

PSFC/RR-97-12

DOE/ET-51013-326

**Local Transport Analysis for
the Alcator C-Mod Tokamak**

Jeffrey M. Schachter

Plasma Science and Fusion Center
Massachusetts Institute of Technology
Cambridge, MA 02139

June 1997

This work was supported by the U. S. Department of Energy Contract No. DE-AC02-78ET51013. Reproduction, translation, publication, use and disposal, in whole or in part by or for the United States government is permitted.

Local Transport Analysis for the Alcator C-Mod Tokamak

by

Jeffrey Marc Schachter

Submitted to the Department of Nuclear Engineering
on June 28, 1997, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Nuclear Engineering

Abstract

Two complementary approaches were used to characterize transport on the Alcator C-Mod tokamak. The first was an empirical analysis of the scaling of transport with ρ^* , the ion Larmor radius normalized to the plasma size. The second was a comparison of the transport predictions from the IFS-PPPL model of ion temperature gradient (ITG) driven turbulence to observations on C-Mod.

The ρ^* scaling experiments on C-Mod extend the range of plasma parameters over which the dimensionless scaling approach has been tested in both magnetic field (to 8 T) and density (to $\langle n_e \rangle = 3.8 \times 10^{20}/\text{m}^3$). In L-Mode, scaling of the global confinement time τ_E showed a Bohm-like dependence on ρ^* ($B\tau_E \propto \rho^{*-2}$) as did the scaling of the local one-fluid effective diffusivity ($\chi_{\text{eff}} \propto \chi_B \rho^{*0}$), though mismatched radiated power profiles prevent a definitive statement. H-Mode plasmas showed a gyroBohm scaling both globally ($B\tau_E \propto \rho^{*-3}$) and locally ($\chi_{\text{eff}} \propto \chi_B \rho^{*1}$). Determining the scaling of the individual electron and ion channels was not possible due to the large uncertainty in the electron-ion coupling associated with the high density.

The IFS-PPPL model is a comprehensive first-principles description of ITG turbulence driven transport. Its numeric solution of the linear gyrokinetic equation indicates that the ITG mode is the predominant instability in the plasma core. Non-linear numerical treatments of the gyrofluid behavior of ITG modes provide parametric models for χ_i and χ_e useful for experimental comparison. Transport from the ITG turbulence is described by a marginal stability model; C-Mod plasmas are predicted to be close to the threshold, so they provide a rigorous test bed for the theory.

Comparisons of the model's predictions for T_i and T_e were conducted on C-Mod for both L- and H-Mode plasmas. In L-Mode the T_i predictions were systematically low, with an RMS difference of $\Delta_{\text{RMS}} = 25 - 30\%$ measured over the range $(\frac{a}{4}, \frac{3a}{4})$. Predictions of T_i in H-Mode were more accurate, giving an RMS difference of $\Delta_{\text{RMS}} = 5 - 20\%$ over the same range, well within the estimated uncertainty. Predictions of T_e did not show a systematic difference between L- and H-Mode. For some shots the was accurate ($\Delta_{\text{RMS}} < 10\%$) while for others it was very inaccurate ($\Delta_{\text{RMS}} > 50\%$). In all cases, the model predicted marginally stable T_i profiles, with $1.0 \leq \langle L_{T,\text{crit}}/L_T \rangle \leq 1.2$.

Thesis Supervisor: Martin Greenwald

Title: Principal Scientist, Plasma Fusion Center

*For my family, who sparked in me the love for science that began this journey
and for Kristin, whose love helped me see it through*

Contents

1	Introduction	19
1.1	Fusion as a Power Source	19
1.1.1	Tokamaks	20
1.2	The Importance of Transport	22
1.3	Alcator C-Mod	23
1.4	Thesis Outline	25
2	Physics of Transport	27
2.1	Transport Basics	27
2.2	Theoretical Transport Analysis	29
2.2.1	Classical Transport	30
2.2.2	Neoclassical Transport	32
2.2.3	Anomalous Transport	36
2.3	Empirical Transport Analysis	40
2.3.1	Global Analysis	41
2.3.2	Local Analysis	42
3	TRANSP and Local Transport Analysis	47
3.1	Calculating χ from Energy Balance	47
3.1.1	Error Analysis	51
3.1.2	Exchange Power Q_{ei}	53
3.1.3	One-Fluid Effective Diffusivity χ_{eff}	54
3.2	Particle Balance	55
3.3	Magnetic Diffusion	57
3.3.1	Resistivity model	58
3.4	MHD Equilibrium	63
3.5	ICRF Power Deposition	64
3.5.1	Code Physics	64
3.5.2	Singularity Problem in SPRUCE	65
3.5.3	Total RF Power Absorption	69
3.5.4	Minority Fraction	70
3.6	Ion Temperature	76

