{\phi j}^{intrin} = \frac{2}{\pi^{0.5}} M_{orbl,j}(\rho) \sqrt{\frac{\left[2kT_j(\rho)\right]}{m_j}} \tag{13}$$

A similar equation defines the intrinsic IOL rotation velocity associated with the carbon impurity by exchanging the “j” subscripts with “k” subscripts. Because of explicit mass dependence in Eq. (13), and the implicit mass dependence in M_{orb}, the intrinsic rotation velocity of carbon is nearly an order of magnitude smaller than for deuterium.

The momentum balance Equations (3) and (4), take collisional, viscous, inertial, external, and electromagnetic forces into account explicitly, but do not explicitly account for ion orbit loss of angular momentum. Components of these momentum balance equation are used to infer experimental momentum transfer rates, which have a large influence on the interpretation of edge transport. For consistency, in the transport calculations utilizing the momentum balance equations, when “experimental” carbon and deuterium rotation velocities are either measured or constructed from experimental data, they must be corrected for IOL effects. To do so, the carbon toroidal intrinsic IOL rotation velocity is subtracted from the measured carbon toroidal rotation velocity.

$$\hat{V}_{\phi k} = V_{\phi k}^{exp} - V_{\phi k}^{intrinsic} \tag{14}$$

Similarly, the deuterium intrinsic IOL toroidal rotation velocity $V_{\phi j}^{intrin}$ must be subtracted from the constructed deuterium toroidal velocity.

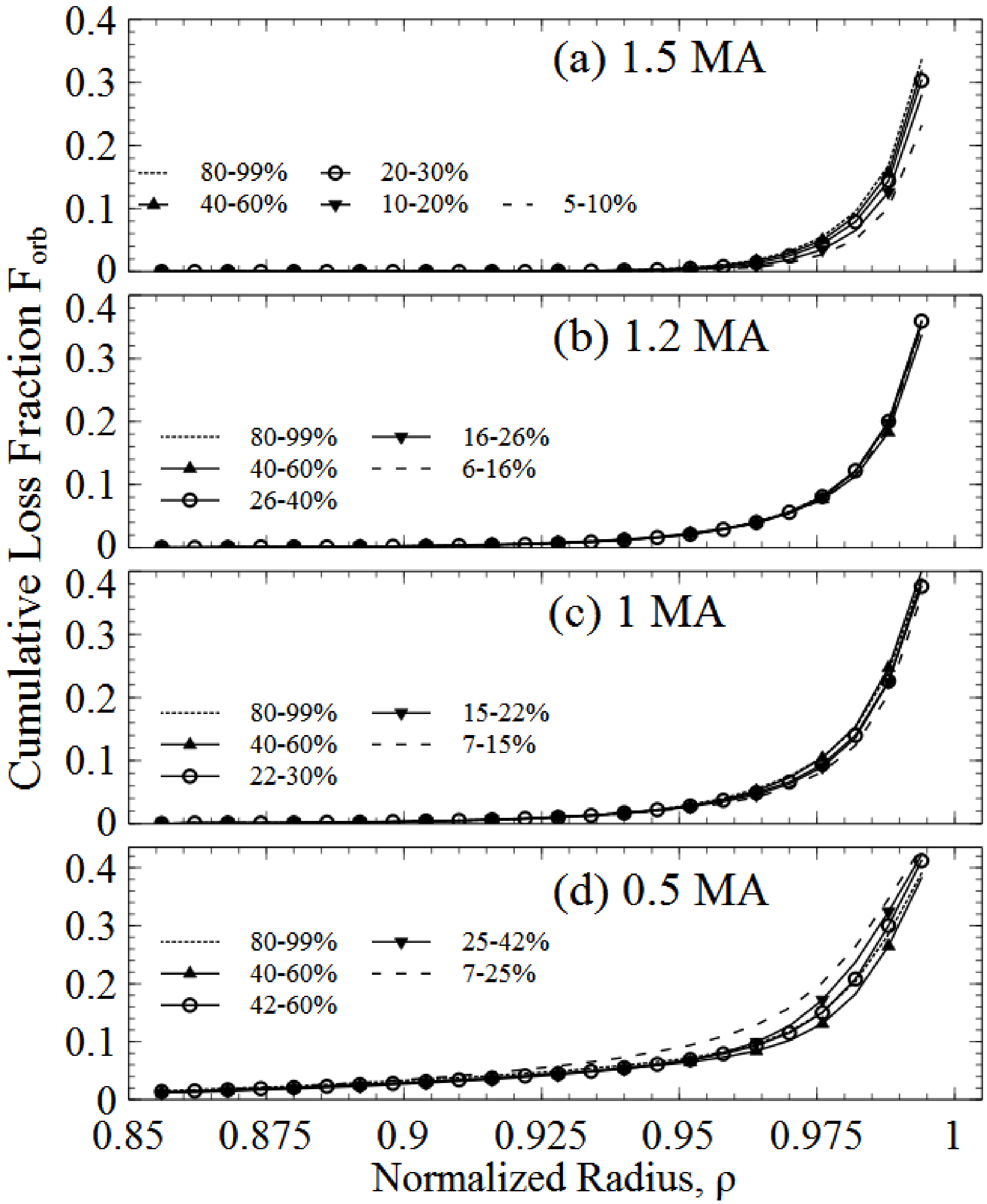


Figure 26: Ion Loss Fraction - The cumulative ion loss fraction as a function of radius with $R_{loss}^{iol} = 0.5$.

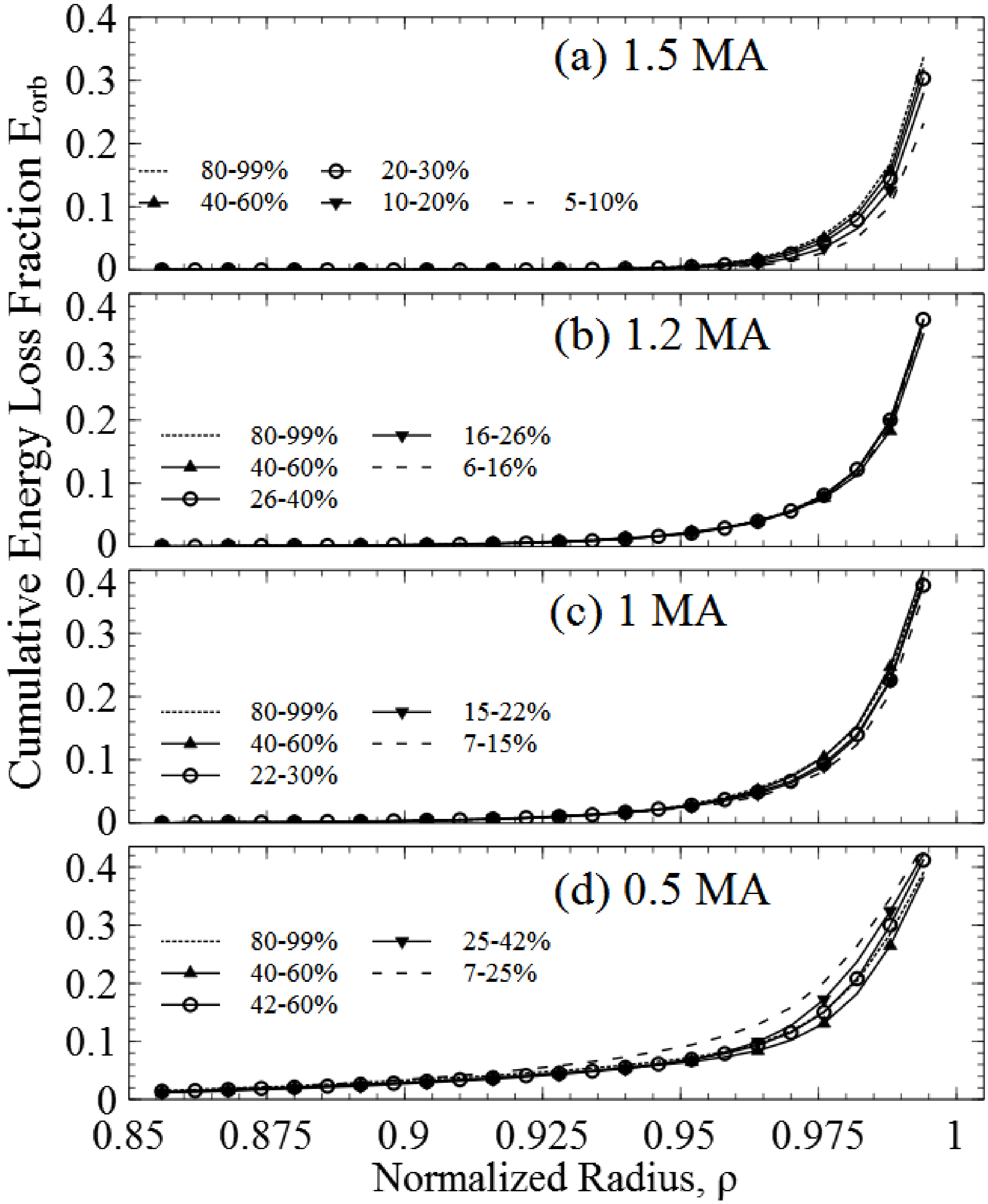


Figure 27: Energy Loss Fraction - The cumulative energy loss fraction as a function of radius with $R_{loss}^{iol} = 0.5$.

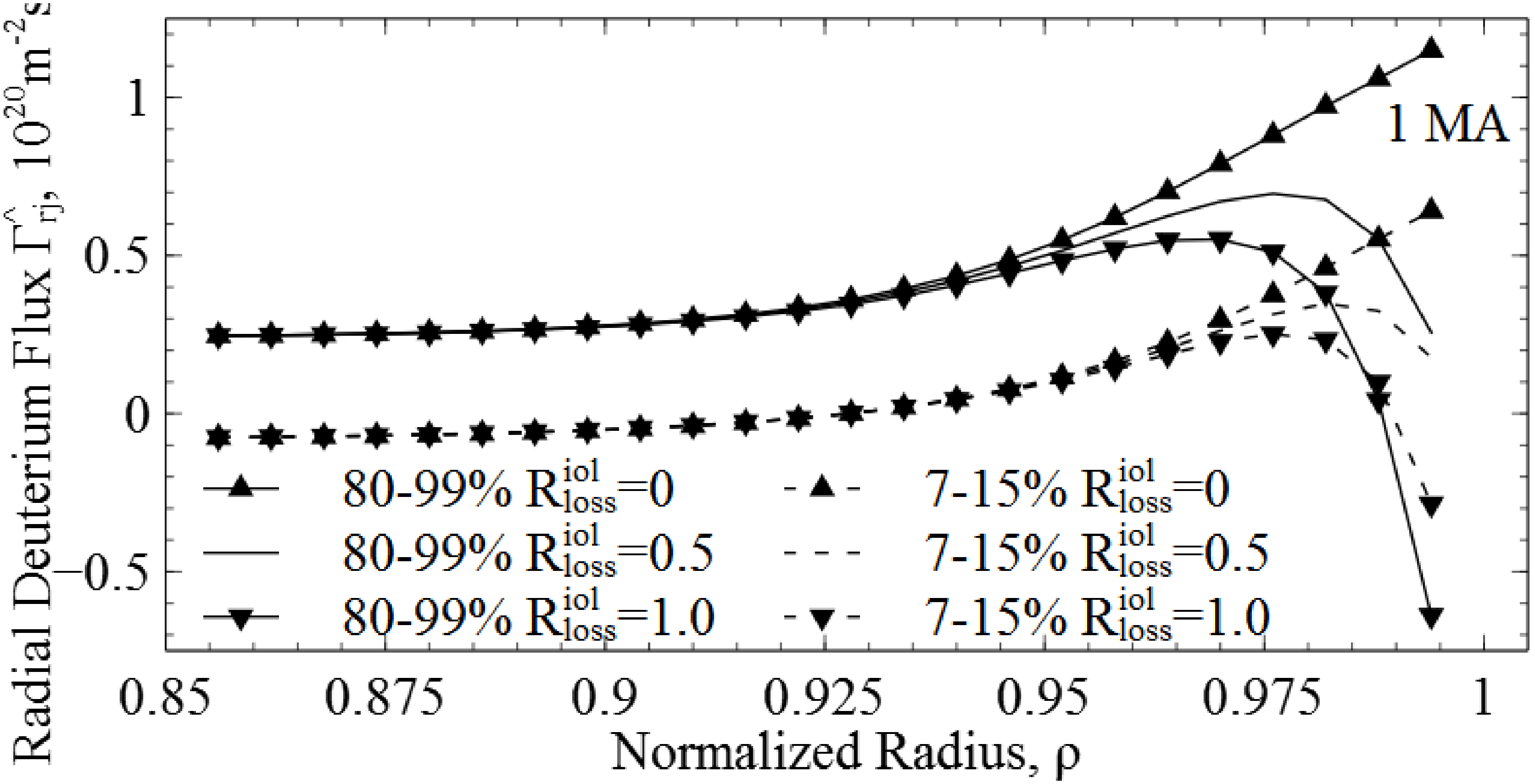


Figure 28: Radial Ion Flux Comparison - Total radial deuterium flux just before (80-99%, solid line) and just after (7-15%, dashed line) an ELM for the 1.0 MA shot, for different values of R_{loss}^{iol}. Comparison of 100% of ion orbit loss particles (R_{loss}^{iol} = 0) particles returning to the plasma, 50% of ion orbit loss particles returning to the plasma (R_{loss}^{iol} = 0.5), and 0% of ion orbit loss particles returning to the plasma (R_{loss}^{iol} = 1.0).

CHAPTER 9

RADIAL PARTICLE TRANSPORT INTERPRETATION

The experimental data presented in Chapter 8 is interpreted in this chapter in terms of the momentum and particle balance constraints of Chapter 9. The first step in the interpretive analysis is to examine the deuterium radial particle fluxes calculated from the continuity equation (Eq. (8)) and corrected for IOL (as shown in Eq. (12)). These corrected deuterium radial ion flux profiles are shown in Figure 29.

The outward radial ion fluxes peak in the sharp gradient region due to ionization of recycling neutrals, and decrease just inside of the separatrix because of the large quantity of particles being lost through ion orbit loss. Significant overshoot behavior is found in some of the shots. Aside from this, a general increase is seen during the inter-ELM period, with noticeable overshoot. The radial ion flux values increase across the edge with lower current, with the largest increase occurring in the sharp gradient region. An inward shift of the sharp gradient region with lower current can also be detected. This shift, and the increase in pedestal width that accompanies it, have been seen in several parameters, and will have a significant effect on the structure of the interpreted edge transport profiles.

The radial ion flux profile of the first interval in each shot demonstrates interesting behavior, being negative and inward early in the 1.5 MA and 1 MA shot. This relates back to the large Type I ELMs that occur just before the first intervals. These ELMs expel significant quantities of ions, which must be replaced to rebuild the density pedestal before the next ELM. This early influx of particles is largely a result of this refilling and rebuilding mechanism. Although the negative fluxes are found for the first interval in the 1 and 1.5 MA shots, they are not found in the 0.5 MA shot. This may be the case for several reasons, and two likely candidates are the lower time resolution available for this shot, or the possibility that the outward flux from the ELM event may still be influencing the flux in this interval (recall, the end of the longest leading ELM event in the 0.5 MA shot was determined more aggressively due to a relative lack of data for this shot, and some residual effects may have a larger effect in the first interval of this shot than in others, due to the ELM being closer in time to the first interval). The different geometry of the 1.2 MA shot may contribute to its lack of a negative radial ion flux profile in its first interval. The influxes seen early in the inter-ELM period in the radial ion flux profile may be related to the decreasing densities also found early in the inter-ELM.

9.1 The Diffusive Particle Flux

The total radial ion flux can be calculated from particle balance using the continuity equation, the known neutral beam source, and the calculated recycling neutral ionization source. As discussed, a fraction of the ion flux so calculated is in the form of free-streaming ion-orbit-loss particles, which must be compensated by an inward ion current, both of which reduce the ion flux actually being transported in the plasma. When this ion-orbit-loss correction is made, the remaining net ion flux in the plasma (shown in Figure 29) is described by the pinch-diffusion relation $\Gamma_{rj} = n_j \left(D_j L_{pj}^{-1} + V_{rj}^{pinch} \right)$ [Eq. (5)], where $D_j = \left(\frac{m_j T_j \nu_{jk}}{\left(e_j B_\theta \right)^2} \right) \left(1 + \frac{\nu_{dj}}{\nu_{jk}} - \frac{e_j}{e_k} \right)$ [Eq. (6)]. The diffusive flux is the first term of the pinch-diffusion relation: $\Gamma_{rj}^{dif} = n_j D_j L_{pj}^{-1} = n_j D_j (L_{nj}^{-1} + L_{Tj}^{-1})$. The quantity ν_{jk} is the deuterium-carbon collision frequency which can be calculated using the measured density and temperature. The quantity ν_{dj} is the deuterium toroidal angular momentum transport frequency due to viscous, inertial, charge-exchange and possibly other mechanisms, and may be referred to as the drag frequency. This is not presently known from first-principles but can be interpreted from the toroidal rotation velocities of both species, if both were measured. In these shots, only the carbon toroidal rotation velocity $V_{\phi k}^{\exp}$ was measured, and it was necessary to use a perturbation estimate{Stacey, 2008 #8} of the difference between the carbon and deuterium toroidal velocities ΔV_{ϕ}^{pert} to obtain an estimate of the toroidal deuterium rotation velocity. The estimated deuterium toroidal rotation velocity $V_{\phi j}^{pert} = V_{\phi k}^{\exp} - \Delta V_{\phi}^{pert}$ was then corrected for the calculated intrinsic rotation due to ion orbit loss{Stacey, 2014 #196} (from Eq. (13)) to obtain the IOL-corrected deuterium toroidal rotation velocity.

$$\hat{V}_{\phi j} = V_{\phi k}^{exp} - \Delta V_{\phi}^{pert.} - V_{\phi j}^{intrinsic} \qquad (15)$$

In the perturbation method [38], a leading order expression for the drag frequencies of both ion species is derived from toroidal momentum balance (Eq. (4)).

$$\nu_{dj} = \frac{\left[\left(B_\theta e_j \hat{\Gamma}_{rj} + M_{\phi j} + n_j e_j E_\phi^A\right) + \left(B_\theta e_k \hat{\Gamma}_{rk} + M_{\phi k} + n_k e_k E_\phi^A\right)\right]}{\left[\left(n_j m_j + n_k m_k\right)\hat{V}_{\phi k}\right]} \qquad (16)$$

Equations (14), (15), and (12) are used to represent the carbon and deuterium corrected toroidal rotation velocities and the corrected radial ion flux, respectively. The first order perturbation estimate of the difference between the deuterium and carbon rotation velocities can be obtained from Eq. (4) using perturbation theory.

$$\Delta V_{\phi}^{pert.} = \frac{\left[\left(B_\theta e_j \hat{\Gamma}_{rj} + M_{\phi j} + n_j e_j E_\phi^A\right) - n_j m_j \nu_{dj} \hat{V}_{\phi k}\right]}{\left[n_j m_j \left(\nu_{jk} + \nu_{dj}\right)\right]} \qquad (17)$$

This provides an estimate of the drag frequency, ν_{dj}, and the velocity difference, ΔV_{ϕ}^{pert} which should be trustworthy when the deuterium toroidal rotation velocity is similar to that of carbon. The drag frequency profiles calculated using this method, and also the collisional ion-impurity momentum transfer frequency v_{jk} profiles are shown in Figure 30.

The structures of both of these profiles have a substantial impact on transport in the edge. The drag frequency is dominated by the collisional frequency in the flattop region in all of the shots, with a peak that is located near the radial location of the top of the pedestal. Past the top of the pedestal, further out into the sharp gradient region, the drag frequency becomes significant, surpassing the collisional frequency in all but the 1.5 MA shot. During the inter-ELM period, the collisional frequency ($\sim n/T^{3/2}$) increases in the flattop region, but generally decreases in the sharp gradient region (as confinement time increases). The drag frequency exhibits overshoot behavior during the inter-ELM period in the sharp gradient region where it is large, increasing towards, then past, the last interval profile, subsequently returning to it before the next ELM.

The relationship between the two variables has a strong dependence on current. As current decreases, the collisional frequency decreases, and becomes closer to the drag frequency profile in the flattop region, while still dominating it. Near the top of the pedestal, the two profiles become more equal as current decreases, and in the sharp gradients region, a large peak in the drag frequency rises. While this peak is less than the collisional frequency in the 1.5 MA shot, it becomes progressively larger with decreasing current, and dominates the collisional frequency in the sharp gradient region of the 0.5 MA shot. The peak value of the collisional frequency is associated with the radial location of the top of the density pedestal, which shifts slightly inward during the inter-ELM period, and widens considerably as current decreases. The location of the left slope of the drag frequency peak is also affected by decreasing current in a similar way, moving inward between ELMs and with decreasing current. However the right slope of the drag frequency peak is less sensitive to the location of the sharp gradient region, only shifting slightly inward with large decreases in plasma current.

Near the separatrix, the relationship between the two frequencies is strongly dependent on the value of the current. The drag frequency becomes significantly larger than the collisional frequency near the separatrix in all but the 1.5 MA shot. The cause of this peak near the edge can be determined from Eq. (16), which is used to interpret the drag frequency from the carbon toroidal rotation velocity (Figure 31), the radial ion flux (Figure 29), and other, less influential factors. The radial ion flux in the numerator of Eq. (16) is large in the sharp gradient region, and is increased by two terms found in the denominator: the density, which decreases in the sharp gradient region, and the radially decreasing carbon toroidal rotation velocity. As the drag frequency profile approaches the separatrix, the severe reduction in the radial ion flux related to IOL causes it to also decrease.

The corrected deuterium toroidal rotation velocities found from Eq. (15), and the deuterium toroidal IOL intrinsic rotation velocity calculated from Eq. (13) are shown in Figure 32. The corrected deuterium toroidal rotation velocity profiles decrease throughout the edge, and are mostly influenced by the intrinsic rotation near the separatrix, where the intrinsic rotation causes a small dip in the corrected rotation. The intrinsic rotation velocity profiles have similar structures, inter-ELM evolutions and trends with current, to the loss fraction profiles in Figure 26. Upon examination, it is clear that the deuterium toroidal rotation velocity profiles mirror many characteristics of the carbon toroidal rotation velocity profiles in Figure 24, justifying the use of the perturbation calculation (which assumes the difference between the toroidal rotation velocity profiles of the two species is relatively small). The broadening of the intrinsic rotation peaks with lower current in the sharp gradient region is a structural feature also found in the drag frequency, and other variables. In all cases, a contributor is the broadening area of IOL influence as current decreases, shown in Figure 26.

The corrected carbon toroidal rotation velocity profiles, and the carbon intrinsic IOL toroidal rotation velocity profiles, are shown in Figure 31. The main difference between the corrected toroidal rotation profiles for the two ion species is the presence of a larger intrinsic IOL rotation velocity correction for deuterium. This is due to a larger cumulative momentum loss fraction M_{orb} from IOL for the deuterium species in the far edge, which causes a deeper depression near the separatrix. The reasons for this difference are discussed in 8.2.

Once the drag frequency is interpreted from the experimental data, it is possible to calculate the value of the deuterium diffusion coefficient of Eq. (5) that is required by momentum balance. This parameter incorporates the ion composite momentum transfer frequency ν_{dj} (Figure 30, Eq. (16)), the momentum exchange frequency ν_{jk} (also Figure 30) evaluated for the ion temperatures, as well as other variables. Figure 33 shows the deuterium diffusion coefficient profiles for the four shots.

It is clear from a comparison of Figure 30 and Figure 33 that the collisional frequency ν_{jk} dominates the deuterium diffusion coefficient structure in the flattop, and the composite toroidal angular momentum transport (drag frequency) dominates the structure in the steep gradient region. This follows from examining Eq. (5), which shows that $D_j \sim T_j(\nu_{dj} + \nu_{jk})$. It is also clear that the peak magnitude of the deuterium diffusion coefficient decreases with lower plasma current, and its radial location moves inward. An increase in the deuterium diffusion coefficient between ELMs is seen in the flattop region, where the collisional frequency dominates, and an increase between ELMs with overshoot behavior is seen in the sharp gradient region, where the drag frequency dominates (in all but the 1.5 MA shot).

9.2 Pinch Velocity

The other term in the pinch-diffusion relation for the radial ion flux of Eq. (4) $\Gamma_{rj} = n_j \left(D_j L_{pj}^{-1} + V_{rj}^{pinch} \right)$ is the (mostly) electromagnetic pinch term: $V_{rj}^{pinch} \simeq \left(m_j / e_j B_\theta^2 \right) \left(\left(\nu_{dj} + \nu_{jk} \right) \left(E_r + V_{\theta j} B_\varphi \right) + \nu_{jk} V_{\phi k} B_\theta \right)$. This term depends on the collision and toroidal angular momentum transport frequencies given in Figure 30, the measured radial electric field, the carbon toroidal rotation velocity, and the unmeasured deuterium poloidal rotation velocity which can be determined by using the deuterium radial force balance equation[37].

$$B_\phi V_{\theta j}^{interp} = B_\theta \hat{V}_{\phi j}^{perturb} - E_r + \frac{1}{n_j e_j} \frac{\partial p_j}{\partial r} \tag{18}$$

The interpreted deuterium poloidal rotation velocity profile evolution is shown in Figure 34. The deuterium poloidal velocity profile is small and positive through most of the edge, and generally increases at each point during the inter-ELM period. For the three higher current shots, the profiles only change significantly between ELMs in the sharp gradient region. For these shots, the velocities increase by a small amount, and the locations of the peak values shift slightly inward. For the 0.5 MA shot, despite a similar small increase in velocity between ELMs, an unexpected, large inward shift in the location of the peak structure occurs. This large change in location is partially associated with the widening of the pressure pedestal.

The profile values are largely driven by a combination of the pressure gradient term and the radial electric field (Eq. (18)). In the flattop region, the pressure gradient term is small, and the radial electric field term overcomes the deuterium toroidal velocity term to keep the deuterium poloidal velocity profile positive. In the sharp gradient region, the negative deuterium toroidal rotation velocity term becomes smaller, and the positive radial electric field combines with the positive pressure gradient term to drive a significant rise in the deuterium poloidal rotation velocity profile. Near the separatrix, the pressure gradient term continues to rise, while the radial electric field term contributes less as its magnitude decreases, damping the rise at the separatrix. The 0.5 MA shot shows the peak structure in a significantly different location, and its outside slope has a different profile, than in the other shots. The inward shift in the radial location of the peak is due to the widening of the pedestal, providing a larger area over which the pressure gradient term is large and increasing. The low rotation values near the separatrix (after the peak) are related to the wide pedestal, and the increasingly negative electric field with decreasing current. The pressure gradient spans more of the edge in the 0.5 MA shot, then is much smaller near the separatrix, providing a wider poloidal velocity peak that begins further inside, and slowly tapers off near the separatrix (with a small gradient). This allows the negative toroidal velocity term to balance the radial electric field term more effectively, decreasing the value of the deuterium poloidal rotation velocity profile at the separatrix.

The expression for the pinch velocity (Eq. (7)) required by momentum balance was obtained by combining the radial and toroidal momentum balance components, Equations (3) and (4), into Equation (7). The first two components of the pinch velocity term relate to the external applied torques, while the third (E_r term) concerns the electrostatic field forces, and the fourth ($V_{\phi j}$ term) and fifth ($V_{\theta j}$ term) terms take the VxB forces into account. All the values necessary to calculate the pinch velocity have been interpreted from the experimental data, and the values of the pinch velocity are shown in Figure 35.

The pinch velocity profile represents the non-diffusive transport forces on the ions, and it is small throughout most of the inner edge. However, it becomes large and important in the sharp gradient region, where it forms a deep negative (inward direction) well. This interpretation of the experimental data suggests that a large inward electromagnetic force is acting on the ions in the far edge, and that it is due to non-diffusive forces such as electrostatic fields and VxB forces. The pinch velocity evolution generally deepens the negative well structure found in the sharp gradient region between ELMs, with a larger decrease and a smaller overshoot behavior seen in the lower current shots. The magnitude of the pinch velocity and the width of the negative well are both strongly affected by current, with the maximum speed increasing ~600% and the well width increasing by ~50% from the 1.5 MA to the 0.5 MA shot. As with the diffusion coefficient, the reduction in the radial ion flux due to IOL contributes to the interpreted pinch velocity returning to smaller values in the far edge (through the drag frequency), along with a similar pattern (return to small magnitudes at the separatrix) in the radial electric field.

The values of the pinch velocity (Figure 35) peak near the separatrix, and become significantly more negative with decreasing current. It is useful to examine the components of the pinch velocity to determine what causes this structure, and these components are compared in Figure 36 for the first and last intervals of each shot. A brief glance at the components in Figure 36 shows that the radial electric field and the deuterium poloidal rotation velocity terms are the most influential on the total pinch velocity. This confirms the main dependence of the pinch velocity derived from Eq. (7), $V_{rj}^{pinch} \sim (\nu_{dj} + \nu_{jk})(E_r + V_{\theta j} B_\varphi)$, and largely explains the structures seen in Figure 35.

Generally in the first intervals, in the flattop region, a negative carbon toroidal and a deuterium poloidal rotation velocity term oppose a positive radial electric field term, the difference being small and negative. Near the end of the inter-ELM period, in the same place, the negative radial electric field term and toroidal carbon rotation velocity term have eclipsed the poloidal carbon rotation velocity term, and combine to result in a small positive radial ion flux. In the sharp gradient region, however, the main pinch velocity drivers are both negative throughout the inter-ELM period. The radial electric field and the deuterium poloidal rotation velocity are both significant and negative, and outweigh the other contributions to the pinch velocity. It is important to note that the radial electric field and the deuterium poloidal rotation velocity profiles, which both peak in the edge, are multiplied in Eq. (7) by the drag frequency, which also has a peak in the edge. This multiplicative effect is a reason for the large values contributed by the radial electric field and deuterium poloidal rotation velocity terms.

One important feature of the pinch velocity profile is the width and depth of the negative well in the sharp gradient region. The properties of this negative well structure dictate how much of the edge is affected by this large inward pinch, and partially determine the strength of the effect. The dominant driver of well width, and the linked property of well depth, is the extent of radial co-location among the minima of the pinch velocity components. In the first, post-ELM interval, these peak magnitudes tend to align relatively well, driving a relatively narrow well with a maximum possible depth. However in the last, pre-ELM, 80-99% interval, the peak magnitudes of the components misalign, broadening the well, but lessening its depth. This evolution between ELMs, and the change in alignment, is best illustrated in the 1 MA and 0.5 MA shots in Figure 36, where the minima are well aligned in the first interval, but the alignment is degraded by the end of the shot. The increase in the pedestal width with lower current, a trend that has been well established, also increases the opportunity for the pinch velocity components to be misaligned. However, different sensitivities to changing pedestal width among the pinch velocity components, also plays a role.

The depth of the pinch velocity negative well is dependent both on the alignment of pinch velocity component minima, and the interaction among parameters inside the component to determine the magnitude of the minima. The peak magnitude of the pinch velocity increases with decreasing current, and between ELMs, and the three parameters to be considered for responsibility are the radial electric field, the deuterium poloidal rotation velocity, and the drag frequency. Regarding the increase in the pinch velocity between ELMs, all three variables have small magnitude increases that are responsible for the observed increase. The increase in drag frequency across the inter-ELM period in the sharp gradient region for each shot is responsible for a good portion of the inter-ELM increase, but this is mitigated by the overshoot behavior in the drag frequency that is seen in the higher current shots (shown in Figure 30). The magnitudes of the radial electric field change somewhat between ELMs, and the deuterium poloidal rotation velocity profiles increase slightly near the separatrix between ELMs. These increases play a significant role in the evolution of the pinch velocity between ELMs, but variation in radial location and alignment of the peaks in the two main pinch velocity components, and in the constituent variables of the components, still play a large role in inter-ELM evolution of the pinch velocity.

A much larger increase in pinch velocity is seen with decreasing current. As the current decreases from 1.5 MA to 0.5 MA, the pinch velocity peak magnitude increases by a factor of 10. This is largely driven by the increase in the radial electric field profile with decreasing current, shown in Figure 25, and a better alignment between it and the multiplying drag frequency. Another driver is the better alignment between the drag frequency and the deuterium poloidal rotation velocity profiles. These changes in alignment are non-uniform, and are a result of differing sensitivities to a widening pedestal, and other, lesser influences. Additionally, the elimination of the overshoot behavior seen in the drag frequency profile evolution by the 0.5 MA shot maximizes the potential magnitude of the quantities being multiplied. The deep, wide well seen in the last interval for the

9.3 Components of the Radial Ion Particle Flux in the Plasma

By separating the pressure gradient in Eq. (5) into its constituent temperature and density gradients, the radial particle flux in the plasma can be written

$$\Gamma_{rj} = n_j D_j L_{nj}^{-1} + n_j D_j L_{Tj}^{-1} + n_j V_{rj}^{pinch}; \qquad L_X^{-1} = \frac{-dX / \partial r}{X} \tag{19}$$

The three components of the radial particle flux in the plasma are shown in Figure 37. In all shots, there is a large inward pinch flux and a large outward diffusive flux (mostly driven by the density gradient) that almost balance to produce a small net flux. (There also a radial ion flux of free-streaming ion-orbit-loss particles that increases strongly for $\rho > 0.95$, but these are not shown in Figure 37.)

The most striking feature of Figure 37 is that in the sharp gradient regions in the edge there are large inward electromagnetic pinch fluxes $\Gamma_{v_p} = n_j V_{rj}^{pinch}$ and large outward diffusive fluxes $\Gamma_{nj} = n_j D_j L_{nj}^{-1}$ and $\Gamma_{Tj} = n_j D_j L_{Tj}^{-1}$ that almost balance each other to produce a much smaller net outward flux (we speak of “fluxes” for convenience; there are actually strong diffusive and electromagnetic forces that almost balance, and drive a small net outward flux composed of plasma ions. In the flattop region, both the diffusive and electromagnetic fluxes (forces) are relatively smaller, but add to produce a net inward flux (force) immediately after the ELM which evolves to a small net outward flux (force) later in the inter-ELM period. We further note that the component of the diffusive flux (force) due to the density gradient is generally larger than the component of the diffusive flux (force) due to the temperature gradient.

The recovery of the inward electromagnetic pinch flux following the ELM is more or less immediate (within the resolution of the analysis—see Figure 16), while the diffusive flux recovers more slowly as the density and temperature pedestal are re-established, and it rebuilds during the inter-ELM period. It is notable that a significant change that occurs between ELMs is found on the inward side of the density-driven and pinch-driven fluxes (forces). As the inter-ELM period progresses, the pedestal widens, and the outward diffusive forces extend further inward, softening the slope of the inward side of the peak in the diffusive flux (forces). This migration is matched by the broadening non-diffusive pinch flux (force), which broadens its area of effect to match the expanded diffusive flux (force) that now exists near the top of the pedestal. The two main components of the pinch velocity combine (and possibly misalign) to broaden the pinch velocity negative well during the inter-ELM period, and match the broadened diffusive flux (force).

9.4 Transport Interpretation Without IOL Correction

The results shown in Figure 29 through Figure 37 are based on an IOL correction assuming that 50% of the ion particles crossing the separatrix on loss orbits do not return to the plasma (R_{loss}^{iol} = 0.5), and are lost from the ion population being transported outward in the plasma edge. The effect of the assumed R_{loss}^{iol} on the ion orbit loss fractions is shown in Figure 26 and on the radial particle flux in Figure 28.

The assumption usually implicitly made by neglecting IOL is that none of the ions crossing the separatrix are lost from the plasma (R_{loss}^{iol} = 0), but instead all return to be transported outward. Relative to this, increasing the assumed loss rate of ions crossing the separatrix to 50% (R_{loss}^{iol} = 0.5) drives a number of important differences in several of the quantities shown in Figure 29 through Figure 37. Those quantities which are most affected by this assumption (neglecting IOL) are the interpreted deuterium drag frequency and thermal diffusivity. The effects of neglecting IOL on these quantities are shown for the 1 MA shot with R_{loss}^{iol} = 0.0 in Figure 38, as the characteristics of the profiles for these parameters are roughly representative of those in the other three shots.

Comparing the upper plot in Figure 38 (R_{loss}^{iol} = 0.0), with the 1 MA case in Figure 30 (R_{loss}^{iol} = 0.5), the effect of neglecting IOL is to sharply over-estimate the interpreted momentum transport (drag) frequency for $\rho > 0.75$ where the transport of momentum by IOL is significant. The consequence of neglecting IOL on the interpreted diffusion coefficient then follows directly from Eq. (6), and may be seen by comparing the center plot in Figure 38 with Figure 33 – the interpreted diffusion coefficient is significantly overestimated for $\rho > 0.75$, where the momentum transport by IOL is significant.

When it is assumed that no particles are lost from the plasma through IOL, the radial ion flux profiles of the first and last interval for the 1 MA shot are shown in Figure 28 (along with the profiles for $R_{loss}^{iol} = 0.5$ and 1.0). For $R_{loss}^{iol} = 0$, the peak values of the radial ion flux are nearly doubled to account for the particles required to be present by the continuity equation (Eq. (8)), but that in reality have escaped confinement through IOL, and are no longer part of the plasma. This is reflected in the disappearance of the peak structure found in Figure 29, and it being replaced by a continuous upward trend that extends to the separatrix. This new structure follows the inward slope of the old peak structure, but does not decrease near the separatrix, as it must account for the full density from the continuity equation at each point. In the 0.5 MA shot, where IOL is more influential further inward, the radial ion flux is larger across almost the entire edge. By not taking IOL into account, this critical edge ion transport parameter, which is instrumental in calculating several other parameters, is greatly overestimated near the separatrix. This over-interpretation of radially outward transport, or underestimation of radially inward transport for certain parameters, when neglecting the effects of IOL, is a problem in most estimates of transport in the plasma edge.

As the (interpreted) drag frequency is derived from the radial ion flux, it too loses the peak structure seen in Figure 30, and increases consistently to the separatrix, as shown in the upper plot of Figure 38. This is a result of attempting to create a momentum transport level that could explain the apparent, but inflated, large outward fluxes in the region, while still taking into account the small deuterium density near the separatrix (the actors in the momentum transport). The profiles of the interpreted deuterium diffusion coefficient for no IOL and the 1 MA shot are shown in the middle plot of Figure 38. The prominent peak structures present in the profiles of Figure 33 are absent, and instead, the continuous increase to the separatrix seen in the drag frequency and the radial ion flux are mirrored. Although IOL has little effect on the ion temperature or collisional toroidal momentum transfer frequency, its effect on the drag frequency is sufficiently large to strongly affect the diffusion coefficient. Again, the balance equations are forcing the diffusion coefficient to account for a large outward flux of ions (that no longer exist in the plasma) in a low density region of the plasma.

The interpreted pinch velocity profiles calculated neglecting IOL (R_{loss}^{iol} = 0) are shown in the lower plot of Figure 38, which should be compared with the 1 MA case in Figure 35 (R_{loss}^{iol}= 0.5). The neglect of IOL makes the interpreted inward pinch much more negative for $\rho > 0.75$. The interpreted pinch velocity loses the well structure found near the separatrix in Figure 35, and instead decreases continuously to the separatrix. The pinch velocity neglecting IOL has a larger magnitude in the edge than the pinch velocity taking it into account.

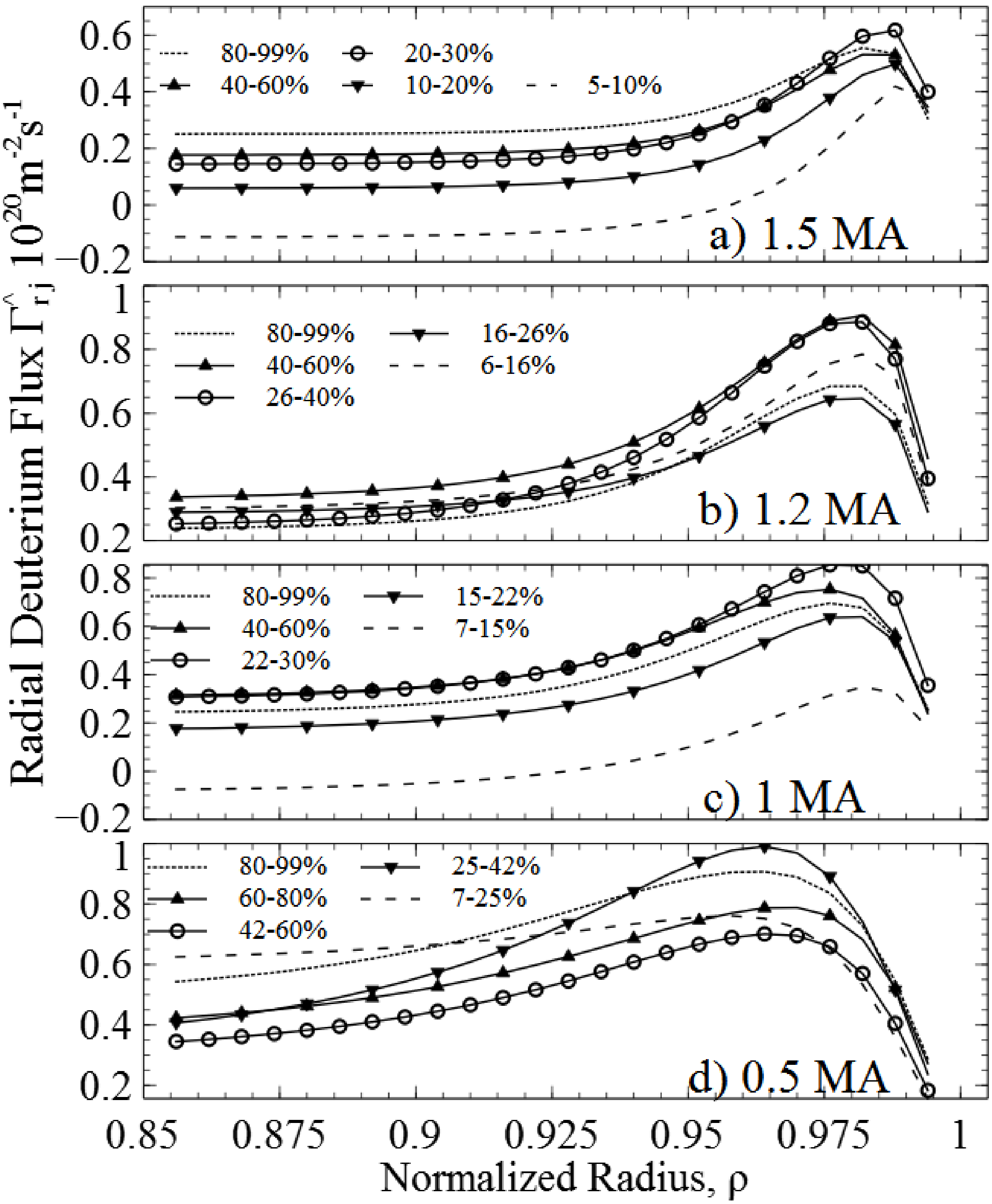


Figure 29: Corrected Radial Ion Flux - IOL-corrected radial deuterium ion fluxes for $R_{loss}^{iol} = 0.5$.

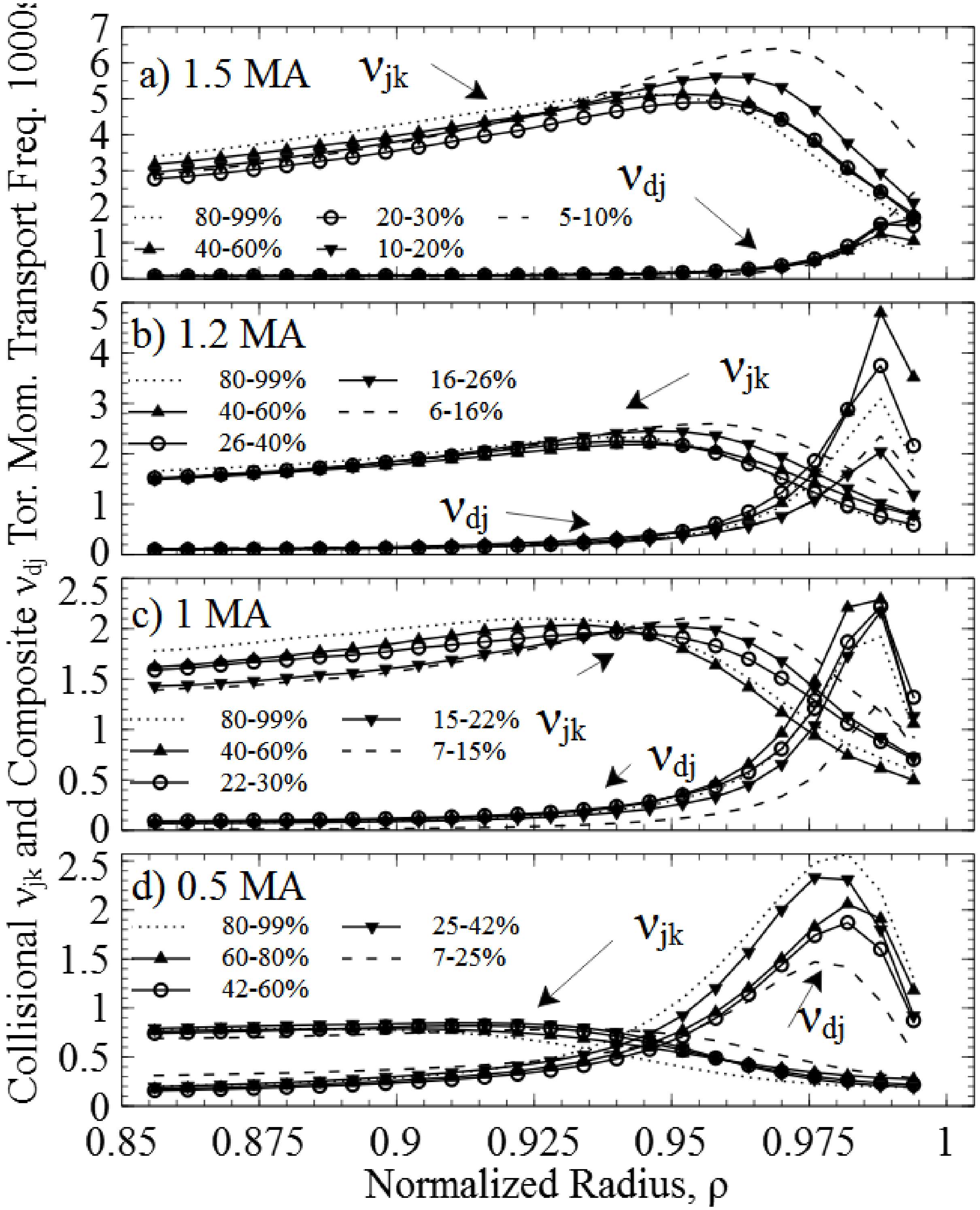


Figure 30: Momentum Transport - The ion-impurity collisional toroidal momentum transfer frequency ν_{jk} and the composite toroidal momentum transport frequency ν_{dj} profiles evolve across the composite inter-ELM period for $R_{loss}^{iol} = 0.5$.

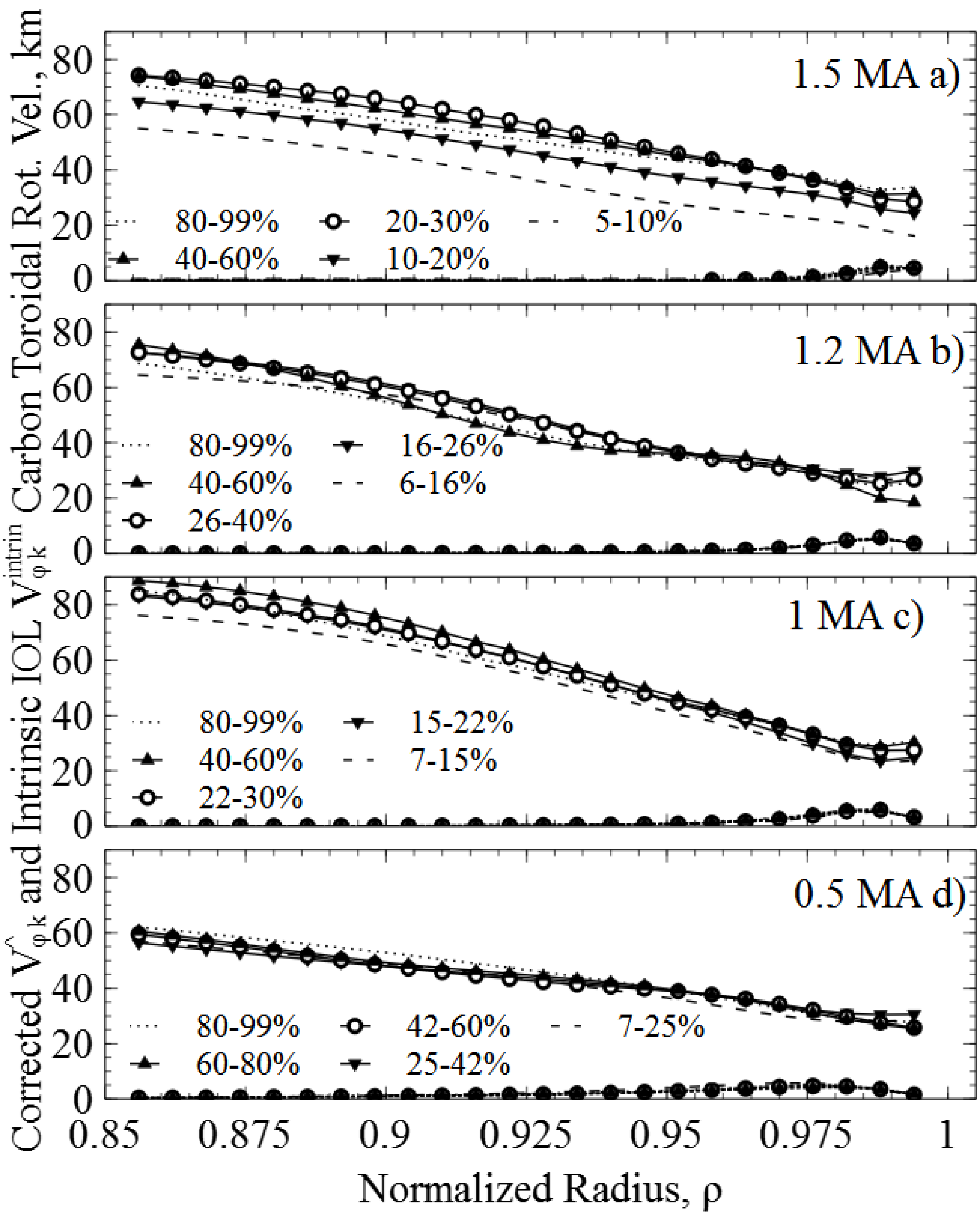


Figure 31: Corrected Carbon Rotation Velocity - The measured carbon toroidal rotation velocity less the carbon intrinsic IOL toroidal rotation velocity is represented by the upper set of profiles in the four plots. The carbon intrinsic IOL toroidal rotation velocity itself is represented by the lower set of profiles in the four plots. All quantities are shown for $R_{loss}^{iol} = 0.5$.