3.6.1	Introduction	76
3.6.2	Modelling χ_i	77
3.6.3	T_i from T_e	78
3.6.4	Testing T_i Models	79
4	ρ^* Scaling	89
4.1	Background and Motivation	89
4.1.1	Limitations of Dimensional Scaling	89
4.1.2	Dimensional Analysis and Scale Invariance	91
4.1.3	Dimensionless Scaling	92
4.2	Physics of ρ^* Scaling	93
4.3	ρ^* Scaling Experiments	95
4.3.1	ρ^* Scaling on Alcator C-Mod	97
4.4	Methods	98
4.4.1	Plasma Measurements	98
4.4.2	Dimensionless Parameters	99
4.4.3	Calculation of τ_E	103
4.4.4	Calculation of χ	107
4.4.5	Regression Analysis	108
4.4.6	Field Ratio and Sensitivity	109
4.5	Results	110
4.5.1	L-Mode Global Analysis	110
4.5.2	L-Mode Local Analysis	125
4.5.3	H-Mode	146
4.6	ν^* Scaling	163
4.6.1	L-Mode	163
4.6.2	H-Mode	169
4.7	Discussion	174
4.7.1	Summary of ρ^* Scaling Results	174
4.7.2	ρ^* Scaling on Other Tokamaks	175
5	Testing the IFS-PPPL Model of ITG Turbulence	179
5.1	The IFS-PPPL Model	179
5.1.1	ITG Modes	180
5.1.2	Details of the IFS-PPPL Model	182
5.1.3	Marginality and C-Mod	184
5.2	Methods	187
5.2.1	Transport Simulation Using the IFS-PPPL Model	187
5.2.2	Comparison to Experiment	188
5.3	Results	190
5.3.1	Optimal Input Parameters	190
5.3.2	Examination of Individual Shots	191

5.3.3	Boundary Condition and Edge Temperature	204
5.3.4	T_e Predictions and Sawteeth	204
5.3.5	Z_{eff} Profile Shape	204
5.3.6	Toroidal Rotation	206
5.4	Summary and Discussion	208
6	Conclusions and Future Work	213
6.1	Summary	213
6.2	Suggested Future Experiments	215
A	TRANSP and MDSplus	217
A.1	Motivation	217
A.2	TRANSP Specifics	220
A.3	MDSplus Specifics	224
A.3.1	Inputs	224
A.3.2	Outputs	226
A.3.3	Namelist	226
A.3.4	Other Nodes	227
A.3.5	Trees	227
A.3.6	Run Management	227
A.3.7	Coordinate Systems	230
A.4	Running TRANSP	232
A.4.1	Choosing the TRANSP Run	234
A.4.2	Loading Sources	235
A.4.3	Processing Inputs	238
A.4.4	Namelist	243
A.4.5	Submitting the Run	248
A.5	Examining Outputs	249
A.5.1	Multigraph	249
A.5.2	Other Tools	251
A.5.3	Scalar Database	253
B	TRANSP Reference	255
B.1	DOCLIST.DOC	257
B.2	GP.DOC	260
B.3	MDSPLUS.SETUP.DOC	265
B.4	MDSPLUS.TRANSP.DOC	271
B.5	MULTIGRAPH.DOC	276
B.6	POSTRANSP.DOC	286
B.7	PRETRANSP.DOC	289
B.8	PSLICE.DOC	296
B.9	SOURCES.DOC	298
B.10	TDI.DOC	303

B.11 TDI_OBSOLETE.DOC	308
B.12 TRANSP_COORDINATES.DOC	309
B.13 TRANSP_FILE_AREAS.DOC	314
B.14 TRANSP_SCALARS.DOC	316
B.15 UFILES_AND_IDL.DOC	319
B.16 UPDATING_TRANSP.DOC	322
B.17 USER_DEFAULTS.DOC	327
Bibliography	328