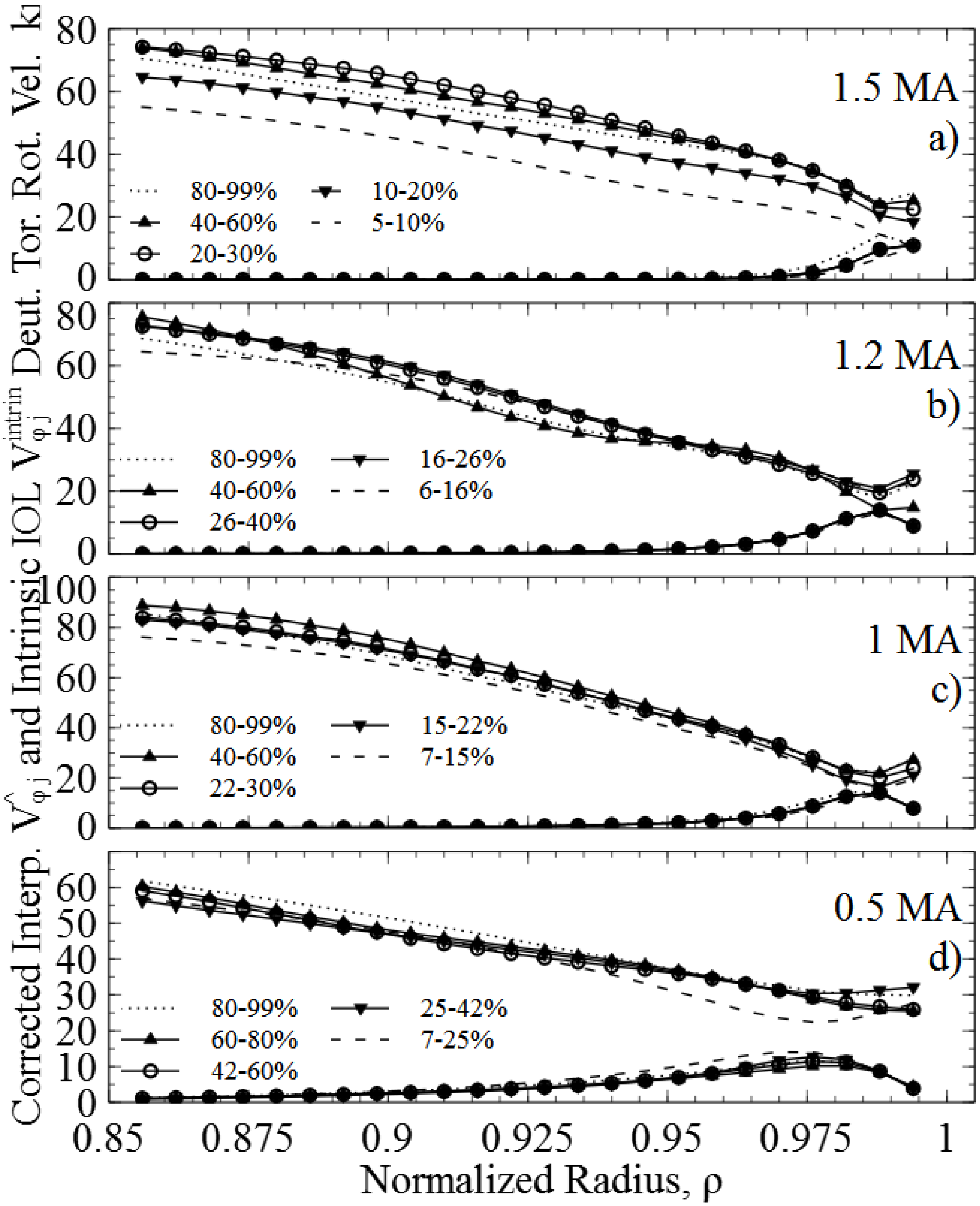


Figure 32: Corrected Deuterium Toroidal Velocity - The deuterium toroidal rotation velocity profiles corrected for IOL effects are shown (upper trends at left), along with the intrinsic rotation velocity profile (lower trends at left) for $R_{loss}^{iol} = 0.5$

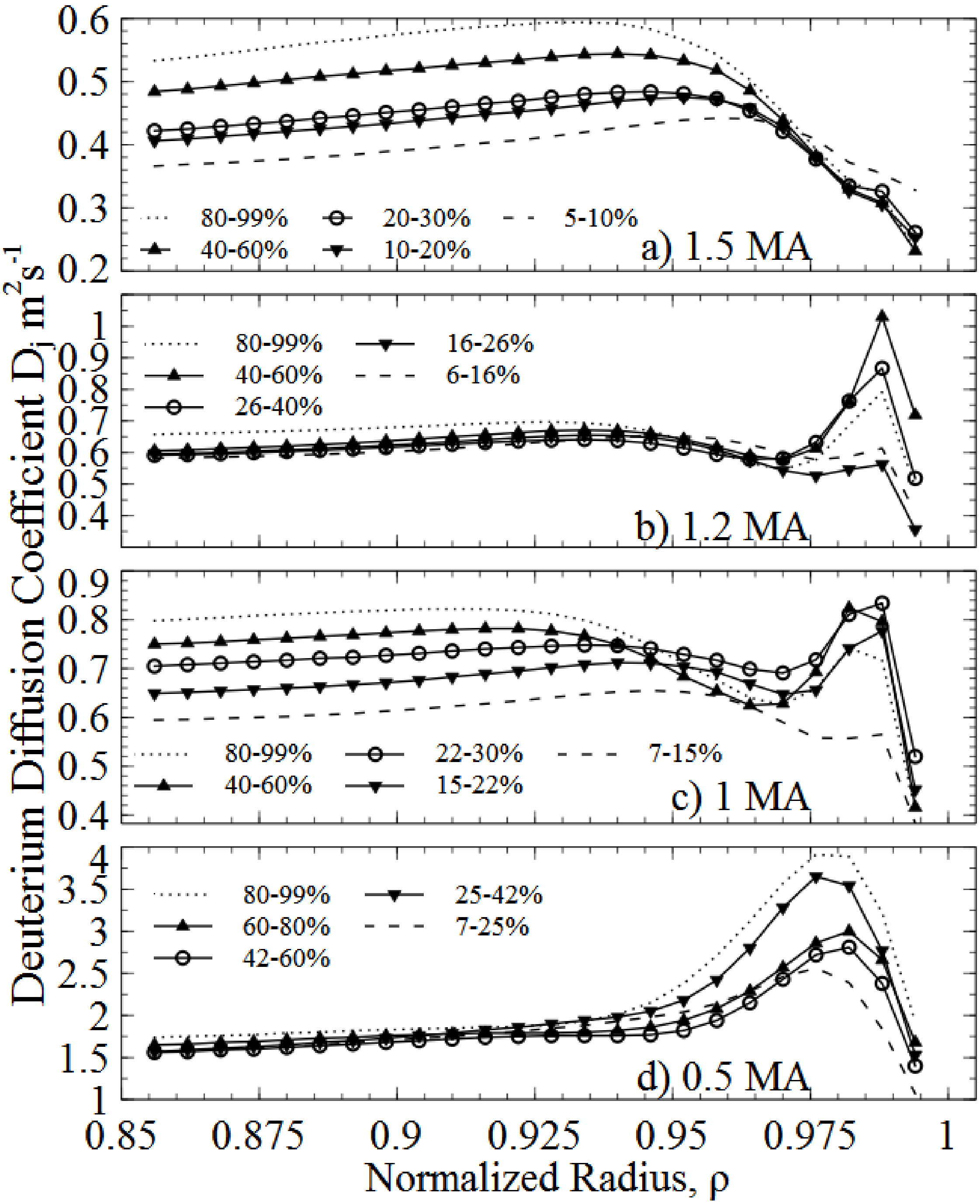


Figure 33: Deuterium Diffusion Coefficient - The interpreted deuterium diffusion coefficient profiles for $R_{loss}^{iol} = 0.5$

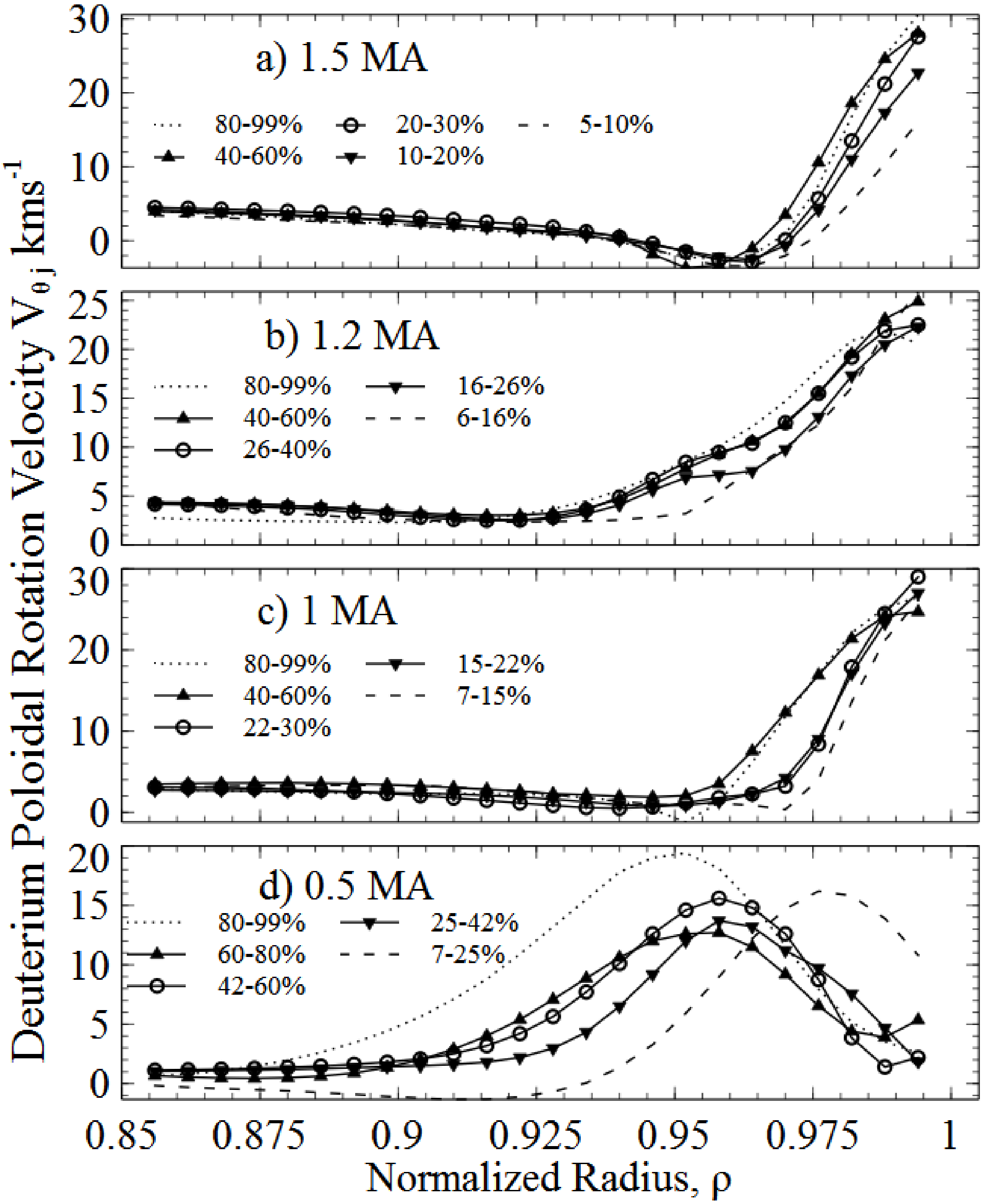


Figure 34: Deuterium Poloidal Rotation Velocity - The inferred deuterium poloidal rotation velocity profiles for $R_{loss}^{iol} = 0.5$.

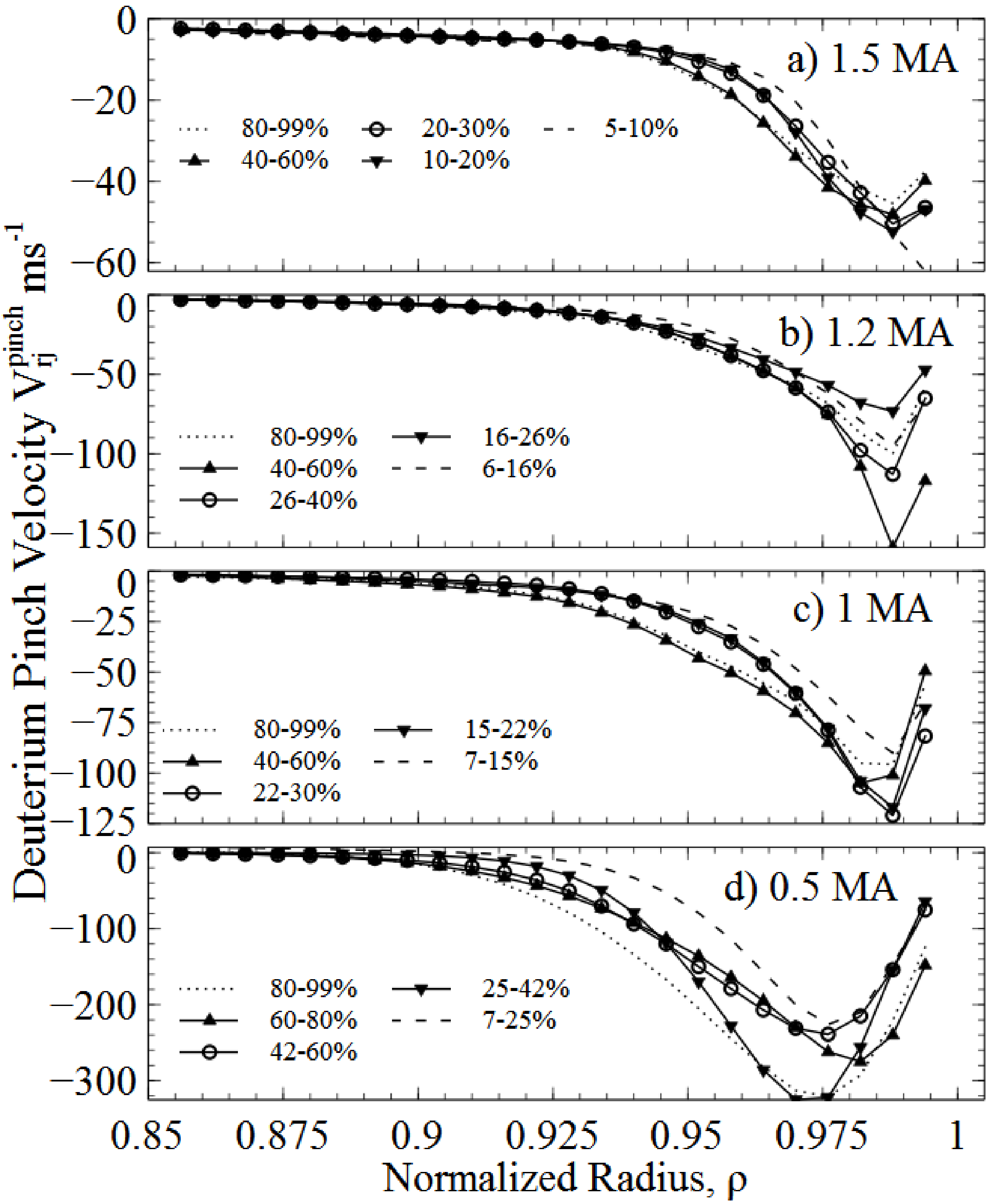


Figure 35: Deuterium Pinch Velocity - The evolution of the pinch velocity profiles for $R_{loss}^{iol} = 0.5$.

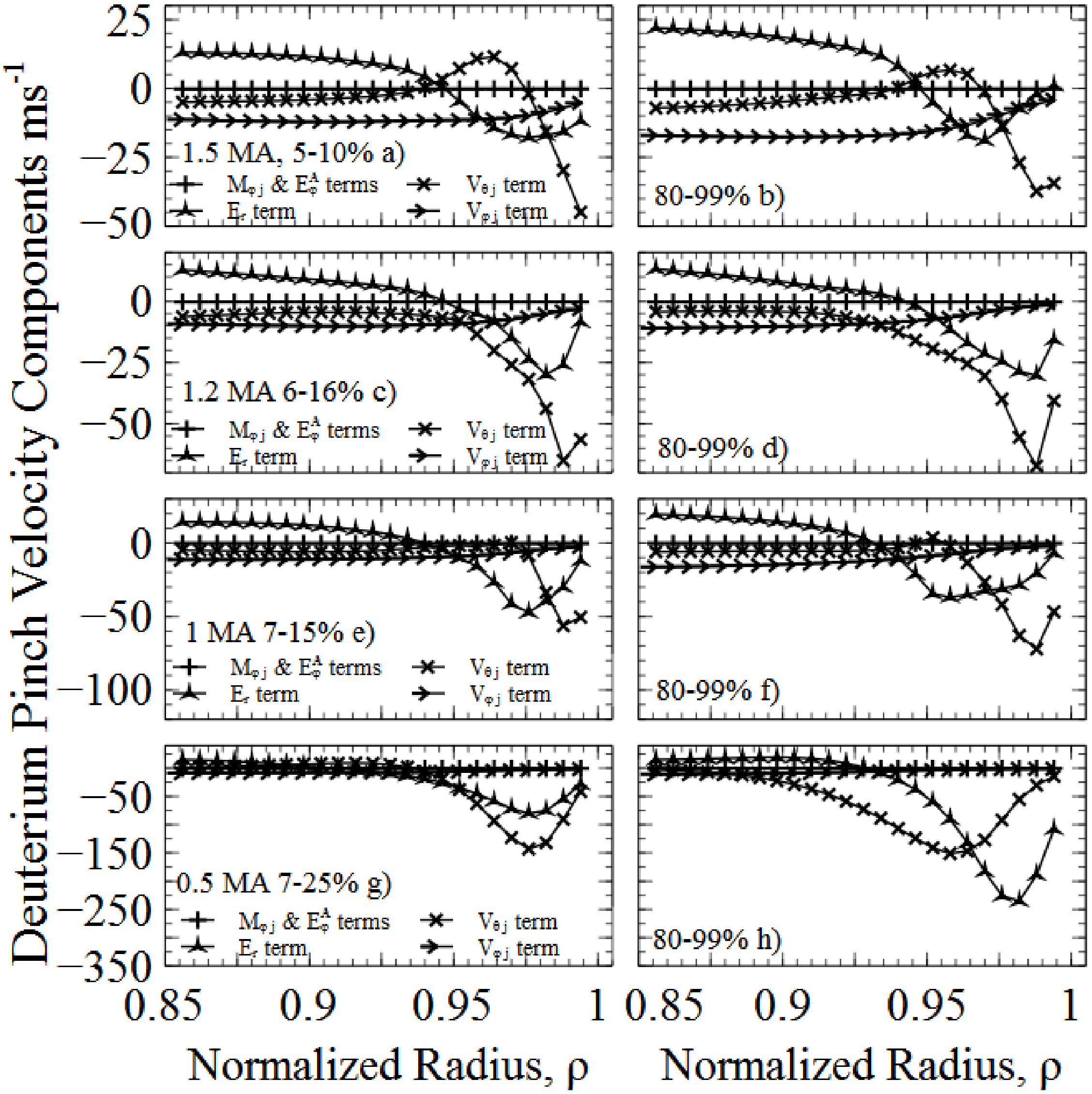


Figure 36: Pinch Velocity Components - The components of the pinch velocity for the first and last intervals of the four shots and $R_{loss}^{iol} = 0.5$.

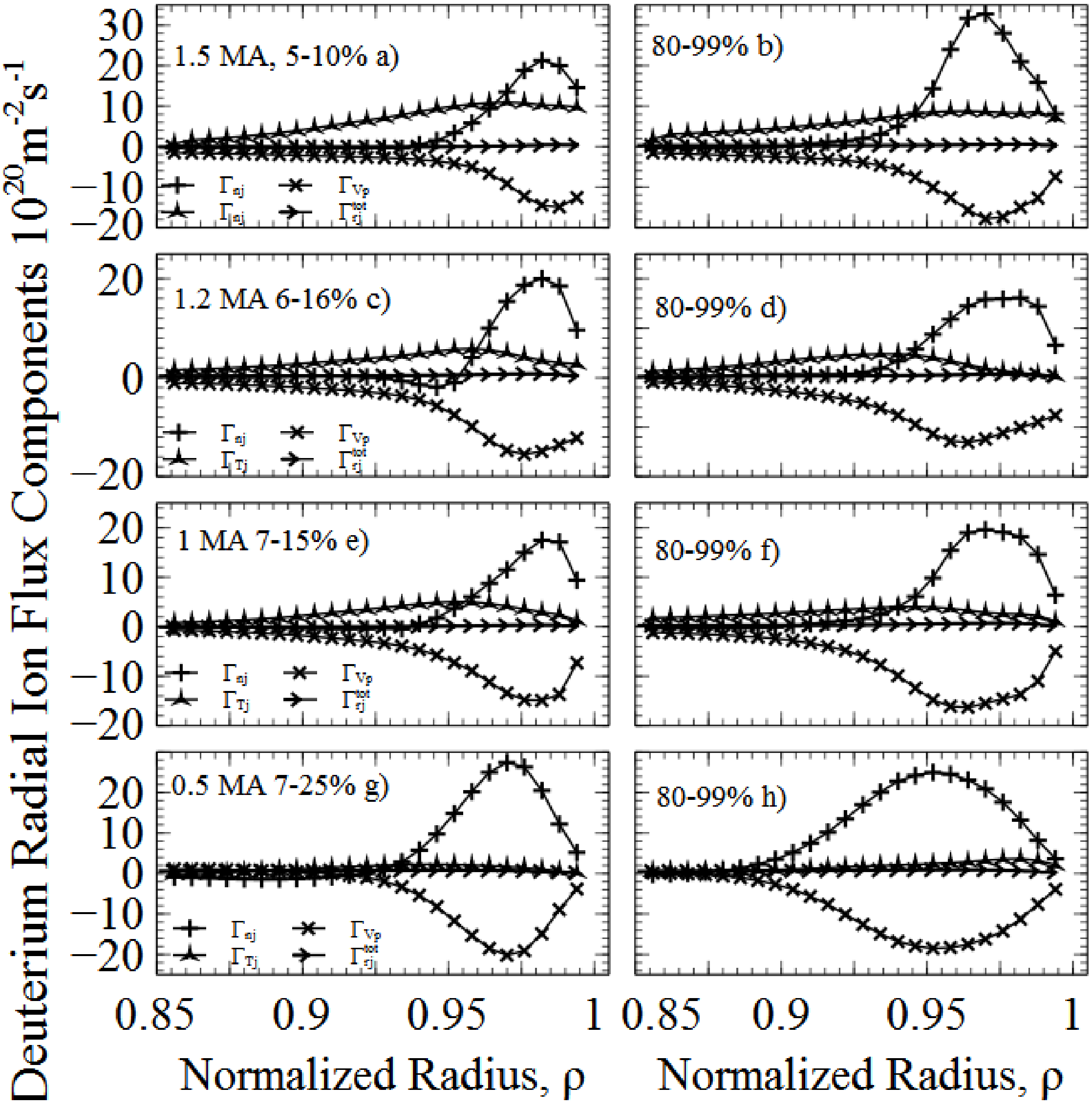


Figure 37: Radial Ion Flux Components - The components of the radial ion particle flux of Eq. (19), along with the total radial ion particle flux from Eq. (12) for $R_{loss}^{iol} = 0.5$.

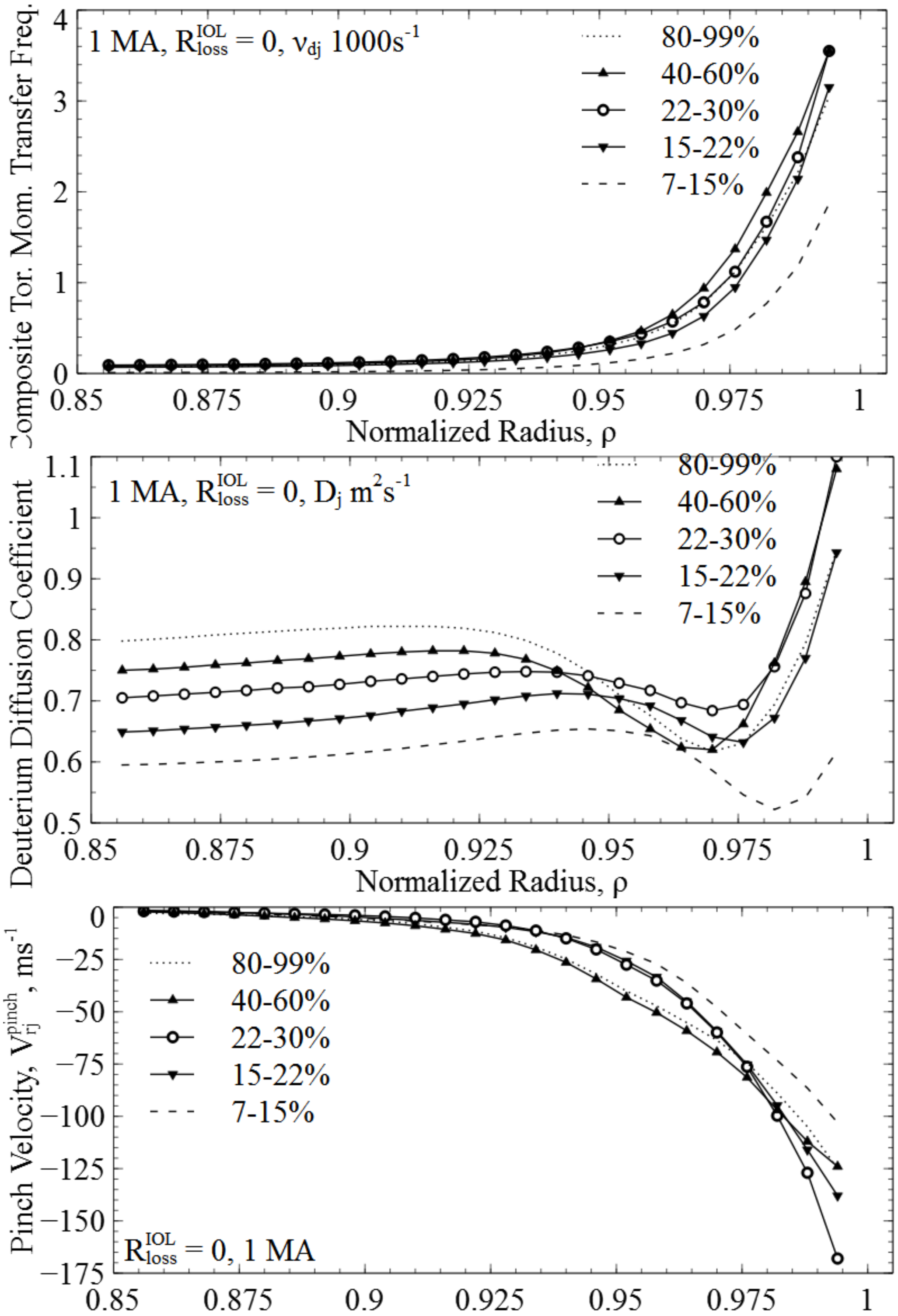


Figure 38: Drag Frequency and Diffusion Coefficient for $R_{loss}^{iol} = 0$. The deuterium drag frequency and diffusion coefficient for the 1 MA shot and $R_{loss}^{iol} = 0$, implying that all particles leaving the plasma through IOL return. These plots can be compared to the plots for the 1 MA shot in Figure 30 and Figure 33 to determine the changes caused by decreasing R_{loss}^{iol} to 0 from 0.5.

CHAPTER 10

INTERPRETATION OF THERMAL DIFFUSIVITIES

10.1 Ion and Electron Deuterium Diffusivities

The radial component of the energy balance equation for the electrons and main ion species can be solved for the ion and electron heat fluxes [39]. For the ions,

$$\frac{\partial Q_j}{\partial r} \equiv \frac{\partial}{\partial r}\left(q_j + \frac{3}{2}\hat{\Gamma}_j T_j\right) = \frac{\partial}{\partial t}\left(\frac{3}{2} n_j T_j\right) + q_{nbj} - q_{je} - n_e n_0^c \left\langle \sigma \upsilon \right\rangle_{cx} \frac{3}{2}\left(T_j - T_0^c\right) \qquad (20)$$

The last three terms on the right are the neutral beam heating of ions, the electron heating by ions, and the charge-exchange heating of recycling neutrals by ions. A similar equation holds for the electron heat flux involving the radiative and ionization heat sinks.

$$\frac{\partial Q_e}{\partial r} \equiv \frac{\partial}{\partial r}\left(q_e + \frac{3}{2}\hat{\Gamma}_e T_e\right) = \frac{-\partial}{\partial t}\left(\frac{3}{2} n_e T_e\right) + q_{nbe} + q_{je} - n_e n_k L_k\left(T_e\right) - n_e n_0 \left\langle \sigma v \right\rangle_{ion} E_{ion} \qquad (21)$$

In these equations, the neutral beam heating is indicated by q_{nb}, the ion to electron heat flux is given by q_{je}, and the last terms represent charge exchange cooling of the ions, and radiation cooling of the electrons, respectively. Once the heat flux is available, the thermal diffusivity for each species can be inferred [38] from it and the density and temperature profiles.

$$\chi_j^{exp} = \frac{\left(\hat{Q}_j - 1.5\hat{\Gamma}_{rj} T_j\right)}{-n_j\left(\partial T_j / \partial r\right)} \qquad (22)$$

A similar equation can be found for the electron thermal diffusivity. Both thermal diffusivities were evaluated using experimental data and the energy IOL factor E_{orb}, shown in Figure 27.

. The results of the calculation are shown in Figure 39 and Figure 40. (For the deuterium thermal diffusivity, the 25-42% interval in the 0.5 MA shot is unavailable due to difficulty fitting the data to calculate this parameter, while the inner ρ = 0.05 of the 7-25% interval for the same shot is omitted due to numerical error arising from division by the derivative of a nearly flat temperature gradient.)

Neither the electron nor the deuterium thermal diffusivities vary much over the inter-ELM period, except near the separatrix. The deuterium thermal diffusivity is small and decreasing with radius throughout the flattop region.

There is a stronger inverse correlation of the thermal diffusivity with plasma current, especially near the separatrix. The magnitudes of the diffusivities increase significantly from ~1 to ~3 m^2s^{-1} with decreasing current from 1.5 to 0.5 MA. In the higher current shots, there is an increase in the edge diffusivity near the separatrix in the pre-ELM (80-99%) interval. In the lower current shots, the increase in χ_i in the pre-ELM interval (80-99%) profile near the separatrix is not observed, and the χ_i value at the separatrix decreases during the inter-ELM period in the 0.5 MA shot. The electron thermal diffusivities generally increase throughout the inter-ELM period throughout the edge, and the decrease in χ_i near the separatrix (due in part to IOL) is not seen in the electron χ_e. There is some slight evidence of a transport barrier in the electron diffusivity, but not in the ion thermal diffusivity.

10.2 IOL Effects on the Deuterium Thermal Diffusivity

As was the case for the ion transport quantities, the interpreted experimental thermal diffusivities are based on the assumption that 50% of the ions and ion energy crossing the separatrix outward (R_{loss}^{iol} = 0.5) on IOL loss orbits do not return to the plasma edge to be transported outward. Relative to this assumption of 50% return of IOL ions and energy, the usual neglect of IOL (implicit assumption that 100% of the ions and energy exiting the edge plasma returns to the edge plasma to be transported outwards

(R_{loss}^{iol} = 0.0)) leads to significant differences in the inferred experimental thermal diffusivities. In Figure 41, the interpreted deuterium thermal diffusivity profiles are shown for the 1 MA shot and R_{loss}^{iol} = 0.0. When compared with the 1 MA case in Figure 39, it is clear that similar to the ion transport quantities, excluding IOL effects causes the interpreted deuterium thermal diffusivities to be much larger near the separatrix, as the interpretation attempts to account for the energy being removed by IOL as energy being removed by thermal diffusion.

The shapes of the no IOL deuterium thermal diffusivity profiles in Figure 41 have a well structure in the sharp gradient region that has in the past been interpreted as a “transport barrier” created by a reduction in thermal diffusivity. However, once IOL is taken into account (shown in Figure 39), this well structure disappears for the deuterium thermal diffusivity profiles.

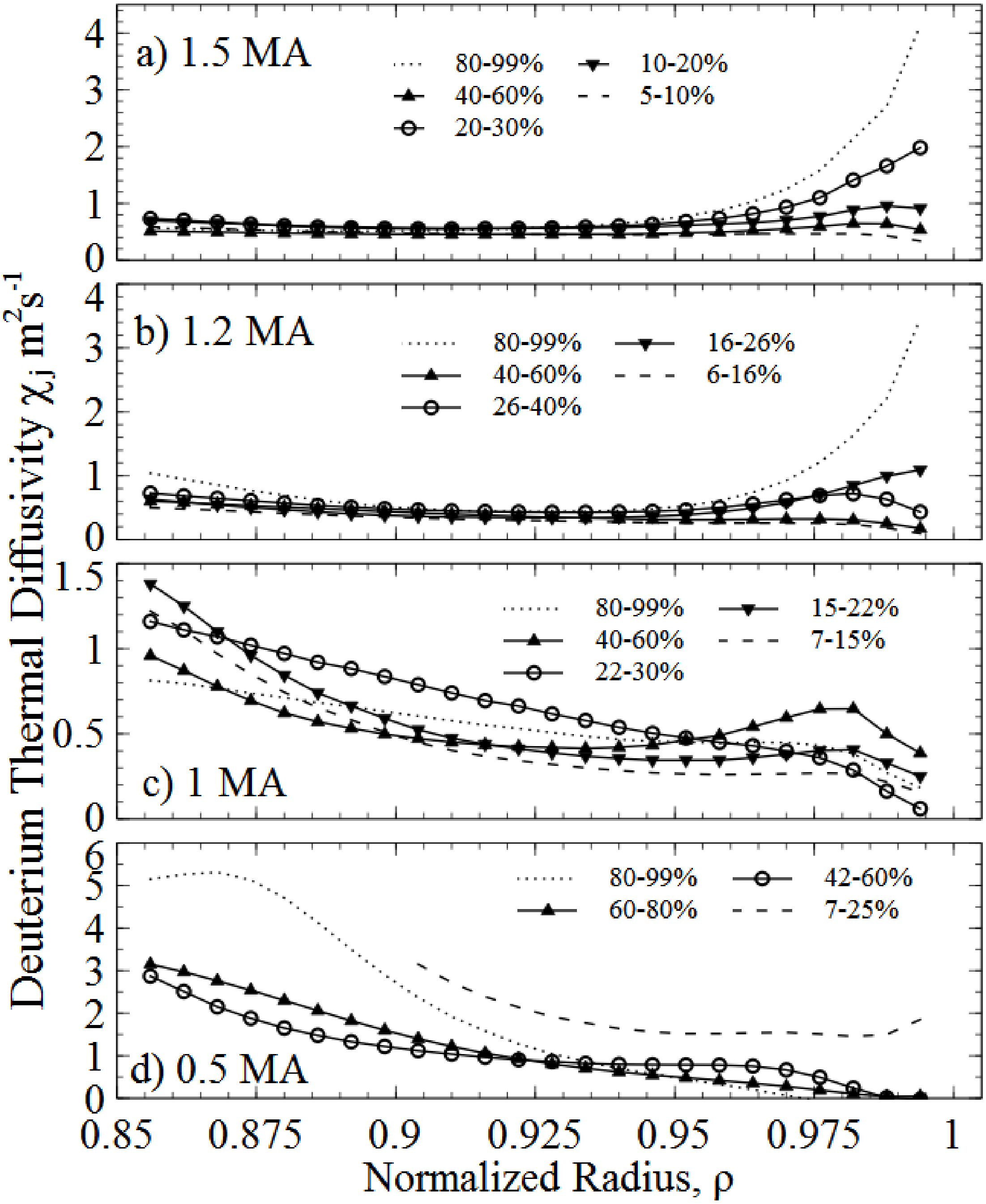


Figure 39: Deuterium Heat Diffusivity - The deuterium thermal heat diffusivity ($R_{loss}^{iol} = 0.5$).

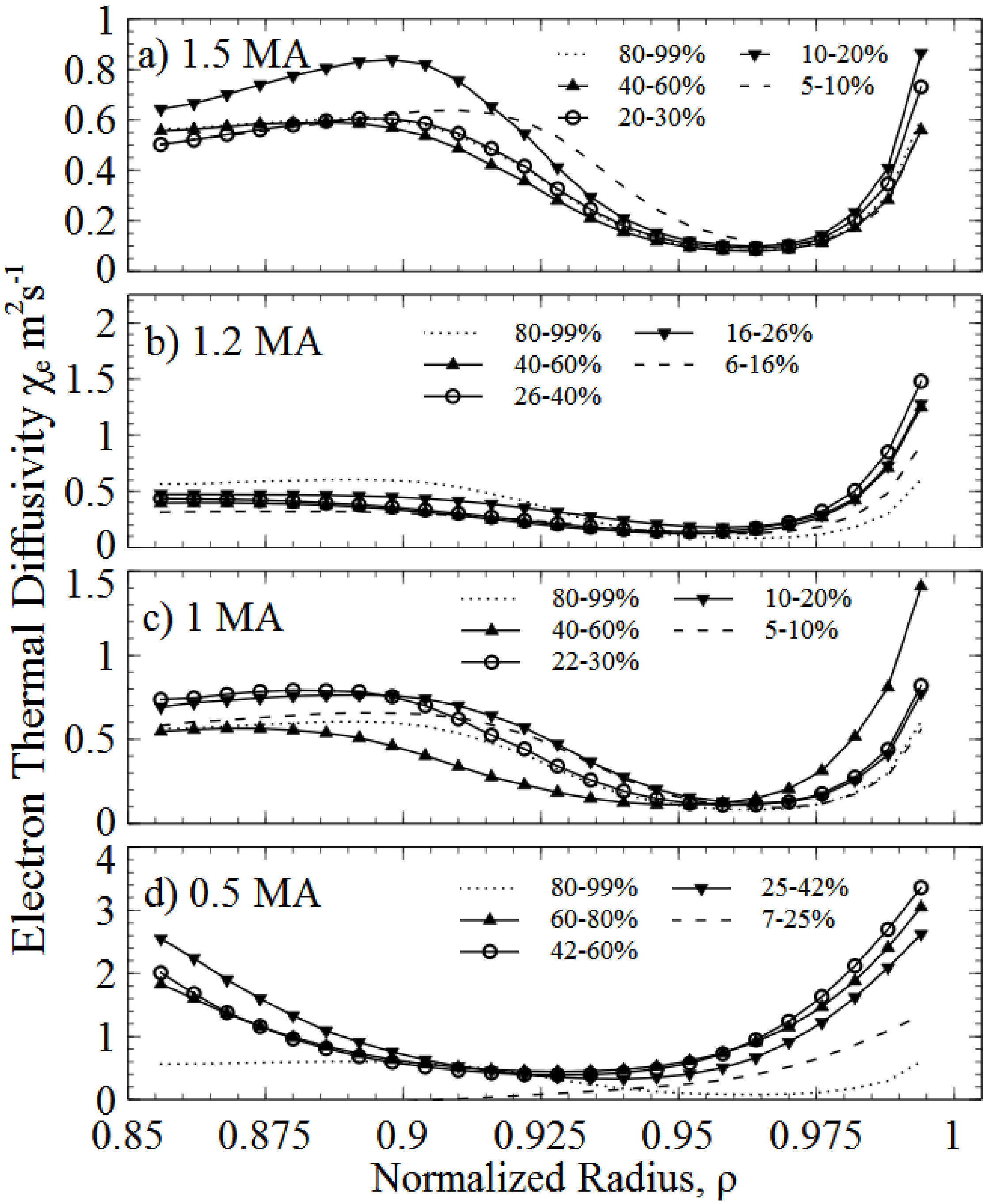


Figure 40: Electron Heat Diffusivity - The electron thermal heat diffusivity ($R_{loss}^{iol} = 0.5$).

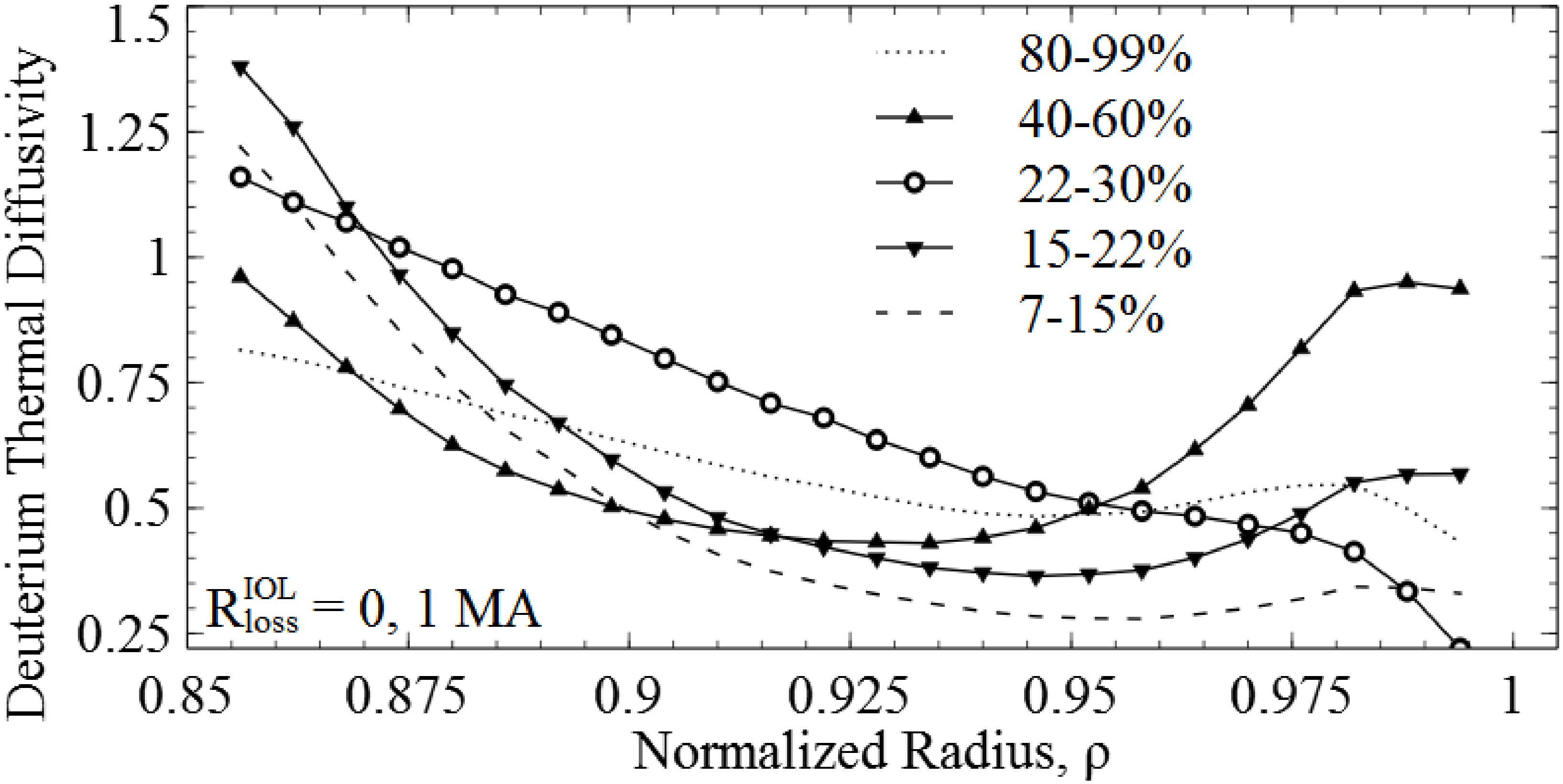


Figure 41: Thermal Diffusivity for $R_{loss}^{iol} = 0$ - The thermal diffusivity profiles for deuterium in the 1 MA shot and for $R_{loss}^{iol} = 0$ are shown. This assumes that all particles leaving the plasma through IOL return. Thise plots can be compared to the 1 MA plot in Figure 39 to determine the changes caused by decreasing R_{loss}^{iol} to 0 from 0.5.

CHAPTER 11

RESULTS AND CONCLUSIONS

Evolution of measured profiles of densities, temperatures and velocities in the edge pedestal region between successive ELM events has been analyzed and interpreted in terms of the constraints imposed by particle, momentum and energy balance in order to gain insights regarding the underlying evolution of transport processes in the edge pedestal between ELM events. In order to maximize the time resolution of the analysis while maintaining adequate data points within each time interval, the data from successive inter-ELM periods during an otherwise steady-state phase of the discharges were combined into a composite inter-ELM period for the purpose of increasing the number of data points in the analysis.

Available evidence strongly suggests that, after the particle and energy expulsion from the plasma edge caused by the ELM event, the plasma conditions in the edge pedestal rebuild rapidly (< 10 ms) to $\geq$ 50% of the pre-ELM values and then evolve further to the pre-ELM "asymptotic" values on a slower time scale. The available evidence also strongly supports a major causative role for inward electromagnetic pinch forces in the pedestal rebuild (which requires an inward flow of ions up a steep pressure gradient) and in the evolution of density profiles between ELMs (reduction of density just inside of the separatrix and inward shift of the steep gradient location). The achievable time resolution is adequate to resolve the evolution of plasma conditions after the initial rebuild to the asymptotic pre-ELM state, but not the initial rebuild (< 10 ms) itself While there is indication of a net inward ion particle flux over the initial pedestal rebuild period (< 10 ms) after the ELM event, the evolving net ion particle flux over most of the inter-ELM period has a relatively smaller outward value which is the difference between a large outward diffusive flux and a large inward electromagnetic pinch flux

Momentum balance prescribes a form for both the electromagnetic pinch velocity and the particle diffusion coefficient. The toroidal angular momentum transport “drag” frequency (due to viscosity, inertial effects, charge-exchange and anomalous processes), which was interpreted from the measured toroidal rotation velocity, is an important component of both the pinch velocity and the particle diffusion coefficient. This drag frequency dominated the diffusion coefficient in the edge pedestal in the lower current shots (but not in the higher collisionality 1.5 MA shot), while the interspecies collision frequency dominated in the flattop region (and throughout for the 1.5 MA shot). The radial electric field and the poloidal velocity VxB force dominated the electromagnetic pinch. The inward pinch velocity in the edge pedestal increased in magnitude over the inter-ELM period (suggesting that control of the radial electric field or the poloidal rotation might be a way to control or at least affect ELMs), and the magnitude of the inward pinch velocity was strongly inverse-correlated to the magnitude of the plasma current.

The Fourier heat conduction relation was used to interpret ion and electron thermal diffusivities from measured temperature gradients. Neither one varied substantially over the inter-ELM period. The electron thermal conductivity rose sharply at the separatrix, but otherwise there was not much structure in the thermal diffusivity profiles.

Ion-orbit-loss was calculated and taken into account in reducing the radial ion particle flux in the plasma, which in turn significantly affected the interpretation of particle and thermal transport by reducing the number of ions being transported to account for those lost by free-streaming. Reference [35] further discusses the effect of IOL on the interpretations.

In summary, the measured evolution of edge plasma profiles between ELMs can be understood in terms of particle, momentum and energy conservation constraints, using an ion toroidal angular momentum transport coefficient and ion and electron thermal diffusivities that were interpreted from experiment, and using experimental values of the rotation velocities and radial electrical field. Thus, identification of the theoretical mechanisms responsible for the ion toroidal angular momentum transport and for the ion and electron thermal diffusivities, together with the development of a calculation of rotation velocities and the radial electric field, would lead to a first-principles predictive capability for the evolution of plasma edge profiles between ELMs and perhaps to additional insights about the control of ELMs.

CHAPTER 12

FUTURE WORK

The explanatory power of this type of analysis could increase significantly if progress continues in several important research areas. Different approaches could also be taken to examine the same processes, and their findings could combine with those of this research, to present a clearer picture of edge transport. Some of this research is underway at Georgia Tech, while other projects have been proposed.

The CER system at DIII-D incorporates cutting-edge diagnostics and is very flexible to meet the needs of a wide array of research projects. The use of its new capabilities to measure deuterium rotation velocities and other properties in a research project similar to this one would be beneficial. This could enhance the accuracy of the interpretation of the all-important drag frequency and provide a direct measurement of the poloidal rotation velocity to use in calculating the pinch forces. The very high time resolution capabilities of the CER system and the ability to rapidly move the plasma across the edge CER viewing chords could be used in tandem. This would be very useful to future studies examining edge transport, IOL, and the radial electric field, and it may allow researchers to obtain some evidence of causation for the various edge transport relationships important to pedestal structure and ELM mitigation.