List of Figures

1.1	Schematic of a tokamak	21
1.2	Cross section of the Alcator C-Mod tokamak	24
2.1	Magnetic field geometry in a tokamak	33
2.2	Banana orbits	35
2.3	Eddies formed by perturbations in potential	38
2.4	Transport driven by eddies	39
2.5	Comparison of χ_i from various models of transport to experiment . .	40
2.6	Marginal stability model of transport	45
3.1	Models, inputs, and outputs in TRANSP	48
3.2	Error in χ_e from errors in T_e and T_i	52
3.3	Errors in χ_i from errors in T_e and T_i	52
3.4	Electron convected power for three values of τ_p	56
3.5	Electron radial velocity for three values of τ_p	56
3.6	Comparison of Spitzer and neoclassical resistivity models for shot 960116024.	60
3.7	Comparison of Spitzer and neoclassical resistivity models for shot 960130028.	62
3.8	Flux surface mappings from EFIT and VMEC	63
3.9	ICRF power deposition from old and new SPRUCE	67
3.10	Time history of the RF power deposition within $r/a = 0.5$	68

3.11 Fractional RF power absorption for 2 nd harmonic H minority heating	71
3.12 Fractional RF power absorption for ³ He minority heating	72
3.13 Tail energies for two shots used in f_{\min} scan	74
3.14 ICRF power absorption and relative heating as a function of minority concentration	75
3.15 Calculation of T_i from T_e using two different methods	80
3.16 Time history of shot 950317023	81
3.17 Calculated T_i profile compared to measured profile	82
3.18 χ_i and χ_{eff} from different T_i profiles, run 950317	83
3.19 Volume-integrated conducted power density $P_{\text{cond}}(r) = \int_0^r dV \nabla \cdot \mathbf{q}$ calculated from different T_i profiles, run 950317	84
3.20 Measured and calculated T_i profiles for shot 960229042	86
3.21 Neutron rates calculated for T_i profiles from figure 3.20	86
3.22 χ_i and χ_{eff} profiles for T_i profiles from figure 3.20	87
4.1 Variations in density profile shape with line-integral fixed	105
4.2 Regression of τ_E against engineering parameters for L-Mode	111
4.3 Regression of τ_E against dimensionless parameters for L-Mode	111
4.4 $q_{\psi 95}$ and ν^* vs. β_p for L-Mode shots	113
4.5 Preliminary checks for global scaling, L-Mode group a	115
4.6 $B\tau_E$ versus ρ^* for L-Mode group a	116
4.7 Sensitivity of fit to ν^* and β_p , Group a	117
4.8 Preliminary checks for global scaling, L-Mode group b	119
4.9 $B\tau_E$ versus ρ^* for L-Mode group b	120
4.10 τ_E versus B_t for L-Mode group b	120
4.11 Sensitivity of fit to ν^* and β_p , Group b	121
4.12 Global Analysis for L-Mode group c	123
4.13 Global Analysis for L-Mode group e	124

4.14	Time history of shots in L-Mode group a	127
4.15	Dimensionless parameter profiles for group a shots	128
4.16	Local analysis plots for ρ^* scaling L-Mode group a	131
4.17	Dimensionless parameter profiles for group b shots	133
4.18	Time history of shots in L-Mode group b	134
4.19	Local analysis plots for ρ^* scaling L-Mode group b	135
4.20	Dimensionless parameter profiles for group c shots	137
4.21	Time history of shots in L-Mode group c	138
4.22	Local analysis plots for ρ^* scaling L-Mode group c	139
4.23	Time history of shots in L-Mode group e	141
4.24	Dimensionless parameter profiles for group e shots	142
4.25	Local analysis plots for ρ^* scaling L-Mode group e	144
4.26	Values of α obtained from global and local scaling in L-Mode	145
4.27	Regression of τ_E against engineering parameters for H-Mode	147
4.28	H-factor s of shots in H-Mode dataset. $H \equiv \tau_E/\tau_{E,ITER89-P}$	147
4.29	Range of P_{RF} and I_p covered by \mathcal{H}	148
4.30	Regression of τ_E against dimensionless parameters for H-Mode	148
4.31	Distribution of H-Mode shots	149
4.32	Global Analysis for H-Mode group a	151
4.33	Time history of shots in H-Mode group a	152
4.34	Dimensionless parameter profiles for H-Mode group a shots	154
4.35	Local analysis plots for ρ^* scaling H-Mode group a	156
4.36	Volume-integrated ICRF power deposition at $t=0.78$ s for shot 960118016 in H-Mode, group a	157
4.37	Minority slowing down at $t=0.78$ s for shot 960118016 in H-Mode, group a	157
4.38	Time history of shots in H-Mode group b	159