Further development of the ion orbit loss models and a better understanding of the effect of ion orbit loss on plasma rotation and transport would improve the explanatory power of this type of research. Given the large effect of ion orbit loss assumptions on interpreted transport quantities and plasma rotation, the capability to better estimate parameters such as R_{loss}^{iol} would be important for predictive modeling and other efforts. An improved understanding of IOL may also allow increased insights into the processes that determine the radial electric field structure, and uncover significant edge rotation effects.

The outsized effect that the drag frequency has on edge transport in the sharp gradient region has been well documented in this research. This parameter may hold the key to better control of the edge pedestal, its structure, and new methods of reliable ELM mitigation. There are a number of proposed drivers of this momentum transfer, and a better understanding of how they work, and how they relate to the drag frequency, may enable direct control of the drag frequency, and through it, edge transport and pedestal structure.

APPENDIX A

GUIDE TO PROFILE FITTING USING PROFILES.PY, THE DIII-D DATABASES, AND THE DIII-D NETWORK

A.1 Introduction

A.1.1 Notes

In this guide, “researcher” will refer to the reader, and a General Atomics (GA) contact (or other fitting expert) will be referred to using the terms “expert” or “GA contact”. The following things will also be assumed:

- The researcher has been granted access to the GA network, including the Cybele gateway server, the server nodes on the GA network, and the internal website.
- The researcher accesses the GA network through a virtual desktop (thin client) on the GA Cybele server, which is accessed remotely using the NoMachine client, from a Windows machine.
- FTP access to the user’s directory on the GA network is gained using Putty and WinSCP.
- The researcher is familiar with the GA data analysis tools, especially reviewplus, and is able to use them

Take note of the following statements.

- This guide is built from a Windows user experience. However, once on the GA servers, this should no longer matter.
- In the figures that accompany this guide, certain features are specified with circled red numbers. These highlighted features will be referenced by both the

figure number and the (red) location/feature number. For instance, Figure 44 Location 1 will be referenced as Figure 44.1.

- Once access to the GA network has been granted, instructions for installing the remote access tools are available at the following link:

  https://diii-d.gat.com/diii-d/Computing/internet_portal

- Documentation for the python scripts is available at this link (if the browser cannot access the linked page, see #3 in A.10):

  https://diii-d.gat.com/~osborne/python/Doc/python_d3d.html#profiles.py

A.1.2 The DIII-D diagnostics systems relevant to this guide

The Charge-Exchange Recombination system at DIII-D[28] measures ion properties, and the Thomson Scattering system[29] measures electron properties. For a detailed review of the inner workings of the CER system (other common acronyms for similar systems are CXR or CXRS), please see Reference [4] or Chapter 4. The CER system measures the impurity density, the ion temperature, and the carbon poloidal and toroidal rotation velocities. The Thomson system measures electron temperature and electron density. (Other important data is measured by additional diagnostics systems on DIII-D, but those values are less relevant to the task of fitting, and will not be reviewed here.)

The radial electric field is directly calculated from these five measured values using the equilibrium carbon radial force balance equation[25]. The electron pressure is calculated using the measured electron density and temperature. These eight measured and constructed properties (the electron density, temperature and pressure; the carbon density, temperature, and poloidal and toroidal rotation velocities; and the radial electric field) are involved in the fitting process. The data for the measured variables, collected directly from the diagnostics systems, is displayed in an x-y scatter plot with error bars, as is the electron pressure data. The radial electric field is displayed as a profile. All of these plots can all be seen in Figure 51.

A.1.3 The purpose of the fitting process

The measured data includes outliers, artifacts, and a number of other structures that partially obscure a clear trend. The goal of the curve fitting is to take a set of x-y data, and craft a smooth, fitted profile that replicates the overall trend of the data, taking into account physical constraints and a number of other concepts. The smooth, continuous fitted profiles can then be used for purposes poorly suited to experimental data. These purposes include taking derivatives, approximating values in areas where there is a gap in the measured data, and using the experimental data trends while minimizing the influence of artifacts, outliers, and erroneous measurements.

Despite the advantages of a smooth fitted profile, it is important to remember that the profile fitted to the data is not actually measured data. However, because it is assumed to accurately represent the measured data, it is imperative for the researcher to carefully approximate the measured data using the profiles.

A.1.4 The fitting process and the role of the expert

Due to the sometimes-imprecise nature of approximating a large group of points with a curve, and the semi-automatic nature of the fitting process, which prevents precise control of the profiles, experience is often necessary to ensure the fitting process produces a representative curve. This “touch” comes with time, experience, practice, and through gaining an understanding of how the fitting system works and responds to various inputs. The process of becoming proficient at curve fitting is not unlike learning to code in an elegant fashion.

As a beginner, the researcher will usually be unable to excel at curve fitting, and will require assistance and constructive criticism of their products from an expert, whose feedback is very important. The expert should generally have a good knowledge of how to identify legitimate data trends, and how to reproduce them with a fitted profile despite of numerous possible complications, such as artifacts or incorrect measurements. They also can help the researcher identify priorities for fitting, based on the type of research being conducted.

For up to a year, the researcher should consult regularly with an expert to discuss the researcher's efforts, determine what errors or omissions are being made, and identify methods of producing better fits. Consultation with the expert and seeking assistance from them is important for time efficiency. Without help, the inexperienced researcher may struggle with a problem for which an easy solution exists, or spend a significant amount of time working on a fit that poorly represents the data or is influenced by an erroneous measurement.

For even longer after learning the process, the researcher should check with the expert before finalizing any fits. This is useful both for feedback on technique and as a resource for particularly difficult fits. Discussions with other researchers who are also working on fitting are beneficial to all parties, as feedback, ideas, and solutions can be exchanged. Fellow researchers can also fulfill important roles as "sounding boards", and assist in solving problems, allowing all parties to gain expertise more quickly.

A.1.5 Data Limitations

There are two main external constraints on fits – the amount of data available to fit, and the known physical characteristics of the variable being fit. Both constraints are very important for accurate fitting, and their influences become clearer with experience.

Some data sets have less data than others to guide the fit, and researchers must respect this scarcity, avoiding overinterpretation of the existing data. This challenge is regularly faced in the edge pedestal region, but data may be scarce in other areas as well. This scarcity may be due to physical limitations of the data gathering systems, system settings, or operational choices. When these gaps occur in the pedestal region, where large gradients in plasma properties are found, it becomes more difficult to accurately fit the data in this important location. Nonetheless, it is important to avoid creating structure (a profile fit is characterized by structure, i.e. hills, valleys, cliffs, flat areas, etc.) in these gaps that is not directly correlated with the measured data. Low numbers of data points can also complicate dealing successfully with outliers, which appear much more influential in sparse data sets.

The known physical behavior of the parameter being fitted must also be taken into account. Often, due to outliers, systematic error, or radial gaps, data trends appear to suggest fit structure that is not realistic. Data sets may include two data points located near the same radial location but having very different y-values, implying a near-vertical gradient. A horizontal variation of this would involve two data points located near the same y-values in the pedestal region, but having different radial locations, implying a flat profile, which for many parameters, is unlikely to be found in the gradient region. The presence of these trends in the data usually suggests that one of the two measurements is inaccurate, or a gap in the radial data distribution is present. Knowledge of the general appearance and behavior of edge parameters often enable the identification of aberrations in the data trends. To address the presence of the aberration, it is best to consult the expert. They usually can provide assistance determining if the aberration is a systematic error that can be corrected, an outlier than can be ignored or removed, or a feature that must be included in the fit.

There are sources of variation within the diagnostics systems at DIII-D. Multiple sets of diagnostics are available for analysis, and for the CER system alone, three sets of viewing chords comprise the main CER data gathering system (see Figure 7). Slight systematic differences among data from these sets of diagnostics systems may not always be fully corrected, and may complicate the interpretation of data trends. The accuracy of diagnostics measurements may also be degraded for certain parameters or areas of the plasma. For this research, in most cases, the expert determined that the data from the tangential CER chords was slightly more reliable than the data from the vertical CER chords, and the data for at least one parameter was unreliable in the inner 50% of the plasma radius. The researcher should always consult with an expert to determine to what degree reliability needs to be considered when fitting a set of data. It is important to prioritize the influence of the most reliable data on the fit, and minimize the influence of unreliable data. The expert is often familiar with the reliability of various data sets for specific types of research, and they may have developed techniques to mitigate the impact on fits of specific types and patterns of problematic data.

A.1.6 Useful Techniques for Data Fitting

Throughout the fitting process, it is convenient to imagine a situation where the researcher is asked to explain the reasoning behind their fitting decisions to a neutral third party (i.e. these three data points formed an arc and an arc is expected to be in this region for this parameter, so I fit this data with an arc structure). If the researcher imagines that the third party may be skeptical of the reasoning given or decision made, it may be prudent to revisit the decision. This helps the researcher be conservative in the task of profile fitting. Conservatism is an ally in fitting, and it is best to err on the side of less information (structure) in the fit, rather than more, to avoid misinterpretation or over-interpretation.

A useful system is to assign a “grade” to the data fit, where a 100 is a perfect fit that is physically reasonable and captures all the structure in the data trends. The author generally considers a fitted profile grade of 91-96 (or A- to A) to describe a well-fit data set, and to indicate an appropriate point to cease work on the fit. A fit graded lower than 91 can generally be improved slightly to become a good fit, while a fit graded higher than 96 often indicates that over-interpretation of the data set trends may be present. This is a subjective concept, but it has been a valuable tool in the author’s fitting work.

A.2 Setting up GA access

A.2.1 The GA Network and Databases

The General Atomics network incorporates several types of permissions and access. The credentials used to access the various parts of the network generally have the same username and different passwords.

The gateway which allows access to the GA network is the Cybele server, which has a password or passphrase (the Cybele password). This is the server that runs the thin client remotely accessed through Nomachine. When working on the Cybele thin client, the user is essentially on a GA computer in the San Diego DIII-D network.

In order to log onto a server node, which is where the computing and other analysis is done at GA, the user’s server node password must be used. The server nodes include benten, venus, and several other servers, and the password is generally the same for each. The login commands will be described below. Once the user is logged onto a server node, it should be possible to run many of the data analysis tools that GA maintains, and access the MDSPlus database.

The website fusion.gat.com has an internal users portal which is accessed using the website password. (The server node password may serve as the website password, but the server node password may also need to be reset to a new website password, using the links on the website, to gain access.) There is a wealth of information available on the DIII-D internal website, including DIII-D system diagrams, software documentation, and experiment schedules and details.

Once logged onto the server nodes, the user can access several other servers used for various procedures and codes that each have their own passwords. Two of these specialty servers are relevant to this process and are discussed later. If access to any of these other servers is required, it is best to coordinate the request with a GA contact.

The GA network can be summarized for remote users as follows: Cybele = gateway; server node = functional servers; website = documentation and information; specialty servers = special tasks.

Basic remote access to the DIII-D network is available using a number of different options, the best of which is the NoMachine client. There is a guide to accessing the network remotely, and instructions for setting up the NoMachine client, at the following link:

https://diii-d.gat.com/diii-d/Computing/internet_portal

This page must be accessed using the GA website password. Also found there are instructions on configuring file transfer software, which allows remote access to the user's General Atomics directory. The user's directory on the General Atomics network is the location where the user's files are stored on the network. Many of the files automatically stored here by GA scripts are important, and must be transferred to the user's local computer.

The thin client on the Cybele server is a Linux Red Hat desktop. This desktop includes a Linux command line interface where most interactions with the General Atomics systems are carried out. In order to efficiently work within Linux, it is helpful to be familiar with a few basic command line operations. Given that very few students today recognize the commands "dosshell" or "ls", which were important commands for computer interaction in a bygone era, and whose cousins are still important in Linux command line operation, it may be useful to do a quick online refresher on Linux commands.

A.2.2 Steps to Enable Use of GA IDL and python scripts

There are a number of configuration files that must be present in a user's remote directory in order to fully operate the python and idl scripts that are integral to the fitting process. The types of files, configurations, and permissions necessary to use these programs vary, and it is best to coordinate with a contact at GA and/or GA tech support in order to determine how to obtain the necessary access. For fitting, clarify that in addition to using the basic GA data analysis tools, access to the python scripts and idl scripts is also required. There are some instructions on the fusion.gat.com website and in the python documentation for gaining this access, but they may not always work for remote users.

In order to operate the profiles.py and other python scripts, access to two of the specialty servers is required: the huez server, and a relational database server. A request for access to these servers should only be made once fitting activity begins, and the request should be coordinated with a General Atomics contact. Dr. Tom Osborne at GA is responsible for accounts on these two specialty servers, and he should be contacted to request an account. The researcher should request an account for the purposes of doing spline fits with “profiles.py”, and generally provide information including the user’s GA username, a brief overview of planned research, and the user’s advisor’s information.

If profile fitting will not be part of a user’s research, but access to fits completed by others is necessary, simple “read” access to the pgadmin3 tables may suffice. This access is generally available to collaborators using a simple username and password, found in the profiles.py or general GA python scripts documentation. Using this read access, users can find fits and shots that are useful and view them without needing to access the fitting routines.

A.3 Shot and Time Selection

Again, this document assumes that the researcher has access to the GA network through Nomachine, and is familiar with the main data analysis tools – efittools, reviewplus, and gaprofiles. To access these tools, bring up the command prompt on the remote desktop (through NoMachine on Cybele), and login to a server node using the command “ssh –Y *server name*”. In the example below, the newer, 64-bit server “venus” is accessed, which is one of several server nodes.

```
[*username*@cybele ~] $ ssh –Y venus
```

The server will then prompt the user for their venus server node password, which is usually the same for all the server nodes. Enter it, and a command prompt will open on the venus server node, where further commands can be entered.

NOTE A space and ampersand “ &” should always be entered after entering a program name (e.g. “program1 &”). This will allow commands to be entered at the prompt while the program is running. If this is not done, any program that is open will have to be exited to enter further commands at the prompt.

NOTE When using the GA tools and systems, and something does not work as expected, always examine the command line output of the program being run. It will usually provide information that can assist in solving the problem.

A.3.1 Shot Selection and Research Purpose

This is a topic that is unique to each individual. The profiles.py script is the main tool for fitting and has a large number of possible uses. For this guide, the profiles.py script will be used to consolidate data from different time periods into one composite time period, divide the composite period into intervals, and fit profiles to the data for each interval. The author has completed this process to study transport evolution between ELMs, and also has used the tools to analyze multiple successive periods of absolute time (studying transport evolution across the L-H transition). This guide will focus on the inter-ELM analysis method, as it is the one with which the author is most familiar, but most of the steps are generally applicable. In some cases where the procedure for analyzing time periods differs from that for interval analysis, additional instructions will be given.

A.3.2 Time Period Selection – Analysis of Time Blocks

If the goal of the research is to analyze blocks of time, or any other purpose for which the construction of a composite inter-ELM period from various time periods is not important, the endpoints of the period to be analyzed can simply be recorded, and the researcher may skip to A.4. However, it is still important to examine the time series data for the plasma, note the presence of any large aberrations during the selected time period, and consider their effects on the researcher's analyses. If a quasi-steady-state period unrelated to ELMs is being sought, a similar process as is described in the next subsection may be followed, but with the listed parameters being changed to those seen as important for the phenomenon under consideration.

A.3.3 Time Period Selection – Inter-ELM evolution

After the shot has been selected, the first step is to examine the macroscopic stability of the shot. This is important in many types of research examining time evolution of parameters. An important part of constructing a defensible composite inter-ELM period is to make sure the plasma is in a steady-state or quasi-steady state during the time examined. A quasi-steady-state plasma can be seen during ELMing H-mode in certain shots, when ELMs are destabilizing periodically, and very little else is happening in the plasma. Although this is not technically steady-state, if similar pedestal rebuilding trends follow each ELM, and the only things that are changing are related to the ELMs, it can be said that the plasma is in a quasi-steady-state. This assumption relies upon the inter-ELM periods in the quasi-steady-state periods being broadly similar in form and character.

A.3.3.1 Time Period Selection – Inter-ELM evolution – reviewplus setup

There are several variables generally examined to determine if the plasma is in a steady state. They can be accessed through the MDSPlus database using the reviewplus tool. In order to view them together, as is shown in the following graphics, the tool "reviewplus" is useful, and its name should be entered at the command prompt:

```
venusa 31: reviewplus &
```

Click through any popup screens and a screen similar to that shown in Figure 44 should appear. Go to Figure 44.1 and enter the relevant shot number, then click “update”. This will ensure that any variables displayed will be for the entered shot number, unless overridden in another window.

Go to Figure 44.2, and click “edit”, and in the dropdown menu, select “Set Signals”. Another window should pop up that looks like Figure 45 (but without the “Z Data Signal” fields filled in). At Figure 45.1 in the “Z Data Signal” field, on rows 0 through 5, enter the following signal names: “pinj”, “density”, “betan”, prmtan_neped”, “fs06”, and “n1rms”. Then enter “atime” in each of the five “X or Y Data Signal” fields. At Figure 45.2, click “apply”. The value “ok” should pop up in each of the five rows’ “status” fields, and the still open window shown in Figure 44 should fill in, and resemble Figure 46. If “error” is shown, there is a problem with the data for that parameter. Some parameters do not have data for every shot. However, if a specific parameter’s data is important to the research being conducted, and an “error” appears in the status field, or the data does not appear for the entire shot length, it is best to consult with the expert and/or check X.5 for possible solutions. To make the window in Figure 45 close, click “done” at Figure 45.2.

NOTE it is always preferable to close the windows in the GA data analysis tools without using the “x” in the top right hand corner. Finding the “done” or “close” option within the window will provide a better experience.

Once the window with the profiles appears, the zoom can be changed to get a clearer view. Under Figure 46.1, ensure the buttons for “zoom” and “crosshairs” are selected, and near Figure 46.2, maximize the screen. The coordinates of the cursor will be displayed at Figure 46.4; if they are not, the window is too large in relation to the NoMachine window, and the NoMachine window must be enlarged (see A.10 and the second note after this paragraph). Next, click with the left mouse button at the top left of the “pinj” data, hold the button, and drag the rectangle around all of the data, with a minimum of extra space in the box, as shown in Figure 46.3. Release the mouse button. This should increase the zoom on the data to a reasonable level, as seen in Figure 47.

NOTE In the reviewplus window, to go back to the previous zoom on a graph, the right mouse button acts as an “undo” button for changing graph views.

NOTE The DIII-D data analysis tool windows have a fixed size. Changing the size of the Nomachine window on the local desktop will not change their size. The only way they can be fully seen is if the local monitor’s resolution is sufficiently high, and the Nomachine window is set to be large enough. On the desktop seen Figure 47, the bottom of the reviewplus window is obstructed because the monitor’s resolution setting, 1280x720, is too small in the vertical direction (the full window can be seen in Figure 46). A vertical resolution of 800 pixels seems to be sufficient to display the full window. If buttons are missing, or a tool is difficult to understand, make sure that the full window is visible! For more details, see A.10.

A.3.3.2 Time period selection – Inter-ELM transport – the five variables

The five variables required to be in quasi-steady-state for inter-ELM edge transport analysis are shown in Figure 45 through Figure 48; they are the injected power, the line-average density, the normalized beta, the electron density at the top of the pedestal, the ELM diagnostic, and the RMS power diagnostic. Anomalies in any of the five variables generally disqualify the local area from being quasi-steady-state, but exceptions may be made in consultation with the researcher’s advisor or GA contact. The selected period of quasi-steady-state operation is usually the largest time period in the shot for which the five variables have overlapping periods of routine behavior (i.e. not continuously increasing, or with random large spikes, etc.). For the shot displayed in Figure 47, the overall quasi-steady-state period is from 2600-3800 ms.

pinj: injected power - It is important to ensure that the auxiliary power input to the plasma remains the same across the time selected for analysis. There may be inverse-delta-function-type artifacts in the profile (see in Figure 47), but these can usually be ignored. Also, there may exist periodic patterns, but as long as the variation within these patterns is relatively small, and they repeat themselves, they can be ignored. In Figure 47, there is a peak in injected power before 2000 ms. After this feature, there are some minor artifacts and a small pattern, but the injected power seems to be in steady state until the shot ends. A good steady state period for pinj in Figure 47 is from ~1800-5000 ms.

line-averaged density (density): This quantity is a measure of the plasma density. Although it fluctuates with ELMs, a series of ELM-induced patterns with similar minima, maxima, and drops, show that the plasma has entered quasi-steady-state H-mode operation. In Figure 47, the density throughout the ELM pattern seems to climb until about 2600 ms. At that time the patterns begin being roughly similar and stay at a relatively constant density until around 3800 ms, when a large dip breaks the pattern. A good quasi-steady-state period for density is 2600-3800 ms. This was the "controlling variable" for quasi-steady-state period selection in this shot, as it was in relative steady state the shortest time of any variable.

normalized beta (betan): Beta is a measure of the relative strengths of the kinetic pressure and the magnetic pressure. Again, fluctuations characteristic of building pressure gradients between ELMs are seen, but the values are relatively constant from around 2200 ms to about 3800 ms in Figure 47, where the large dip again ends the quasi-steady-state period. A good quasi-steady-state period for betan is 2200-3800 ms.

electron density at the top of the pedestal (prmtan_neped): this variable is the pedestal density measured at the point determined to be the top of the pedestal by a data fit. The location of the top of the pedestal is sometimes slightly inaccurate, but the parameter's accuracy is sufficient for the purposes of this process. In Figure 47, this variable has a consistent pattern with similar magnitude peaks and troughs from about 2000 ms to about 3900 ms, where the large drop is again seen. A good quasi-steady-state period for prmtan_neped is 2000-3900 ms.

filterscope channel 6 (fs06): This parameter is a diagnostic that is used to indicate the occurrence of an ELM through measuring the light emitted when charge-exchange interactions occur between the plasma ions and the neutrals outside the separatrix and near the divertor. There are a number of these diagnostics with different viewing locations, and the researcher’s advisor and GA contact should be consulted to determine which location provides the best data for the purpose of the research being done. In the case of studying inter-ELM periods, the fs06 parameter is a diagnostic close to the divertor, but outside the private flux region, making it a good indicator of ELMs. These signals normally approximate a delta function when viewed on 1 second timescale, which is realistic given the short duration of an ELM event. After the transition to H-mode around 1600 ms, the first ELM in Figure 47 is not very delta-like, but the next six are closer to this shape, until around 3800 ms, where the peaks broaden again. The large ELM just before 4000 ms is abnormally extended, and seems to be the catalyst for the large drops that mark the end of the quasi-steady-state periods for the other parameters. A larger ELM will have more area under the curve during the delta function-like signal. For reference, the ELMs in Figure 47 are very large ELM events. The period for fs06 in which ELMs were similar was from 2300-3800 ms, and each inter-ELM period during that time was viable for analysis, assuming the rest of the plasma was simultaneously in quasi-steady-state operation.

indicator of MHD instability in the plasma (n1rms): This variable indicates the level of MHD activity and instability in the plasma associated with various modes. Large fluctuations in this parameter can indicate other ongoing processes in the plasma that may affect the aforementioned and other important parameters in unpredictable ways. It is desirable to have a minimum area under the curve for this variable; flat lines on the x-axis are best, delta functions are ok, broad peaks are bad, and a bubble as seen near 4000 ms in Figure 47 is unacceptable. Although there is more MHD activity in this shot than is desirable, it seems that the period from about 1800 to about 3800 is relatively calm. The large bubble around 4000 ms in this parameter is likely a partial driver of the abnormalities in the other profiles seen around that same time, illustrating how MHD activity can affect many parameters. A good quasi-steady-state period for n1rms is from 1800 ms to 3800 ms.

After carefully examining each variable, and determining if overlapping steady-state periods exist and if they are of sufficient length to analyze the shot, it is important to double-check the selection with the researcher's advisor and GA contact. Be mindful of the purpose of the research, and try to select a time period that has the least opportunity for erroneous or abnormal readings, and also which provides the most data for analysis. For analysis of inter-ELM transport, it is generally good to start the time period just before an ELM and end it just after an ELM, so that a set of full inter-ELM periods is available. Also, the horizontal crosshair can be useful in trying to determine constancy in a variable. When these tasks are complete, record the beginning and ending times of the quasi-steady-state period that is to be used.

A.4 The Tools of the Trade - profiles.py, pgadmin3, and Profplot, and how to use them.

This is the point where the fitting process begins in earnest. The pgadmin3 program will be used to alter the inputs to a spline fit, the resulting data and profiles will be viewed on the Profplot tool, and the fits will be updated using the profiles.py script. Each of these tools will be discussed in detail in this section. Using the analogy of a piece of industrial equipment, think of profiles.py as the machinery, pgadmin3 as the control panel, and Profplot as the diagnostics display.

The structure of this section will follow the procedure for creating a run. First profiles.py will be introduced, then pgadmin3, and then Profplot. Next, using profiles.py to create a run will be reviewed, followed by instructions to view the run in pgadmin3 and Profplot. Finally, editing and fitting the data using the three tools will be covered.

A.4.1 profiles.py

The main function of the software is to gather all the diagnostic measurements for a certain time period or set of time periods, and display them together. Then, the electron data is automatically fit with a tanh fit, and the ion data is semi-automatically fit with a spline fit. The latter fits usually must be improved by the researcher. In each case, the script weights large clumps of data and attempts to create a curve that passes as close to the center of the error bar for as many data points as possible.

The rapidity of the Thomson system measurement of electron properties allows a large amount of data to be collected during even relatively short time periods, enabling the automatic tanh fit of the electron data to be relatively good in most cases. In contrast, the CER system provides the ion data, and the properties of the measurement techniques and electronics systems often generate much less data than the Thomson system for the same time period. Due to this reality, the data for the ions often exhibits trends that are harder to identify, and are more vulnerable to misidentification resulting from inaccurate measurements or outliers. For this reason, a user-adjustable spline fit is employed to fit the ion data. An automatic attempt is made to fit the data, but expert input and guidance from the researcher is generally necessary to adjust, correct, and tune the profiles in order to best represent the true data trends.

When manually fitting ion data, the semi-automatic spline fit takes the radial location of the specified spline knots, and attempts to run the curve through them, while still taking the presence and location of the data into consideration automatically. Therefore, the user has substantial, but ultimately limited control over the curve. This causes the experience of fitting the profile to the data to be quite difficult in many circumstances. The two main inputs used in fitting the ion data for each variable are the spline knots and the range of the fit, and for the electron data, the range can also be adjusted.

The radial location where the researcher wants to place the spline knots are specified by their radial coordinate. The range variable sets the radial bounds within which the script attempts to fit the curve to the data. If a clump of data points outside the separatrix (where the data may be less reliable) is causing a fit to have a non-physical slope near the separatrix, the range can be reduced so that the offending clump of points is not taken into account. There are a huge number of other inputs and capabilities of the profiles.py program, but the knots and range of the ion data fits are the inputs that researchers will spend the vast majority of their time altering.

The documentation for profiles.py is easily accessible, and it should be read a few times before beginning work. It can be found at the following url.

https://diii-d.gat.com/~osborne/python/Doc/python_d3d.html#profiles.py

If the browser is unable to access this page, please see A.10. Some basic documentation can be found by entering

```
profiles.py -h
```

A.4.2 pgadmin3

When the profiles.py script is run, it reads in the inputs and other properties from a spreadsheet-like table, where the inputs for each run are entered into a single row. The script then reads the data from the row, and performs its functions accordingly. The program used to access and edit the inputs is called pgadmin3, and it is an important part of the fitting procedure. To open pgadmin3, the program name should be entered at the command prompt.

```
venusa 32: pgadmin3 &
```

Click through any pop-ups, and after the program has opened, it becomes necessary to log into the huez server. Although a view of the program while not logged in is unavailable, Figure 49.1 shows the information to enter in order to log into the server and access the profiles.py inputs. This information is covered in the documentation for the python scripts at the aforementioned address. The profiles.py documentation is on the page, and a discussion about the huez server is at the top. If there is a problem, it may be necessary to consult a GA contact. Once the information has been entered, and the user is logged into the huez server, it is necessary to navigate the tree structure in Figure 49.2 until the “Tables” tab can be opened. Under the “Tables” tab is an item called “profile_runs”. Select this item, and then click the icon under Figure 49.3 with the purple funnel, which will bring up the window in Figure 49.4. In the blank area for filters, a variety of filters can be entered that correspond with fields in the input row. In order to simply look at the different profile fits already completed for a shot, enter

```
shot=*shot number*
```

in the field. The field follows boolean logic, so long “and” and ”or” statements can be used to filter the shots using the variables in the profiles.py input row. It is easiest to stick with simple “and” and/or “or” constructions, as the syntax becomes complicated with long statements. When using “or” statements, the entire conditional statement must be restated (i.e. shot=144987 and userid=’smithjs’ or shot=144987 and userid=’epstein’).

Upon clicking “OK” below Figure 49.4, the input table for profiles.py will open, and any runs that match the specified filter will be displayed. Figure 50 shows the results when “shot=144977” is entered into the filter field.

A.4.3 Creating a profiles.py run

First, the procedure for creating a run/row (a profiles.py run is controlled using a pgadmin3 row, hence the use of both run and row) will be reviewed. There are two main ways to create a run, either by copying an existing run, or creating a new run. For several reasons, it is recommended to copy an existing run made by a GA researcher. Much of the information that the user needs to specify can be taken from fits completed by a GA expert, and in most cases where a GA researcher has completed a fit, certain settings will be present that improve the quality of the data to be fit. This includes eradicating malfunctioning diagnostic chords and preselecting the data sets that give the most accurate, clean, and reliable view of the trends present for the selected shot. These procedures are beyond the scope of this paper, so it is recommended to copy a row with certain fields pre-filled.

There are also disadvantages to copying the data from a different run. Different data settings are used for different purposes, and the data analysis goals of the other research’s work may not line up with the ones for the current project. It is always important to check with a GA contact to ensure the suitability of the values that will be copied over from a previous run.

Setting up a new run from nothing has the disadvantage of resetting all values to default values. The default values are discussed in the documentation for profiles.py. The choice whether to start a new row, or copy a previous row, and how to update the various settings should be discussed with a GA contact.

In order to copy a row in pgadmin3, the shot, timeid, runid, and userid values must be known. To create a new row, at least the shot, timeid, runid, tmin and tmax values must be known. The process to determine these variables is described in the following paragraphs, with example entries:

shot: The shot number (e.g. 189889), should be already known.

timeid: This value is customarily set at the value of the midpoint of the maximum and minimum time being examined. So, if the time period from 2000 to 4000 ms is being examined, the timeid variable should be set to 3000.

runid: This variable is a text string to designate the purpose of the run, and distinguish it from other runs. It is required to limit the runid of a row to five alphanumeric characters, as any more will cause difficulty in retrieving the data and writing it to the MDSPlus database. If a row were examining the first 20-30% of a composite inter-ELM period, a good runid choice would be e2030 (ELM 20-30%), or e203a for a second iteration of the 20-30% fit.

userid: This is automatically set to identify the user who creates the row, and only they can edit or run the row.

In order to copy a row in pgadmin3, the shot, timeid, runid, and userid of the original row must be specified, in addition to the shot, timeid, and runid of the new row to be created. These values are all shown in the pgadmin3 table in Figure 50. The userid of the user making the row is automatically entered into the row when it is created, and cannot be changed. For example, if the run to be copied was made by user “epstein” for shot 189889 with a timeid of 3050 and a runid of j2030, and the user wished to create a new row for shot 189889 (a run can also be copied to a different shot, although this may create problems) with a timeid of 3000 and runid of s2030, the command to copy the row would be:

```
profiles.py –s 189889 timeid=3000 runid='s2030' copy_shot=189889 copy_timeid=3050
copy_runid='j2030' copy_userid='epstein'
```

After running this command, a message should appear confirming the creation of the new row with the same parameters as the original row in the command prompt.

NOTE If any process, window, or tool is not functioning properly, or as it is expected to, check the command line window, and see if the tools are producing error messages that may help identify the problem.

In order to create a new row in pgadmin3, the shot, timeid, runid, tmin, and tmax of the run to be created must be specified.

tmin: This is the starting time for the analysis period (for inter-ELM analysis, the quasi-steady-state period).

tmax: This is the ending time for the analysis period.

Given a tmin of 2000 and a tmax of 4000, shot 189889, and a runid of s2030, the correct timeid would be 3000, and the command to set up a new pgadmin3 row as specified would be:

```
profiles.py –s 189889 timeid=3000 runid='s2030' tmin=2000 tmax=4000
```

This command should result in a message saying that the computer is setting up a row to control profile.py for shot 189889, and listing the input parameters for confirmation.

Once the row has been created, it can be viewed in the pgadmin3 program. To locate it, follow the direction in A.4.2 once again, but enter in the filter "shot=189889 and userid=*researcher userid* and timeid=3000 and runid='j2030'" - the properties of the run just created in the previous paragraphs. This should bring up the row in pgadmin3 corresponding to the new run.

A.4.4 Running the profiles.py script for the first time on a row

Before running the script for the first time, there are several additional checks of field values in the pgadmin3 row that should be done in order to ease the process and ensure its accuracy. Many of the values in the fields are automatically generated and a field should never be changed without reason or just for convenience. In pgadmin3, the field name is listed in the first row, and the format for the entries is listed immediately below. Only the creator of a row can edit the row. Changing a value, then pressing enter/return, will save the value in the spreadsheet.

Note that in pgadmin3 the "undo" function is limited, so take care to record previous values when changing a value. To copy and paste in pgadmin3 is slightly more complicated than usual, as a selection must be made inside a field. Highlight the value in the field, and use either the keyboard shortcuts (ctrl+C, ctrl+V) or the menu inside the program to copy and paste.

For a new row, the default values for each of the variables in the row will be present, and they are discussed in the profile.py documentation. It is important to check each of the following variables, as well as any others recommended by the expert, to ensure that they have a proper value. For copied rows, several values need to be changed to ensure a good run, and they can be identified by checking with the expert. Also, a common default value that commands the program to automatically determine the value for a field is "-1", and this command can be used with many fields.

NOTE If using the number pad to input values, number lock must be on to input values in pgadmin3, but must be off to input values in the Profplot.py tool. If number pad keystrokes are not having their expected effect, check the status of the number lock.

complete: When profiles.py is run from the command prompt, it will run for each row created by the current user with a value of 0 in the "complete" field. If the value is anything other than 0, profiles.py will not run the row. After completing a final run where the fit data is written to the MDSPlus database, profiles.py will automatically set this variable to 1. This field is often useful to denote the status of the fitting of a row. If work is ongoing, then a -2 could be used, but if it is complete and awaiting finalization, a -5, etc. Feel free to create a system of notation, but do not manually set the value to 1, as this has a high likelihood of causing confusion.

NOTE If a “final run” is interrupted, often the “complete” field will be changed to 1 in the server, but not show a “1” value to the user, leaving them to wonder why profiles.py is not running using that row when “complete” appears to be 0. To fix this, it is necessary to manually set the “complete” field to some integer, save the value, then change it back to 0, save the value again, then try to run profiles.py.

shot: This is the shot number.

timeid: This value is customarily set at the value at the midpoint of the maximum and minimum time being examined.

runid: This is a label related to the specific run.

userid: This is automatically set to identify the user who created the row, and only they can edit or run the row.

write_mds: This variable should be set to 0 until the fitting process is fully complete and the user’s GA contact gives their approval of the fits. Once this variable is set to 1, and profiles.py is run, the data from the fitted profile will be written to MDSPlus for other researchers to use. It is set to 0 by default for a newly created row, but may not be zero for a row copied from another run. Running profiles.py with this value as 1 will set the “complete” variable to 1 also. Writing data to MDSPlus should only be done once the fits are complete, as there are a limited number of slots for runs in the database for each shot, and the data on MDSPlus is difficult to delete as a remote researcher. Check with the GA contact before writing to MDSPlus. It is usually possible to overwrite an older MDSPlus row for the same shot, timeid, and runid without a problem. This may be useful when an older, complete fit needs to be updated to take new information into account, or in similar scenarios.

*tmin: This is the lower bound of the time period being analyzed. If analyzing inter-ELM periods, this should be set to the beginning of the quasi-steady-state period of the shot. If not, then this can be set to the starting time of the analysis period.

*tmax: This is the upper bound of the time period being analyzed. Similar to the “tmin” value, this will have different values depending on the type of analysis being conducted.

*elmsync: This is a very important variable when using profiles.py. If analyzing inter-ELM time slices, then this should be set to 1. “profiles.py” will automatically detect inter-ELM periods between “tmin” and “tmax”, and combine the data measured during them into a composite inter-ELM period. If other types of research are being pursued, this value should be set to 0. Setting this variable to zero will simply combine all measurements between “tmin” and “tmax” into a composite period, with no regard to ELMs or other phenomena that occur.

write_mds: This should be set to zero for the first run of a row, and should remain at 0 until fitting is complete

adjust_zts: This should be set to -1, so that profiles.py will automatically determine its value.

shift_ions: this should be set to 0

which_prof_id: this should be set to “tisplvtspl”

run_onetwo: this should be set to 0 until it is time to write the complete fits to MDSPlus

nubeam: this should be set to 1

ne_co2norm: this should be set to -1, so that profiles.py will automatically determine its value

*elm_phase_min: when examining inter-ELM periods, and elmsync = 1, this and elm_phase_max determine what percentage of the composite inter-ELM period will be represented by the current row. A decimal is input to determine the starting point of the fraction of the composite inter-ELM period represented by the current row. It is generally best to have a large difference between the two values avoid complications. A good starting point for this variable is 0% (0)

*elm_phase_max: similar to elm_phase_min, this is the ending point of the fraction of the composite inter-ELM period being represented by this row. A good initial value for this variable is 30% (30% = 0.3).

Once these variables have been checked, and set to the correct values when necessary, the row can be run. To do so, enter the command

```
v “profiles.py –r *shot number*”
```

at the server node command prompt, substituting the desired shot number into the command. Recall that all the runs with “complete” fields of 0 for the referenced shot will be run consecutively after giving this command, so be sure to only have a “0” value in the “complete” fields of those rows that are intended to be run. Additionally, only rows created by the user running profiles.py will be run for the shot. Do not forget the “v” or the quotation marks, which ensure that the computational load of the run is balanced across available server nodes.

Running profiles.py for a row with run_onetwo = 0 usually takes 3-5 minutes. If the run completes successfully, a message to that effect will be printed to the command line window, and the fit data will be written to the user’s remote directory. If it does not succeed, an error message will be printed to the command line window, and nothing will be written to the remote directory. A failure to complete a run will be addressed in A.5.1. If the run completes successfully, proceed to the next section.

It is useful to watch the text that is written to the terminal screen, as it updates the viewer on the status of the run, and reiterates much of the information entered in the row, like spline knots. In addition, other information, such as the interim value of ne_co2norm (which is only written to pgadmin during the process of writing the data to MDSPlus), can be found by looking though the text that is output at the command line.

When running profiles.py for a final time, and writing the data to MDSPlus while running onetwo during the profiles.py run, the run may take considerably longer than a normal run. In the event that the run takes longer than an hour, the user should manually cancel the shot using the “CTRL+C” combination in the terminal window and try again, as the run may never complete. If this problem happens routinely, and a run allowed to continue for some hours still fails, the expert should be consulted.

A.4.5 The Profplot graphing tool

Profplot is a graphing tool that is currently the best option available to view the data and fits. It can be temperamental and has a steep learning curve, but it is lightweight and powerful. It can be run by entering the program name at the server node command prompt, remembering to capitalize the first letter.

```
Profplot
```

If Profplot opens with windows, the user must click on the title bar of “PGPLOT Window 1” and press “Alt+e”. Only after completing this step will interaction with the input panel of Profplot be possible. It is also necessary to close the graph windows. This can be viewed as either interacting with the graphs, a mode entered when the graphs are drawn onscreen, or interacting with the control panel, which is achieved by switching modes. To re-enter the graph interaction mode, the graphs must be plotted again from the control panel, upon which interaction with the control panel is no longer possible. Both modes cannot be active simultaneously. To recap, Profplot only allows user interaction with one part at a time. If interacting with the input panel, the user must click "plot" to replot the graphs, and interact with them. If interacting with the graphs, the user must click on the title bar of "PGPLOT Window 1" and press "Alt+e" to be able to interact with the input panel again or close graph windows. The graph windows in "PGPLOT Window 1" can be maximized by maximizing the "PGPLOT Window 1", switching back to control panel mode, and replotting the graphs, which will then fill the maximized data. Interaction is only possible with "PGPLOT Window 1". All other graph windows may only be viewed and not modified. However, it is possible to change the graphs that appear in "PGPLOT Window 1".

If "Plot" is clicked in the Profplot control panel, and nothing happens, check the command line window. If an error is being shown, part of the profiles.py run has failed, and this issue addressed in A.5.1.

The input panel (Figure 51) for Profplot will be discussed from top to bottom. The "shot", "Timeid", and "Runid" fields should be filled in with the appropriate data for the run to be viewed. If the run is already written to the MDS server, then the radio button "MDS+" should be checked. If the user wishes to view another user's run, but it is not written to the MDS+ data server, it will not be viewable. If the run to be viewed is not yet written to MDS+, and has been created by the current user, the radio button "File" must be selected to be able to view it. The "Profile Plots" box should be checked in order to view the data to be fit and the existing profiles. To interact with other windows (checkboxes in Figure 51.1, i.e. "ELM time plot", "Er, v*ExB Plots", etc.), and to do more than just view them, they must be the first box checked. For instance, if interaction with the "ELM time plot" window is desired, the "Profile Plots" box must be unchecked in Figure 51.1. This will cause “PGPLOT Window 1” to appear with the “ELM time plot” instead of the “Profile plots”, allowing interaction.

It is very important that the radio button next to “rhob” be checked. Profplot defaults to “psin”, but “rhob” is the radial scale used in GTEDGE, so it is the one that needs to be used when fitting the data. The small but important difference between the two is that rhob is a normalized minor radius, whereas psin is a radial measurement based on percent of flux surface enclosed. “psimin” and “psimax” set the minimum and maximum value for the x-axis that will be displayed in each graph.

The “ELM time plot” option displays the time characteristics of a run. Figure 10 shows a sample ELM time plot window. This view is mainly useful when examining inter-ELM phenomena, as it gives a graphical representation of the time periods being analyzed for the current row/run. The green boxes are the times being combined into a composite period, the pink crossed circles are the locations of CER measurements in time, the red circles are the “beginning” of the inter-ELM period, and the blue/black signal is the fs06 signal described earlier.

The “Er,v*ExB Plots” box generates the graph for the radial electric field, as well as a number of other variables, and should be checked. Everything else should generally be left as is in Figure 51. Beginners should keep the first three boxes checked to display these plots.

The “overlay” function is useful for determining the effects of changing fitting parameters. When this box is checked, and “plot” is clicked, the current data and profiles are displayed on the same plot as the previous data and profiles, but in a different color and trendline style. This can be useful in understanding the difference made by small changes to spline knot sets, or when directly comparing a current data set and fit to a data set and fit completed by an expert. It is invaluable when changing the fitting parameters discussed in A.6.

After clicking “Plot”, the graphs should show up onscreen, or be updated if they were already present. Similar to reviewplus, a left click and hold creates a zoom box, and a right click is an “undo zoom” button. The views seen in the “PGPLOT Windows” are also deposited in the user’s remote directory as a color postscript file. This feature can be used to send high-quality reproductions of graphs to colleagues before they are written to MDS+, and to create a image with a customized zoom. The file will be written to the user’s directory as follows

p*shot number*_*timeid*_*runid**graph number*.cps

The graph numbers start with 0 (PGPLOT Window 1 is graph number 0), and the PGPLOT Window 1 picture for the run of shot 189889 with timeid = 3000 and runid = ‘j2030’ would be written to the user’s remote directory as “p189889_3000_j20300.cps” After retrieving the file using a file transfer program like winscp, remove the “c” in the file extension to read the file using a general postscript file viewer. It may take several minutes for the file to appear in the user’s folder after the figures are generated using Profplot.

A.5 Advanced Setup

A.5.1 Errors in the profiles.py run

Errors in the profiles.py run are usually discovered in one of two ways. Either the profiles.py run from **Error! Reference source not found.** does not complete and gives an error message in the command line window, or an error is printed to the command line window while trying to graph the results of a run using Profplot. There are many and sundry reasons why an error may occur, but this guide will only attempt to discuss two of the most common causes. For other types of errors, it is necessary to seek the assistance of the expert.

A.5.1.1 Data Errors

These types of errors occur when there is an insufficient amount of data in the time period specified for analysis. The profiles.py script requires a certain amount of data to calculate the values of indirect measurements, and if an insufficient amount of data is present, an error will occur. These errors are usually directly related to the number of measurements inside a given analysis period. This can usually be solved by increasing the cumulative length of the analysis period.

For a run that represents a fraction of a composite inter-ELM period, a researcher can try several solutions:

- increase the length of time of the quasi-steady-state period by slightly relaxing the boundaries, thereby increasing the number of whole inter-ELM periods for analysis, and increasing the amount of data in the fraction of the composite inter-ELM period
- increase the width of the specified fraction of the composite inter-ELM period (designated using the elm_phase_min and elm_phase_max variables in the pgadmin3 row)
- change the data selection criteria by altering the cer_twin field (discussed in A.6)
- allow a wider variety of ELMs to be considered by altering the elm_size_min and elm_size_max fields (discussed in A.6)

It is clear that all of these solutions serve to increase the amount of data in the composite period. An easy way to determine if the quantity of data is problematic is to set the limits of the composite period to be very large, such as 10-90% of a composite inter-ELM period.

If the composite period consists of a block of time, rather than a set of fractions of inter-ELM periods, and errors occur, then similar actions can be taken that increase the amount of data available for analysis. A good rule is that for each block of time, there must be at least three CER measurements to obtain sufficient data. The time characteristics of the CER measurements are further discussed under “cer_twin” in A.6. To learn the frequency of either CER or TS measurements, the graphs in reviewplus must be altered to not show a continuous trend, but instead discrete measured points. This can be done by locating a variable known to be measured by the CER or TS systems, and plotting that variable in reviewplus, then removing the trendline, and examining the results. Two useful parameters for this are the signals “cerftit9” and “tste_tan”. The menu to adjust the plot properties can be accessed by right clicking on the reviewplus plots, holding the button down, and choosing the “ranges dialog” option. If there are at least three TS and three CER data points in a block of time, and profiles.py runs successfully, it has sufficient data for analysis.

Another common data error is when the profiles.py script runs, and the graphs are displayed, but the data is unfittable either due to its sparseness or the insistence of the profile being fit to maintain a non-physical shape such as a straight line or a structure having a maximum value 1000% of the highest measured value for the parameter. In those cases, additional data must be used to succeed in fitting.

In all of the following examples of this problem, the profiles have not been fit to the data, the left set of figures is the tanh-fit electron data graphs, and the right set of plots are the CER spline-fit ion data graphs, which are the subject of this guide. An example of a period with too little data for reliable fitting is shown in Figure 54. Note that in the outer 20% of the upper right hand graph there is essentially no data to fit. A zoomed view of the second data trend in the lower right hand graph (the gray points), which must also be fit, is shown in Figure 55. This data set was successfully processed by profiles.py, but there is an insufficient quantity to reliably fit with spline profiles. Note that if there is an insufficient quantity of data for just one parameter, the entire data set must be broadened. Figure 56 and Figure 57 show a data set that barely has enough data to fit, and ordinarily would be widened unless no more data was available. Figure 58 and Figure 59 show a data set that is a bit sparse near the separatrix, but generally has enough data to reliably fit.

A.5.1.2 Fitting Errors

These types of errors are more difficult to detect on the initial run, but easier to detect during fitting. They occur when the fit of the data as specified in the pgadmin3 row (discussed in the next section) causes numerical errors in the profiles.py run, which end the profiles.py run prematurely. A row with this error also may be run by profile.py with little problem, but not display in Profplot. This type of error can often be distinguished from data scarcity errors by the process referenced above of increasing the analysis period to be very large. If the error still occurs, a fitting error is likely to blame.

This error is usually associated with an overabundance of spline knots in a given fit, unusual values for the fit range, an input error, or an unusual data formation. When copying fitted rows from one shot to another this error often occurs, as the setting and knots for the original shot may not work well with the data from the second shot. It can also be the result of an accidental keystroke, such as setting the range for a fit from rho = 0 to rho = 102 rather than rho = 1.02.

When this error is encountered on the first run, it is best to review the values in the pgadmin3 row, and reduce the complexity of the fitting instructions. For instance, if this error is occurring, and the spline knots for a fit (that were copied over from an original row) are {0,0.5,0.55,0.9,0.92,0.921,0.924,0.95,0.99,0.991,1.02}, a solution may be to reset the spline knot set to {0,0.2,0.4,0.6,0.8,1.0}. Unusual ranges also trigger this issue often, and setting the range to {0,1.02} from whatever it previously was may solve the issue. If this error is encountered during fitting, it is usually best to go back a step, and try to understand what action caused the error to occur. Narrowly spaced {0.921,0.922,0.923} or excessive spline knot placement is a common culprit. If these steps do not solve the problem, it is best to seek help from the expert.

A.5.2 Accessing the actual time data for inter-ELM analysis with profiles.py, and its uses

Figure 53 illustrates an important function of the profiles.py script. It locates the ELMs based on the fs06 D_α signal and the elm_size_min and elm_size_max variables, and defines each complete inter-ELM period that occurs during the analysis period, during which the plasma should be in quasi-steady-state. Then, it finds the portion of each inter-ELM period specified with the elm_phase_min and elm_phase_max variables, and selects the data measured during each, according to the criteria set by cer_twin (discussed in A.6). Then, all the data measured during each fractional inter-ELM period is combined into a composite fractional inter-ELM period.

In order to complete this task, the profiles.py script records the ELM starting and ending times, and the starting and ending times of the fractional width of each inter-ELM period, as specified using elm_phase_min and elm_phase_max. These times are written to a file in the user’s remote directory that is designated the same way as the postscript files that contain the figures from Profplot. For a run in shot 189889 designated by timeid = 3000 and runid = ‘j2030’, when this row is run, a file is placed in the remote directory named time_windows_189889_3000_j2030

This contains the starting and ending points of the row-specified fraction of the inter-ELM periods in the analysis period for shot 189889. This data is required to use the automation process for the input file generation and input file balancing for GTEDGE, so it is useful to have these files on hand. However, by varying the elm_phase_min and elm_phase_max values, the user can obtain time coordinates for any fractional point in any inter-ELM period in the analysis period of the shot. Values of 0 and 1 can be entered to obtain the lengths of each inter-ELM period, and other techniques can be used to gather such data which is useful for many purposes.

It should be noted that profiles.py takes 0% of the inter-ELM periods (i.e. the starting points of the inter-ELM periods) to be at the locations of the red circles in Figure 53, which are located at the peak of the fs06 D_α signal. However, it is abundantly clear that the major transport disruption caused by the ELM takes some time to dissipate, and any analysis of inter-ELM transport which starts at 0% will be polluted by intra-ELM event transport. To determine where the ELM event ends, for each leading ELM event, requires a careful examination of the fs06 D_α signal, and noting where the fs06 D_α signal again approaches steady-state after each leading ELM. If the “ending time” of each leading ELM event is recorded, then it can be compared against the beginning time of the analysis period for the corresponding inter-ELM period. For instance, if the second ELM in Figure 53 “ended” at 2615 ms, but the 0%-10% (first) fractional of the inter-ELM period following that ELM began at 2600 ms, there would almost certainly be at least a 10-15 ms overlap between the fractional “inter-ELM” analysis period and the actual ELM event, polluting the transport analysis. The starting time of the first fractional inter-ELM period would have to be delayed to 5%-10%, or 7.5%-10%, in order to avoid being contemporaneous with the ELM event.

This is just one example of the usefulness of the time_windows data generated by the profiles.py script for each run. A second example is a part of the process described in A.5.3. In sum, the profiles.py script can be queried for the starting and ending times of the analysis periods it uses, whether fractional composite inter-ELM periods, or blocks of time. That concrete time data can then be used to examine experimental data and determine what occurs inside the analysis period.

A.5.3 Setting up custom time slices for inter-ELM analysis

A feature of the profiles.py script is the ability to set a block of time, or a set of time blocks to be made into a composite data set. When elmsyncid is set to 0, the measurements during a block of time are combined into a composite data set, and when it is set to 1, the script calculates where the fractional inter-ELM periods are located in time based on the elm_phase_min and elm_phase_max values, and combines those blocks of time to produce a composite data set. However, it is also possible to provide profiles.py with a set of time periods, and use the script to combine these specified time periods into a composite data set. This is useful for many purposes, and one is illustrated in the fifth graph of Figure 48.

In this shot, many of the ELMs exhibited “dithering” behavior[18], characterized by a series of smaller fs06 D_α peaks after the initial one indicating a Type I ELM. These inter-ELM periods were unsuitable for analysis, but it was important to be able to use this shot to study inter-ELM transport evolution. To do so, those inter-ELM periods which did not exhibit dithering (such as the one between 3100 and 3200 ms), were selected for inter-ELM transport analysis. By running profiles.py on the period from 1500 ms to 5100 ms on this shot with elm_phase_min = 0 and elm_phase_max = 1, profiles.py generated list of starting and ending times for each inter-ELM period between the two times. The beginning and ending times were compared to the fs06 D_α signal, and those inter-ELM periods without dithering behavior were selected. Next, the process described in the previous section was used to ensure that the first fractional inter-ELM analysis period did not overlap with any of the leading dither-free ELMs, and the widths of the fractional composite inter-ELM periods were set. Using the starting and ending times of each inter-ELM period provided by profiles.py, the ending time of each leading ELM, and an Excel spreadsheet, a set of divisions of the composite inter-ELM period that were useful to the research being done were specified. A time_windows file was then constructed with the appropriate syntax and name, with the beginning and ending time of each fraction of each inter-ELM period. The timeid and runid must match a row previously created using normal setup procedures. Changes to specific variables in the pgadmin row that were necessary for profiles.py to accept the external time_windows file are outlined below. Profiles.py was then run to combine this set of time periods into a composite time period, and fitting commenced normally. However, if a custom set of intervals is used, the “ELM time plot” in Profplot becomes useless.

In order for profiles.py to accept an externally generated time_windows file that specifies a set of intervals to be combined into a composite inter-ELM period, the “tmin”, “tmax”, “elmsyncid” and “elmsync” values must be changed. Although the documentation specifies a certain method, the author worked out a slightly different procedure that worked successfully. Set the variables as follows: tmin = -1, tmax = -1, elmsync = -1, and elmsyncid = time_windows

Then, the beginning and ending times of the desired intervals must be written to a text file in a two column format with the file name

time_windows_*shot number*_*timeid*_*runid*

There is no file extension (e.g. filename not filename.txt) for this file. The row created in pgadmin3 must have the same labels as the time_windows file (e.g. shot number, timeid, and runid – if this is not the case, an error will appear when trying to run profiles.py for that shot). The time_windows file must also be placed in the user's remote directory. Having completed these steps, profiles.py should work well with custom sets of time intervals.

A.6 The pgadmin3 row and its entries

Once the setting up of a new row in pgadmin3 is completed, good fractional composite inter-ELM periods are found, and profiles.py has been run for each row (one row exists per fractional composite inter-ELM period), the fitting process begins. It is recommended to set up each row, and run profiles.py for each of them, before beginning to fit the data for any row. This is an excellent time to review the profiles.py documentation and read about the various parameters that can be varied in its use. The GA documentation provides a good explanation of the purpose of many of the different parameters, and a postscript will be provided here for selected variables that are important and/or don't seem to be fully treated in the documentation. Selected troubleshooting for each variable will be designated with an asterisk. pgadmin3 fields reviewed in previous sections will not be re-explained, for efficiency.

A.6.1 General settings

quality: generally signifies the quality of the data. This field can be sometimes used as a designator for certain rows. For instance, if the user is working on fitting a set of rows, the quality can be set to 6 for each row. Then, when pulling up rows in pgadmin3, if "shot=189889 and userid='epstein' and quality=6" is entered in Figure 49.4, it will allow the user to view only those rows. This becomes useful when many runs are entered for one shot, and an easy way to separate those of current interest is valuable.

elm_size_min and elm_size_max: these variables specify to profiles.py how large of a fs06 D_α signal should be considered an ELM. It seems that they are normalized to the largest fs06 D_α spike during the analysis period. Therefore, for values of 0.2 and 1.0, if an ELM fs06 D_α spike were between 20% and 100% of the largest fs06 D_α spike in the quasi-steady-state period, it would be considered an ELM, and start an inter-ELM period according to profiles.py.

cer_twin: this string represents how strict the conditions for inclusion are for collected data in the specified time frame. (What follows is a general description of how this function works, but details regarding the methods the CER system uses to label data may vary from system to system or shot to shot.) The CER systems that measure the ion properties average the collected data over the CER integrating time, which varies from shot to shot. (By using the method described in Section A.5.3 to alter the reviewplus graph results for CER and TS measurements, an idea of the minimum time separation, and therefore integrating time for the CER system, may be obtained. However, always confirm this parameter with an expert.) The data is then labeled with the midpoint of the integrating time. So if the integrating time is 10 ms, then data labeled with 150 ms is actually gathered from 145 ms to 155 ms.

If cer_twin is set to ‘strict’, then any data points including any measurements from times outside the specified boundaries is thrown out. If it is set to ‘in’, then any data points for which the midpoint of the integrating time is inside the boundaries is kept. If the setting is ‘extended’ then any data point with any part of the integrating time inside the boundaries is kept. For example, if the time period boundaries for one interval are 100 to 200 ms, the integrating time is 10 ms, and the setting is ‘strict’, only data labeled $105<t<195$ will be included. If ‘in’, then $100<t<200$ is kept, and if ‘extended’ then $95<t<205$ is kept. When dealing with short time periods, like slices of a composite inter-ELM period of a shot with a rapid ELM frequency, the integrating time will have significant effects on the amount of data available for a fit. When doing an analysis with consecutive time periods and fine time resolution, it is best to use the ‘strict’ condition to ensure that any overlap between the sections is avoided. Beginners should leave the setting on ‘strict’.

adjust_zts: this value sometimes is changed to alter the radial location of the measured data. It should be set to -1, and will be automatically filled in by profiles.py when the finalized data is written to MDSPlus. Users should never change this variable without consulting with their GA contacts. Sometimes for a certain run, this value will be an outlier from the other values for the shot, and cause inaccuracies in the data positioning relative to the separatrix.

NOTE: Ensure that after setting write_mds to 1 for a shot and running profiles.py, if the fit on that shot is revisited, set adjust_zts back to -1.

run_onetwo: setting this value to 1 will run the onetwo code, which is a transport code that calculates various information about the row that is important for a complete MDSPlus entry, but is not especially relevant to the fitting process. It will take somewhere between seven and sixty minutes to run this code along with profiles.py, and therefore this value should only be set to one once the data is ready to be written to MDSPlus.

nubeam: this variable tells profiles.py to run the neutral beam code nubeam, which calculates several important neutral beam parameters, such as the beam pressure. It usually does not add much to the run time of profiles.py, and therefore should be set to one while fitting profiles.

ne_co2norm: this variable corrects the density data, and should be set to -1. When the data is written to MDSPlus, profiles.py will automatically write the correct value in this field.

NOTE: Ensure that after setting write_mds to 1 for a shot and running profiles.py, if the fit on that shot is revisited, set ne_co2norm back to -1. Also, when copying another completed row, this value often must be reset to -1.

A.6.2 Electron Profile Fitting settings

The next set of parameters are the main tool to alter the tanh fits for the electron data that are automatically generated by profiles.py. Consult with the expert when altering the electron data fits.

ne_psimax, te_psimax, pe_psimax: these variables determine the radial extent of the electron density, temperature, and pressure fits, and are one of the few ways to tune the electron data tanh fits. If the electron data automatic tanh fits are imprecise, change the range of these fits using the psimax variable in coordination with a GA contact. Psi is a measure of radius (related to enclosed flux surface percent) that is used by many researchers, although usually normalized radius (rhob) is used by Dr. Stacey and GTEDGE. These values should be set to -1 to be automatically determined, and only altered if a problem becomes apparent.

A.6.3 Refining the data chosen for analysis

The next set of parameters alter the data retrieved for the ions and electrons, and can omit those diagnostic channels that are inaccurate. The process of matching chord data with a chord name is difficult, but can be done by using the diagnostic overlays function in efitviewer and through trial and error (running profiles.py, plotting the data, omitting the chord, running profiles.py again, overlaying the data from the new run, seeing what data point is no longer present, etc.) and. The radial “location” of the Thomson view chords (for the TS electron data) and the CER chords (for the CER ion data) are shown using this diagnostic overlays function, and can be used to identify the general radial location of the data provided by a certain chord. This can then be confirmed using profiles.py and excluding data from that chord by following the stated procedure. It is always important to check with the expert before altering the sets of "bad" chords.

bad_*: these variables - bad_ts_core, bad_ts_tan, bad_ti, bad_fz, bad_vtor, and bad_vpol – specify diagnostic channels that are providing "bad" data. When copying a GA researcher's row, this data is often included, as the expert has often analyzed the accuracy of data from the individual channels. This is a major benefit to copying a researcher's row, since the process of determining a "bad" channel is often difficult and requires large amounts of expertise in the diagnostics systems. *It is important to confirm with a GA contact that the set of "bad" channels is appropriate for the current research purpose when copying them from another row, and also if modifying the set of "bad" channels.

A.6.4 Ion profile fitting parameters

After these variables, there are a number of fields related to the fitting process for the ion temperature (ti), impurity fraction (fz), toroidal carbon rotation velocity (vtor), and poloidal carbon rotation velocity (vpol) profiles. There are several fields that are the same for the different variables, which will be discussed in common, while other fields that are not shared will be discussed as needed.

*_knots (ti_knots, fz_knts, vtor_knots, vpol_knots): this field contains the user-defined spline knots the computer will take into account when creating the profile fit. A "curly bracket" will begin and end the set of knots, and each will be separated by a comma. The knots are input as a radius, and the user essentially tells the computer "I want the fit to go through the data points at or near this radial location, whatever magnitude they may be".

Generally, the knots should start at the fitting range minimum and end at the maximum, with one knot at each extreme. Positive values place a spline knot at the indicated radius, while negative knots instruct the profiles.py script to automatically place however many knots are included in the subset (this seems to be the case). As an example, for a profile fit ranging from psin=0 to psin=1.2, an acceptable set of knots would be (0, 0.5, 0.8, 0.9, 1.2). If negative knots were used, it seems that (-0.5,-0.8) would instruct profiles.py to automatically place two knots to generate a spline fit for the profile. Placing a spline knot at the minimum and maximum range of the fit it always a good idea, but as the user becomes more advanced at manipulating the profile fits, these minimum and maximum knots may be altered for specific purposes. For trends that are clear, without outliers or much obfuscation, automatic spline fitting may be useful.

NOTE The radial location of the spline knots specified by the researcher is read by profiles.py as being <u>psin values, not rho values</u>, even though the rhob box may be checked in Profplot and the data is viewed versus normalized radius (rho). The location of the spline knots (in the psi coordinates) will always appear inward of their rho coordinates, and spline knots are indicated by arrows on the x-axis of the plots (see the figures past Figure 54 for examples).

*_range: this field contains the range that the profile fit will be generated over. It again uses “curly brackets” and two radial locations separated by a comma. For the great majority of fits, the range will begin at 0, and although the maximum range may vary, it should always be very near 1.0, the radial location of the separatrix. The presence of data outside the separatrix may induce desirable or undesirable structure in the profile fit, and the maximum range may be tuned for that purpose, but it is important to remember that the fit should be accurate throughout the entire range being fit. Therefore, if the range is increased to 1.02 in order to improve the profile fit (through the spline fit taking data points outside the separatrix into account), it is still important that the fit be accurate through 1.02.

NOTE the radial location of the range is again psin not rhob. This should not be disruptive to fitting, but will explain small inconsistencies in the radial locations of the emplaced knots and range extremes, and those settings input into pgadmin3.

ti_views and fz_views: these fields determine which CER chords profiles.py will collect data from to generate the data profile. The three main types of CER chords are tangential, core, and vertical, represented by the letters T, C, and V, respectively. It is probable that only two sets of chords can be used at one time. This field is usually altered to improve the clarity of the data profile, or remove systematic data sets that may be known to be less reliable. CER chord set selection should be discussed with an expert. For general purposes, the fz_views setting, the tangential (T) chords are most reliable, and when more data is needed in the edge region, the vertical (V) chords can be added, making the fz_views = ‘TV’. Recall that, although difficult to achieve, specific chords can be added in to the data set using a combination of these parameters and the bad_* parameters, and by using the diagnostic overlays in efitviewer, if necessary for a specific reason.

include_vpol: it seems that this field controls the generation of the vpol fit, and therefore should remain set to 1

Other variables in pgadmin3 exist, but they are either of lesser importance to profile fitting, or the author has little experience working with them. Consult the documentation and the expert if more information is desired regarding the other fitting parameters and features in profiles.py.

A.7 Profile Fitting

Once the row has been made, profiles.py run initially, and the fits viewed in Profplot, the process of tuning and improving the fits can begin. The colors of the data points and trendlines in Profplot correspond to the legend to identify what each trend and data point is. The legend is in the form xyz, where x is the variable name, y is the item being shown, and z is the x-axis parameter. For example, the electron density in the top left graph of Figure 54 shows netanhrhob for the black line, which is the electron density (x=ne) tanh fit (y=tanh) against rhob (z=rhob). The data in the same graph is red, and labeled nedatrhob, indicating that the red data points are showing the electron density (x=ne) data points (y=dat) against rhob (z=rhob). For the ion fits, such as the fz data in the upper right graph in Figure 54, the data is shown the same way, but the fits used are different. For instance, in this graph, the fz fit is the black line, and the label indicates it is a trend fitting fz (x=fz) using a spline fit (y=spl) against rhob (z=rhob). electron data and trends are denoted in the legends of the graphs as netanrhob, tetanhrhob, and petanhrhob, standing for the variable name, the fit type, and the x-axis parameter, respectively. These three trends are shown on the left side of the six-plot screen. The calculated radial electric field profile is shown as a blue trendline in the bottom right plot, along with the carbon toroidal (usually red data points) and poloidal (usually grey data points) velocities. This is seen in Figure 51.9. Recall that the radial electric field profile is calculated from the fits of the other variables, and cannot be altered directly.

A.7.1 TANH fitted, electron density profiles

A.7.1.1 Electron density (Figure 51.2)

The accuracy of this fit is of high importance not only because the profiles are derived from the fitted data, but also because it determines the location of the separatrix relative to the data in many of the plots. Since there will usually be a large number of data points for this profile, it should be relatively simple to discern a data trend identifiable as the well-known electron density profiles for H-mode (as shown in Figure 54 in the top left graph) (or L-mode, etc., depending on the focus of the research.). Unless an exotic behavior is being investigated, in H-mode, the profile should have a relatively constant slope from the core to the top of the pedestal (the profile may be slightly hollow, rising a bit from the inner edge to the top of the pedestal), with a drastic slope change at the top of the pedestal. The profile should rapidly decrease until the bottom of the pedestal, where the profile should become nearly flat. The separatrix should be located near the bottom of the pedestal. The location that is defined as the separatrix is often determined based on the tanh electron density fit, so make sure that it accurately reflects the data trends. It is also important to make sure that the density fit does not curve upward before the separatrix, as this may be a sign that the ne_psimax field needs adjustment, or that another important variable, like ne_co2norm or adjust_zts are imprecise. Consult the GA contact when encountering problems with the electron density fits. An example of an electron density fit needing improvement is given in Figure 60. In this fit, the ne_psimax had been set to 1.1 somehow, and this large value damaged the accuracy of the fit. It was changed to -1, to be set automatically, and the fit improved greatly, resulting in the electron density fit in Figure 61.

A.7.1.2 Electron Temperature (Figure 51.3)

This profile shares many characteristics with the electron density fit. The main ones are the user's inability to alter the fit significantly, and the usual presence of large numbers of data points that should enable an automatic, accurate tanh fit that clearly follows the data trends. Again, it is good to check and make sure that the location of the separatrix is appropriate based on the researcher's general knowledge of the shapes of normal electron temperature profiles, and that the profile slope does not become positive between the foot of the pedestal and the separatrix. Any other anomalies, or poor fits of the data trends, should be discussed with an expert. In the event of a bad fit, the te_psimax parameter may need to be adjusted in consultation with a fitting expert.

A.7.1.3 Electron Pressure (Figure 51.4)

This quantity is directly proportional to the product of the electron temperature and electron density, and should be a strong fit if its constituents are as well. Similar guidelines as those for the other electron properties apply to this parameter.

A.7.2 Spline Fitted Ion data Profiles

These profiles will consume most of the time in the fitting process. It is recommended to go slow, be patient, and often record changes in fitting parameters. This is useful as the researcher may want to retrace their steps to a previous fit, and try a different path. Also, it is best when beginning to make one change at a time, and see how it affects the fit. This knowledge is important for fitting and learning how to use the profiles.py tool, and it is clearer when done one step at a time.

Several basic examples will be given for the various ion data sets and profiles from the author’s research. These are simply a few of the problems and solutions that can be found, and the researcher is encouraged to try different methods to solve any problems they encounter. It is generally best to follow this process in creating the fits:

1. examine the initial fits generated by the first profiles.py run for a row
2. identify areas of concern in the fits

3. adjust one fit at a time, one change at a time, keeping an eye on the radial electric field profile (blue profile in the bottom right graph when the “Er v*ExB plots” box is checked in the Profplot control panel) and the location of the emplaced spline knots (small vertical arrows on the x-axis, up for the upper trend and down for the lower trend in plots with two trends, clearly seen on the x-axis in Figure 21)
4. move through the rows/runs, completing steps 1-4 for each run
5. after completing the fitting for the set, apply any lessons learned or solutions found when fitting any of the runs to other similar situations in the set
6. ensure consistency in the fits and that similar data trends and similar problems in similar runs are being treated in similar ways. This is especially important when the various time periods being analyzed will be compared against each other. The goal is to introduce as little differentiation through the fit as possible, and let data trends drive differentiation.
7. take some time away from the fits (1+ days if possible), and ask a fellow researcher or fitter to look them over
8. take another look at the fits, and submit them to the expert for feedback and approval
9. implement the suggestions
10. finalize the fits by running onetwo and writing the run to MDSPlus.

To understand the relationship between the fit and the spline knots emplaced by the researcher, it is useful to recall the metal-clad cable that connects the handset to the telephone in a public phone booth. The cable is flexible to a point, after which is becomes rigid and can no longer be easily bent. The spline knots can be thought of as points where the cable is bent. The spline fits handle small to substantial gradients well, but have difficulty approximating the data as the trends approach a right angle. This is to be expected given the trigonometric nature of the spline fit is based on the tangent function.

The spline knots should initially be thought of as points where the researcher wishes the fit to change slopes, or wishes the fit to take co-located data into account more strongly. However, their actual functionality is far more complex. Keeping in mind the public telephone cable analogy, it should follow that when two or more spline knots are located near each other, they will interact, and this interaction, along with the effect of the nearby data, can produce unexpected results. Due to this, it is best to avoid locating knots within rho=0.05 of each other as a beginner. Consider two knots on a trend, with some structure between them, like a shallow rise. If they are brought more closely together, the rise will become higher and narrower, and if they are drawn apart, the rise will become lower and wider. The structure in the trend is influenced by the data present, and if there were a valley in the data between the two knots, the inverse would occur.

In the case of three knots, the effects of their interaction is less predictable, but if the data in the area resembled a mountain with valleys on each side, the changes resulting from shifting their relative locations would be analogous to the two-knot case. Bringing the outer knots toward the center knot would increase the height of the mountain, decrease its width, and increase the depth and decrease the width of the valleys on each side. If the center point were brought closer to one of the outer two points, the mountain and valley structure would become asymmetric. One useful example of a structure that is fit well using a three-knot-type system is the edge pedestal, which will be seen frequently if fitting H-mode data. One data point is situated inside of the top of the pedestal, another is situated in the middle of the pedestal, and a third is placed at the separatrix. By varying their relative locations, and through having edge pedestal-shaped data trends, the fit can be relatively easily tuned to match the actual data trend. Another useful three-knot interaction is one that forms a well defined rise, with the knots on each side forming the “shoulders” of the mountain, and the middle knot demarcating the peak. This type of interaction (inverse) can be seen in Figure 76. As seen in Figure 75, previously, there were only two knots in the edge pedestal well region, and as a result, the well was shallow and wide, missing the data point in the center of the well structure. Adding a third knot in the center of the well helped structure a larger “valley”, deepening it enough (and narrowing it accordingly) to catch the data point at 0.98 in Figure 76.

Finally, some very useful tips will be given here that are difficult to classify, and are most appropriate for users with some fitting experience.

- If the fit does not seem to take a clump of data into account, add a spline knot at the radial location of the data. If it still does not work, adding a second spline knot very close to the first ($\text{rho}_2 = \text{rho}_1 + 0.001$) may help without drastically changing the fit of the data. However, if too many spline knots are added to a fit, it may result in a “fit” like that in the top right hand graph of Figure 63.
- If the curvature of the fit is different from what would result in a good fit (i.e. one side of a parabola resembles an exponential curve rather than a parabolic curve), try to reduce the spline knots or increase their number to be either odd or even, whichever it is currently not.
- Try not to emplace a spline knot at a radial location where there is not data, as the results are often unpredictable
- Before writing the fit data to MDSPlus, see what spline knots can be removed without changing the fit. Spline knots can “build up” during the fitting process, and unnecessary ones often remain. These may cause difficulties with the radial

electric field profile and other fits, and add excessive complexity to the fit. It is always best to fit data using the minimum number of spline knots.

- In the author's experience, the radial location of the knots can be specified using up to decimal places, and after that it seems that profiles.py will round the input values up to the thousandths.
- Recall that the location of the separatrix relative to the data as shown in Profplot has some small uncertainty associated with it, and therefore it is important, where possible, to ensure that the fit is consistent with measurements which nominally lie slightly outside the separatrix, but may actually be inside.
- Sometimes all areas of a fit cannot be 95 A fits due to the nature of the interaction between the data and the spline fit, a problem demonstrated in Figure 76, where the edge fit is excellent, but further inward, the fit is not as good. Once a significant amount of time has been spent trying to improve these fits, and success has not been achieved, it is important to remember to put the most emphasis on those areas of the fit that are most relevant to the research being conducted. In the cases discussed in this guide, that would be 0.85 < rho < 1.0, where the fit in Figure 76 is excellent.
- Last but not least, view the data points in the context of their error bars! It is always preferable to have the fit go through the exact center of the data point error bars, but if not possible, the fit is still acceptable in many cases.

A.7.2.1 Fitting the Impurity Fraction data (Figure 51.5)

This quantity represents the percent of the ion density that consists of impurity ions, represented as carbon in GTEDGE. It is very important to get this fit as accurate as possible, as it has a major impact on the radial electric field profile, which cannot be directly modified. The fz data generally has an asymmetric parabolic shape in H-mode operation, with a gentle rise from rho = 0 to rho ~ 0.8, and a plateau where it is relatively steady until about rho ~ 0.95. After rho ~ 0.95, the data shows a steep slope going down to the separatrix. Often, outside the separatrix, the fz data levels out at the value at the bottom of the pedestal.

The author has found it useful to fit the fz data backwards, starting at the separatrix, and moving inward. This is due to the fact that for this variable, most of the difficulty of achieving a good fit is related to achieving a sharp gradient in the pedestal region, a small gradient at the separatrix, and an asymmetric parabolic profile inside of rho ~ 0.95. The fit for this parameter may be initialized by using a set of spline knots with one at the beginning, one at a low radial value (0.1 < rho < 0.5), one at the top of the data trend, one at the top of the pedestal, one in the pedestal, one at the bottom of the pedestal, and one at or outside the separatrix. An example set could be, for Figure 62 {0,0.2,0.8,0.9,0.95,1.01} with a range of {0,1.01}.

Due to characteristics of the CER system, and depending on the entry in the field fz_views in pgadmin3, there often can be a kind of double trend in the fz graphs. A f-z graph showing this type of issue is given in Figure 62, although the trends are sufficiently close on the y-axis to be somewhat merged. The two trends are sometimes slightly offset radially as well, making them more difficult to distinguish from normal data points. Also, data scarcity in the edge can often limit the continuity of these trends from being shown in the pedestal region, disguising the existence of two separate trends. In these cases, if there is a clear double trend, it is generally best to fit the lower trend rather than the upper trend. If it cannot definitively determined that there is a double trend, and there are two sets of data at different magnitudes near the same radial location, it is best to send the profile between them. It is also a good idea to consult the expert for assistance in determining the best trend to fit or to clean up the data in the plot.

It is desirable to avoid the presence of large slopes at the separatrix (despite their presence just inside the separatrix) for several reasons, including data outside the separatrix that suggests this is inaccurate, for computational reasons, and because of known properties of the fz profile. In order to avoid this, most of the fits in the examples will turn up slightly at the separatrix, either reducing the slope drastically or bringing it near zero. This can be achieved by increasing the range of the fz fit to a value greater than 1.0 in instances where there is data outside the separatrix, but it often comes at the cost of softening the slope of the fit in the pedestal, which the data normally suggests should be steep. However, it is also important to avoid inducing a positive slope between the pedestal and the separatrix unless absolutely required by the data. The main issue with a positive slope in this area near the separatrix is the negative effect is has on the radial electric field profile. Approximating the fz fit slope at or near zero in this area is acceptable for two reasons. The first is that since the magnitudes of the fz profile are very small in this region, the actual magnitudes will not have a significant impact on calculations. However, the slope and structure of the fit will usually have an immediate and direct effect on the radial electric field profile. The second is the known shape of the fz fit, which generally does not have an increase right before the separatrix. This is an example of constraining the profile fit using what is known about the realistic behavior of the variable.

Just inside of the separatrix, the next difficulty is to induce a large enough slope in the spline fit in the edge pedestal to match the data trend. Sometimes, in order to induce a sufficiently large slope, it is necessary to decrease the range to a value lower than 1.0, although this must be balanced against the result of a large negative slope at the separatrix. This is an example of a three-knot system, where the data trend is asymmetric and the right knot must be moved inward against the central knot to steepen the right flank of the mountain. The difficulty is in avoiding a simultaneous deepening and narrowing of the right valley while also avoiding inducing much structure on the left side of the central mountain, where the trend should be relatively flat) However, if decreasing the range below rho=1.0 can be avoided, it is best to do so. As a user gains experience, they will see how small changes such as this cause major effects in the fits, and must be done incrementally in order to determine the effect an action will have on the fits.

Finally, the area inside of the pedestal must be fit. The data in this area usually takes the form of either a rise or a plateau, with a gentle slope on the left. The structure in this area can be controlled with a small number of spline knots, and the correct shape can generally be induced by varying the locations of the leftmost of the three pedestal knots, the knot at rho ~ 0.8, and the 0.1 < rho < 0.5 knot, relative to each other.

NOTE fitting guidelines in this document are intended for data similar to that shown in the example figures, which is ELMing H-mode DIII-D shot data. Items such as spline knot locations, structures, and spline knot combination strategies are very subjectively determined, and may be entirely inapplicable to the researcher's cases. Be sure to use this as a complementary guide to the general characteristics and dynamics of fitting, and not as a definitive source of information.

Several examples of f-z fit problems are shown in the Figures. Figure 64 through Figure 67 show two runs in the same shot that both had the same problem, and were both fixed in similar ways. (This is often the case for a shot, hence instruction 5 in A.7.2.) In Figure 64, largely due to a large data clump that creates a discontinuity in the data trend at rho ~ 0.75, there is a concave shape at the top of the profile that does not match the data trend. To eliminate this concave shape, the spline knot at rho = 0.5 was removed, resulting in a much better shape for the profile (shown in Figure 65), but one that still missed the data trend magnitude. This is an example of how having a smaller number of points, and changing the number from odd to even or even to odd, can affect the curvature of the profile fit. This is also an example of eliminating a two-knot interaction, which introduced the concave structure into the fit, as there existed another knot at rho ~

0.85 (shown by the arrows in Figure 64). After the left knot of the two-knot interaction was removed, the structure was eliminated. If the rho = 0.5 knot had been changed to rho = 0.6, the depression would have become deeper and narrower.

The run shown in Figure 66 had a similar discrepancy between the data trends and the profile fits at the top of the profile. After working with a GA expert to examine the data for this run, and better understand the apparent discontinuity at rho ~ 0.8, it was determined that the large data clump at rho ~ 0.75 was less reliable than the other data, and it was removed (using the method described in Section A.5.1.1). This eliminated the seeming discontinuity in the data, and an excellent fit for the entire range was achieved, which is shown in Figure 67 (more fitting was done to achieve this fit than is described in this paragraph, but the description of the workflow is generally accurate and is useful for illustrative purposes) (another fit problem that was fixed was the fit being slightly low on the large data clump in the edge pedestal in Figure 66). These procedures were followed for almost all of the runs for the shot, achieving great fits in all cases.

A.7.2.2 Fitting the Ion Temperature (Figure 51.6)

This data is generally well fit by profiles.py after the initial run. Problems most often occur in the edge pedestal region and near the separatrix. If this fit is insufficient, then it may be useful to use several negative values in the ti_knots field to get an automatic fit with two or three knots (removing all knots may work as well for this parameter). If these steps do not solve the problem, then it may be necessary to fit the profile using the general fitting techniques that the user will utilize on the other spline fits.

Often, the range on this fit will need to be adjusted either to affect the profile value at the separatrix, or to prevent unrealistic values from being present in the fit outside the separatrix. Figure 68 shows both of these problems in one ion temperature fit.

The fit is quite good through most of the plasma, but near the separatrix, there is a sudden decrease in an area void of data. This created three problems. The data outside the separatrix, while less reliable, strongly contradicts this fit, and it seems unrealistic to see such a large drop in a profile that generally is quite smooth to the separatrix. Also, there is no data to justify such a deviation from the general trend. For these reasons, in order to fix this problem, the range, which was previously set at {0,1.0} was extended to {0,1.02}. This created the fit seen in Figure 69, which is in accordance with the data outside the separatrix and what the ion temperature profile is expected to look like. The price for this improvement was a slightly worse fit of the clump near rho ~ 0.9, however, given the dual trends seen in the t-i data in these plots, and the data scarcity in the edge pedestal, this fit seemed to be about an 89 to the author. This was accepted as finished after a considerable amount of work on the fit was completed, and the fit did not improve. However, the fit does pass within the error bars of the data in the clump.

On this fit, it is also easy to overinterpret the data, as often the double trend phenomenon seen in the fz profiles is present. It is important to resist this urge, and ensure that the profile remains smooth and without extraneous structure that may have little justification in the data. Simply recall the “grade” system, and know that higher than about a 94 on this fit is approaching the territory of overinterpreting the data. This is because the data would imply a significant amount of structure, but the actual ion temperature trend is generally a smoothly varying line with a barely noticeable pedestal. Again, generally, it is desirable to have a low negative or zero slope fit at the separatrix, which matches general physical knowledge of the ion temperature fits seen on research reactors.

A.7.2.3 Fitting the Carbon Toroidal Rotation Velocity (Figure 51.7)

In this graph, there are three different profiles. The carbon toroidal rotation velocity (vtor) is the one with red data points and generally is above the other two. This plot is similar to the ion temperature plot in that it often can be fit successfully with one or two automatic spline knots (or none) and a bit of adjustment of the range. However, there is often less data available for the vtor fit than the ion temperature fit, making the danger of overinterpreting the data significant. This is relevant because the vtor profile is an important fit that has direct effects on the radial electric field fit, as well as other calculated values. Again, it is important that the user not be tempted to overinterpret the data, and it is also good to have a low negative slope or zero fit at the separatrix. It often seems like the slope of the vtor profile changes at a few points near the edge, but again, combining the lack of data with the different trends, it is good to be conservative for this fit.

When the carbon toroidal rotation velocity fit is inaccurate, it is usually a poor fit in the edge pedestal region, and it may not accurately represent or pass through the data groupings in this region. There also may be structure in the fit that may not be a result of data trends. This fit sometimes is inaccurate near the separatrix as well. In the author’s experience, with some patience, it is usually possible to obtain a good fit of the carbon toroidal velocity data by using automatically placed knots (negative knot values) or no knots at all, and good choices for the range of the fit.   However, in some cases it is necessary to do a traditional fit for the carbon toroidal velocity, and tune the fit manually.

In Figure 70, a carbon toroidal rotation velocity fit is shown that is poor at low radii and also misses the data trend in the edge pedestal. After adding a spline knot at rho = 0.2, and removing one at rho = 0.95, the fit improves greatly, becoming the fit in Figure 71. Again, the fit is not perfect, nearly missing the data point on the separatrix, but working on the fit further may have resulted in introducing additional unsupported structure.

A.7.2.4 Carbon Poloidal Rotation Velocity (Figure 51.8)

The carbon poloidal rotation velocity (vpol) is another important profile in its own right, and also due to its contribution to the radial electric field calculation. It is the profile with the grey data points in the graph with three profiles, on the bottom right of the Profplot screen. The carbon rotation velocity is measured, and while most of the velocity is in the toroidal direction, a trigonometric fraction of it is in the poloidal direction. These measurements are what make up the vpol profile, which often has very few data points, and large gaps between data groups. These gaps often manifest in the most important parts of the profile, right near the separatrix. This situation makes fitting the vpol data difficult due to its large importance in edge transport and the lack of information available to create a fit with. Fortunately, there usually seems to be enough points to provide a general outline of the data trends, and allow a conservative fit to be made. This trend generally consists of (from the separatrix in) a high value near the separatrix, decreasing strongly from near zero to form a negative well right around the edge pedestal, and then increasing from zero to form the inner edge of the well inside the pedestal. This well structure varies between asymmetric to more of a symmetric shape, and the fit must be closely based on the data present for that run. Inside of the well structure, the structure of the poloidal carbon rotation velocity profile varies. The poloidal carbon rotation data is relatively unreliable inside of about rhob=0.5. Although the fit should not wildly move about in that region, the fit near the core for vpol is especially devalued compared to that in the edge. Given the relative lack of data in this profile, its importance in edge transport, and the unreliability of data inside of rho = 0.5, it is especially important to be conservative in the fit, and not overinterpret any possible data trends.

Problems in the poloidal fit can occur anywhere in the edge, from rho = 0.5 to rho = 1.0, and beyond the separatrix. The main areas of concern are the shape (or presence) of the negative well in the edge pedestal region, the steepness of the inside and outside wall of the negative well, the depth of the negative well, and the fit of the data inside of the negative well. In some cases, there may be data indicating the presence of a well by having a downward trend on either side of the pedestal region, but lacking data indicating the depth. In those cases, it is best to conservatively estimate the depth of the well by trying to accurately fit the downward trends bracketing where the negative well should be, and allowing the algorithm to determine the depth for consistency. The outer wall of the well may be partly determined by data slightly outside the separatrix, which is ok to use as a guide, and should be fitted accurately with the profile. The inside wall of the well should be determined the same way, using the data trends as a guide. Often, there will be a shallow rise in the poloidal rotation velocity data just inside of the edge pedestal, which should be conservatively fit. Although complex structure is often present in the data trends of the poloidal carbon velocity, it should be possible to manipulate the profile to get a relatively accurate fit. However, it is important to remember to observe how the changes in the knots for the poloidal carbon velocity fit affect the radial electric field profile, and backtrack if the radial electric field profile becomes damaged.

Several examples are shown in the figures for correcting errors in the poloidal carbon velocity and radial electric field profiles. In Figure 72, the upper graph (zoomed out to show the radial electric field profile in full) shows that the radial electric field profile has some unusual structure between rho = 0.8 and rho = 0.9, before the normal negative well structure. The lower graph in the same figure zooms in to obtain a clearer view of the poloidal rotation velocity profile, which needs improvement. The spline knot at rho = 0.92 seems to be forcing a change of direction in the poloidal carbon velocity fit that causes the well structure to be found too far inward. After removing this knot, the lower graph in Figure 73 shows that the profile fits the data trend much better, and the expected carbon poloidal velocity well structure is located in the correct place. Note that the depth of the well cannot be determined from the data, and the fit must be done solely with the information available from the slope of the edges of the well structure. The radial electric field in the upper graph of Figure 30 appears to have a much smoother trend, ensuring that the chosen carbon poloidal velocity fit is acceptable.

A three step progression in fitting the carbon poloidal velocity is shown in Figure 74 through Figure 76. In Figure 74, the fit of the data trends is reasonably good, but the fit is unacceptable beyond the separatrix. This is due to the range of the fit ending in the blank space between the two carbon poloidal velocity data groups at rho = 1.01 and rho = 1.04, probably at rho = 1.02. To solve this problem, the range of the fit is extended to rho = 1.1 (this is an especially wide fit, beginners should limit the right extreme of their ranges to less than rho = 1.1, and preferably, to less than rho = 1.05 in most circumstances). This change results in the fit shown in Figure 75, which shows a good fit of the data trends outside the separatrix and inside the well structure. However, since there exists a data point which probably indicates the depth of the well structure (unlike in Figure 72 and Figure 73) at around rho = 0.98, it is important to make sure the fit matches the “specified” well depth. This is done by changing the knot system defining the well structure in the edge pedestal of Figure 75 from a two knot system to a three knot system in Figure 76. This additional knot, and the interaction among the three knots, better structures the edge pedestal well, and successfully matches the data trends.

A.7.2.5 Radial Electric Field (Figure 51.9)

The radial electric field profile (er) is generated based on the four spline fits mentioned above, but is mainly influenced by the impurity fraction profile, and next most importantly, the carbon poloidal rotation velocity profile. It is proportional to the profile shown in blue in the lower right graph along with the rotation velocities. The er profile is very important in edge transport, and care must be taken to ensure a good fit and smooth profile, even if that requires a redo of perfectly good profile fits for one of the velocity profiles or the impurity fraction. The structure of the radial electric field profile generally begins as a high value on the left, and slowly decreases until the edge pedestal region. In this region, a relatively deep, clearly defined well structure forms, which ends around the separatrix. The depth of the well, its location relative to the radius, and the structure of the slope leading into the well from the left will vary with different shot data. Any deviation or abrupt change to this structure must be viewed with concern, as it will likely be directly replicated in the radial electric field values. Figure 52.1 shows the actual radial electric field profile, which is shown in PGPLOT2 window of Profplot when both the “Profile plots” and the “Er, v*ExB plots” boxes are checked in the Profplot control panel.

The radial electric profile shape shown in Figure 51 (the actual radial electric field is shown in Figure 52), Figure 64, and Figure 73 are acceptable, while the one in Figure 72 probably has too much structure in most cases. When a radial electric field profile deviates from the smooth, continuous, single-well structure, the culprit is usually an overly complex fit for the poloidal rotation velocity or the impurity fraction, and the extra structure in the radial electric field fit is almost always co-located radially with the offending part of the other fit. Changing knots in the impurity fraction and poloidal rotation velocity fits is one way to determine where the extra radial electric field structure is coming from. Once the offending fit is found, in most cases, the same data trends must be fit using a different spline knot pattern that does not damage the radial electric field fit. Sometimes removing one or two knots from the offending fit, or removing extra, low-impact knots from it, will correct the problem. Rarely, it will be found that the carbon toroidal rotation velocity fit or the ion temperature fit are at fault. Examples for fixing radial electric field fit problems caused by the poloidal carbon rotation velocity are given in the preceding section.

A.8 Final Checks

Once the fitting process has been completed, before extracting the data for analysis, it is useful to check each fit in each run one more time for these items

- Is the data physically realistic? Does the ion temperature decrease across the edge? Are wells that are generally known to exist generally in the right places?
- Are there any structures in the fits that are not representative of data trends? In the case that they are emplaced for a reason, is the reasoning defensible to a third party?
- Are adjust_zts and ne_co2norm set to -1 before writing to MDSPlus?
- Are the fits as good as possible outside the area of emphasis of the current research (i.e. rho < 0.8 and rho > 1.0 for the examples in this guide)?
- Is the radial electric field profile smooth and relatively feature-free, until the negative well in the edge pedestal? Does the well profile extend smoothly past the separatrix?
- Has nubeam been set to 1 during the fitting process?

After these have been checked, it is good to check the beam pressure levels, which generally appears as a magenta trend in the lower left plot in Profplot. It should be much smaller than the total electron pressure. If it is not, contact the expert. Finally, after receiving approval from the expert, set write_mds, nubeam, and run_onetwo to 1, and complete to 0, for all the rows ready to be written to MDSPlus, run profiles.py for that shot, and take a break while it runs! You've earned it!

A.9 Data Extraction

This step will vary depending on the type of analysis being carried out. The automation guide associated with this guide will contain detailed instructions on extracting the data for inter-ELM evolution analysis through a set of specialized scripts developed for this purpose. The expert may recommend other methods of data extraction for other research purposes.

A.10 Troubleshooting

1. The nomachine window doesn’t show the full remote desktop, or windows are being cut off.

The Nomachine interface is generally user friendly, and technical assistance or clues are often available using a thorough Google search. However, display problems may occur when there is a mismatch between client resolutions, monitor capabilities, the virtual desktop resolutions, and the various configurations of these three properties used when accessing the GA servers from different machines. To attempt to remedy these problems, open the nomachine shortcut that shows an available “configure…” menu in the bottom left (Figure 42.2). After clicking on the “configure…” button, in the “general” tab, select “fullscreen” from the drop-down menu under “display” (Figure 43.1). Click “Save” then “Ok”. This is the setting the author has have had success with.

Now, if when the nomachine client opens, a bar across the top listing open applications is not visible, the resolution of the remote desktop must be adjusted. Right clicking on the open desktop area will bring up a menu with the “open in terminal” option. After left clicking on this option, a terminal will appear. Type

```
xrandr -q
```

to list the available resolutions for the remote desktop. Choose one that is lower than the local desktop size, and type it in using

```
xrandr –s ####x###
```

replacing the pound signs with the new desired resolution. This is illustrated in Figure 43.3, and the remote desktop should change size accordingly. Be aware that choosing a remote desktop resolution lower than 800 pixels tall on either the local desktop or remote desktop will, in some occasions, result in the bottom of windows in the remote desktop being cut off in some GA analysis tools. This will prevent the selection or use of buttons and features below the bottom of the remote desktop screen.

Now that the remote desktop resolution has been set to a low enough level to fit on the local screen, the nomachine client must be closed, and disconnect must be chosen, rather than terminate, in the open window (Figure 43.1). The next time Nomachine is opened, the full remote screen should appear on the local desktop, including the “task bar”.

2. the password assigned by GA in my email doesn’t grant access through nomachine.

It may be useful to open the “putty” program, as described in the GA remote access documentation, and change passwords through it. Once logged on to the Cybele server using the putty interface, type

```
passwd
```

which will then allow the Cybele password to be reset. The new password must comply with the strict requirements for Cybele passwords, and may not reset correctly if they are not met. Also, the linux system often does not print asterisks to the screen when entering the passwords into the system, and often the cursor will remain in the same place and nothing will appear to have been entered, when keyboard input has actually been recorded. This method of changing passwords can also be used to change the <u>server node password</u>.

3. parts of the fusion.gat.com password-protected internal website are inaccessible through my browser.

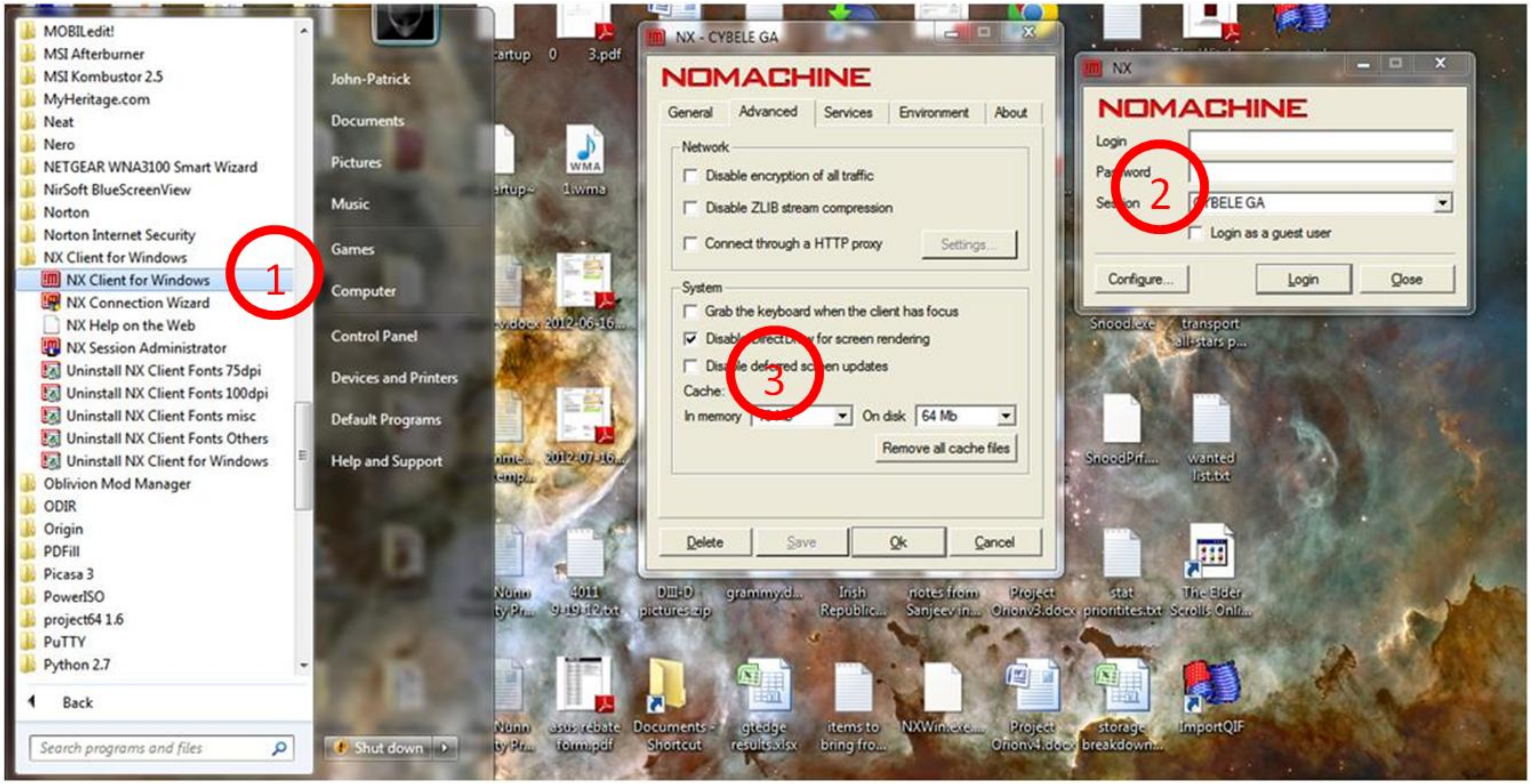
Open the website using the Firefox browser in the Nomachine client on the remote desktop. If the page exists and there are no technical problems, the full content of the website should be available this way.

4. rendering the Profplot plots on the screen takes a long time

There is an important setting to change in the Nomachine program that may save much frustration and time. Go to the start menu (not the shortcut to the preconfigured program), and find the "NX Client for Windows" Link, shown in Figure 42.1. When it opens, select the Cybele session from the drop-down menu in Figure 42.2, and click the "configure…" button in Figure 42.2. In the window that opens, in the "Advanced" tab, ensure the box labeled "Disable DirectDraw for screen rendering" is checked. This box is near Figure 42.3. Click "Save" in the window, then exit the program. Then run the Nomachine program as usual.