4.39	$B\tau_E$ versus ρ^* for H-Mode group b	160
4.40	Dimensionless parameter profiles for H-Mode group b shots	161
4.41	Local analysis plots for ρ^* scaling H-Mode group b	162
4.42	Distribution of L-Mode shots for ν^* scaling shot selection	164
4.43	$B\tau_E$ versus ν^* for L-Mode	165
4.44	Dimensionless parameter profiles for L-Mode ν^* scaling	167
4.45	Scaling of χ with ν^* for L-Mode	168
4.46	Distribution of H-Mode shots for ν^* scaling shot selection	170
4.47	$B\tau_E$ versus ν^* for H-Mode	171
4.48	Dimensionless parameter profiles for H-Mode ν^* scaling	172
4.49	Scaling of χ with ν^* for H-Mode	173
5.1	Simple picture of the ITG Mode	181
5.2	Proximity to marginal stability threshold for C-Mod, DIII-D, TFTR, and ITER	186
5.3	H-factor vs. edge T_e for shots used in comparison	188
5.4	Determining optimal dt and n_s for simulations	192
5.5	Comparison of IFS-PPPL model for shot 960301009	194
5.6	Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960301009	195
5.7	Comparison of IFS-PPPL Model for Shot 960229042	196
5.8	Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960229042	197
5.9	Comparison of IFS-PPPL Model for Shot 960125004	199
5.10	Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960125004	200
5.11	Comparison of IFS-PPPL Model for Shot 960116024	201
5.12	Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960116024	202

5.13	Checking sensitivity of prediction to T_i/T_e . The dotted line shows the T_i prediction obtained while also predicting T_e . The dashed line shows the T_i prediction obtained using the measured T_e profile. Even though T_e is poorly predicted for this shot, the T_i prediction is not significantly affected.	203
5.14	Comparison of estimated T_i profile to predicted profile using different r_b	205
5.15	Effects on prediction of ad hoc sawtooth model	206
5.16	Effect of Z_{eff} profile shape on IFS-PPPL model prediction, shot 960116024	207
5.17	Effect of toroidal rotation on IFS-PPPL model prediction	209
5.18	Possible ways in which the IFS-PPPL model can be wrong	210
5.19	Plotting RMS difference between predicted and measured profiles . . .	212
A.1	Dataflow through TRANSP	218
A.2	Traditional steps in TRANSP run	222
A.3	TRANSP steps in MDSplus environment	223
A.4	TRANSP tree structure	225
A.5	Pasted TRANSP Run	228
A.6	Coordinate system shown on typical C-Mod plasma	231
A.7	PRETRANSP main screen	234
A.8	Symmetric Fourier decomposition	236
A.9	Asymmetric Fourier decomposition	236
A.10	Processing Widget	239
A.11	Comparison of smoothing kernels	240
A.12	Smoothing and cut times	242
A.13	Multigraph program	250
A.14	PSLICE	252

List of Tables

1.1	Alcator C-Mod parameters	25
3.1	Terms in TRANSP energy balance calculation	50
3.2	Assumed values of f_{RF} for different heating schemes when measurement is not possible	70
4.1	Some tokamak plasma dimensionless parameters	93
4.2	Named ρ^* scalings	96
4.3	ITER and Alcator C-Mod parameters	98
4.4	Diagnostics and plasma parameters for ρ^* scaling analysis	100
4.5	Errors in parameters making up ν^*	102
4.6	L-Mode shot groups for global scaling	112
4.7	Global scaling for L-Mode groups studied	125
4.8	Parameters for L-Mode, group a	126
4.9	Parameters for L-Mode, group b	132
4.10	Parameters for L-Mode, group c	136
4.11	Parameters for L-Mode, group e	140
4.12	Summary of ρ^* scaling in L-Mode	143
4.13	H-Mode shot groups for global scaling	150
4.14	Parameters for H-Mode, group a	153
4.15	Parameters for H-Mode, group b	158

LIST OF TABLES

17

4.16	Summary of ρ^* scaling in H-Mode	160
4.17	L-Mode shot groups for global ν^* scaling	165
4.18	Parameters for L-Mode, group a	166
4.19	H-Mode shot groups for global ν^* scaling	169
4.20	Parameters for H-Mode, group a	171
5.1	Parameters for comparison of P_{in}/P_{ITG} for Alcator C-Mod, DIII-D, TFTR, and ITER	185
5.2	Shots for IFS-PPPL model comparison	189
5.3	Values of $\nu \equiv 2.1RnT_e^{-1.5} T_i^{0.5}$ for C-Mod plasmas	190
5.4	Summary of IFS-PPPL model comparison	210
A.1	Run Status Values	229
A.2	Input Coordinate Systems	232
A.3	Output Coordinate Systems	233
A.4	Mapping Nodes	233
A.5	Usual TRANSP Inputs	237
A.6	Namelist settings for ion temperature control	247
B.1	Online Documentation	256