5. when in reviewplus, and trying to bring up the data for a common or important variable, an "error" appears in the status row of the input panel, or the data for some variables is unavailable for the entire length of the shot.

While checking with the GA contact is always advisable in this case, it may be possible to view the data by changing the format of the data name, and calling the data specifically from an MDSPlus data tree. This can be done by adding a "/" in front of the signal name, and entering an efit tree branch in the “tree” area. For the variable in row 0 of Figure 45, this would be done by changing “pinj” to ”/pinj” and entering “EFIT03” in the “tree” field.



Network

☐ Disable encryption of all traffic

☐ Disable ZLIB stream compression

☐ Connect through a HTTP proxy

System

☐ Grab the keyboard when the client has focus

☑ Disable DirectDraw for screen rendering

☐ Disable deferred screen updates

Cache:

In memory

On disk 64 Mb

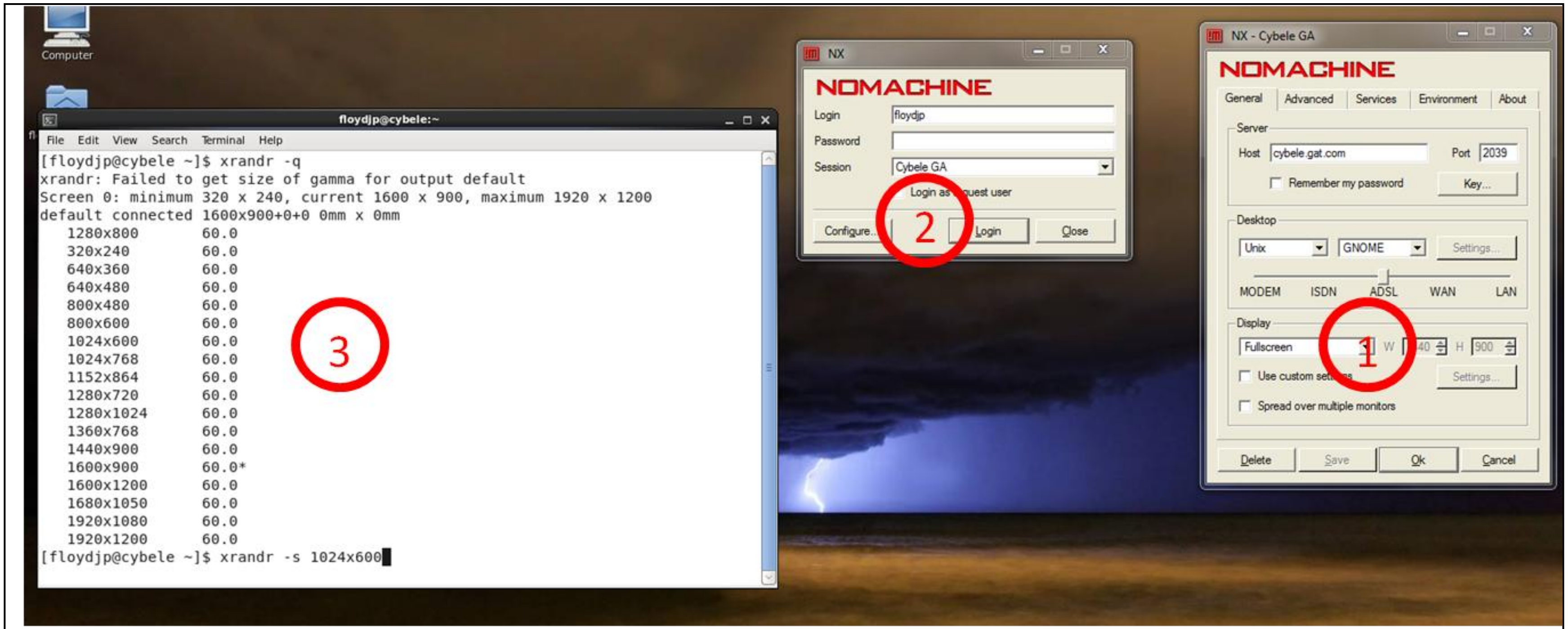
Login

Password

Session CYBELE GA

☐ Login as a guest user

Figure 42: A: Important setting change for Nomachine



Login: floydjp
Password:
Session: Cybele GA
☐ Login as a guest user

Server
Host: cybele.gat.com
Port: 2039
☐ Remember my password

Desktop
Unix
GNOME

Display
Fullscreen
W: 1440
H: 900
☐ Use custom settings
☐ Spread over multiple monitors

floydjp@cybele:~

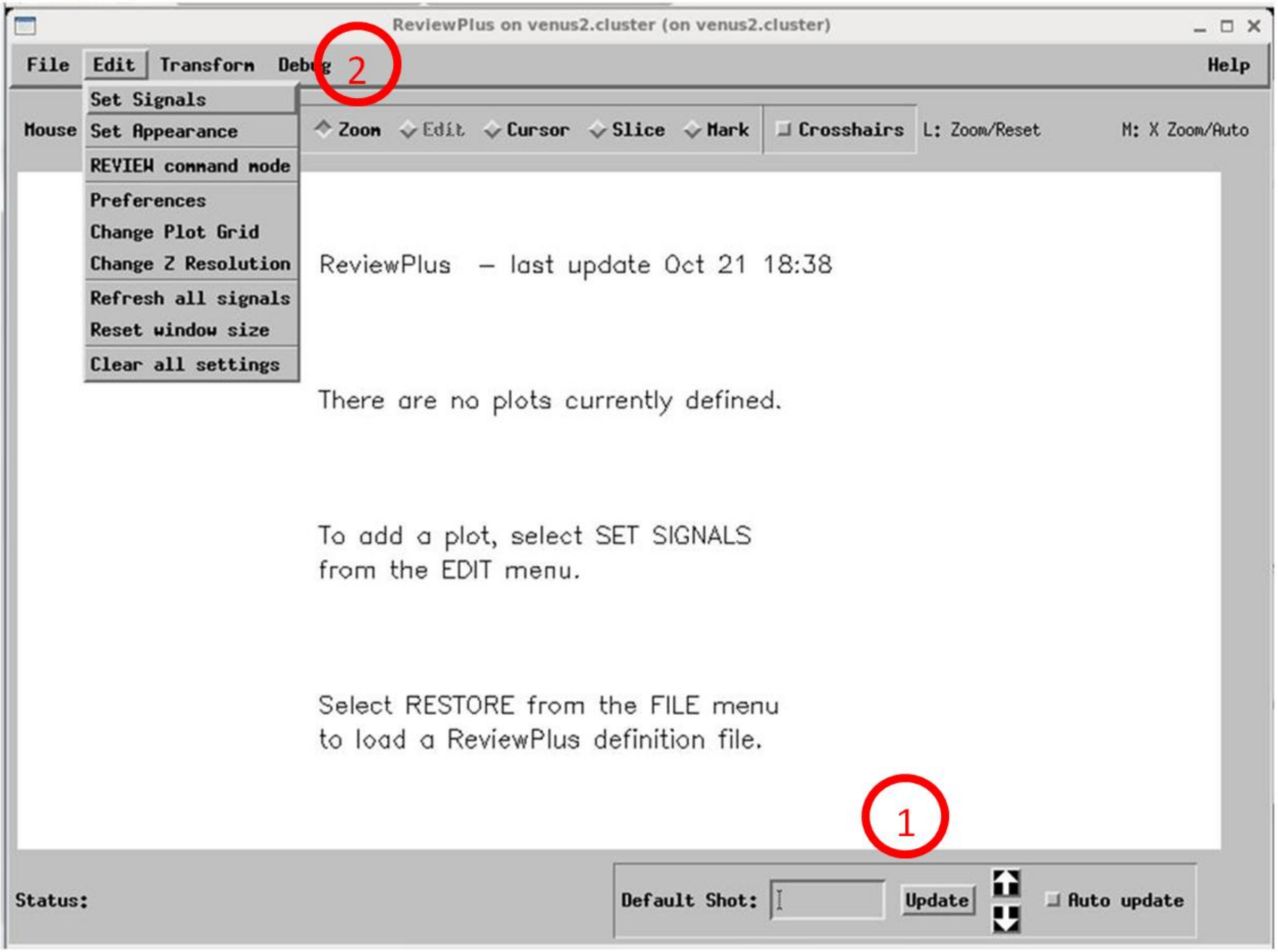
File Edit View Search Terminal Help

```
[floydjp@cybele ~]$ xrandr -q
xrandr: Failed to get size of gamma for output default
Screen 0: minimum 320 x 240, current 1600 x 900, maximum 1920 x 1200
default connected 1600x900+0+0 0mm x 0mm
   1280x800       60.0
   320x240        60.0
   640x360        60.0
   640x480        60.0
   800x480        60.0
   800x600        60.0
   1024x600       60.0
   1024x768       60.0
   1152x864       60.0
   1280x720       60.0
   1280x1024      60.0
   1360x768       60.0
   1440x900       60.0
   1600x900       60.0*
   1600x1200      60.0
   1680x1050      60.0
   1920x1080      60.0
   1920x1200      60.0
[floydjp@cybele ~]$ xrandr -s 1024x600
```

NX - cybele.gat.com-1158

Press the disconnect button to disconnect the running session. You will be able to resume the session at later time. Press the terminate button to exit the session and close all the running programs.

Figure 43: Setting up nomachine



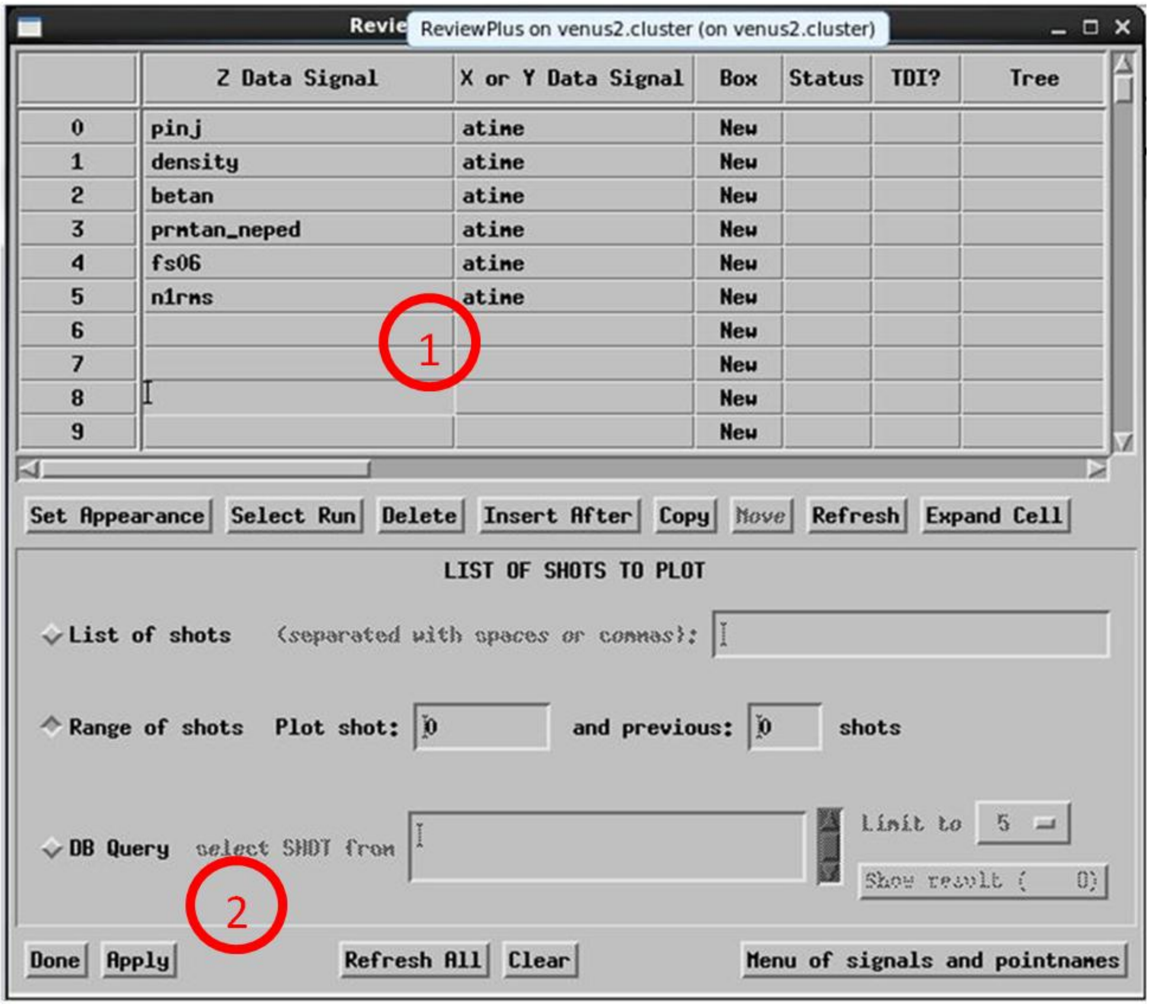
ReviewPlus — last update Oct 21 18:38

There are no plots currently defined.

To add a plot, select SET SIGNALS
from the EDIT menu.

Select RESTORE from the FILE menu
to load a ReviewPlus definition file.

Figure 44: Reviewplus opening screen



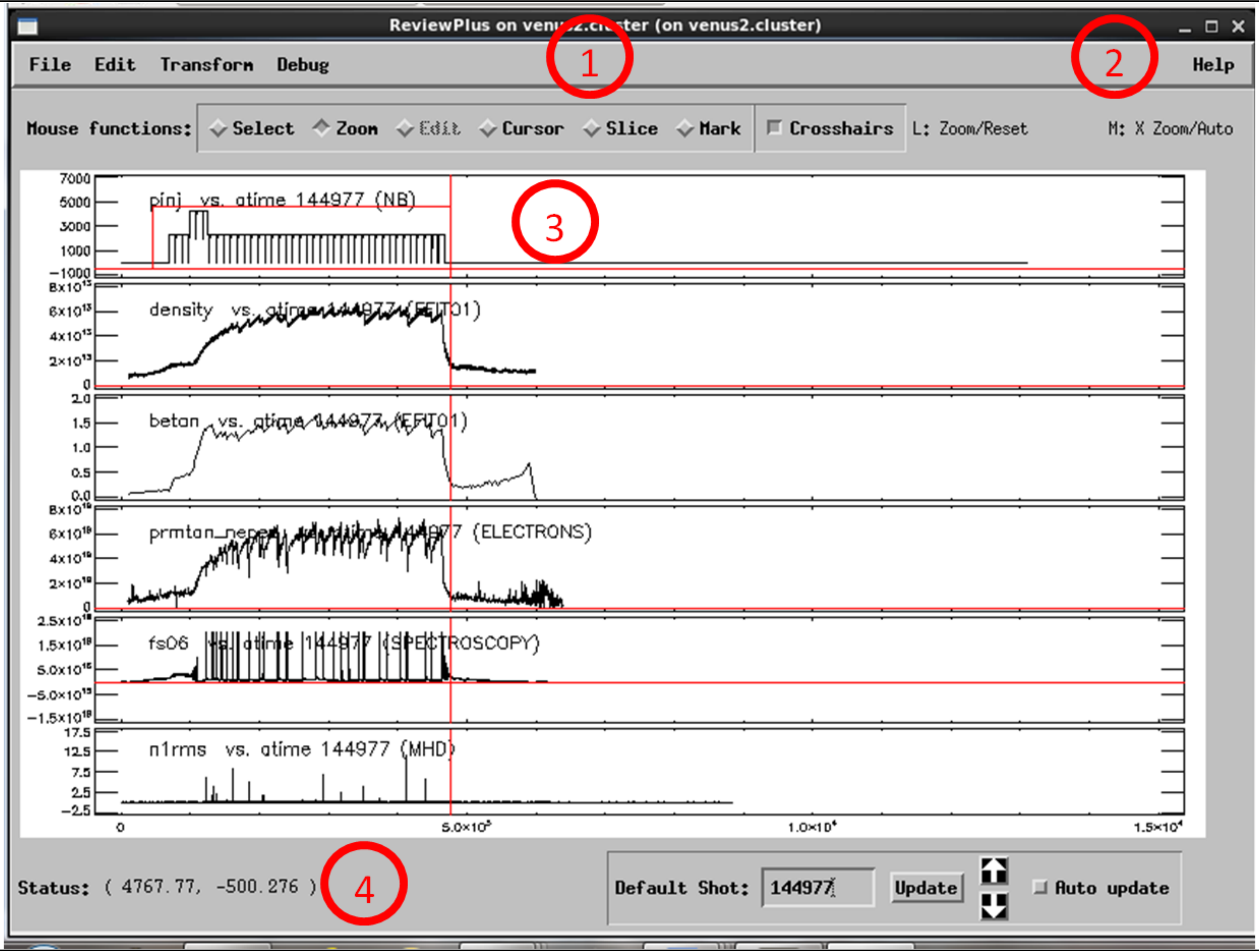
	Z Data Signal	X or Y Data Signal	Box	Status	TDI?	Tree
0	pinj	atime	New			
1	density	atime	New			
2	betan	atime	New			
3	prmtan_neped	atime	New			
4	fs06	atime	New			
5	n1rms	atime	New			
6			New			
7			New			
8			New			
9			New			

◇ List of shots (separated with spaces or commas):

◇ Range of shots Plot shot: 0 and previous: 0 shots

◇ DB Query select SHOT from Limit to 5 Show result (0)

Figure 45: Set Signals menu



Status: (4767.77, -500.276)

Default Shot: 144977

Figure 46: Initial reviewplus graphs

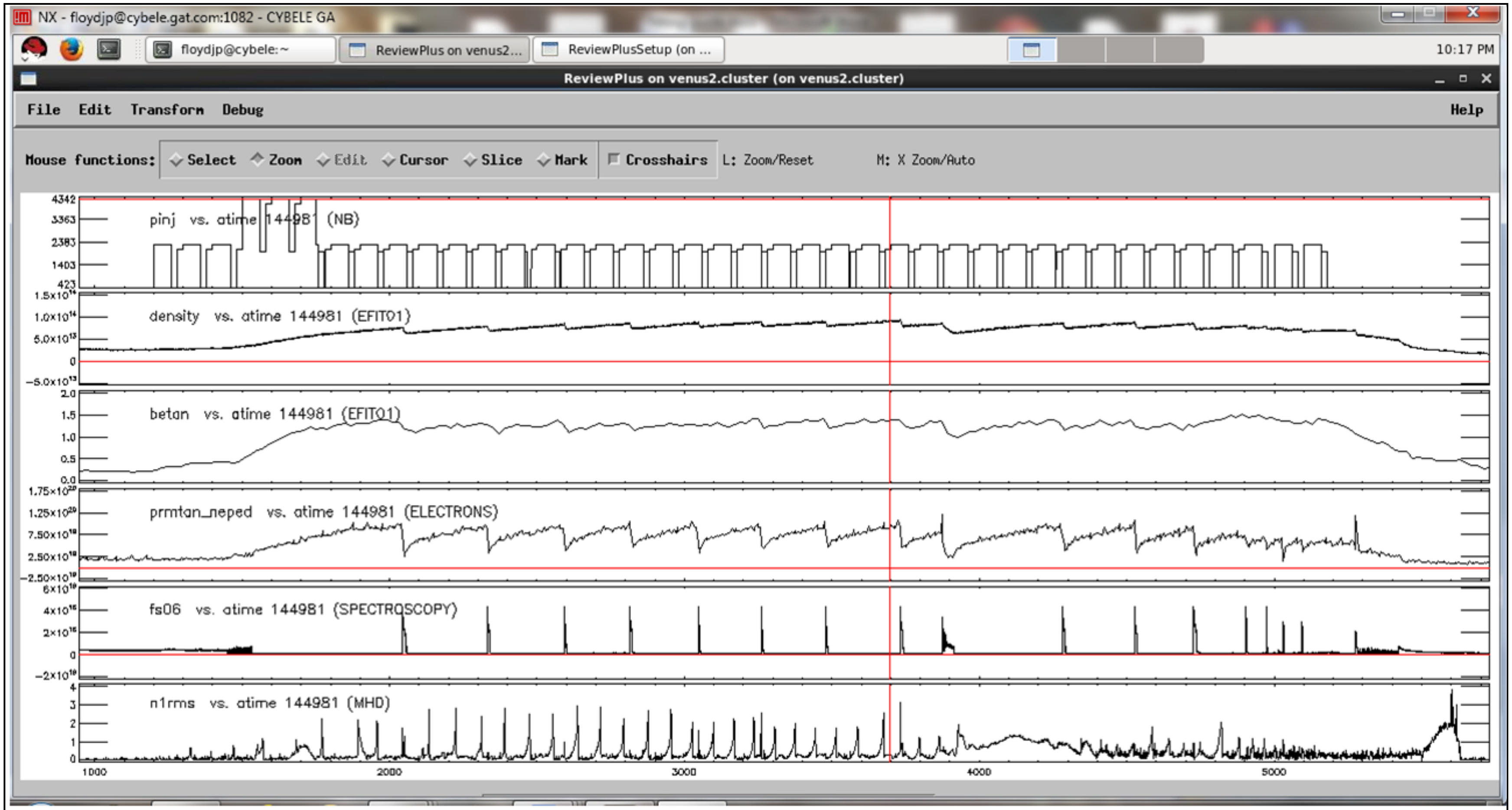


Figure 47: Reviewplus graphs are shown at a good zoom level for the given data. This is for DIII-D Shot 144981. Note the resolution-caused differences on the bottom of the graph from Figure 46.

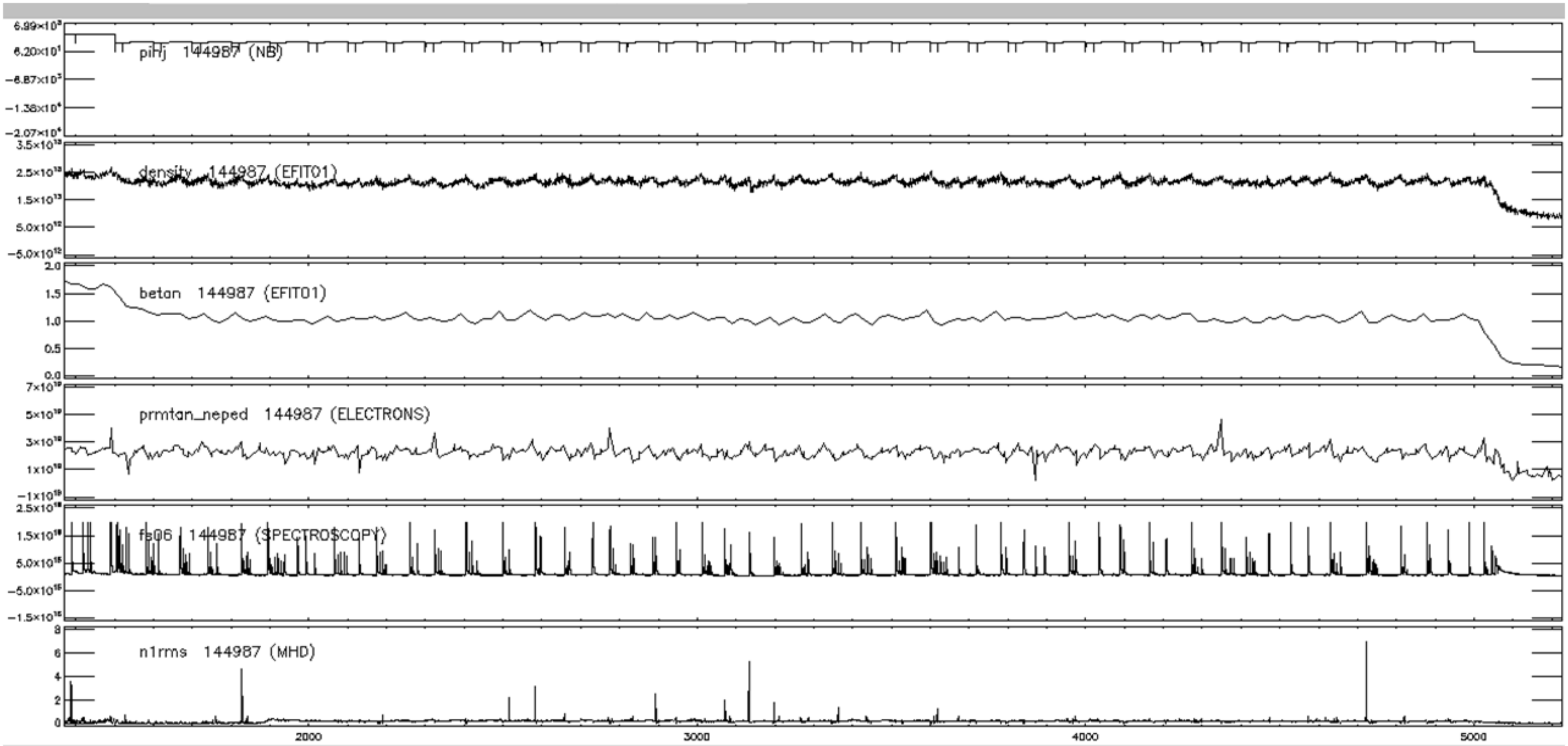
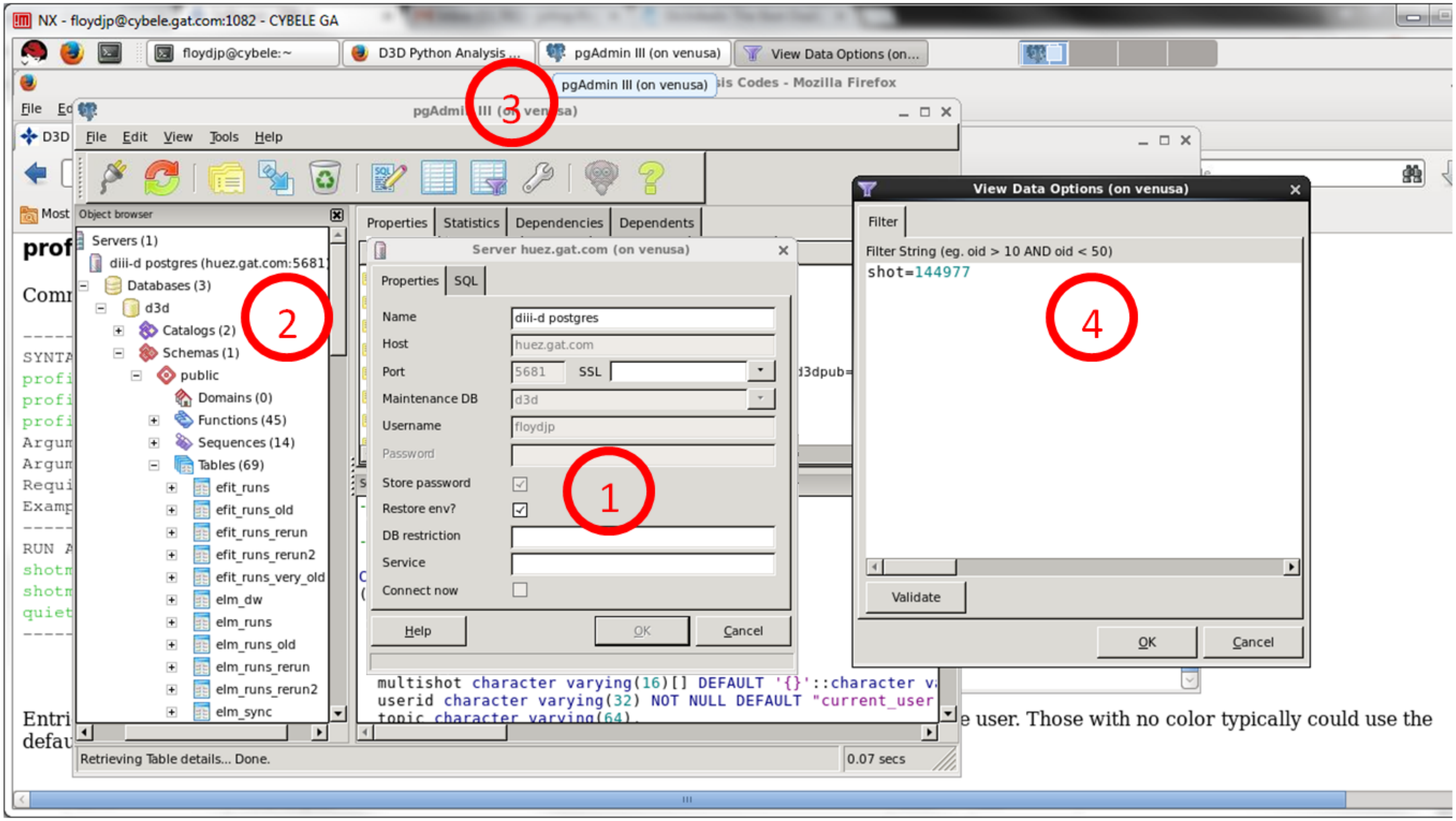


Figure 48: Variables for DIII-D shot 144987 – with many problems.



Server huez.gat.com (on venusa)

Name	diii-d postgres
Host	huez.gat.com
Port	5681
SSL	
Maintenance DB	d3d
Username	floydjp
Password	
Store password	☑
Restore env?	☑
DB restriction	
Service	
Connect now	☐

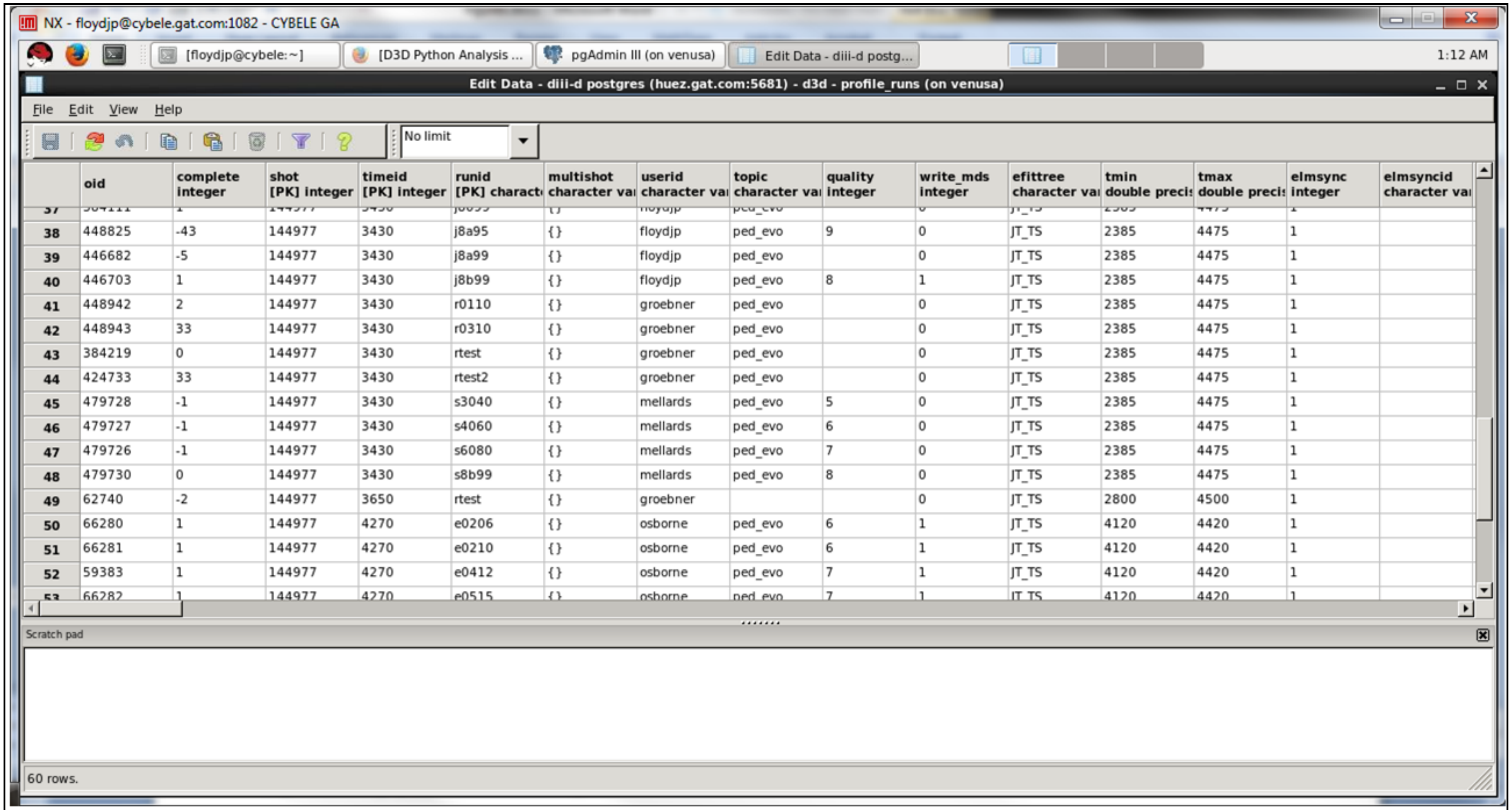
View Data Options (on venusa)

Filter String (eg. oid > 10 AND oid < 50)

```
shot=144977
```

```
multishot character varying(16)[] DEFAULT '{}'::character v
userid character varying(32) NOT NULL DEFAULT "current_user
topic character varying(64)
```

Figure 49: Setting up pgadmin3 and filtering results

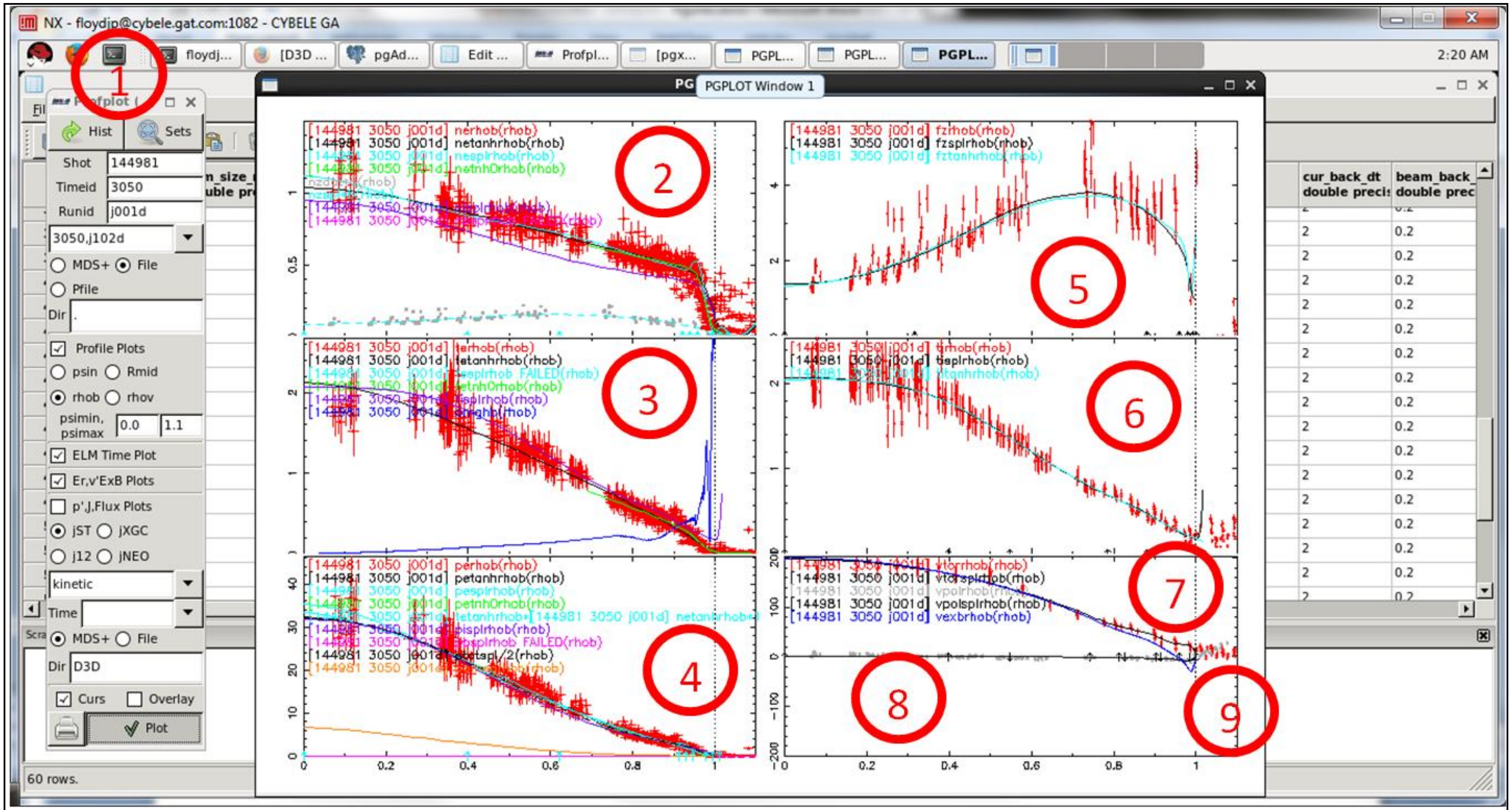


Edit Data - diii-d postgres (huez.gat.com:5681) - d3d - profile_runs (on venusa)

	oid	complete integer	shot [PK] integer	timeid [PK] integer	runid [PK] characte	multishot character va	userid character va	topic character va	quality integer	write_mds integer	efittree character va	tmin double precis	tmax double precis	elmsync integer	elmsyncid character va
38	448825	-43	144977	3430	j8a95	{}	floydjp	ped_evo	9	0	JT_TS	2385	4475	1	
39	446682	-5	144977	3430	j8a99	{}	floydjp	ped_evo		0	JT_TS	2385	4475	1	
40	446703	1	144977	3430	j8b99	{}	floydjp	ped_evo	8	1	JT_TS	2385	4475	1	
41	448942	2	144977	3430	r0110	{}	groebner	ped_evo		0	JT_TS	2385	4475	1	
42	448943	33	144977	3430	r0310	{}	groebner	ped_evo		0	JT_TS	2385	4475	1	
43	384219	0	144977	3430	rtest	{}	groebner	ped_evo		0	JT_TS	2385	4475	1	
44	424733	33	144977	3430	rtest2	{}	groebner	ped_evo		0	JT_TS	2385	4475	1	
45	479728	-1	144977	3430	s3040	{}	mellards	ped_evo	5	0	JT_TS	2385	4475	1	
46	479727	-1	144977	3430	s4060	{}	mellards	ped_evo	6	0	JT_TS	2385	4475	1	
47	479726	-1	144977	3430	s6080	{}	mellards	ped_evo	7	0	JT_TS	2385	4475	1	
48	479730	0	144977	3430	s8b99	{}	mellards	ped_evo	8	0	JT_TS	2385	4475	1	
49	62740	-2	144977	3650	rtest	{}	groebner			0	JT_TS	2800	4500	1	
50	66280	1	144977	4270	e0206	{}	osborne	ped_evo	6	1	JT_TS	4120	4420	1	
51	66281	1	144977	4270	e0210	{}	osborne	ped_evo	6	1	JT_TS	4120	4420	1	
52	59383	1	144977	4270	e0412	{}	osborne	ped_evo	7	1	JT_TS	4120	4420	1	
53	66282	1	144977	4270	e0515	{}	osborne	ped_evo	7	1	JT_TS	4120	4420	1	

60 rows.

Figure 50: profiles.py runs for DIII-D shot 144977



Shot 144981

Timeid 3050

Runid j001d

3050,j102d

○ MDS+ ● File

○ Pfile

Dir .

☑ Profile Plots

○ psin ○ Rmid

● rhob ○ rhov

psimin, psimax 0.0 1.1

☑ ELM Time Plot

☑ Er,v'ExB Plots

☐ p',J,Flux Plots

● jST ○ jXGC

○ j12 ○ jNEO

kinetic

Time

● MDS+ ○ File

Dir D3D

☑ Curs ☐ Overlay

Plot

60 rows.

Figure 51: Working with Profplot

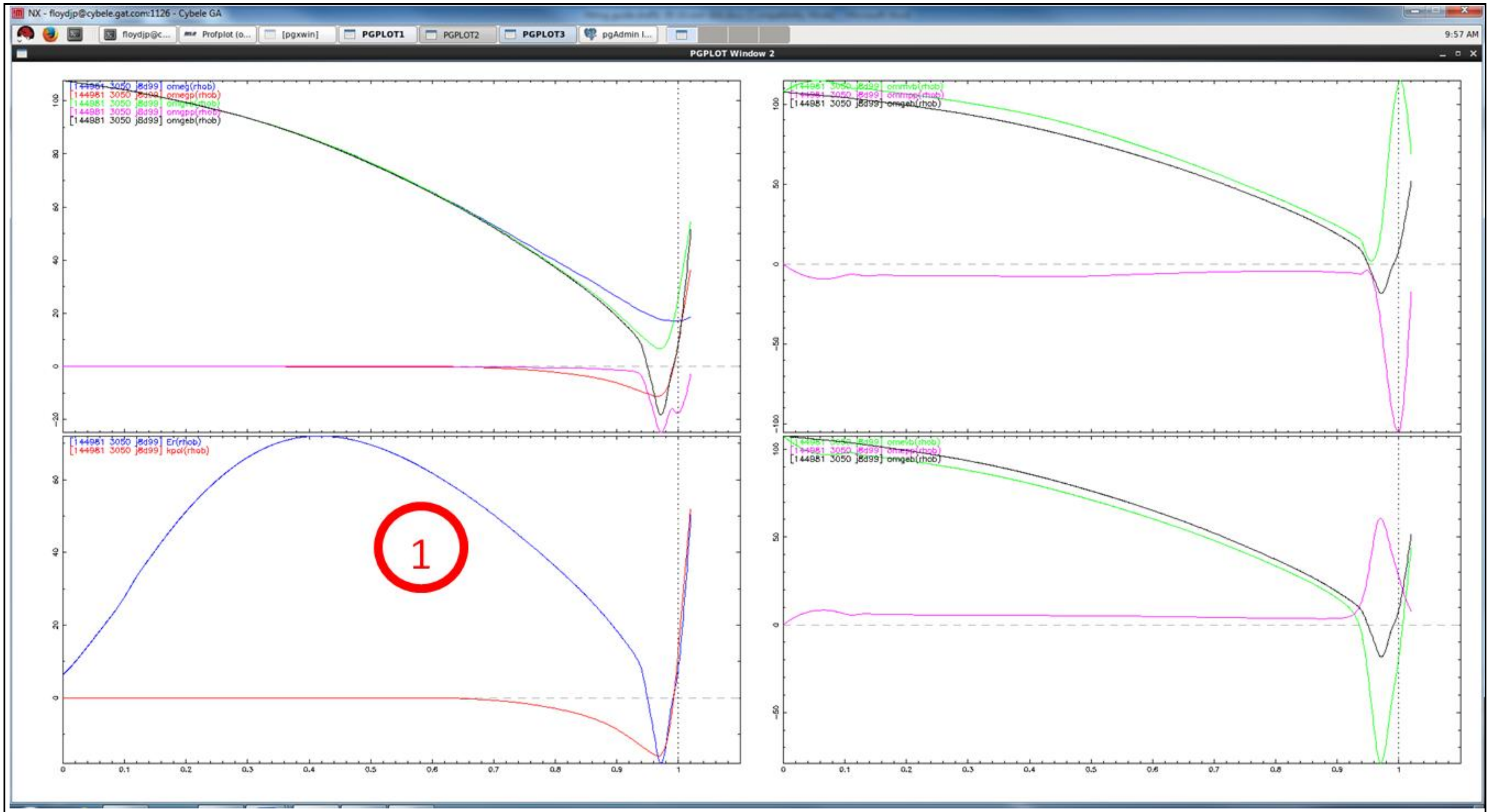


Figure 52: PGPLOT2 Window for Profplot showing the radial electric field

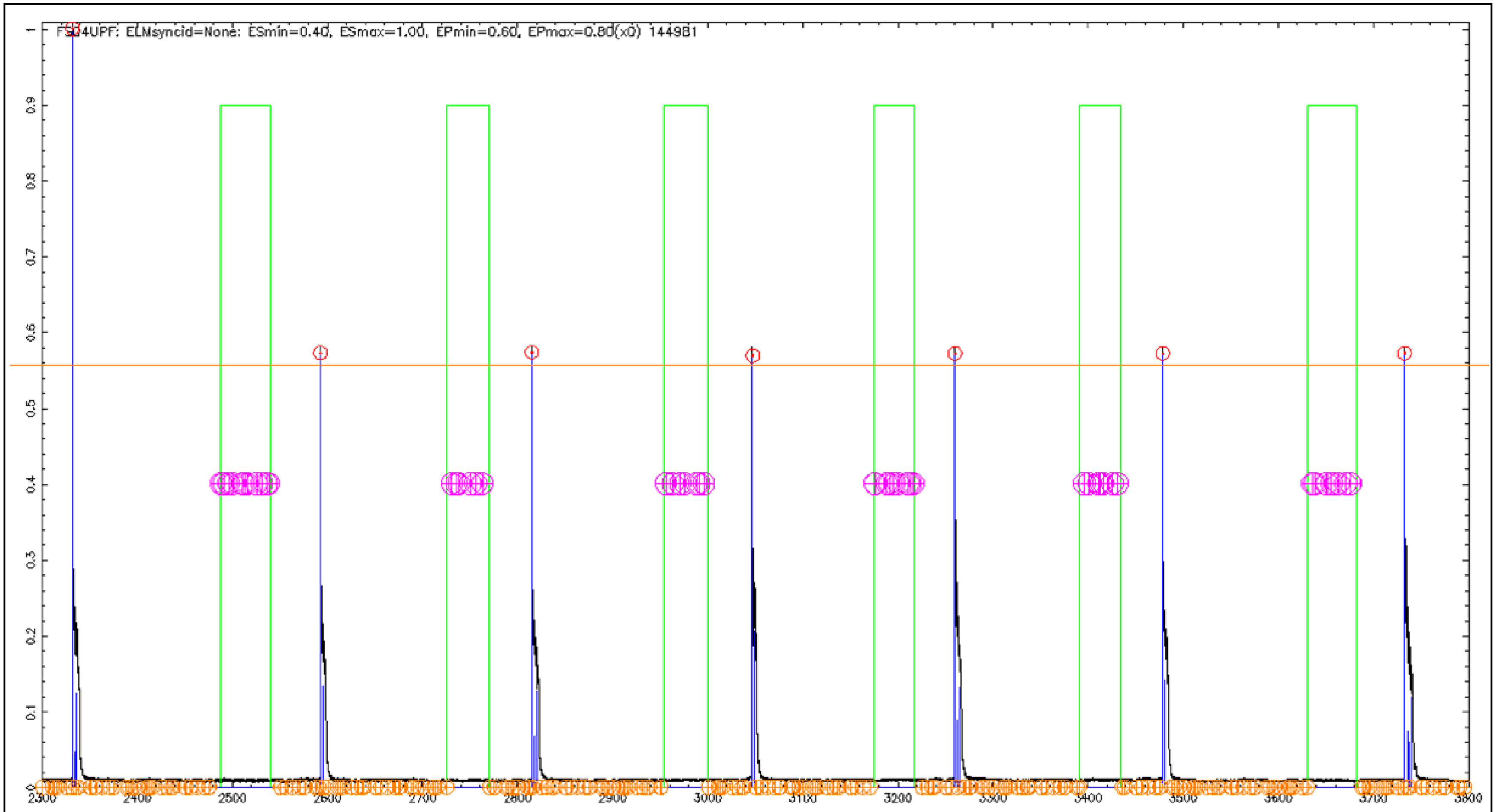


Figure 53: ELM time plot for shot 144981, timeid=3050, runid='j608d' displaying the 60-80% time slices of the inter-ELM periods during the specified quasi-steady-state period that are combined into a 60-80% fraction of the composite inter-ELM period.

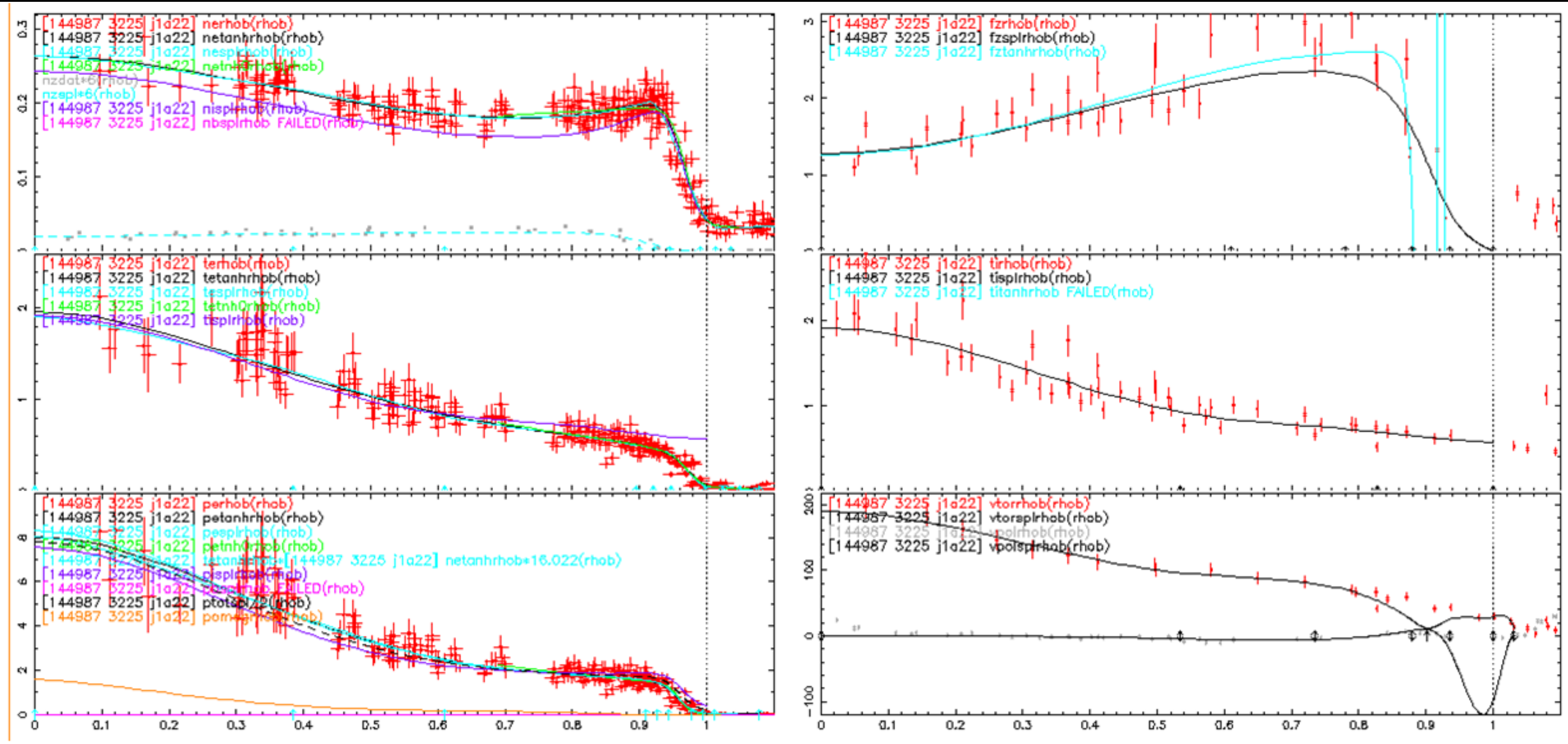


Figure 54: A set of data that is too sparse to fit reliably.

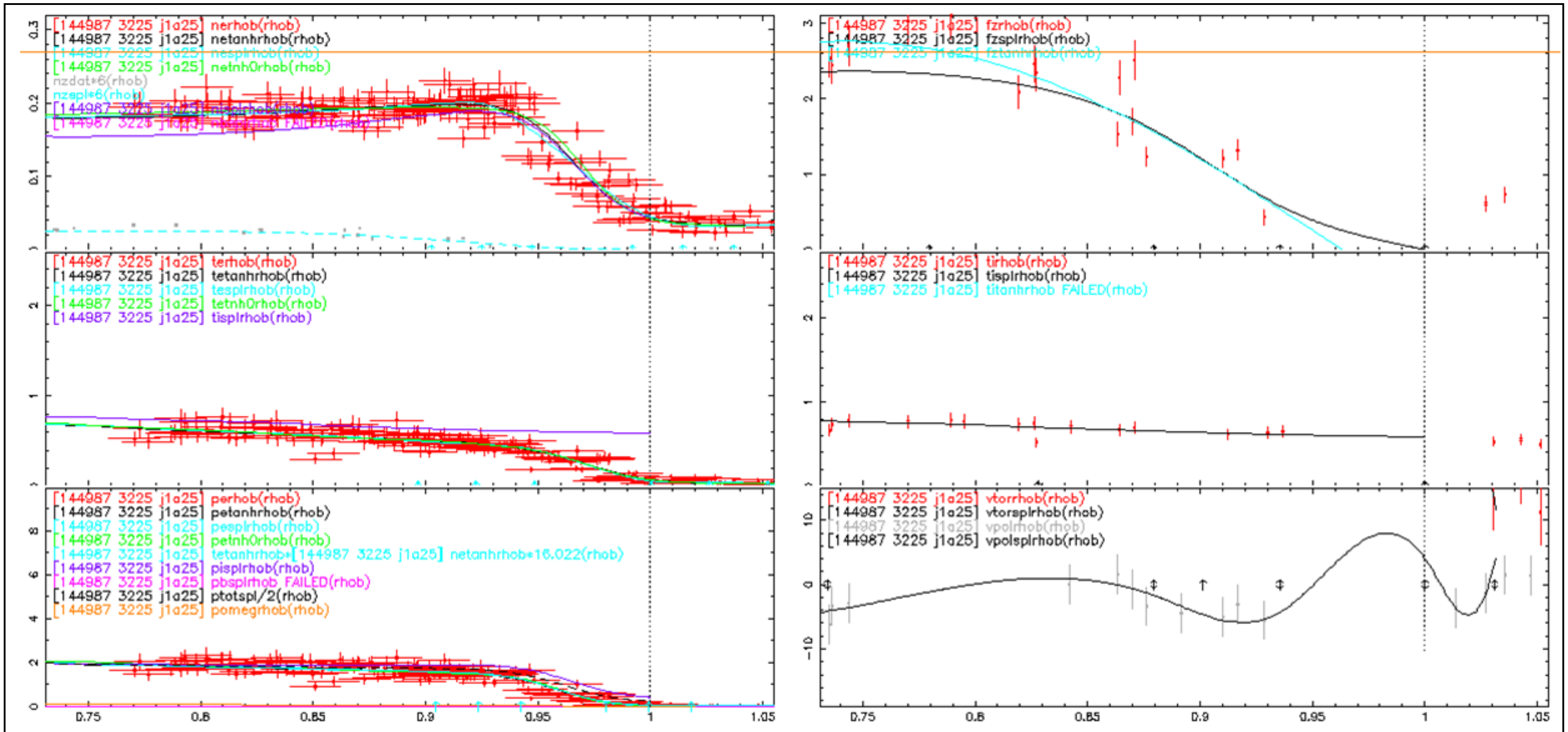


Figure 55: A closer view of the outer 25% of Figure 54.

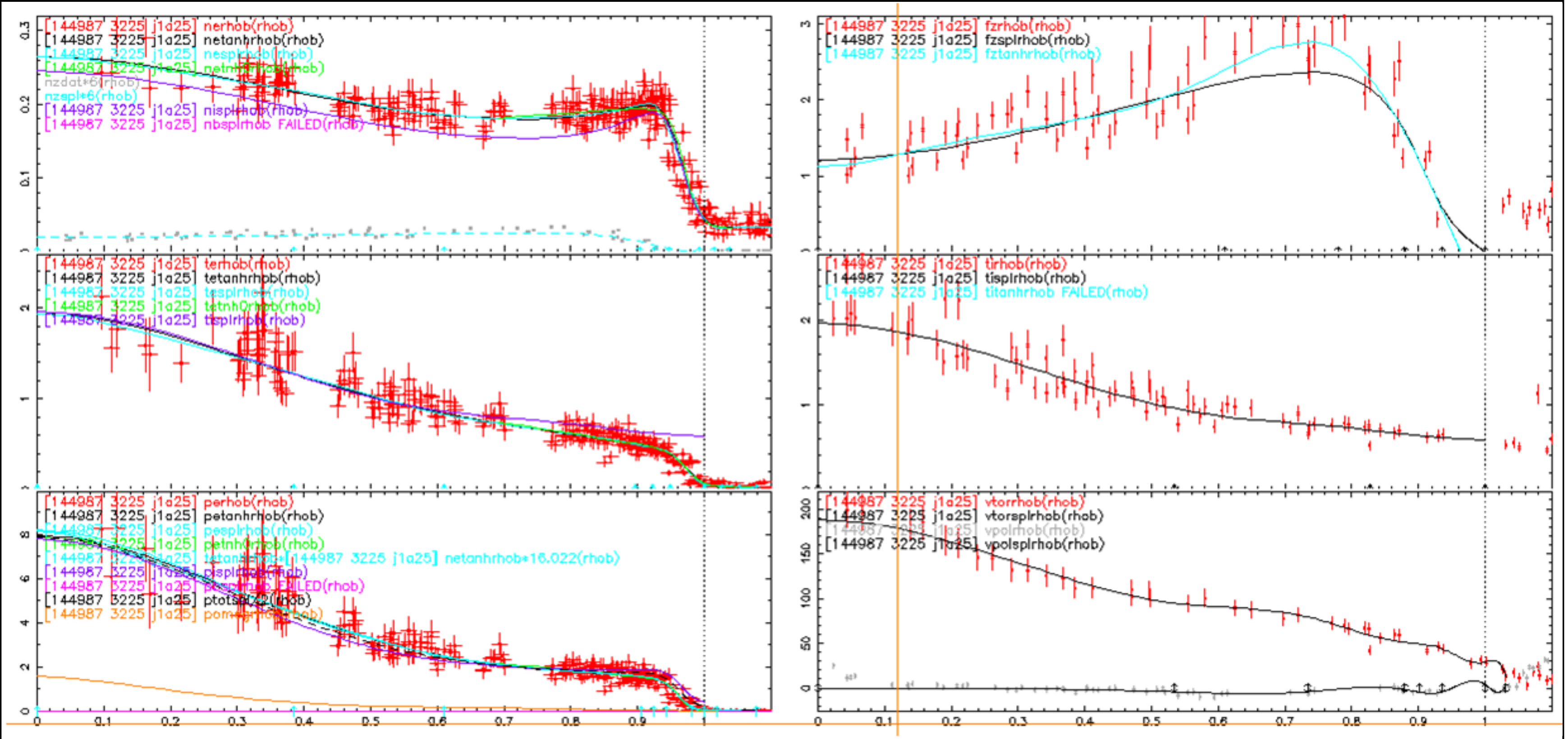


Figure 56: An extremely sparse data set, which is on the edge of being able to be reliably fit

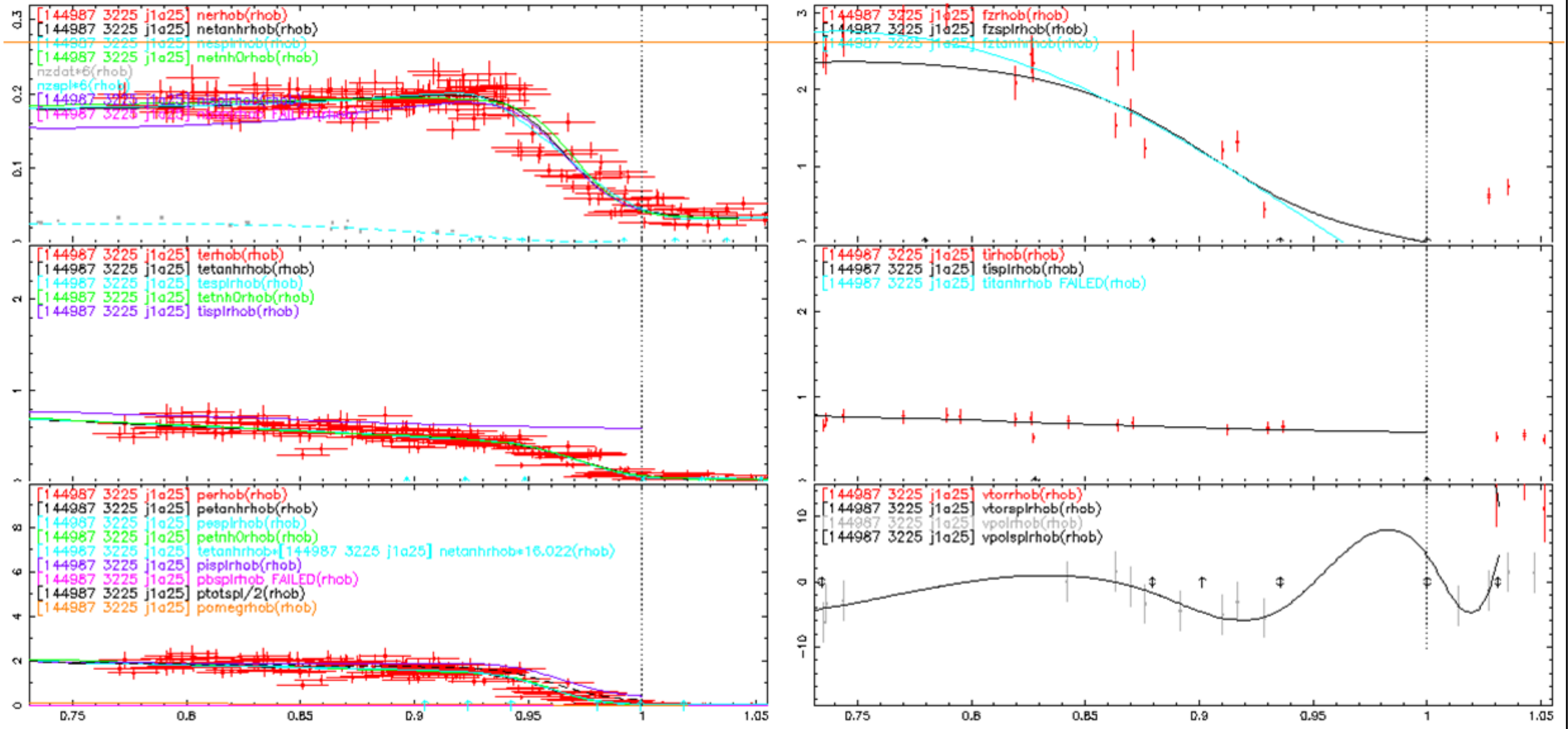


Figure 57: A closer view of the second trend in the lower right hand graph of Figure 56.

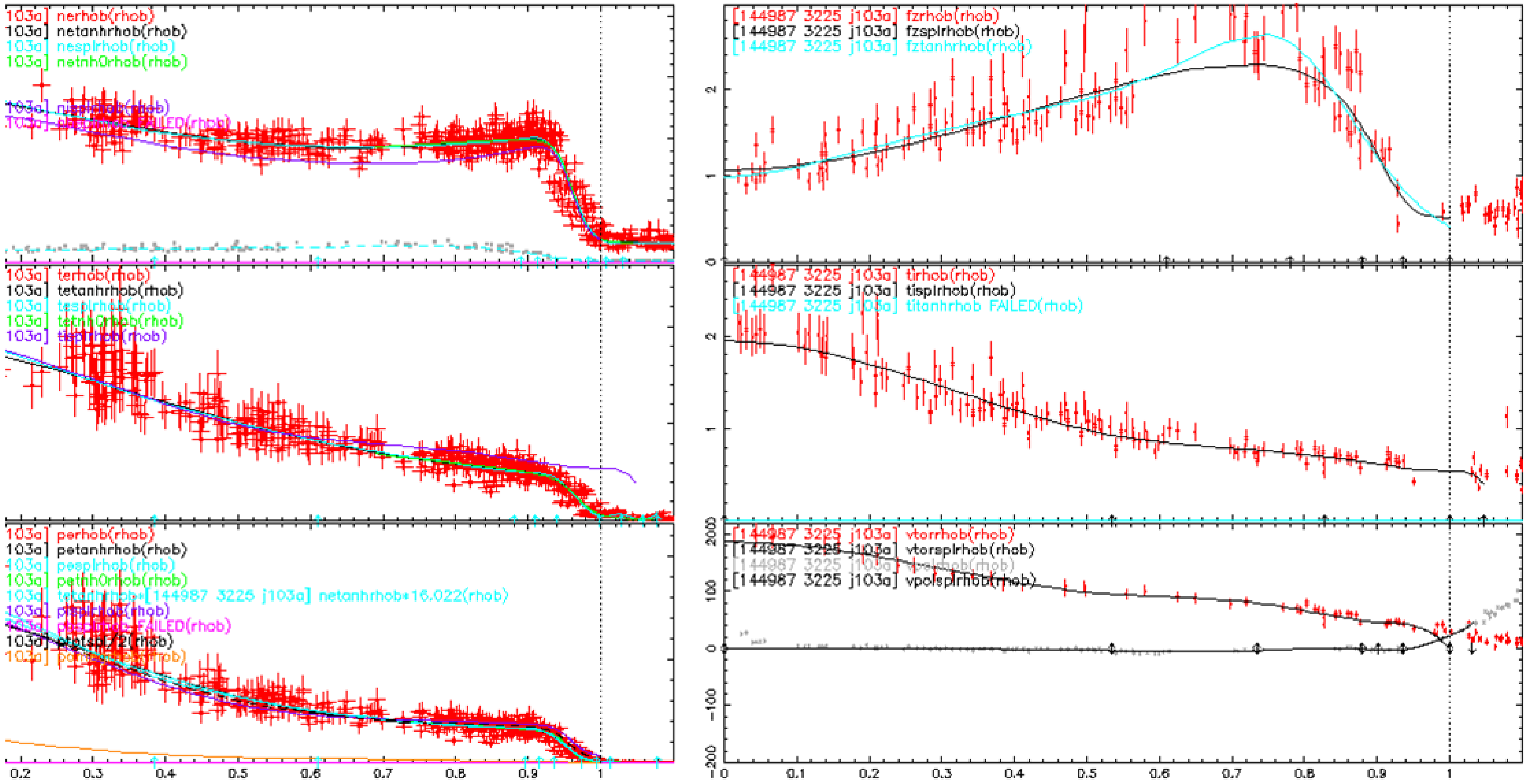


Figure 58: Despite a relative lack of data points near the separatrix, this data set can be reliably fit.

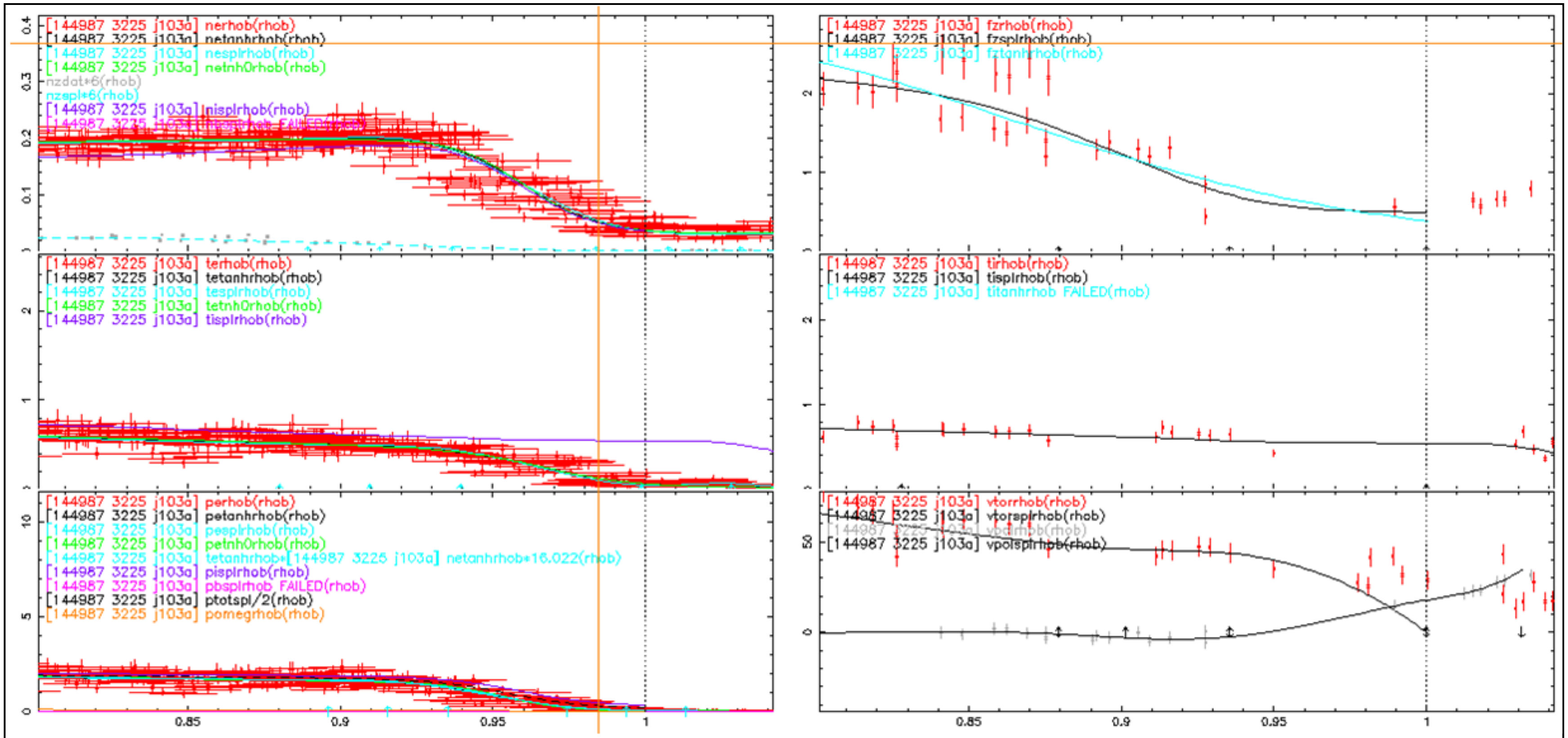


Figure 59: A set of data that can be reliably fit, but more data points near the separatrix would greatly increase the quality of the fit.

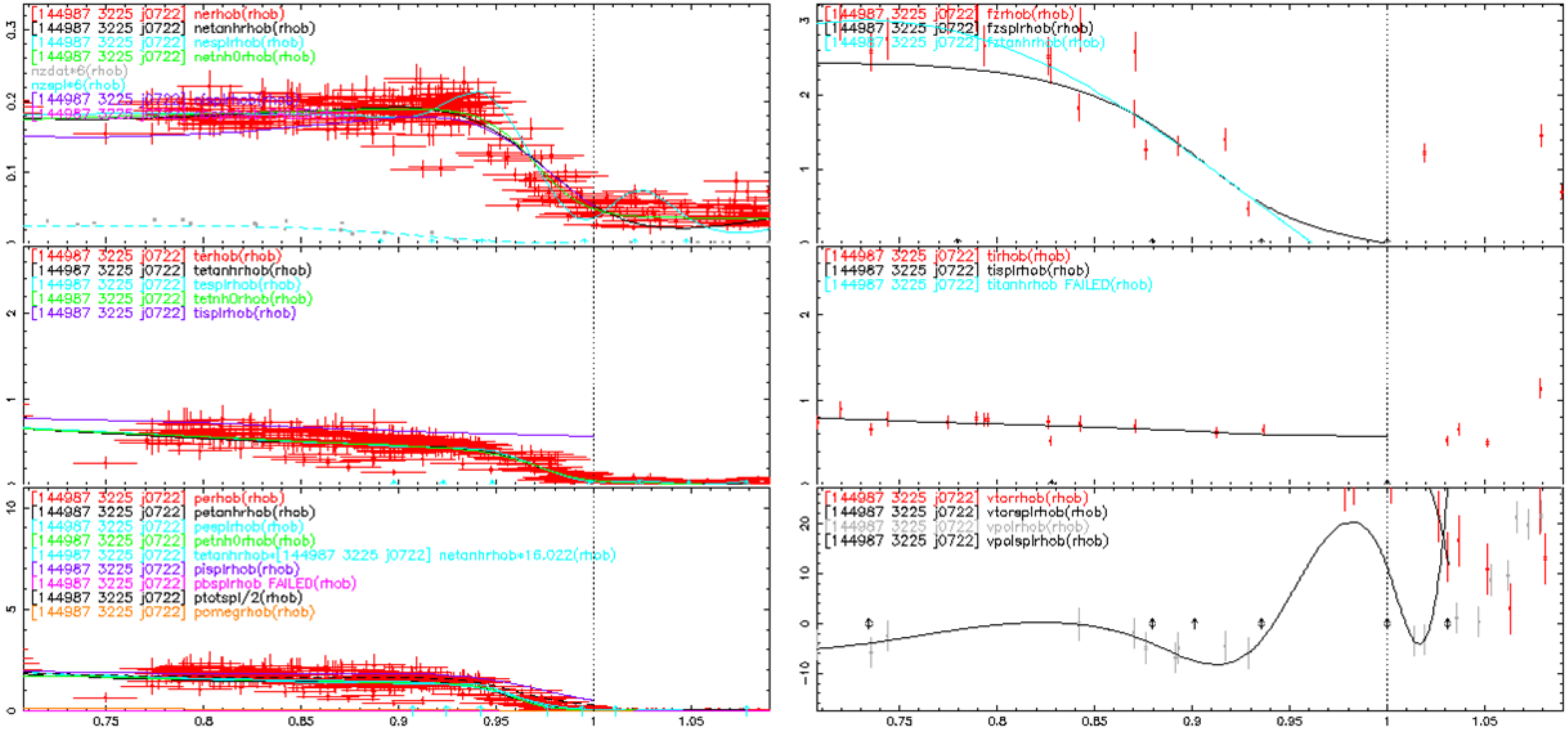


Figure 60: The black electron density tanh fit in the upper left does not do a great job following the data trend, especially in the top of the pedestal and after the separatrix, where an anomalous valley is found.

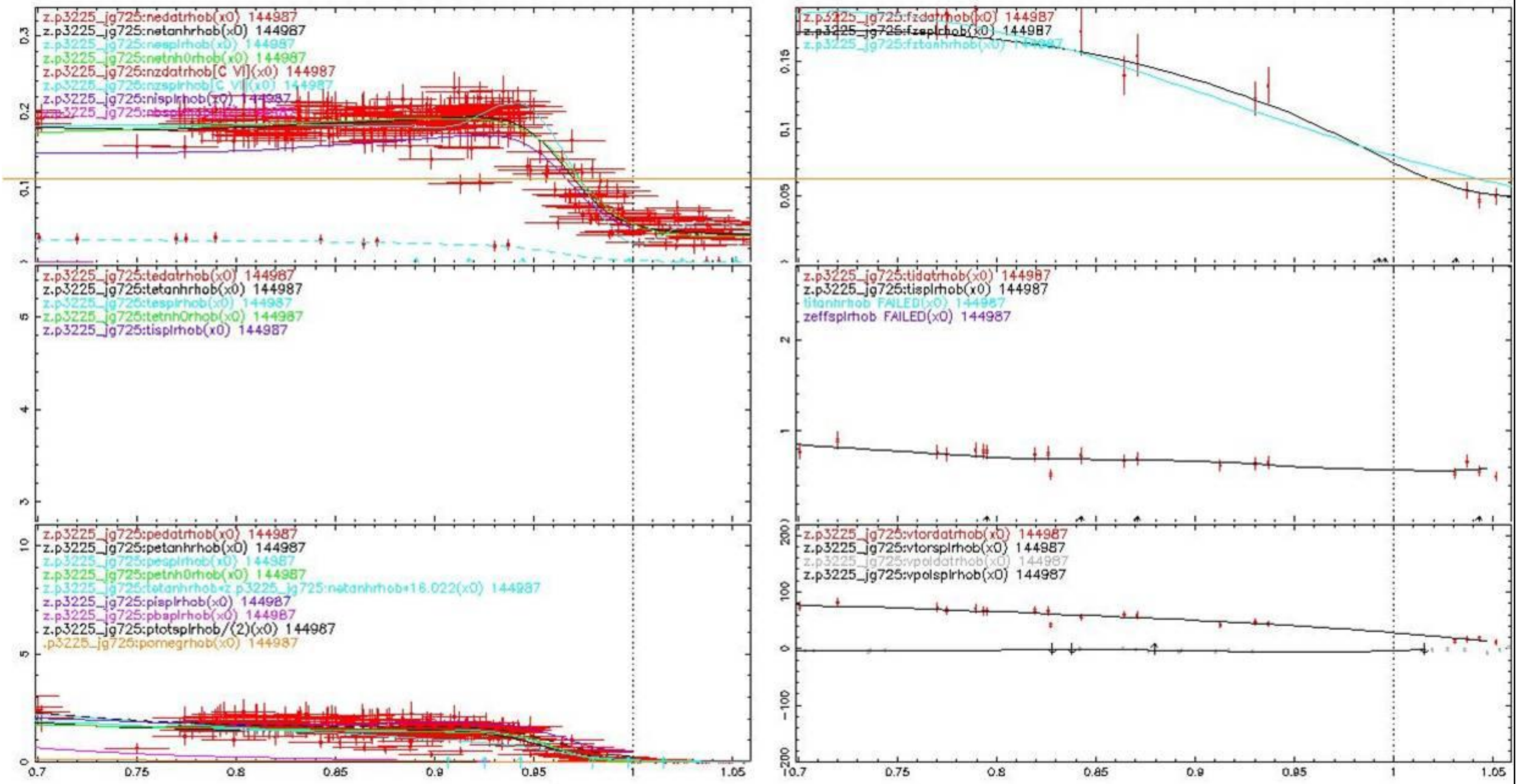


Figure 61: The same electron density graph in the upper left as shown in Figure 60, but with an improved tanh fit (black line).

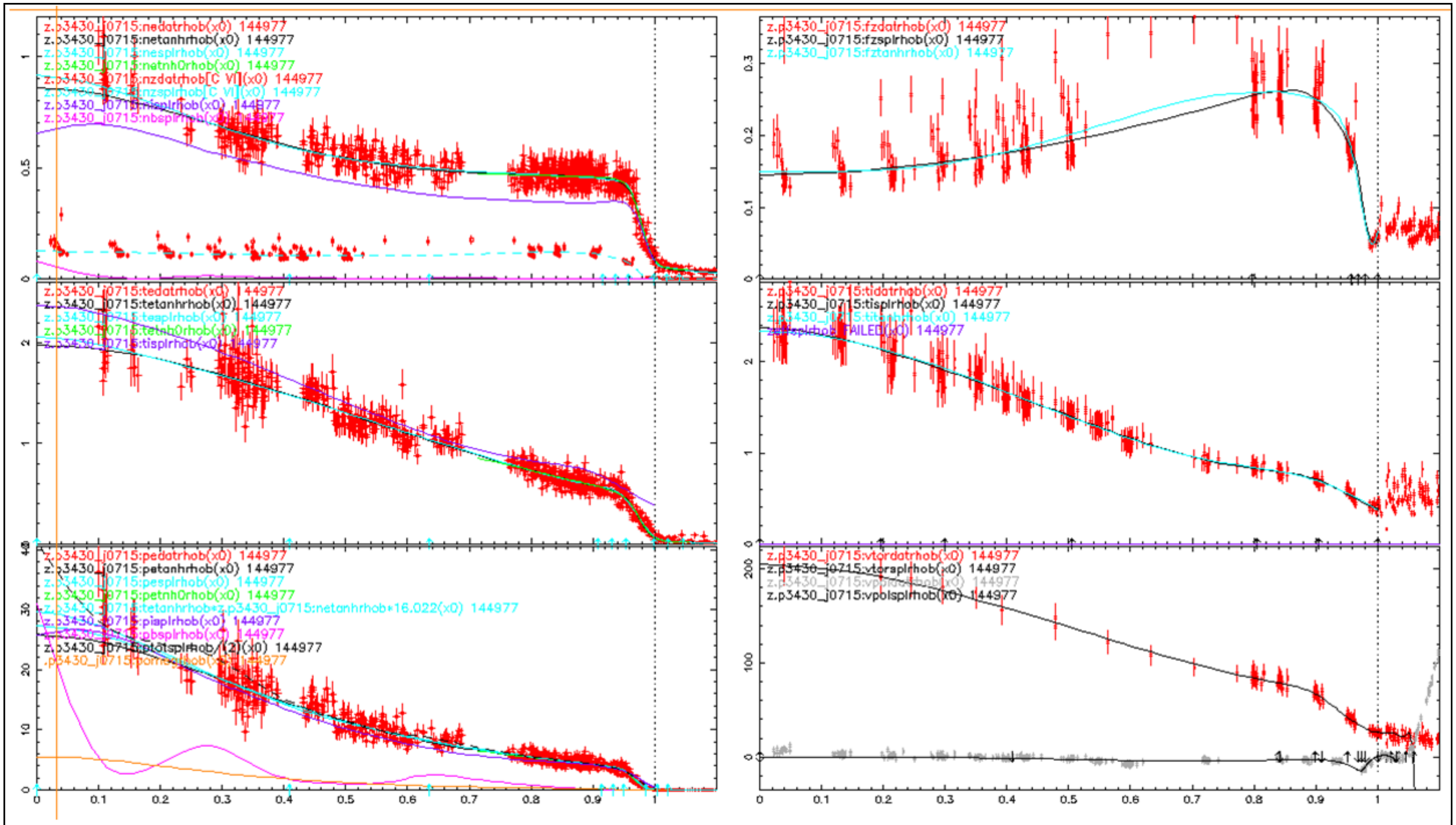


Figure 62: The impurity fraction graph shown in the upper right has a double trend across the entire radius, the upper set of data being about 0.05 above the lower set. Beginnings of a third trend can be seen about 0.1 above the lower set. The ion temperature and velocities fits in the center-right and lower right graphs are excellent across the edge and inside the separatrix

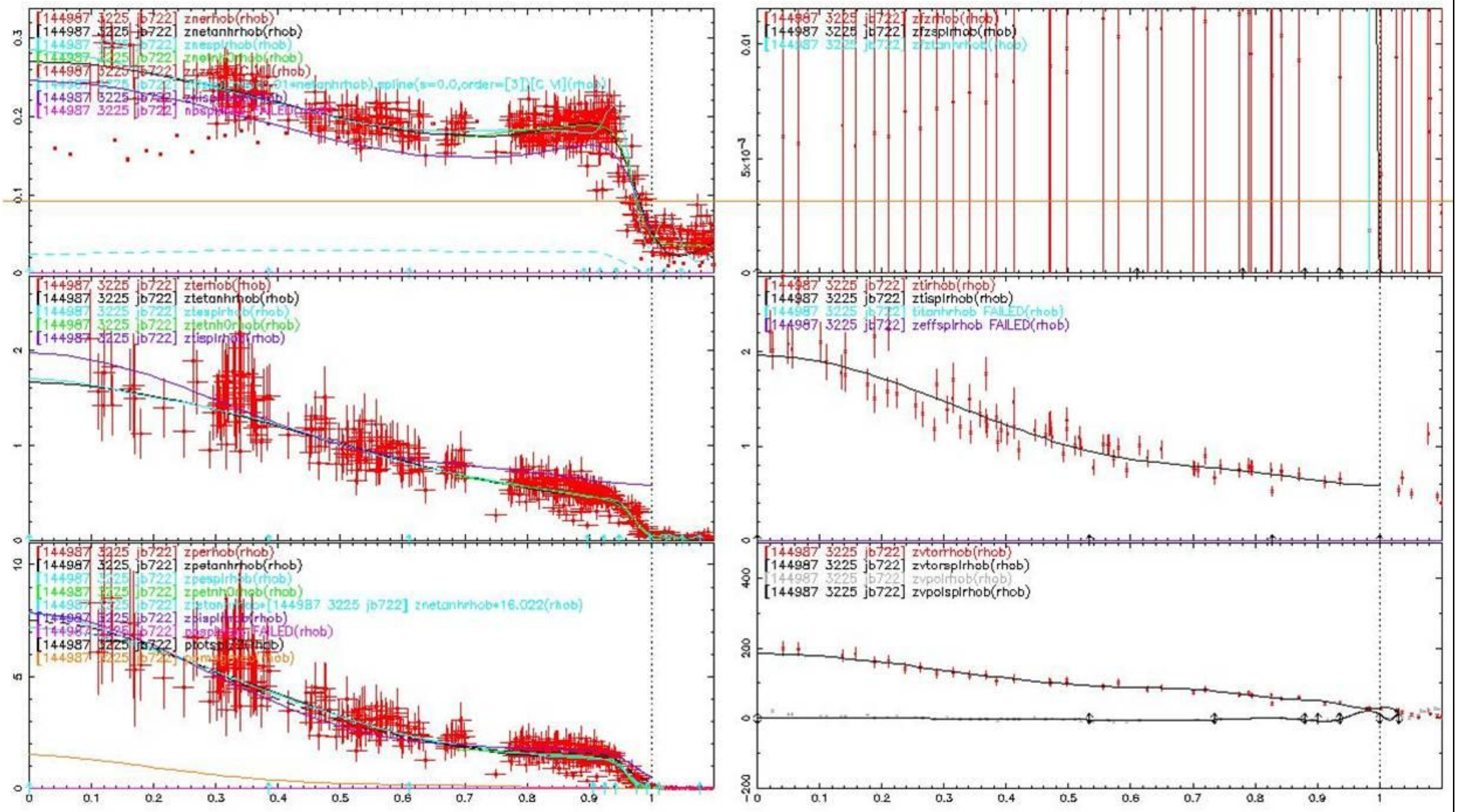


Figure 63: The upper right graph has too many spline knots, resulting in a bad fit.

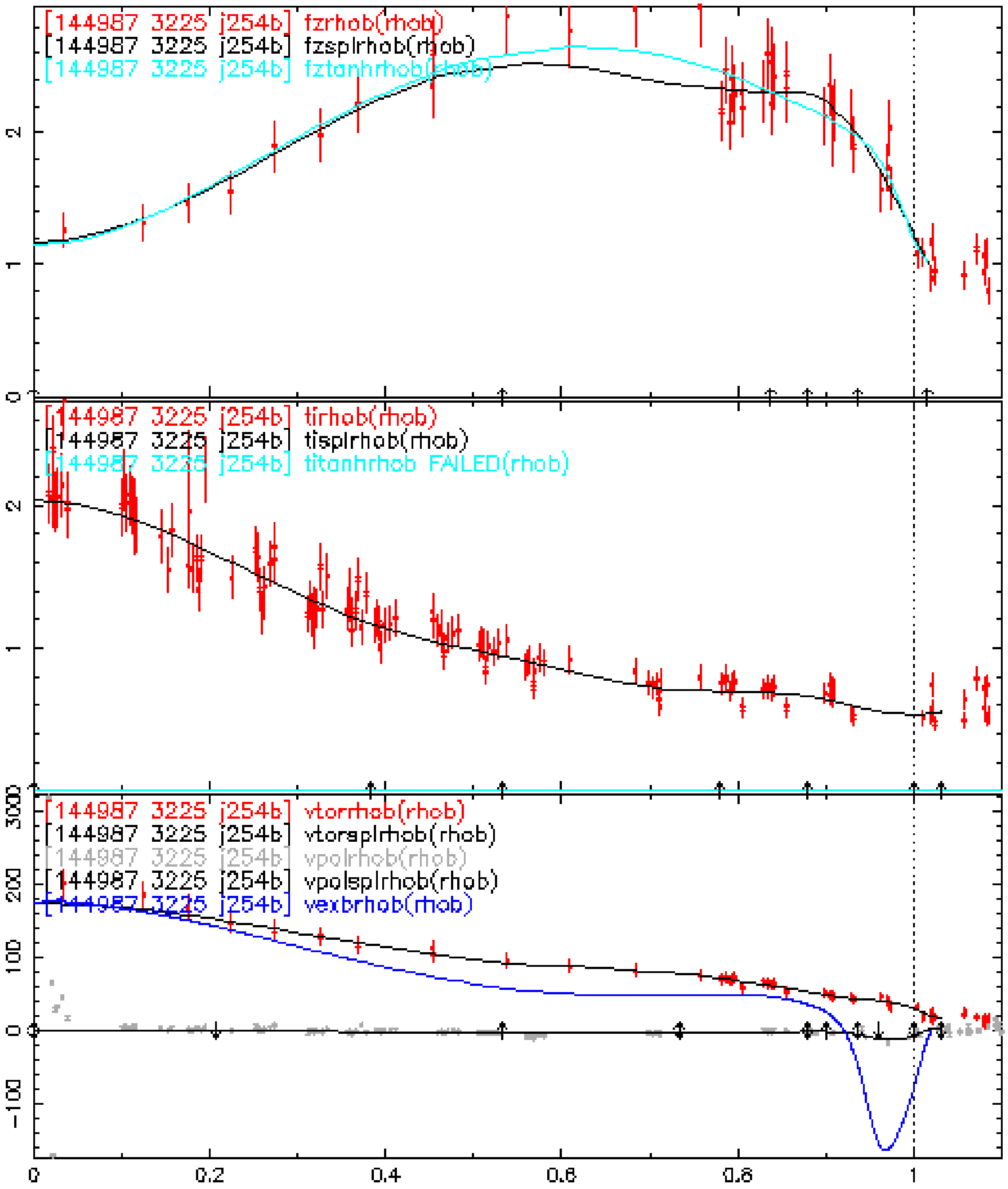


Figure 64: A f-z fit showing a large depression at the top of the data trend that does not match the data.

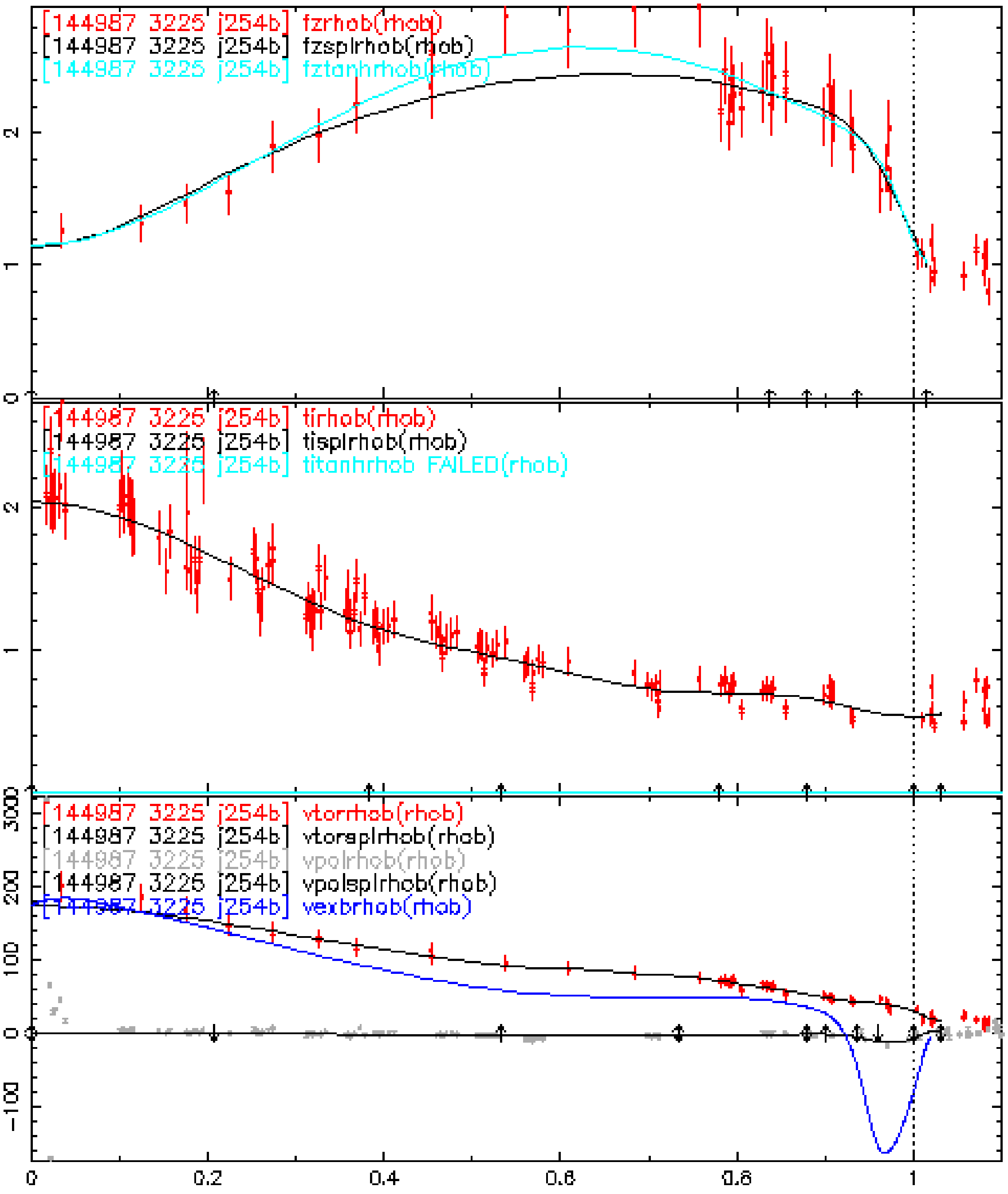


Figure 65: This is the same fz fit as shown in Figure 64, but after the knot at rho=0.5 has been removed.

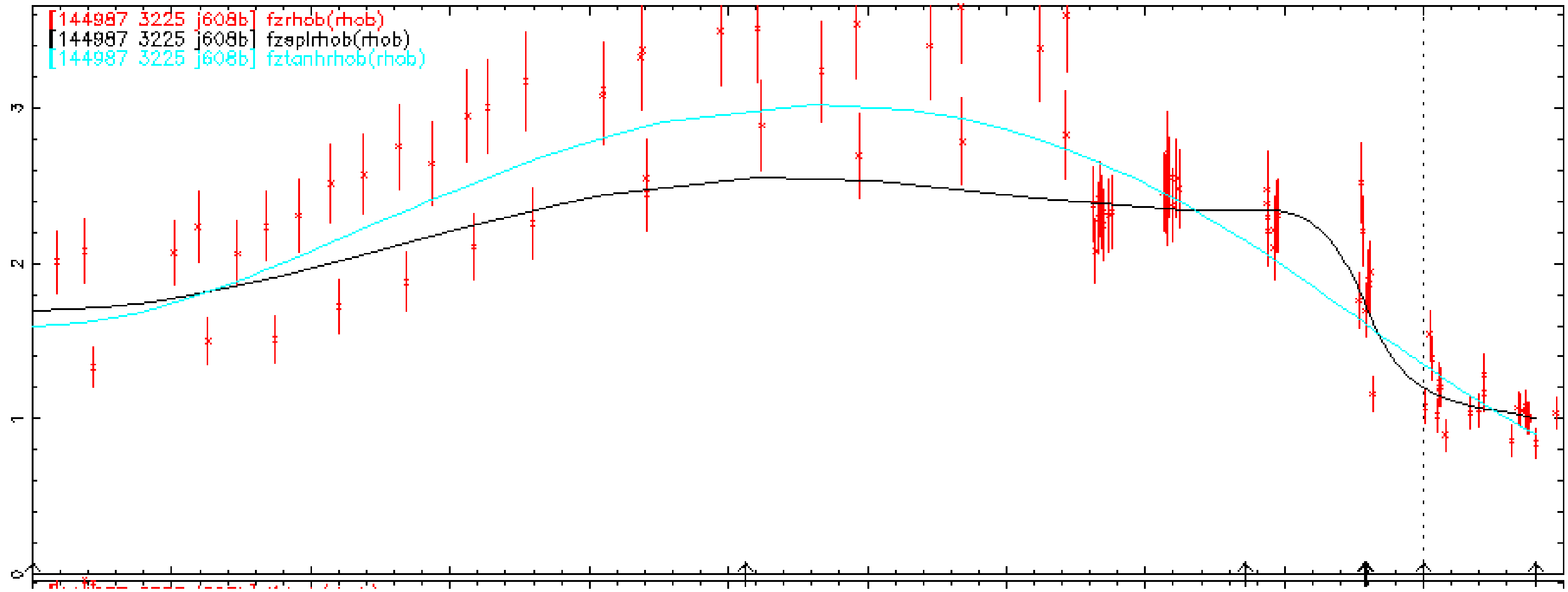


Figure 66: An fz fit that is nearly complete, but is slightly low on the clump of data points in the edge pedestal. A decrease in the range of the fit solved the problem. The "double trend" in fz is shown in this run, and also, an apparent data discontinuity is found at about rho = 0.75. After consulting with GA experts, it was determined that the large clump of data at rho = 0.75 was less reliable, and so it was removed in further analyses for this run, greatly simplifying the fit in the central part of the plasma radius.

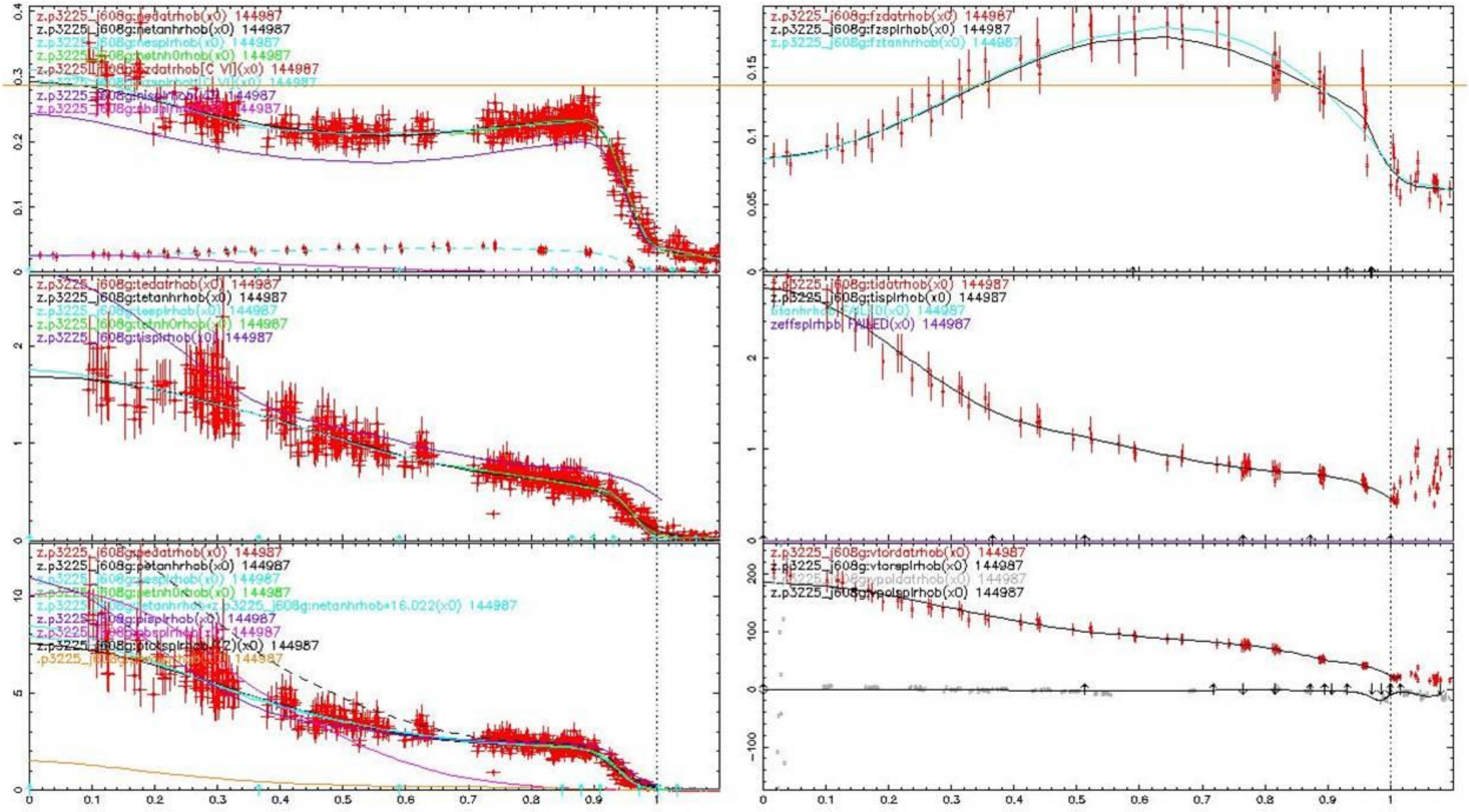


Figure 67: After removing an inaccurate measurement, and tuning the fit, the same f-z plot as is shown in Figure 66 is in the upper right plot.

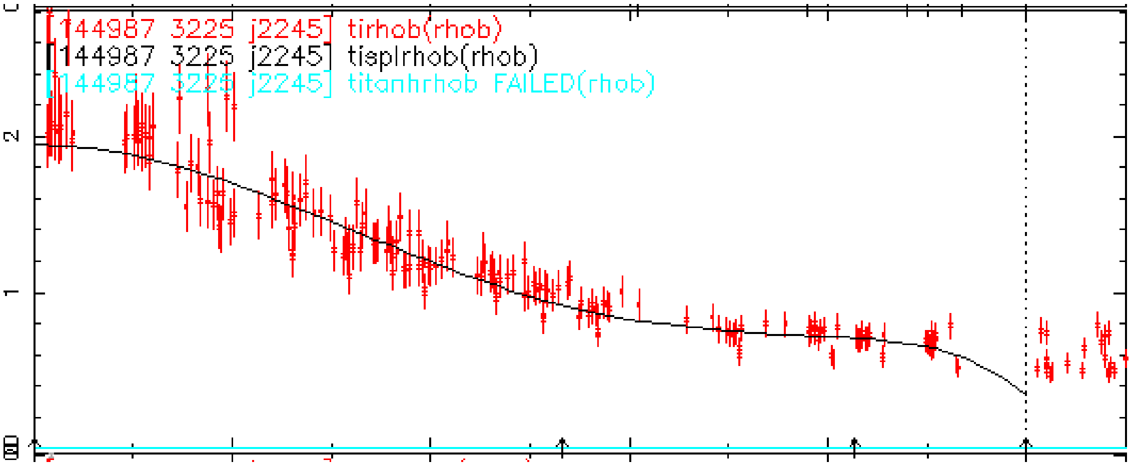


Figure 68: An ion temperature fit that mirrors the trends of the data relatively well until near the separatrix, where a large drop occurs.

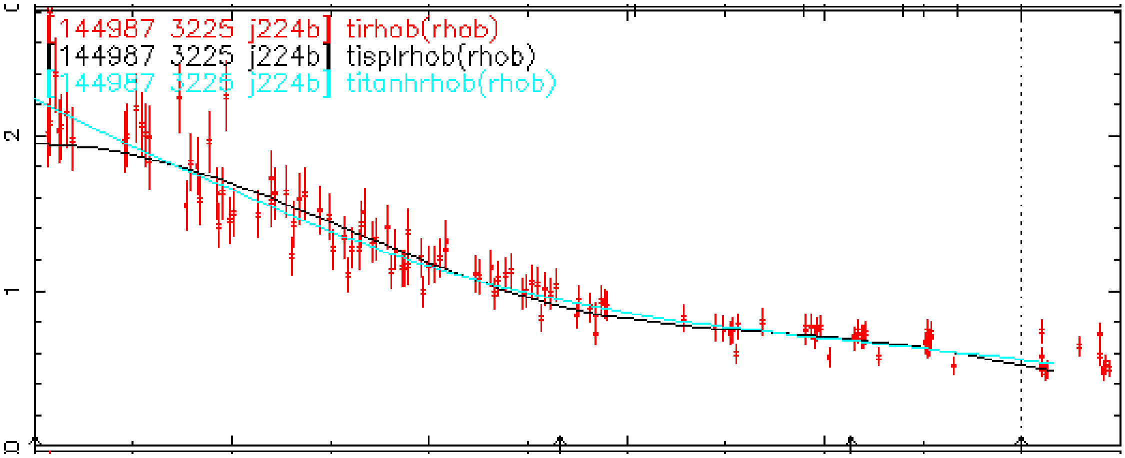


Figure 69: After extending the range, the fit in Figure 68 is more realistic near the separatrix.

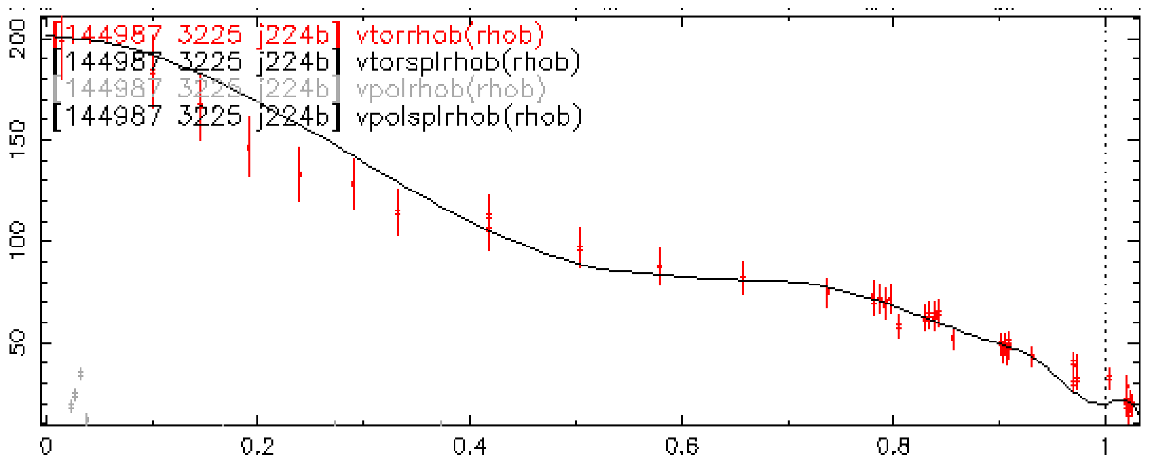


Figure 70: A carbon toroidal rotation velocity fit that is poor both near the core and in the edge pedestal.

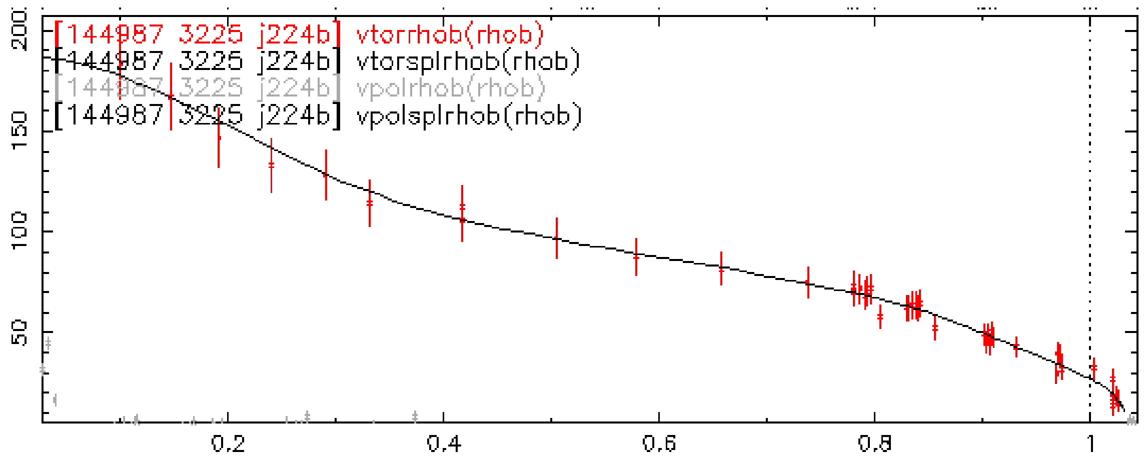


Figure 71: Adding a knot at 0.2 and removing the knot at 0.95 gives a much better fit than in Figure 70.

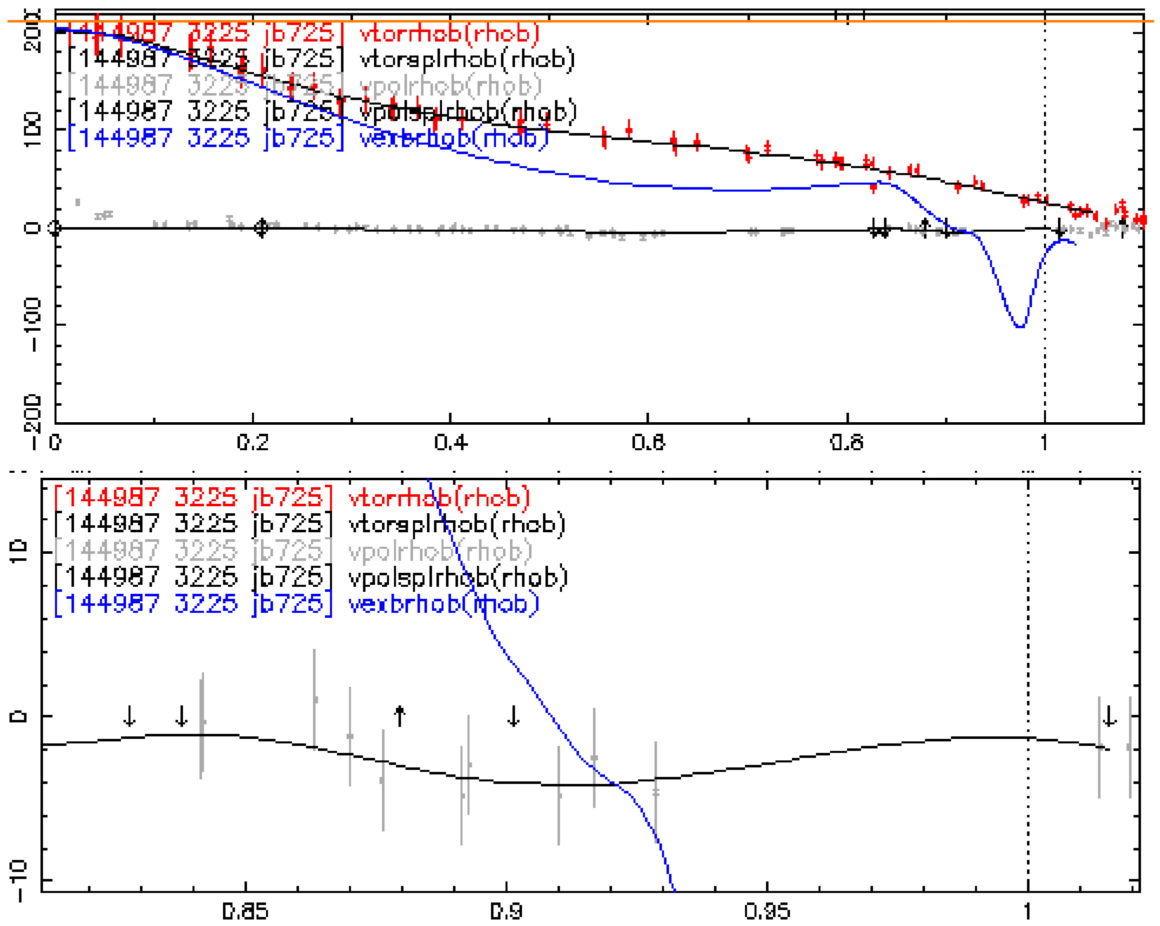


Figure 72: The upper graph shows the carbon toroidal and poloidal rotation velocity data and the radial electric field profile. The lower graph shows a clearer view of the carbon poloidal rotation velocity. The fit is poor for the carbon poloidal rotation velocity, and the radial electric field has a significant amount of unexpected structure.

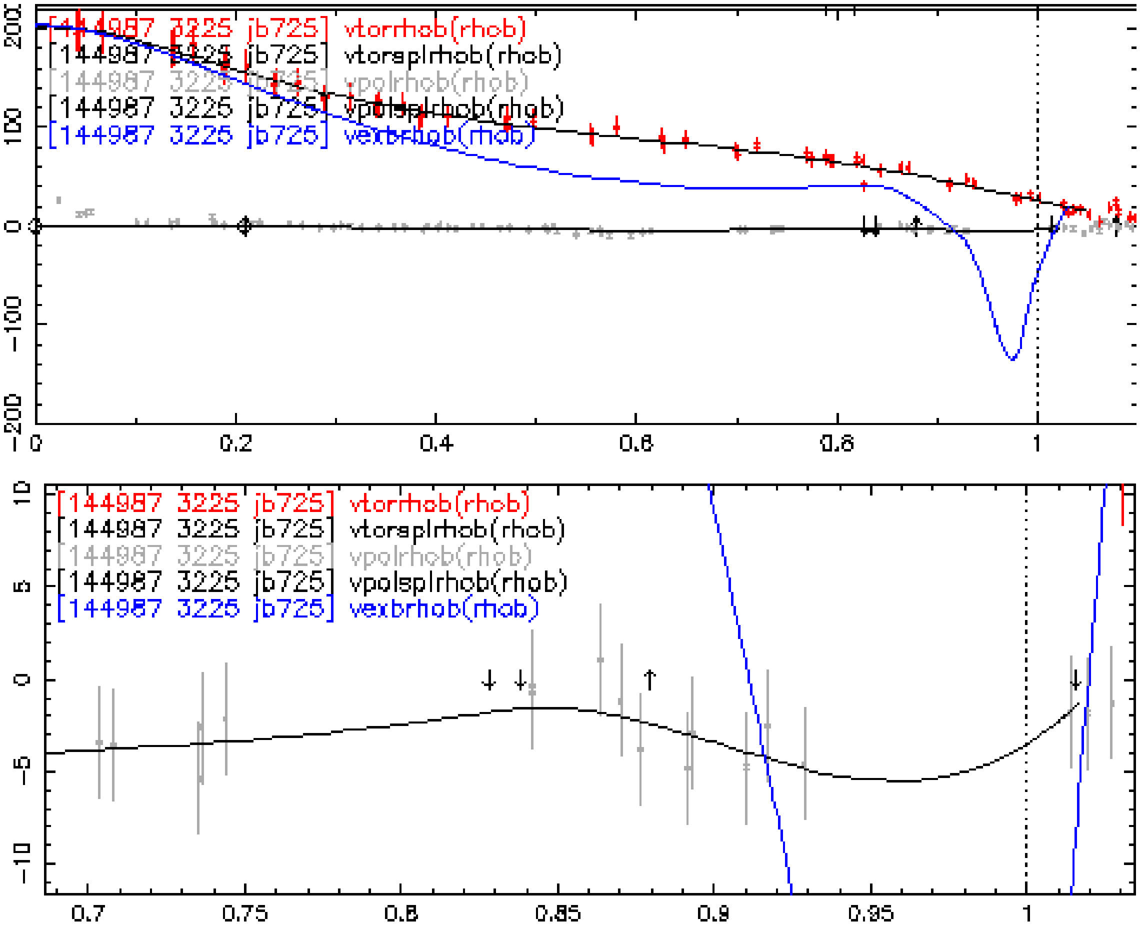


Figure 73: The same plots as in Figure 72, but with the spline knot at rho = 0.92 removed, a much improved carbon poloidal rotation velocity, and a smoother, more regular radial electric field profile.

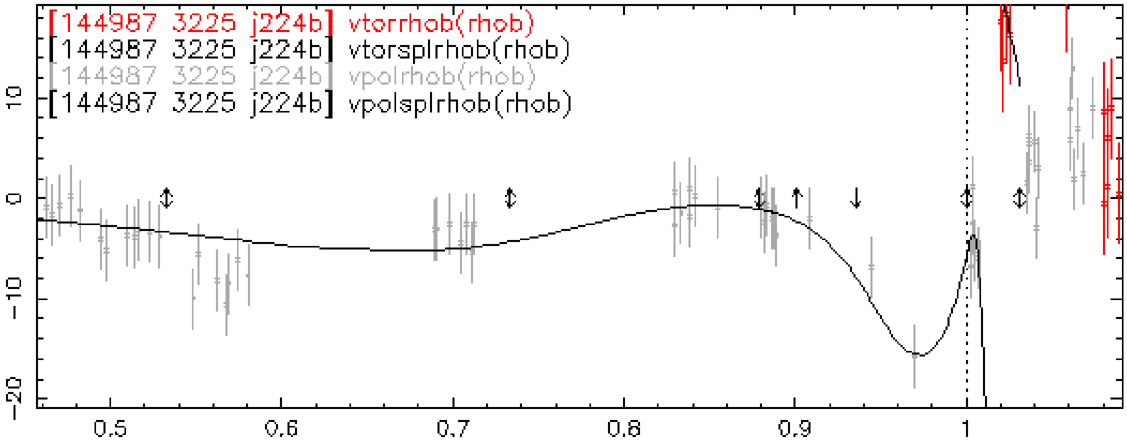


Figure 74: The fit outside the separatrix for the carbon poloidal rotation velocity is bad, and the fit misses the data trend for the inner side of the edge pedestal well structure.

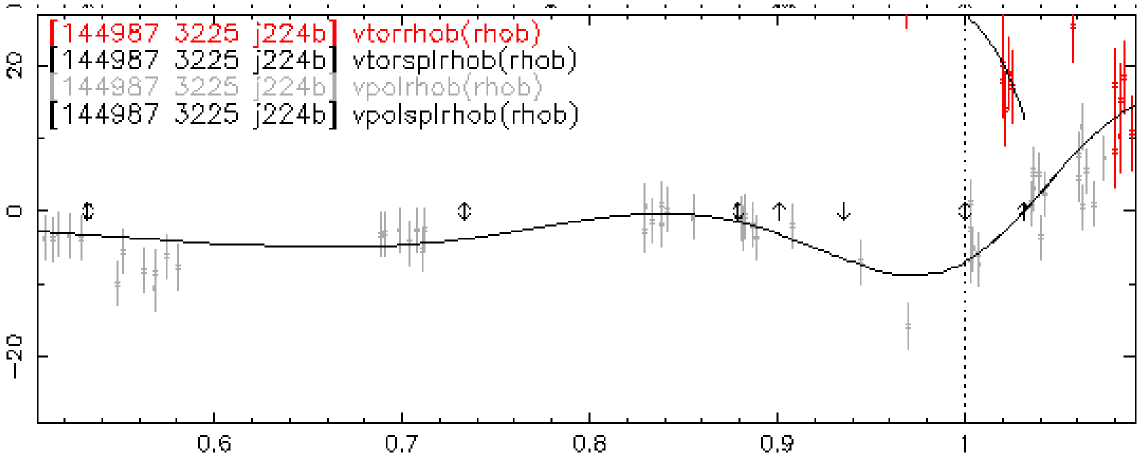


Figure 75: The range has been extended to rho = 1.1, greatly improving the fit outside the separatrix, but eliminating the edge pedestal well structure.

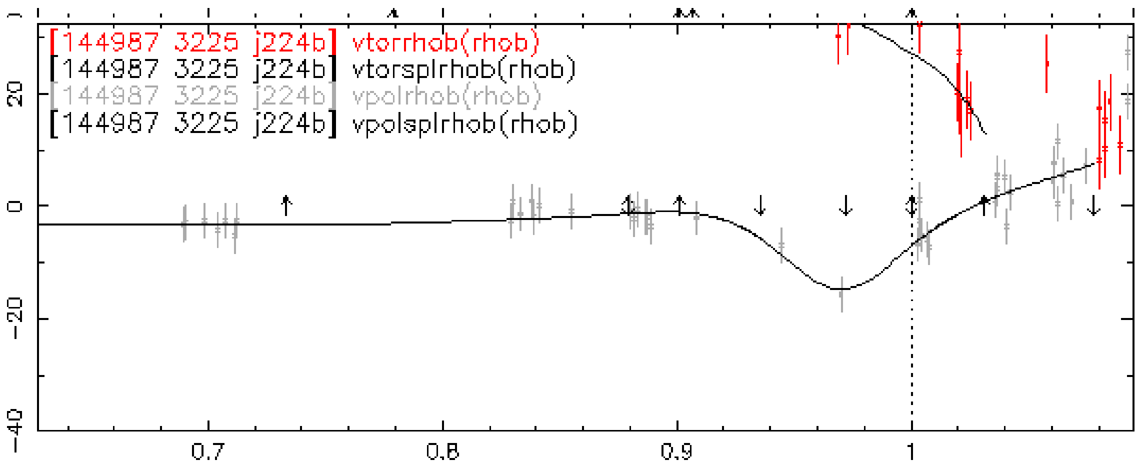


Figure 76: The two-knot system defining the well in Figure 74 has been altered to a three knot system, which better defines the well structure in the edge pedestal for the carbon poloidal rotation velocity fit. This caused the well to deepen from Figure 74, and more accurately capture the data trend (specifically, the data points around rho = 0.98).

APPENDIX B

PRIMER FOR USING THE AUTOMATED TOOLS DEVELOPED FOR EASING AND SHORTENING THE DIII-D TO GTEDGE MODELING WORKFLOW

The following sections describe the modeling workflow that allows use of the experimental DIII-D data in the GTEDGE modeling code, and the improvements that have been made to the process as a result of this research. The process consistently involves two procedures: gathering data from the GA DIII-D MDSPlus database by hand and using codes, and processing data from the fitted profiles. All of the data is then combined, and used to run the GTEDGE integrated modeling code {Stacey, 1998 #165;Stacey, 2000 #167;Stacey, 2001 #166}. The time required to complete the modeling workflow has been shortened considerably, and further reductions are possible through additional automation completed by other researchers. Be aware that many of the automation scripts are written in Matlab, which is necessary to run them. Also, the automation process was created for inter-ELM evolution analysis only, but the tools used may be useful for other types of analyses with further modification.

B.1 Retrieving and using the profile data created in A.7

B.1.1 General Notes

Once the profile fits for the experimental data have been completed, they can be retrieved from the MDSPlus database using a set of scripts developed for this purpose. These python scripts are maintained by R. J. Groebner at General Atomics, and in order to run them, it is necessary to ensure that the user’s account is set up properly, and the correct permissions have been obtained. Contact him in order to complete these steps, and gain access to the scripts.

NOTE all of the files mentioned in this guide will be made available at the frc.gatech.edu website.

B.1.2 Remotely accessing files in the user's directory on the GA network

When using the tools and scripts developed to ease the GTEDGE modeling workflow, it is important to have access not only to the GA network through a NoMachine terminal, but also through an FTP program. Instructions to set up such a program alongside the NoMachine client are found on the GA DIII-D computing instructions website, at the url found in A.1.1. This allows easy transfer of files on and off the local machine, and easier modification of scripts and other files.

B.1.3 Retrieving the profiles and their gradient scale lengths: drvr_pedxax

There are two sets of data required by GTEDGE that can be retrieved from the profile fits. The first consists of the regular profiles fit to the data, and their gradient scale lengths (gradient scale length of X = $X/(\partial X/\partial r)$). The second set consists of the time derivatives of measured and experimentally-derived values that are used by GTEDGE – such as the time derivative of the ion density used in the ion continuity equation (Eq. (8)). All of these variables are retrieved from the completed profile fits by sampling the fits at twenty-five points, which should correspond with the points on the GTEDGE grid. The locations where the fit is sampled are read by Groebner's scripts from the grid.dat file, which should be located in the folder where the scripts are run. To change the edge locations specified in grid.dat, simply replace the existing normalized radial coordinate with the one of interest, taking care to use the same format.

The scripts that retrieve the two sets of data exist in Dr. Groebner's directory on the GA network, and are run through a control script, which is present in the user's directory, and is modified by the user for their purposes. The control script for the profiles and gradient scale lengths is named drvr_pedxax.pro, and is shown in Figure 77. The control script intakes specific runs and the user must enter the shot number, timeid, and runid for each run of interest. The shot number, timeid, and runid must all be entered in their respective fields, and in order. The third line in drvr_pedxax.pro contains a set of shot numbers in brackets, and the shot numbers of the runs of interest should be entered here sequentially. The fourth line contains a set of numbers should be the timeids of the runs of interest, in the same order as the corresponding shot numbers. The fifth line contains a set of strings that should consist of the runids of the runs of interest, and again, must be in the same order as the shot numbers and timeids. When changing drvr_pedxax.pro, do not alter the formatting or delete programming symbols, such as ampersands or commas. Ensure that there are the same number of entries in the shot number, timeid, and runid sets, and repeat these values if they are common among several runs.

The three options in this script useful for this guide are the “dump”, “ps”, and “scalel” switches. The “dump” option writes text files with the profile values to the directory where drvr_pedxax.pro is located. The “ps” option writes a postscript file where the profiles are displayed to the local directory, and shows the postscript file on the desktop. The “scalel” switch causes the script to calculate the scale lengths for the various profiles. When generating data for GTEDGE, the “dump” and “scalel” options should be used; the “ps” option is optional; the “dump” option is also optional when not generating data for GTEDGE.

To run the control scripts, the user must log onto a server node on the GA network, run IDL, and compile and run the scripts. Return to the command prompt on the server node, and using the “cd X” and “cd/” commands to navigate through the file structure of the user’s directory, open the directory where the updated drvr_pedxax.pro and grid.dat are located. Once this is complete, an instance of IDL must be opened, and the file compiled. At the command prompt on the server node, once in the correct directory, enter

```
idl
```

and press return. This action should cause a session of IDL to begin. Once the IDL command prompt is available, compile the updated drvr_pedxax.pro file using the command

```
.r drvr_pedxax.pro
```

and press return. IDL should confirm that the file is compiled (it is best to make sure the file being compiled is closed).

To run the drvr_pedxax.pro script for GTEDGE data generation, run the following command.

```
drvr_pedxax, /scalel, /dump
```

This runs the drvr_pedxax script with the “scalel” and the “dump” options activated, enabling easy transfer of the resulting files to the local computer. The script usually ends its runs with a simple math error, indicating that the end of the list of runs has been reached. Despite the error message, all of the output files will have been written to the user’s directory. If another type of error occurs, such as in syntax, ensure that all these steps have been followed sequentially, that there is the same number of entries in each of the three input sets in drvr_pedxax.pro, and that everything is spelled correctly in the commands.

The following set of files should appear in the local directory for each run, along with .ps files, if that option was selected.

profile fits: .dat files give the xy coordinates for the fitted profile over the entire range, while .fit files are from the fitted profiles, and are sampled at the points specified in grid.dat. All of the files should be of the form p*shot_number*_*timeid*_*runid*_*fit*.*dat or fit*, and the set of files written to the directory should include those with the following strings located in place of the *fit* before the file extension: ersplrhob, fzsplrhob, netanhrhob, nisplrhob, tetanhrhob, tisplrhob, vpolsplrhob, and vtorsplrhob. The form of the *fit* string gives the abbreviated variable name (e.g. “er” for radial electric field), the type of fit (“tanh” or “spl” for hyperbolic tangent or b-spline fit, and “dat” for raw data), and the x-axis values (“rhob” for normalized radius – this should be used with GTEDGE). For example, one set of data should have the following label.

p*shot number*_*timeid*_*runid*_ersplrhob.fit

This would be the radial electric field profile derived from carbon radial momentum balance (Eq. (2)) and the other spline fits, sampled at the points specified in grid.dat, and plotted against the normalized radius. The .fit and .dat files for each of these variables should appear in the local directory after a run of drvr_pedxax.

scale_lengths: The scale lengths fits are extremely similar to the profiles fits in many ways. Several key differences are useful in the process of modeling, and inform the analysis. First, the script never generates .dat files for the scale length fits, but generates .fit files with a “_sc” inserted before the extension. The data in the scale length fit file comes in three columns (xyz) instead of two (xy), where the first (x) is the x-axis values, the second (y) is the gradient scale length values for the variable, and the third (z) is the raw variable value. The columns are also labeled, showing the definition of the gradient scale length and crucially, the units of the raw variable values found in the third row. These definitions are useful to ensure that units and expressions are consistent across the analysis.

B.1.4 The time derivatives of the fitted profiles: drvr_ped_timed

The drvr_ped_timed.pro script calculates several time derivatives at each radial location in the edge. As in the drvr_pedxax.pro script, the drvr_ped_timed.pro script uses the shot number, timeid, and runid of the run, but also utilizes another quantity which positions the run in time. This quantity is “elm_dt, which is written to the pgadmin3 table after the run is written to MDSPlus. It can be found near the end of the row in pgadmin3, and must be collected for each run.

The inputs in the drvr_ped_timed.pro script are similar to those in the drvr_pedxax.pro script, but the former is more flexible in order to accommodate various types of analysis. This guide shows how to use the script to calculate time derivatives between different intervals in the same shot (i.e. inter-ELM evolution analysis), or different time periods in the same shot (i.e. examining how transport changes across the L-H boundary), but other functionality may be available.

Figure 78 shows the two variations of the drvr_ped_timed.pro file: Figure 78.1 and Figure 78.2, respectively. The shot number is entered normally in both instances. In Figure 78.1, several time periods across a plasma event are analyzed, and they are arranged in chronological order in a set. Then, the runid for each of the runs is input as the runid. The elm_dt values for each of the runs are then entered in the order corresponding with the placement of the timeids in a set. In Figure 78.2, a standard inter-ELM analysis is underway, and the shot number and timeid for all of the runs are the same. They are entered, and the runids for each run are entered sequentially in a set. Then, the elm_dt values are entered in the same order as the corresponding runids.

To run drvr_ped_timed.pro, the user should follow the same procedure used for the drvr_pedxax.pro script, by navigating to the directory it shares with grid.dat, entering idl, and recompiling the updated script. To run the time derivatives script, and write the time derivative profiles to the user’s directory, the user must type the following command at the command line in idl:

drvr_ped_timed, /dump

The “ps” switch can be used if desired. Always be sure that in the drvr_ped_timed.pro script that all runs are separately identified by a unique combination of the shot number, timeid, and runid. It is also important to note that in order for the time derivatives to be accurate, the user should not include any runids in the drvr_ped_timed run that may be incorrect, or that will not be used in the final analysis (such as intervals overlapping with the ELM event, or test runs).

The files written to the local directory will be of the form:

p*shot_number*_*timeid*_*runid*_dln*value*_dt.dat

where *value* is either “ne”, “we”, or “wi”, representing electron density, electron energy, and ion energy, respectively. In this case, the .dat files show the xy pairs of the time derivative profiles sampled at locations specified by the grid.dat file, and no .fit files are produced.

B.1.5 The jpf2_fitstoexcelplot_2014.m script

Once the files have been deposited into the remote directory, the user should retrieve the files from their remote directory using WinSCP, or whichever file transfer software they have configured to access their GA directory. There should be a set of files for each runid entered into the scripts that includes .fit files for each parameter profile, their gradient scale length profiles, and time derivative profiles for the three values mentioned in B.1.4. It is generally best to locate all of these files in a single directory.

When all of the files have been assembled in a local directory, copy the “xlsColNum2Str.m” and the “jpf2_fitstoexcelplot_2014.m” files to the same directory as the .fit files. The Matlab scripts with names beginning in “jpf2” are the functional parts of the automation process, while the other files are necessary, but usually will not need to be modified. The function of this script (jpf2_fitstoexcelplot_2014.m) is to write the profiles in the .fit files to an Excel spreadsheet, and create graphs for each profile. The user should open the file, and carefully replace only the shot number and timeid in line 5 with the appropriate data for the shot being examined, without modifying spacing, underscores, or other parts of the code. The runids for the runs being examined should be entered in chronological order in the brackets on line 7, taking care to leave the underscore after each runid.

After making these modifications, the user should run the Matlab script jpf2_fitstoexcelplot_2014.m, and wait for it to complete. An Excel file will be opened, and each parameter will be plotted on a separate worksheet, with the same parameter from each runid plotted and labeled in a graph, showing the evolution with time of the parameter. This provides an easy way to qualitatively gauge the data, and easily find outliers and other problems that may be present. The Excel plot will be named:

p*shot_number*_*timeid*_data.xlsx

and will be saved to the local directory. This visualization is very useful for spotting aberrations in the data or the derivatives that should be addressed before going into the modeling phase, and the profile plots should be examined for such issues.

B.1.6 Special procedures for custom-defined intervals (from A.5.3)

If custom intervals are defined, the user must manually calculate the average time characteristics of the custom interval in order to obtain the elm_dt and other values necessary to calculate time derivatives. Firstly, to calculate the boundaries of the custom interval, follow the procedure in A.5.3, which is briefly reprised here:

> Run profiles.py with a specified interval of 0 to 1, generating a composite time interval which includes each entire inter-ELM period inside the time period being analyzed. This generates a file named:

time_windows_*shot_number*_*timeid*_*runid*

which has no extension, but can be opened with a text editor. It includes in two columns, the starting and ending time of each inter-ELM period within the analysis period. The user must then partition each inter-ELM period into custom intervals, after which the beginning and ending time of each custom interval in each of the inter-ELM periods of the analysis period must also be recorded. In essence, the user is manually reprising the process of creating a composite inter-ELM period since custom intervals are desired. This is only necessary in special situations, such as those when non-sequential sections of an analysis period are desired.

Once the boundaries of each custom interval are recorded for each inter-ELM period in the analysis period, they must be recorded in a "time_windows" file with the exact formatting of the one generated by profiles.py and written to the user directory. The beginning and ending times of the custom intervals of each inter-ELM period must be listed chronologically in the shot, in ascending order in the two columns. After completing the editing, take care not to save the file with an extension, as it will not be recognized by profiles.py.

Once the time boundaries have been calculated for each custom interval, it is possible to calculate the elm_dt value for use in the drvr_ped_timed.pro script. The elm_dt value is calculated using the chronological properties of the custom interval and the composite inter-ELM period. It can be calculated by taking finding the elapsed time from the beginning of the composite inter-ELM period to the middle of the selected custom interval of the inter-ELM period. For example, if the inter-ELM period started at 25 ms, and the custom interval of the selected inter-ELM period stretched from 75 ms to 125 ms, the elm_dt value would be 75 ms. When calculating this variable, be sure to check formulas and methods against the elm_dt values calculated by profiles.py during normal operation.

B.2 Retrieving MDSPlus Data

Additional parameters are required to model the plasma, and some of them are available from the MDSPlus database. The scripts in this section have been written to retrieve this data, process it, and average it to produce a single value representative of each parameter value for each interval, as GTEDGE requires.

B.2.1 The jpf2_getdata_2014b.pro script

The jpf2_getdata_2014b.pro script retrieves all measurements for a given parameter during the quasi-steady-state period for each shot. It is an IDL script, similar to those in the B.1.3, and must similarly be compiled and run from the IDL command prompt. The shot number, and beginning and ending time of the quasi-steady-state analysis period must be updated early in the script. Then, the script calls the measured data for each of the desired parameters from the MDSPlus database, and stores it in a xy format.

The first section of the script, from lines 5 to 9, defines the shot number the data will be retrieved for, and the time period during which measurements are to be selected. From lines 13-32, the script uses the gadat2 idl function (further details on this function and its form can be found at https://diii-d.gat.com/diii-d/Gadat) to retrieve a set of xy pairs from the MDSPlus database, with the x column representing the time the measurements were taken, and the y column representing the values. After the call to the function gadat2, the next variable is the name of the vector containing the measurement times for the parameter, the next is the name of the values vector, and the next is the name of the variable in MDSPlus to be retrieved. The definitions of the other entries are apparent.

It is important to note here that the “tree” the data is retrieved from may have to be altered to obtain good data, retrieve sufficient amounts of it, or limit the data points to a reasonable number. If no tree is specified, and the call to gadat2 ends after entering the ending time, data from the efit01 tree is used. This data has the lowest time resolution, and usually measurements are available every twenty milliseconds. The other main option is efit03, which can be specified using the call “tree=’efit03’” after the ending time is entered. This option usually gives the highest time resolution. In some cases, especially when retrieving those variables measured by TS systems, the EFIT03 tree can provide too much data, producing a very large text file. In those cases, defaulting to the lower time resolution EFIT01 tree is preferable. Additional circumstances may mandate the use of one tree or another, such as equipment outages during a shot that disrupt data collection for a certain tree, or bad data. For instance, sometimes the xy pairs will be returned with a value of zero for every entry. If sufficient data is unavailable using the efit01 or efit03 trees, it is best to check with the GA contact and see what can be done. Note that different trees will provided different quantities of xy pairs, and vector lengths will not match if data from different trees is processed by the same loops.

The script then opens text files for each variable in lines 35-52, which will be filled with the xy values for the various parameters. Next, to ensure the loops run properly, the length of the xy columns of measured data must be determined. Usually, most of the variables have a common length, but several may not. For instance, the injected power (pinj) and ohmic power (poh) often have better data in the efit01 tree, and will be a different length than the other values, while the line average density (nebar_v2), which is measured by the TS system, often has many more data points than the other parameters.

A set of loops is used to write each xy pair to its respective text file, and the length of each set of xy pairs (lines 54-58) is used to set boundaries for the loops (lines 61-89). Finally, the text files are closed, saved, and written to the local directory, providing the user with the full set of GA measurements of a set of given parameters during a specific shot and time period.

To run the script after updating the file, log onto the server node, and navigate to the directory which contains the file. Open IDL as previously instructed, compile the updated file, and run it by typing

```
jpf2_getdata_2014b
```

This should result in a large number of text files being dumped to the directory that the script is located in. They should include text files labeled: aux_power, btor, elongation, intcpt_rin, intcpt_rout, intcpt_zin, intcpt_zout, lineavg_density, major_radius, minor_radius, ohmic_power, plasma_current, q95, ssi95, taumhd, triangularity, xpoint_r, and xpoint_z.txt. Each of these files will have a set of xy pairs identifying the time and value for any measurements found within the period specified in the script.

Errors when running this script generally relate to inconsistencies between the lengths of xy pairs for a parameter and loop boundaries, a lack of data for the selected tree, or some other common programming error. The script can be updated as necessary to deal with these simple problems. A lack of data for a certain parameter, or incorrect values (such as y = -20.00 for each value of the major radius) may occur, and usually indicate the need to draw the data from a different tree. Inspect the sets of data points generated to see if they are all zeros, or contain a these types of improbable values. If data problems are encountered, and changing trees is insufficient to fix the problems, it is generally best to contact the GA expert for assistance in obtaining other data, or improving the existing set.

B.2.2 The jpf2_mdsplus_data_processor_2014a.m script

The GTEDGE integrated modeling code uses one value to represent the parameter value over the entire analysis period for each of those variables mentioned in the previous section. One functional method of obtaining a representative value is to plot the parameter value against time, remove the sections not relevant to the analysis, and visually find a representative value across the remaining time. To be more precise, an averaging script was written to process the data, and reliably perform the task the visual representation is approximating. In simpler analyses, when a single period is being examined, the visual approximation may be sufficiently accurate for the purposes of the research. However, in inter-ELM evolution analysis, when using the intervals described throughout this document, the complexity of the task is increased, and the use of this or a similar script is recommended.

The script takes in the full set of measurements made during the quasi-steady-state period, then filters out those erroneous values (usually represented as zero in the xy pairs) that can skew the data. Then, the script interpolates the existing measured values to a grid with spacing of 0.25 ms to ensure clumping of data does not affect the accuracy of the averaging process, as the raw data is not sampled with a uniform frequency. Next, the script reads in the set of time periods that make up an interval of a composite inter-ELM period, which can be directly taken from the “time_windows” files generated in the user’s directory when profiles.py is run for each interval. For the interval being analyzes, the script removes those xy pairs (now interpolated to a fine grid) that are not included in the set of periods that make up the interval of the composite inter-ELM period (from the time_windows files). Then, the script averages the remaining xy points on the grid, and generates an average value for that interval of the inter-ELM period. All of the sets of xy pairs are also generated and written to the local directory for inspection.

The first step is to retrieve the MDSPlus xy pairs text files written in the previous section, and place them in the same folder as the script. Next, the same should be done with the with the "time_windows" files from the server. Ensure that the "time_windows" files for the intervals are accurate – i.e. they are continuous, and do not repeat. The time_windows files are contained in the user's remote directory in files that follow this convention:

time_windows_*shot number*_*timeid*_*runid*

The user should update the Matlab script on lines 28-30 with the current shot, timeid, and runid information, and run the script. The script will write the following files to the local directory: the full set of measurements (xy_pairs.txt), xy_pairs less any xy points both zero (zero_removed_xy_pairs.txt), the number of deleted xy points (displayed in the command window), the zero_removed_xy_pairs in the timing window for the selected interval (filtered_xy_pairs.txt), the gridpoints for the 0.25 ms grid inside the interval bounds (gridpoints.txt), the filtered_xy_pairs interpolated to a 0.25 ms grid inside the interval bounds (final_xy_pairs.txt), and the average value for each parameter inside the interval bounds (averages.txt).

It is imperative to check the final_xy_pairs.txt file for each parameter, and see if the data contained therein may give misleading averages. For instance, for some parameters such as the confinement time (taumhd), the time can spike by a factor of six or more, wildly inflating the average beyond the normal low values found throughout most of the intervals. Sometimes, manual removal of outliers and other problematic values is absolutely necessary to give quality results. Consult with the academic advisor when taking this step. To fix these issues, sometimes it may be useful to return to the visual method of approximating an average value for an especially problematic parameter, or simply average the filtered_xy_pairs.txt values after removing outliers.

Recall that the outliers are multiplied once they are interpolated on the fine mesh, causing the problem to metastasize if addressed after that step.

B.2.3 Other data for GTEDGE

Once these averages have been generated, it is useful to input them into the Excel spreadsheet labeled

shot number inputs.xlsx

This logs the values retrieved from MDSPlus, and also provides a listing of those variables that must be retrieved manually from the profile fits. Several characteristics of the plasma are relevant to modeling the plasma, but are not automatically generated with sufficient accuracy. These include the value for the electron density, electron temperature, and ion temperature at the top of the edge pedestal, these same parameter values at the separatrix, and also at the center of the plasma. It is important to be consistent in using the profile characteristics to identify the locations of the separatrix and top of the pedestal. In addition, due to the sparse data available in the central part of the plasma, the central plasma density profiles are often not representative of the data trend. It is therefore important to use the data trend to determine a sensible central density value for the plasma, and be consistent by using the same criteria to determine the central density values across the various intervals for the same shot. The radial locations of these measurements are also required. It is recommended to use the Profplot program and the crosshairs option to gather this data, and to enter it into the *shot number*_inputs.xlsx spreadsheet, where spaces are provided for each necessary variable. Once this is complete, the data gathering for the GTEDGE modeling run is over.

To summarize, the data gathered visually from the fits in this section, and the data gathered from MDSPlus in the previous section using the averaging script, should all be entered into the *shot_number*_inputs.xlsx file. This, and the profile fits, will be all that is necessary to construct the GTEDGE input file. These values all being located in the same place speeds their entry into the next script.

B.3 Building the GTEDGE Input File: The jpf2_soldata_constructor_2014_*shot number*_*runid*.m Script.

The purpose of this script is to take a template input file for GTEDGE, and write all of the recently gathered data to it, sparing the researcher much time, effort, and opportunity for error. Several files must be placed in the same directory as the script, including: the results from drvr_pedxax and drvr_ped_timed; and the soldatatemplate.for. Then, the script file must be updated with the average values for the MDSPlus data and the values manually gathered from the profile fits, the processes to obtain both of which are described in the previous section. The script then reads in the entire soldatatemplate.for file as a string, searches the string for the specific locations of the various parameters that must be updated, emplaces the new values, and writes the string to a new file, entitled:

p*shot_number*_*timeid*_*current_date*_*runid*_.txt

A separate input file must be constructed for each runid.

The first part of the script, from line 1 to 30, sets up the variable input vectors to be read. Between lines 23 and 27 are the variable names that will be input into the input file. If any input need to be changed, such as the time derivative variable names, this is the first place to begin modifying the script. Between lines 31 and 55, the variables from MDSPlus should be input in their respective locations and in the existing formats. Between lines 57 and 72, the parameters retrieved from the profile fits are listed, and they should be entered in the correct locations and formats. From lines 74 to 89, several variable values are calculated that use the MDSPlus and fit data as inputs. The rest of the script contains the machinery updates the input file.

This script uses the concept called regular expressions to tell the computer what text patterns are being sought. Regular expressions are extremely useful, but have a steep learning curve. When expressions are encountered in the script that appear to be jibberish, they are usually regular expressions. From lines 90 to 135, the script constructs vectors with the appropriate names, units, and syntaxes of the various parameters to be updated. From lines 136 to 186, the profile fits are read in, any undefined values are noted and removed, and one 3D matrix is created with all of the data from the drvr_pedxax and drvr_ped_timed text files. The rest of the script reads in the soldatatemplate.for file into a string, iteratively searches the string for the expressions to be updated, inserts the new values into the string with the appropriate format in the correct location, and writes the fully updated string to a new file, which is output. This output is a file ready to be run in GTEDGE that includes all the data gathered for the interval. The file will be named

p*shot number*_*timeid*_*date**runid*_.txt

It will have to be renamed to

soldata.for

to be run using GTEDGE, but it is best to avoid changing the name until necessary, for organizational purposes.

Common errors encountered when running this script deal with a modified soldatatemplate.for file, a missing input data file, file extensions being different than the script expects, or error entering values to be written to the input file. Fortran is known for being a very strict language for syntax, and if the spacing, return placements, or other parts of the soldatatemplate.for file are modified, there is a distinct possibility that the input file will no longer be readable by Fortran. To combat these problems, ensure that extra characters or additional significant figures are not entered in the script file, and that when the updated file is written, it remains in a form readable by a Fortran compiler.

If the soldatatemplate.for file is modified, ensure that the script continues to write the values to the input file correctly. Similarly, if adding additional variables to be updated by the script, ensure that the output file is written correctly, and the variables are accurately emplaced. In all cases, if time is available, it is strongly recommended to double check the newly created input file to ensure that the new entries have been transcribed correctly. Often, a number written in the script in a slightly different format, or with different numbers of decimal places, can obstruct the correct operation of the script, and covertly emplace a bad value that is difficult to detect unless the new input file is directly checked.

The script includes some operations that alter the file names of input files, or use them in various ways. If the script does not find the file being used, it will return an error. Ensure that the file names written by Dr. Groebner's scripts are used directly, and that they are not modified. A special case of this problem regards the time derivatives of the density and electron and ion energy. Usually, running the drvr_ped_timed script returns the time derivatives of the electron density and the ion and electron energies. However, certain versions of the drvr_ped_timed script return time derivatives of the ion and electron density, and the ion temperature. If these time derivatives files are placed in the same directory as the script, which looks for the electron density and electron and ion energy time derivatives, an error will occur since, this set of variables is hard-coded into the script. If this set of variables is unavailable, or a similar type of modification needs to be made, the script code must be altered.

B.4 Balancing the Input File: The jpf2_balancer_2014_*shot number*_*timeid*_*runid*.m Script

Once the new input file has been created, it can be used to run GTEDGE. However, the input file must be tuned to make sure that the plasma generated in GTEDGE matches the experimental one being simulated. This process is automated using the procedures and scripts described in this section. This is done in two parts. First, the GTEDGE code is modified to ease the balancing process, and then the actual balancing process is automated using Matlab scripts. It is important to note that these scripts work quite well, but are not completely functional independently. The functionality of the "alpha" version of these scripts is described in this section, and suggestions for improvement are given at the end of the section.

There are three classes of variables relevant to the balancing process. The aim of the process is to match the properties of the simulation, the beta variables, to the experimentally-measured properties of the plasma being modeled, the goal variables. This is performed indirectly, by adjusting the parabolic properties of parameter profiles, the alpha variables, that are averaged, and impact the simulation. In sum, the alpha variables are modified to match the simulation's beta variables to the experimental goal variables. The alpha variables are named, in order of use, hconf, hrat, alphan, alphat2, and cballoon. The beta variables are named taue, lav, centralne, centralte, and the fifth entry in the plasma temperature row, all of which are found in an output of the GTEDGE code. The target variables are taumhd, nebar_v2, ne0, te0, and ped_temp, all of which are found from the MDSPlus data or from the manual retrieval of data from the profile fits, both reviewd in B.2. The first two target variables are retrieved from MDSPlus, and the latter three are taken from the profile fits.

B.4.1 Modifications to GTEDGE to Allow Easy Balancing and Automation

Several minor modifications were made to the GTEDGE files to ease the process of balancing, and allow it to be automated. The first was to input the alpha variable values through an external text file, read into the program at execution, rather than them being hard-coded into the input file, which would require the re-compilation of the entire program every time they were changed. The files soldatatemplate.for and data16 have the appropriate modifications, but the process is summarized here for convenience.

In order change the location of alpha variable input, first the input file where the alpha variables were initially defined was modified to remove these definitional statements. This input file was data16, and in the GTEDGE input file soldata.for, the variable list to be read in from data16 was named pedbar (line 287-289). The pedbar list was located early in the soldata.for file in the definitional section (lines 44-55), and the names of the alpha variables were also removed from the variable list (hconf, hrat, alphan, alphat2, cballoon). Next, the direct definitional statements in the soldata.for input file were commented out (lines 828, 829, 831-833). Then, a new list of variables called balance was created with the five alpha variables (line 91), and a statement was created to read in these variables from a file called balance.txt (lines 110-112). This allowed the variables to be changed without having to recompile the GTEDGE code, and the GTEDGE code to read in the alpha variable values at execution.

The beta variables (taue, lav, centralne, centralte, fifth entry in the plasma temperature row), which are the properties of the simulation, are written to the local directory by the GTEDGE executable after it is run. They can be found in the soldiv.txt file. The beta variables taue, lav, centralne, centralte, and the fifth entry in the plasma temperature row (the pedestal temperature) are to be matched to the experimental and fitted goal values of taumhd, nebar_v2, ne0, te0, and ped_temp. The last goal variable, <u>ped_temp, is the average of the electron and ion temperatures at the top of their respective pedestals</u> as retrieved from the profile fits.

B.4.2 Balancing the Input File Manually

If the user has never balanced a GTEDGE input file before, it is highly recommended that they do so manually before attempting to use the automated process, as the scripts may require some hand-tuning that is difficult to complete without foreknowledge of the fitting process. Balancing is made significantly easier by the implementation of the balance.txt file. The procedure will be given in the following paragraphs. The scripts basically replicate this procedure.

First, it is important to realize that there are varying dependencies among all of the alpha and beta variables. The dependencies can be visualized using brackets, where major dependencies are in the same brackets, and the dependencies decrease the further apart from each other two variables are.

{{(hconf and taue)[(hrat and lav)(alphan and centralne)]}[(alphat2 and centralte)(cballoon and ped_temp)]}

In words, hconf directly affects taue, hrat directly affects lav, alphan directly affects centralne, alphat2 directly affects centralte, and cballoon directly affects ped_temp. In addition, hconf, hrat, and lav mutually affect taue, lav, and centralne, while alphat2 and cballoon mutually affect centralte and ped_temp. The relationships between the alpha and beta variables require an iterative approach to balancing. The most efficient process in use is: a) matching taue to taumhd by altering hconf; b) matching lav to nebar_v2 by altering hrat, and c) matching centralne to ne0 by altering alphan, then repeating b and c until lav matches nebar_v2 and centralne matches ne0 simultaneously. This is the bc loop. Once the bc loop is complete, again match taue to taumhd by altering hconf (the a loop), then repeat the bc loop. Continue repeating this process (the a(bc) loop) until all three beta variables match their goal variables simultaneously. Next, d) match centralte to te0 by altering alphat2, then e) match ped_temp to the experimental ped_temp by altering cballoon. Repeat the de loop until both beta variables match their goal variables simultaneously. Then double back to the a(bc) loop, repeating this process until all five beta variables match their respective goal variables simultaneously. Proceed in this manner, and during the balancing process, record changes to the alpha variables and the resulting beta variable changes. The process can be completed more quickly if the user makes changes to the alpha variables that are informed by the previous alpha-beta variable relationship.

Take heart! With the completion of each outer iteration (((ab)c)(de)), the changes made to the alpha variables should effect less indirect change in the other four beta variables, and slowly, all beta variables will converge to their respective goal variables. All users should complete this process manually at least once to understand the balancing process, and understand the actions the script is directly replicating.

To actually conduct this process, change the alpha variable value in the balance.txt file, save it, and run the GTEDGE executable. Scan the soldiv.txt file for the beta variable being balanced, check the difference between it and the goal variable, note the directionality of the relationship between the alpha and beta variables, and modify the alpha variable accordingly. Run GTEDGE again, and repeat. Once all five beta variables match their respective goal variables, the input file is balanced, and the files output along with soldiv.txt are ready to be analyzed. Usually, if the goal variables and beta variables are within one increment of each other in the third integer from the left in a figure, they are deemed converged (e.g. 3.20=3.21, 1432=1442, or 326=327). This convergence criterion is acceptable given the uncertainties at play in the modeling, and the limited precision of the values from GTEDGE, but a more sophisticated criterion could be developed and implemented.

B.4.3 Balancing the Input File: Operating the jpf2_balancer_2014_*shot_number*_*timeid*_*runid*.m Script.

To complete this same process in a mostly-automated way, the user must ensure that the Matlab script, the balance.txt file, the dataXX files (XX=10-20), and the Matlab subscripts (jpf2_balancer_2014_: GTEDGE_runner.m, incrementer.m, and convergence_checker.m) are in the same directory as the operational GTEDGE executable as placed by the compiler. In the documentation of the code, the alpha variables are called bvariables (balancing variables), the beta variable are called mvariables (moving variables), and the goal variables have the same name (gvariables for goalvariables).

Entering initial alpha variable values in balance.txt that are known to be close to the appropriate values (such as those from another successfully balanced interval from the same shot), is very helpful to the script, and usually allows it to complete its tasks in a shorter time. In any case, the initial alpha values should always be set to reasonable values to ensure efficient progression towards a reasonable solution. The script usually takes between 30 and 60 minutes to run from start to finish, which is artificially extended due to the wait time implemented to allow GTEDGE to run and export results.

B.4.3.1 The Subroutines of the Balancing Script:

jpf2_balancer_2014_GTEDGE_runner.m

The first subroutine is the jpf2_balancer_2014_GTEDGE_runner.m script. This script writes values to balance.txt, runs GTEDGE, and reads the values exported by GTEDGE. It takes in “index” and “newvalue”, which indicate the alpha variable being updated and its new value, respectively. The “writer” section of the subscript (lines 8-28), reads the balance.txt file into a string, finds the definitional statement for the variable to be updated, replaces the definitional statement with the new definitional statement including the updated value, and overwrites balance.txt with a new file including the updated string.

Occasionally, errors will occur in this section of the file. They can often be detected by the GTEDGE code giving a domain error when it tries to run. This is usually due to the search and replace function in the writer section of the script malfunctioning, and causing the script to be unreadable for the next iteration. This will either feed GTEDGE a string for an alpha variable, not input any variable, or not change the alpha variable value at all. Although rare, this can often be repaired by ensuring the balance.txt file is reset to the appropriate format, and making sure that reasonable values are being written to the balance.txt file.

The next section of the subscript, the "runner" section between lines 34 and 38, runs GTEDGE and waits two seconds for it to complete its running and writing activities. The output of the GTEDGE code is written to a text file output.txt. However, if GTEDGE throws an error, it will usually appear in the command window.

The last section of this subscript, the "retriever" section found after line 44, reads the balance.txt and soldiv.txt files, and retrieves the alpha and beta variable values from them. The output of the subscript, bvariables and mvariables, are vectors with the new values of the alpha and beta variables, in the orders: hconf, hrat, alphan, alphat2,cballoon; and taue, lav, centralne, centralte, and ped_temp, respectively.

This section first reads the balance.txt file, and writes the complete current set of alpha variables to a vector. A check could be implemented here to see if the variable that is supposed to be written in the "writer" section has indeed been updated succesfully. The retriever section then reads the soldiv.txt file for the new beta variable values. Again as for all search functions in these scripts, general expressions are used (i.e. line 65), and unique to this sections, tokens are utilized. These tokens would be useful if implemented in the "writer" section and the soldata_constructor script, as they would reduce the possibility for error and unwanted alteration of the text files. All of these variables are written to the bvariables and mvariables vectors, and returned to the main script.

B.4.3.2 The Subroutines of the Balancing Script: jpf2_balancer_2014_convergence_checker.m

The third subroutine is called jpf2_balancer_2014_convergence_checker.m. This script applies the convergence criterion (one unit in the third place from the left) to the beta and goal variables, and determines if they are close enough to move to the next variable. If they are converged, the routine returns a 1, if they are not converged, a 0, and if the value of the beta variable has diverged beyond physical reasonableness, this subroutine returns an error. A more sophisticated method of determining convergence could be implemented into this script if desired.

B.4.3.3 The Subroutines of the Balancing Script:

jpf2_balancer_2014_incrementer

The final subroutine takes in the variable identifier, the goal variable value, and the beta variable value. It compares the value of the goal and beta variables, and makes an initial determination of the size and direction of the adjustment that should be made to the alpha variable to get the first two to match. It takes in the goal variable (gvariable), the beta variable (mvariable), and the variable identifier, and returns the estimated increment of adjustment of the alpha variable, and a currently-unused legacy variable minimum_indicator, which indicates if the beta and goal variables are close to convergence.

The code between lines 3 and 56 examines the goal variable, and determines via the "one thousandth" convergence criterion what qualifies as close to converged. Then, in lines 57-60, the difference between the goal variable and beta variable are calculated, as is the ratio of the difference between the goal and beta variable values and the goal variable value. Based on these quantities, in lines 60-80, the script determines whether the goal variable value is larger or smaller than the beta variable value, and to what degree. Then, the script suggests a multiplier for the alpha variable consistent with the finding, including very small multipliers if the beta and goal variables are close to convergence. This multiplier is then returned to the main script.

This process, and how it is implemented, offers the greatest opportunity for improvement of the code. An active determinination the slope of each of the alpha and beta variable relationship at their current values would enable more precise adjustments for each variable set. Additionally, improving the process of setting the factor of adjustment for the alpha variable could make the process run faster.

For those variables with proportional relationships and positive alpha values, such as hconf, hrat, and cballoon, the process is already fast. These and the other variables still are complicated by possible alpha values less than one, which complicates numerical operations, and may have to be treated separately by the script if the same alpha variable adjustment strategy currently in the code is kept. For the other two variables, however, there are a number of obstacles to improving the script efficiency.

For the alphat2 variable, the alpha-beta variable relationship has an extremely high slope, and it extends across the x-axis into negative territory. This means that not only will the script have to deal with fractions for alpha, but also negative values and fractions, an extremely high sensitivity to changes in the alpha variable, and a parallel alpha variable adjustment strategy that will drive the value across zero (which cannot be done by increase or decrease by a percentage).

The alphan-centralne relationship is highly nonlinear, with various local minima and maxima, areas where centralne is undefined, sections where the relationship crosses the x-axis, and limited sections of the relationship where the desired value of centralne may be attainable. In order to successfully program a script to balance this variable efficiently, this program limits the range of alphan to a section of the relationship where the correct value of centralne is present, which is $4 < \text{alphan} < 7$ in the example scripts. This can be done by examining the diagnostic vector for alphan and centralne that is generated by the main script, and further described in the next section.

*B.4.3.4 The main script: jpf2_balancer_2014_*shot_number*_*timeid*_*runid*.m*

The function of the main script is to manage and record the overall balancing process, and determine when convergence is complete. To setup the script, the user needs to enter the goal variable values into lines 35-42, and be sure the script, subscripts, balance.txt, the executable, and all the other files normally required to run GTEDGE itself are all in the same folder. Also, the name of the script should be changed to reflect the interval it is used to analyze.

Several important record-keeping functions allow for easy troubleshooting and analysis of the balancing process. The balancing_log is the central repository for balancing information during the balancing process. All information regarding the values of alpha, beta, and goal variables, as well as convergence, is written to it, and all the same information is pulled from it. This provides an exhaustive record of each change in alpha variables, the direct effect on the linked beta variable, and the indirect effect on the other beta variables. The balancing_log.xls file contains the balancing_log information, and is continuously updated during the running of the main script. Once all convergence is complete, the script will write the final balancing_log to balancing_log_complete.xls, which provides a full record of the complete balancing process.

The diagnostic vectors change the value of an alpha variable across a wide, predetermined range, and track the response of the beta variables. This is invaluable in balancing a stubborn variable, determining why an error is occurring, or understanding whey the code is not working correctly. Generating this plot takes a significant amount of time, and should not be done more than once per interval. Once the diagnostic vector is complete, it is written to diagnostic_vector.xls, which has the xy pairs illustrating the alpha-beta relationship in a different worksheet for each set of variables. Whether or not these plots are generated is controlled between lines 146 and 150, where 0 indicates that the plot is to be generated. While these plots are being generated, the program will appear to be stuck, but that is not the case.

In lines 14-34 the balancing_log, and its recording to balancing_log.xls is setup. In lines 35-43, the values of the goal variables are input for the specific interval being analyzed. Lines 48-102 are part of a devalued section which sought to determine the relationship between the alpha and beta variables. This has been superseded by the diagnostic vectors, but the code is left in to allow the reactivation of this functionality for future use. This section sequentially increases each alpha variable by 10% to generate an initial data set on the directionality of the direct and indirect alpha-beta relationships.

In lines 130 to 140, the variables tracking number of iterations and convergence are set. In lines 145-150, the switches for the diagnostic plot generation are found, where diag_1 through diag_5 correspond with hconf-taue, hrat-lav, alphan-centralne, alphat2-centralte, cballoon-ped_temp, respectively.

The next section, after line 150, begins the iterative balancing loops that are the core of the program. Each loop is begun by a line of numbers indicating the order the loop occurs, and the code in each checks for convergence. If all the variables supervised by the loop are converged, it proceeds to the next loop. If not, the balancing processes in the loop are begun.

Loop 1 is for outer convergence, which tests to see if all of the variables are converged. Loop 2 is for density convergence, which checks to see if taue, lav, and centralne are converged (inside outer convergence). Loop 3 is the taue convergence loop, which ends and returns to the density convergence loop level. Loop 4 is for e20 convergence, which checks to see if the two variables expressed in Xe20, lav and centralne, are converged (inside density convergence loop). Loop 5 is for lav convergence, and ends, returning to the e20 loop level. Loop 6 is for centralne (ne0) convergence, and also ends, returning to the e20 loop level. Here loop 4, the e20 convergence loop, ends, returning to the density loop level. Next loop 2, the density loop, ends, returning to the outer convergence loop level. Loop 7 is for temperature convergence, and balances the two temperature variables, centralte and ped_temp (inside the outer convergence loop). Loop 8 is for centralte convergence, and it ends, returning to the temperature convergence loop. Loop 9 is for ped_temp convergence, and it ends, returning to the temperature convergence loop, Then, the temperature convergence loop, loop 7, ends, returning to the outer convergence loop. Then the outer convergence loop ends. It is useful to draw out the relationships given here in words, or use this html tag-based schematic to understand the nesting organization of the balancing loops.

<1> <2> 3 <4> 5 6 </4> </2> <7> 8 9 </7> </1>

The functionality for the convergence loops (1, 2, 4, and 7), and the loops that actually balance the variables (3, 5, 6, 8, 9) is different. The convergence loops begin by calling jpf2_balancer_2014_convergence_checker.m to determine if each of the beta and goal variable pairs supervised by that loop are balanced in the current generation. If they are, the loop ends, and the next one begins. If they are not, the next inner loop begins, which eventually will be a balancing loop. After the inner and/or balancing loops run, and the subset of beta and goal variable pairs supervised by the loop are balanced, the convergence loop will check to ensure the convergence is true, and set the convergence indicator to 1. This allows an exit from the “while *indicator*==0” loops that force balancing. Otherwise, the loop will run again, rebalancing the variables and running inner loops until all the variable pairs supervised by the loop are balanced simultaneously.

The balancing loops add additional functionality, but have a similar form to that of the convergence loops. The balancing loops also initially check convergence between the beta and goal variables, and if it is present, skip the working section of the loop. If the beta and goal variables are not converged, then the working section of the loop is engaged.

This working section of each balancing loop will first (line numbers given for the lav loop, loop 5) read the existing goal and beta variable values from the balancing_log (354-355). then call the jpf2_balancer_2014_incrementer subroutine with these values to determine the difference between the goal and beta variable, and the target increment by which the beta variable should be increased (356-357). Next, the row of the balancing log is increased in preparation for generation of new alpha and beta variables (line 358), and if the present iteration is the first for this loop in the current generation of iterations, the alpha variable is simply increased by 10% (lines 359-361). If the current iteration is not the first for the loop, then the alpha variable is modified and the new value is written to the balancing log (lines 363-369; specifics for this adjustment for each of the alpha-beta variable pairs are discussed in the next paragraph). Next, the code for generating the diagnostic vector, and writing it to the diagnostic_vector.xls file is present, and is activated if the value for diag_X is set to 0 (lines 371-381). This location is also where the range of the alpha variable can be modified for the diagnostic vector (line 373).

Once these preparatory steps are complete, the jpf2_balancer_2014_GTEDGE_runner.m subroutine is called, and it writes the new alpha variable to balance.txt, runs GTEDGE, waits for the GTEDGE output, and reads the new alpha and beta variables, returning them back to the main script (lins 384-385). Lines 386-389 are a sanity check on the alpha variable value, to indicate if it is outside of an extreme range (usually 100 < hconf, cballoon < 0.1; 10 < hrat, alphan < 0.01; -1.2 < alphat2 < -0.01). Next, the new inputs and results from GTEDGE, the new alpha and beta variables, are written to the balancing_log record (lines 391-403), and a check is performed to ensure that the alpha variables passed to the jpf2_balancer_2014_GTEDGE_runner.m subroutine are the same as those returned. Then, the beta variable for the loop is checked to see if they are converged, and if the loop can be ended. In either case, the data from the current row of the balancing_log is written to balancing_log.xls. If the beta and goal variables are converged, the loop ends. If not, the balancing loop is repeated.

The manner in which the alpha variable is modified is the most difficult part of the process. In determining how it should be modified, the diagnostic vectors are extremely useful because they clearly show the relationship between alpha variable values and beta variable values. Usually, the hconf-taue, hrat-lav, and cballoon-ped_temp relationships are proportional, have a relatively gentle slope, and exist in the upper right quadrant of the diagnostic plot. For these conditions, the previous alpha value can simply be multiplied by the value returned from the jpf2_balancer_2014_incrementer.m subroutine to generate the new alpha value.

The alphan-centralne relationship is nonlinear, and it is highly recommended to consult the diagnostic vector in order to determine how the previous alpha value should be modified. A sample diagnostic vector is given in Figure 79, and the characteristics of this relationship are similar to those for which code is written in lines 467-481. In Figure 79, there are several features that stand out. First, there appears to be an asymptote just before x = -1 (the constant y-values from -1.5 < x < -1 represent errors returning from GTEDGE), there is a shallow hill structure from -1 < x < 3.75, and then the plot rises towards the positive end of the x range. For this example, the goal variable was 0.35, which was only met by the beta variable in the section of the relationship where the alpha variable had a value x > 3.5. However, there exist a number of local minima and maxima elsewhere in the relationship that would cause havoc with an automated procedure designed to find an x-value that causes a y-value that matches 0.35. Therefore, using the insight provided by this graph, the code in lines 467-481 was modified to restrict the movement of the alpha variable to values between 4 and 7. At values greater than around 7, GTEDGE returned an error code, and the value of 0.35 was not attained by the beta variable with alpha variable values less than 4. This is an extreme case, and the appropriate value may have been found if initial values were set properly, but it instructive to note the complications that may be present in a balancing relationship between two variables.

For the centralte-alphat2 relationship, the modification of the alpha variable was less complicated, but still challenging. As shown in Figure 80, the relationship is slightly nonlinear, veers into negative territory, reasonable temperatures are achieved with fractional values of alpha, and the slope of the relationship is very large. If the algorithm sampled the beta value at negative values of alpha, in order to properly modify the alpha value in the next iteration, the increments returning from jpf2_balancer_2014_incrementer had to be reversed (i.e. make the alpha value LESS negative and smaller to increase the beta value, rather than more negative and larger, which is what the unmodified incrementer output would do). In addition, the increment suggested by the subroutine had to be decreased significantly, and the adjustment to the alpha variable made much finer, as the slope of the centralte-alphat2 relationship was too large for the adjustments suggested by the subroutine. Without the adjustments, the script would oscillate endlessly between values of alphat2 that were slightly too small or slightly too large, unable to adjust alphat2 with the precision necessary to balance the variable

These types of adjustments to the alpha variable modification routine may be necessary due to the wide variation of plasma properties that can be achieved in a tokamak. In general, the following procedure, and code with similar effects of the modifications described above can be useful to 1) improve the speed of the program, and 2) correct any errors in the balancing procedure.

If an error occurs in the script, or it becomes stuck for more than 30 minutes with no activity, the user should push CTRL+C to stop the program if necessary, then examine the error message to determine what went wrong. If the answer is not readily apparent, the user should examine the balancing_log.xls file to determine where the balancing process went awry, look at the balance.txt file to ensure that the values are within acceptable ranges for the alpha variables, and in this context, examine the diagnostic vectors to determine if a problem between the selected alpha variable value and the beta variable is the culprit. After the problem has been identified and fixed, the user should delete the balancing_log.xls file, reset any unreasonable values in balance.txt to reasonable values (a good set is hconf=5.0, hrat=0.5, alphan=0.5, alphat2=-0.5, cballoon=3.0), and run the main script again.

After all of the balancing has been completed, and all convergences are achieved simultaneously, the script writes the balancing log to balancing_log_complete.xls, and the program ends.

B.5 Recommended Future Work

There are a number of improvements that could be made to this workflow, and many of them are noted in the various sections. However, two improvements would seem to be most logical. The first is to enhance the balancing algorithm for non-linear balancing relationships by parsing the diagnostic_vector.xls data for each variable pair, and determining what ranges of x values are appropriate for use in each specific case. For example, the script could be made to read the data that makes up Figure 79, note that the alpha range where the desired beta value is located is around 5, determine the local bounds of the structure that contains this beta value ($4 < x < 7$), and limit the balancing algorithm to this predetermined range of alpha variable values.

The other clear need is an script to retrieve, organize, and plot the results from the GTEDGE simulation, found in the pedestal.txt file. The retrieval algorithms found in the jpf2_balancer_2014_GTEDGE_runner.m subscript (the string search functions using general expressions), and the Excel plotting routines found in the jpf2_fitstoexcelplot_2014.m script, would simply have to be expanded to successfully achieve this task.

Finally, porting this system over to a different language would be beneficial as it would remove the cost associated with Matlab coding. Also, the ability to compile the scripts into independent executables not reliant on the Matlab program to run could improve efficiency.

B.6 Summary of APPEndix B: Modeling Workflow for Reference

1. profile fit data: Update the drvr_pedxax and drvr_ped_timed scripts with the correct shot numbers, timeids, runids, and elm_dts.

   Run the scripts in IDL.

   Retrieve the products of the scripts from the remote directory.

   (recommended) Run the jpf2_fitstoexcelplot_2014 Matlab script to visualize the data.
2. MDSPlus data: Update the jpf2_getdata_2014b script with the shot number and quasi-steady-state period starting and ending times.

   Run the script in IDL.

   Check to ensure that the products contain good data.
3. processing the MDSPlus data: Retrieve the time_windows files from the remote directory for each run, and place them in the same directory as the jpf2_mdsplus_data_processor_2014a script.

   Update this script with the correct shot number, timeid, and runid.

   Run the script with Matlab.

   Check the products of the script for data accuracy, reasonableness, and quantity.

   (recommended) Manually correct problematic data sets, or use the visual trends with time to make reasonable estimates of the average for each interval.

4. Place the products of the drvr_pedxax and drvr_ped_timed scripts, and the soldatatemplate.for file in the same directory as the jpf2_soldata_constructor_2014_*shot_number*_*runid* script.

   Update the shot number, timeid, runid, and the MDPlus average values in the script.

   Run the Matlab script.

   (recommended) Double check the values in the updated input file for accuracy, and ensure it is in the correct Fortran format.

5. Rename the input file soldata.for, and place it in the appropriate directory of the compiler..

   Compile the GTEDGE executable.

6. Place the GTEDGE required files, the balance.txt file, and all four of the jpf2_balancer_2014 Matlab script files in the same directory as the executable.

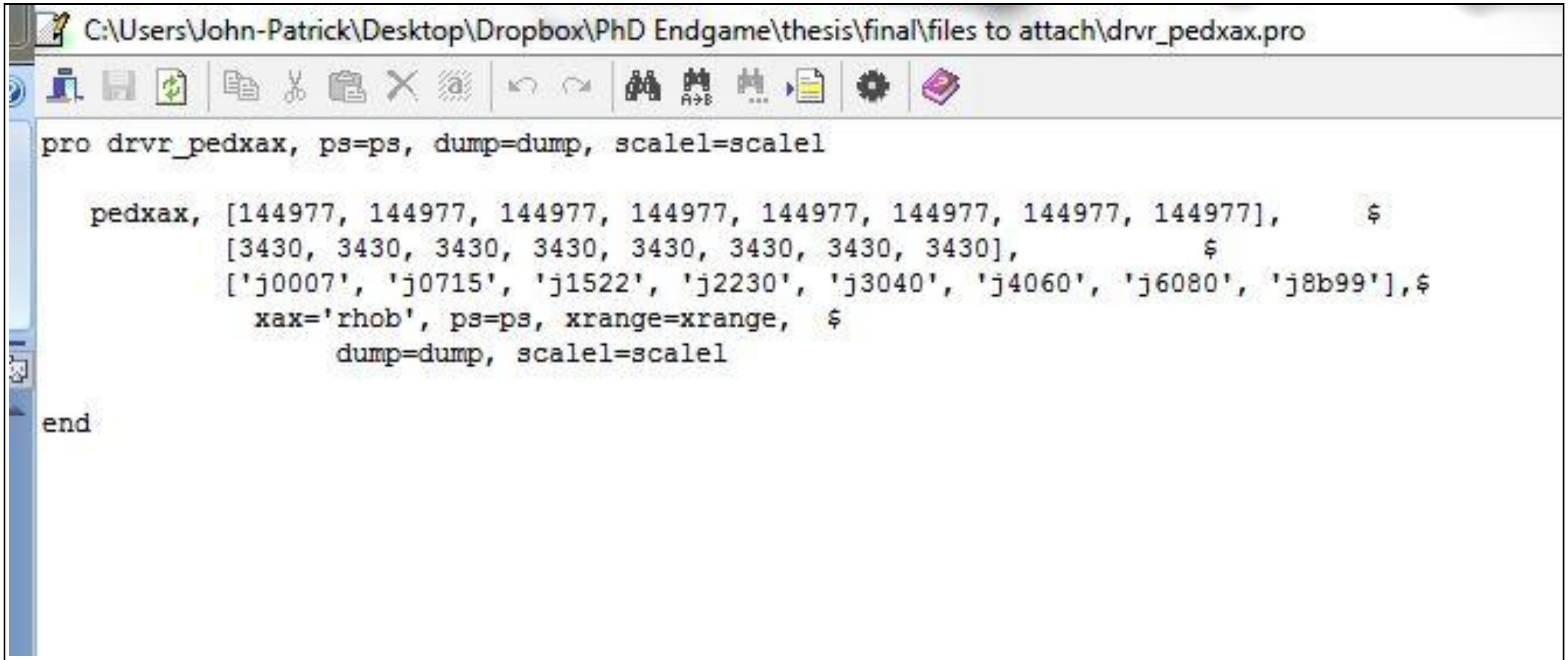
   Update the file name and goal variable values in the jpf2_balancer_2014_*shot_number*_*timeid*_*runid*.m file to the current values.

   Ensure that the alpha variable values in balance.txt are reasonable.

   Run the main balancing script in Matlab.

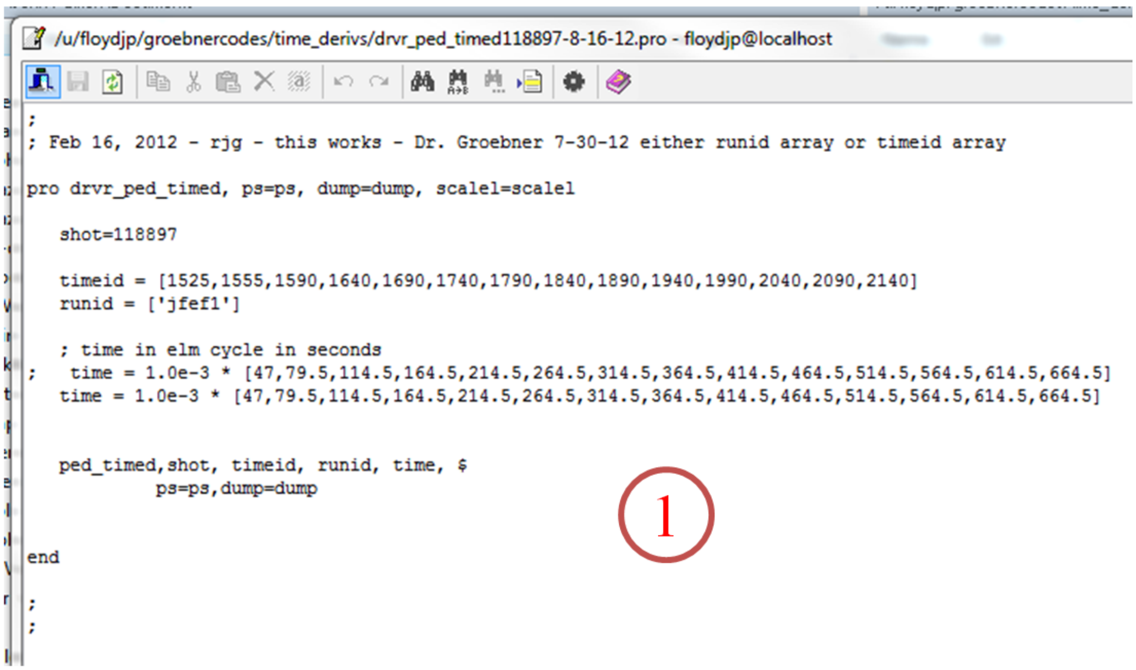
   (optional) Prepare the balancing scripts and executables for multiple intervals at once, and write a script to run them sequentially.

7. Retrieve the pedestal.txt and other output text files from GTEDGE, and use the data therein.

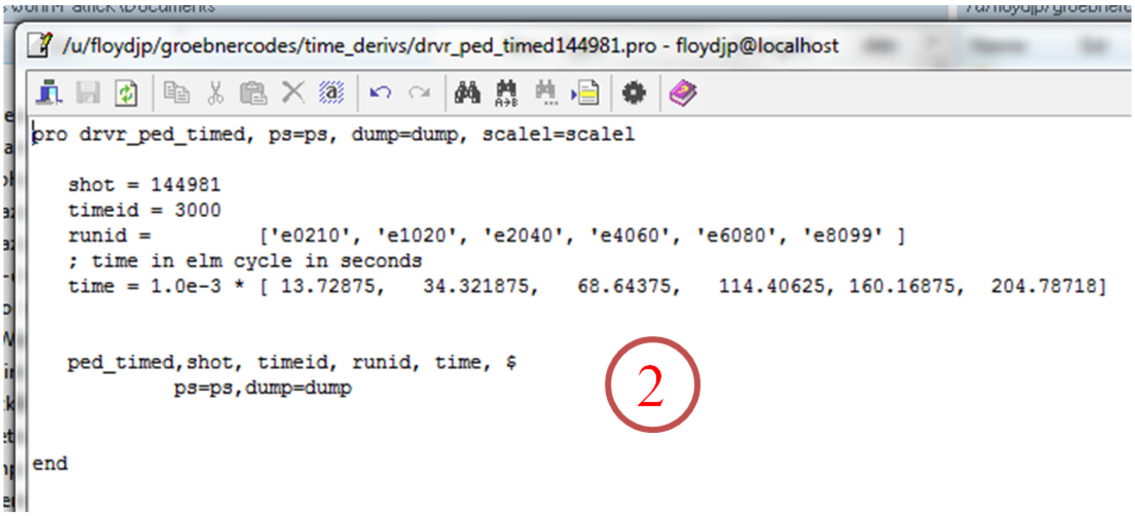


```
pro drvr_pedxax, ps=ps, dump=dump, scale1=scale1

   pedxax, [144977, 144977, 144977, 144977, 144977, 144977, 144977, 144977],     $
           [3430, 3430, 3430, 3430, 3430, 3430, 3430, 3430],           $
           ['j0007', 'j0715', 'j1522', 'j2230', 'j3040', 'j4060', 'j6080', 'j8b99'],$
             xax='rhob', ps=ps, xrange=xrange,  $
                  dump=dump, scale1=scale1

end
```

Figure 77: The drvr_pedxax.pro script.



```
;
; Feb 16, 2012 - rjg - this works - Dr. Groebner 7-30-12 either runid array or timeid array

pro drvr_ped_timed, ps=ps, dump=dump, scalel=scalel

   shot=118897

   timeid = [1525,1555,1590,1640,1690,1740,1790,1840,1890,1940,1990,2040,2090,2140]
   runid = ['jfef1']

   ; time in elm cycle in seconds
;   time = 1.0e-3 * [47,79.5,114.5,164.5,214.5,264.5,314.5,364.5,414.5,464.5,514.5,564.5,614.5,664.5]
   time = 1.0e-3 * [47,79.5,114.5,164.5,214.5,264.5,314.5,364.5,414.5,464.5,514.5,564.5,614.5,664.5]


   ped_timed,shot, timeid, runid, time, $
            ps=ps,dump=dump


end

;
;
```



```
pro drvr_ped_timed, ps=ps, dump=dump, scalel=scalel

   shot = 144981
   timeid = 3000
   runid =          ['e0210', 'e1020', 'e2040', 'e4060', 'e6080', 'e8099' ]
   ; time in elm cycle in seconds
   time = 1.0e-3 * [ 13.72875,    34.321875,    68.64375,    114.40625, 160.16875,   204.78718]


   ped_timed,shot, timeid, runid, time, $
            ps=ps,dump=dump


end
```

Figure 78: Two versions of the drvr_ped_timed.pro script. The upper version is examining a series of independent time periods. The lower version is examining a series of intervals of a composite inter-ELM period.

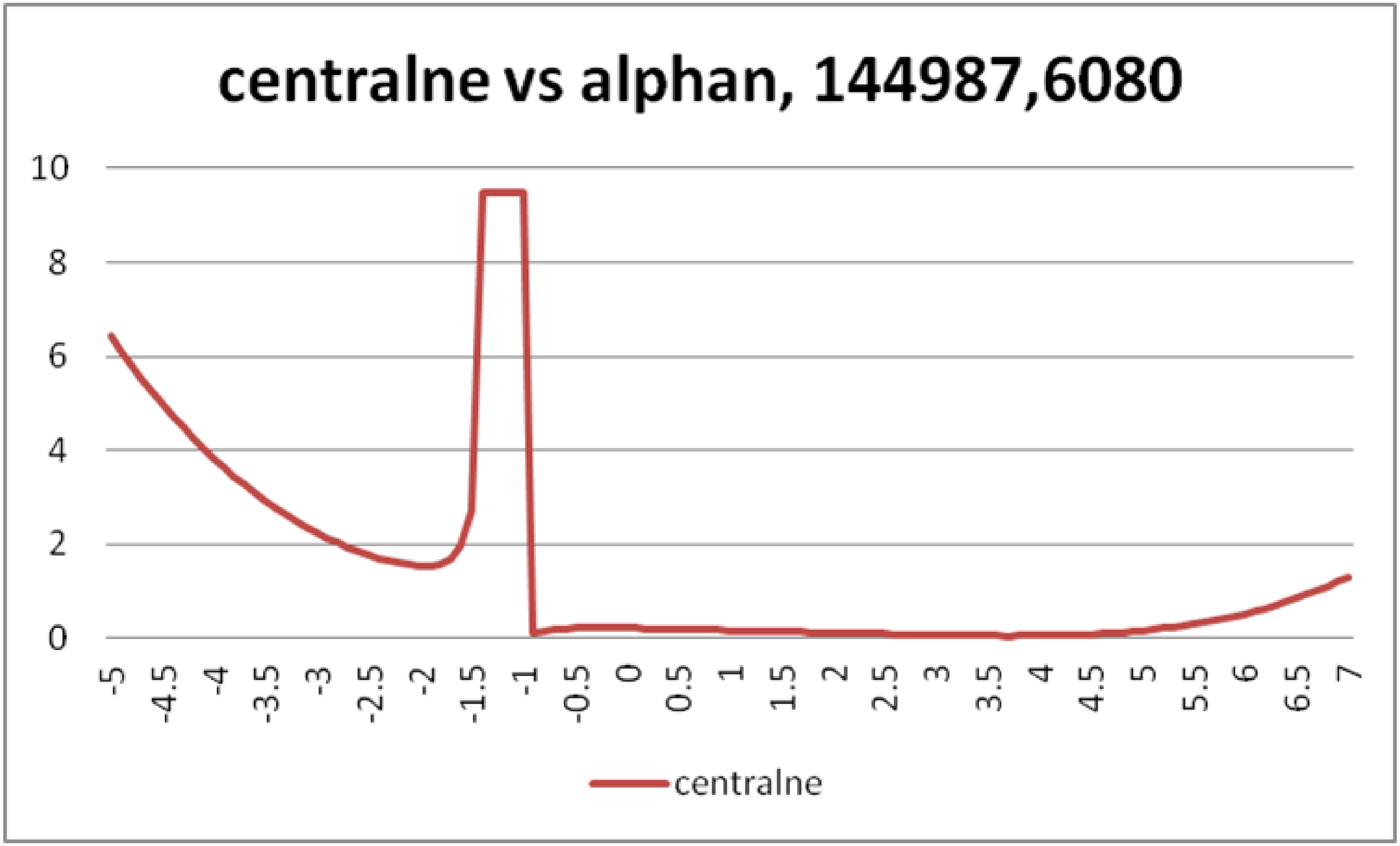


Figure 79: The relationship between the alpha variable alphan and the beta variable centralne. Note the nonlinearity in the profile, and the abundance of local maxima and minima. The goal variable value was 0.35, only reached by the beta profile around alpha value x=5.5.

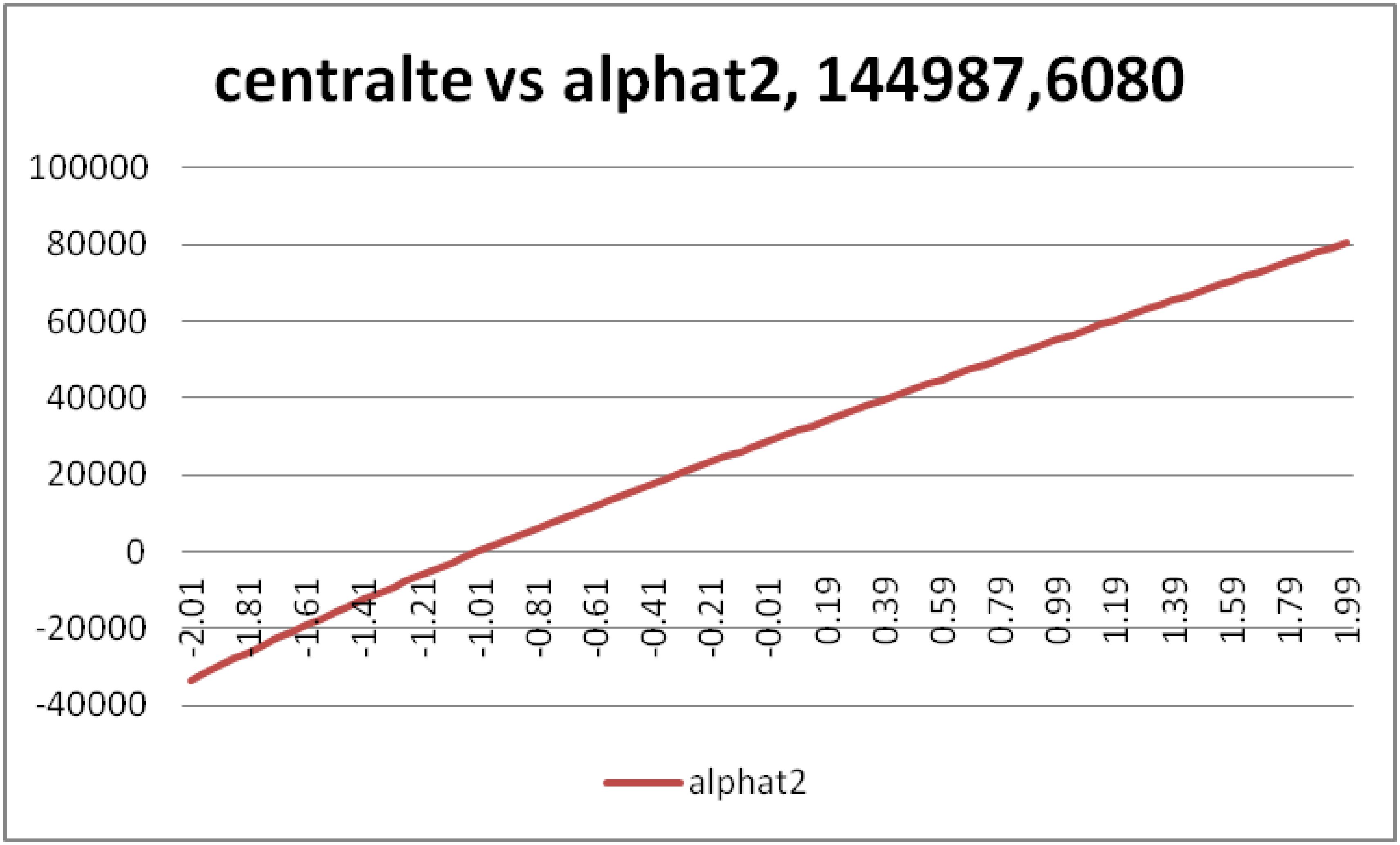


Figure 80: The relationship between alphat2 and centralte. Note the large slope, slight nonlinearity, and proximity of reasonable values of the beta variable to alpha values near zero.

REFERENCES

1. Wilks, T.M., *Experimental Flux Surfaces for GTEDGE*, 2014.
2. Jarvis, O.N. *4.7.4 Nuclear Fusion*. 2004 9/18/2014; Web Edition:[Available from: http://www.kayelaby.npl.co.uk/atomic_and_nuclear_physics/4_7/4_7_4.html.
3. Miller, R.L., et al., *Noncircular, finite aspect ratio, local equilibrium model.* Physics of Plasmas (1994-present), 1998. **5**(4): p. 973-978.
4. Isler, R.C., *An overview of charge-exchange spectroscopy as a plasma diagnostic.* Plasma Physics and Controlled Fusion, 1994. **36**(2): p. 171.
5. Stacey, W.M., et al., *Interpretation of changes in diffusive and non-diffusive transport in the edge plasma during pedestal buildup following a low-high transition in DIII-D.* Physics of Plasmas (1994-present), 2013. **20**(1): p. -.
6. Snyder, P.B., et al., *Edge localized modes and the pedestal: A model based on coupled peeling–ballooning modes.* Physics of Plasmas (1994-present), 2002. **9**(5): p. 2037-2043.
7. Meshik, A.P., C.M. Hohenberg, and O.V. Pravdivtseva, *Record of Cycling Operation of the Natural Nuclear Reactor in the Oklo/Okelobondo Area in Gabon.* Physical Review Letters, 2004. **93**(18): p. 182302.
8. W. M. Stacey, C.L.S., J.-P. Floyd, T. M. Wilks, A. P. Moore, A. T. Bopp, M. D. Hill, S. Tandon, and A. S. Erickson, *Resolution of Fission and Fusion Technology Integration Issues: An Upgraded Design Concept for the Subcritical Advanced Burner Reactor.* Nuclear Technology, 2014. **187**(1): p. 15-43.
9. *Facts and Figures*. 9/18/2014; Available from: https://www.iter.org/factsfigures.

10. Wagner, F., et al., *Regime of Improved Confinement and High Beta in Neutral-Beam-Heated Divertor Discharges of the ASDEX Tokamak.* Physical Review Letters, 1982. **49**(19): p. 1408-1412.

11. Wagner, F., et al., *Development of an Edge Transport Barrier at the H-Mode Transition of ASDEX.* Physical Review Letters, 1984. **53**(15): p. 1453-1456.

12. Groebner, R.J., K.H. Burrell, and R.P. Seraydarian, *Role of edge electric field and poloidal rotation in the <i>L</i>-<i>H</i> transition.* Physical Review Letters, 1990. **64**(25): p. 3015-3018.

13. Greenwald, M., et al., *H mode confinement in Alcator C-Mod.* Nuclear Fusion, 1997. **37**(6): p. 793.

14. Kinsey, J.E., et al., *Burning plasma projections using drift-wave transport models and scalings for the H-mode pedestal.* Nuclear Fusion, 2003. **43**(12): p. 1845.

15. Kotschenreuther, M., et al. in *16th International Conference on Fusion Energy*. 1996. Montreal, Canada: IAEA.

16. Osborne, T.H., et al. in *24th European Physical Society Conference on Controlled Fusion and Plasma Physics*. 1997. Berchtesgaden, Germany: Petit-Lancy: European Physical Society.

17. Suttrop, W., et al., *Identification of plasma-edge-related operational regime boundaries and the effect of edge instability on confinement in ASDEX Upgrade.* Plasma Physics and Controlled Fusion, 1997. **39**(12): p. 2051.

18. Zohm, H., *Edge localized modes (ELMs).* Plasma Physics and Controlled Fusion, 1996. **38**(2): p. 105.

19. Evans, T.E., et al., *RMP ELM suppression in DIII-D plasmas with ITER similar shapes and collisionalities.* Nuclear Fusion, 2008. **48**(2): p. 024002.

20. Baylor, L.R., et al., *Comparison of deuterium pellet injection from different locations on the DIII-D tokamak.* Nuclear Fusion, 2007. **47**(11): p. 1598.

21. Burrell, K.H., et al., *Quiescent H-mode plasmas in the DIII-D tokamak.* Plasma Physics and Controlled Fusion, 2002. **44**(5A): p. A253.

22. Whyte, D.G., et al., *I-mode: an H-mode energy confinement regime with L-mode particle transport in Alcator C-Mod.* Nuclear Fusion, 2010. **50**(10): p. 105005.

23. Grierson, B.A., et al., *Deuterium velocity and temperature measurements on the DIII-D tokamaka).* Review of Scientific Instruments, 2010. **81**(10): p. -.

24. Groebner, R.J., *Information Regarding the Tangential CER Chords*, J.-P. Floyd, Editor 2013.

25. Kim, J., et al., *Rotation characteristics of main ions and impurity ions in <i>H</i>-mode tokamak plasma.* Physical Review Letters, 1994. **72**(14): p. 2199-2202.

26. Stacey, W.M. and R.J. Groebner, *Evolution of the H-mode edge pedestal between ELMs.* Nuclear Fusion, 2011. **51**(6): p. 063024.

27. Osborne, T.H., et al., *Edge stability of stationary ELM-suppressed regimes on DIII-D.* Journal of Physics: Conference Series, 2008. **123**(1): p. 012014.

28. Gohil, P., et al., *The charge exchange recombination diagnostic system on the DIII-D tokamak.* Proceedings The 14th IEEE/NPSS Symposium Fusion Engineering, 1991: p. 1199.

29. Carlstrom, T.N., et al., *Design and operation of the multipulse Thomson scattering diagnostic on DIII-D (invited).* Review of Scientific Instruments, 1992. **63**(10): p. 4901.

30. Stacey, W.M., *A coupled plasma-neutrals model for divertor simulations.* Physics of Plasmas (1994-present), 1998. **5**(4): p. 1015-1026.

31. Stacey, W.M., *A calculation model for density limits in auxiliary heated, gas fueled tokamaks and application to DIII-D model problems.* Physics of Plasmas (1994-present), 2001. **8**(8): p. 3673-3688.

32. Stacey, W.M., *Modelling the neutral density in the edge of the DIII-D plasma.* Nuclear Fusion, 2000. **40**(5): p. 965.

33. Maingi, R., et al., *Comparison of wall/divertor deuterium retention and plasma fueling requirements on the DIII-D, TdeV and ASDEX Upgrade tokamaks.* Journal of Nuclear Materials, 1997. **241–243**(0): p. 672-677.

34. Stacey, W.M., *Ion Particle Transport in the Tokamak Edge Plasma.* Contributions to Plasma Physics, 2008. **48**(1-3): p. 94-98.

35. Stacey, W.M., *The effect of ion orbit loss and X-loss on the interpretation of ion energy and particle transport in the DIII-D edge plasma.* Physics of Plasmas (1994-present), 2011. **18**(10): p. -.

36. Stacey, W.M., et al., *Intrinsic rotation produced by ion orbit loss and X-loss.* Physics of Plasmas (1994-present), 2012. **19**(11): p. -.

37. Stacey, W.M. and B.A. Grierson, *Interpretation of rotation and momentum transport in the DIII-D edge plasma and comparison with neoclassical theory.* Nuclear Fusion, 2014. **54**(7): p. 073021.

38. Stacey, W.M. and R.J. Groebner, *Interpretation of edge pedestal rotation measurements in DIII-D.* Physics of Plasmas (1994-present), 2008. **15**(1): p. -.

39. Stacey, W.M., R.J. Groebner, and T.E. Evans, *Non-diffusive transport in the tokamak edge pedestal.* Nuclear Fusion, 2012. **52**(11): p. 114020.

40. Stacey, W.M., *Structure in the Edge Plasma profiles in Tokamaks.* Contributions to Plasma Physics, 2014. **54**(4-6): p. 524-528.

41. Stacey, W.M., *Effect of ion orbit loss on distribution of particle, energy and momentum sources into the tokamak scrape-off layer.* Nuclear Fusion, 2013. **53**(6): p. 063011.

VITA

JOHN-PATRICK FLOYD II

FLOYD was a resident of Port Saint Joe, Florida, a small, rural, coastal town in the North Florida Panhandle surrounded by natural beauty, for most of his early life. He attended Faith Christian School through 6th grade, and later was the Salutatorian of the Class of 2002 of Port Saint Joe High School. Floyd left home in August 2002 to attend the Georgia Institute of Technology and study Nuclear Engineering, a subject in which he had been interested since boyhood. There he was an active brother of the Kappa Alpha Order, continuing a multigenerational family affiliation, and he returned home over breaks and during the summer semesters to run Tarpon Title, Inc., a company he founded in 2002 with his father and brother. In May 2007, Floyd graduated with a Bachelor of Science degree in Nuclear Engineering and a Minor in International Affairs. Floyd had communicated with Dr. W. M. Stacey for several years regarding a graduate education in fusion energy, and was a student in several of his plasma physics courses. In light of this experience, Floyd chose to study fusion plasma physics, and began graduate studies in the Nuclear Engineering Program in the Woodruff School of Mechanical Engineering at Georgia Tech in August 2007, under the supervision of Dr. Stacey. In 2009, in response to a lifelong interest in international relations, and also encouraged by his undergraduate coursework, Floyd entered the Sam Nunn School of International Affairs at Georgia Tech to seek an additional Master of Science degree in International Affairs. In May of 2011, Floyd earned his Master of Science degree in Nuclear Engineering, publishing original research concerning the adaptation of a model of edge transport in the edge of fusion reactors to be compatible with the powerful, well-established solution methods and codes used to model neutron diffusion. For the 2012-2013 year, Floyd was selected to be a Fellow of the Sam Nunn Security Program, researching technology policy and national security, and in December 2013, Floyd earned his Master of Science degree in International Affairs, specializing in Energy+Technology Policy. Outside the lab, Floyd greatly enjoys football, bay fishing, performance and fine art, traveling, and church.