Chapter 1

Introduction

The development of controlled nuclear fusion as a commercially viable power source has proven to be a very difficult challenge. The promise of a limitless supply of inexpensive energy, with few hazardous by-products, provides the inspiration to tackle the many complex technical and fundamental physics hurdles that stand in the way. In the past 50 years, a considerable amount of knowledge has been amassed on both fronts. Several key factors have been established; these are summarized in § 1.1. The question of *transport*, the general subject of this thesis, is one of the many problems that remain to be solved. § 1.2 provides a concise definition of transport, and presents its relationship to the goal of fusion as a power source. The experiments presented in this thesis were conducted on the Alcator C-Mod tokamak, which is briefly described in § 1.3. § 1.4 presents an outline for the presentation of the results of these experiments, constituting the bulk of this thesis.

1.1 Fusion as a Power Source

The application of fusion to the generation of power is the same as with fission, the other directly nuclear-related energy source. Basically, the energy from the nuclear fusion reactions would be used to boil water to create steam that, as in any other fuel-burning plant, drives turbines and thus generates electricity. Much of the research conducted over the past 50 years has been to establish the conditions under which the fusion reactions can occur.

The fuel of a fusion reactor¹ will be deuterium and tritium, both isotopes of hydrogen, mixed in equal proportions. The nuclear reaction that is exploited is:



This reaction can only occur if the deuterium and tritium nuclei are within the range of the strong nuclear force that can bind them together, which is roughly the size of

¹The term “reactor” is adopted from the fission power plant.

the nuclei, or 10^{-15} m. Ultimately, from a fundamental physics point of view, what makes fusion so difficult is the necessity to overcome the Coulomb repulsion of the two positively charged nuclei to bring them this close together. Quantum mechanical effects allow two nuclei to penetrate each other's potential barriers with a relative energy of several tens of keV.

The most obvious source of nuclei with energies this high, particle acceleration, turns out not to be economical, as the vast majority of nuclei Coulomb scatter at distances far beyond the range of the strong force, and are lost from the system. Instead, a *thermonuclear* approach is employed, in which the nuclei have a thermal distribution of velocities with a temperature of ~ 10 keV. As the nuclei randomly collide they remain in the system, so there is no net energy loss from scattering. The nuclei in the tails of the thermal distribution have sufficient energy to undergo a fusion reaction upon colliding.

The ionization energy of hydrogen isotopes is ~ 13 eV, so at a temperature of 10 keV the deuterium and tritium gas is fully ionized. A fully ionized gas is a type of *plasma*; hence the importance of plasma physics to the quest for fusion energy. Plasmas show all the fluid-like properties of neutral gases, as well as exhibiting electromagnetic behavior from their freely moving charges. The study of plasmas takes place at both the macroscopic, or fluid level, and at the microscopic, or kinetic level. Many important phenomena unique to plasmas can only be understood from the latter point of view.

To create a plasma with an energy in the range of keV, it must be thermally isolated from its surroundings, confined long enough to heat and generate significant fusion power. The most promising approach to containing plasma employs magnetic fields. The Lorentz force restricts the motion of charged particles perpendicular to magnetic field lines, so that a uniform, straight magnetic field will confine a plasma in a cylindrical geometry. The field does not restrict the motion of the particles along it, so they must be prevented from escaping out the ends of the cylinder. One way to accomplish this is to bend the field lines around into a torus so that they form closed loops; this is basically the idea behind a tokamak. For a more detailed discussion of magnetic confinement of fusion plasmas, the interested reader is referred to Hazeltine and Meiss [1].

1.1.1 Tokamaks

A tokamak is a device designed to contain plasma in a particular type of toroidal configuration. First conceived by Lev Artsimovich² [2], tokamaks have proven to be the most successful of the magnetic fusion reactors. A schematic diagram of a tokamak is shown in figure 1.1.

²Artsimovich coined the name tokamak from the Russian "TOroidalnaya KAmera ee MAgnitaya Katushka" that translates to "toroidal chamber with magnetic coil".

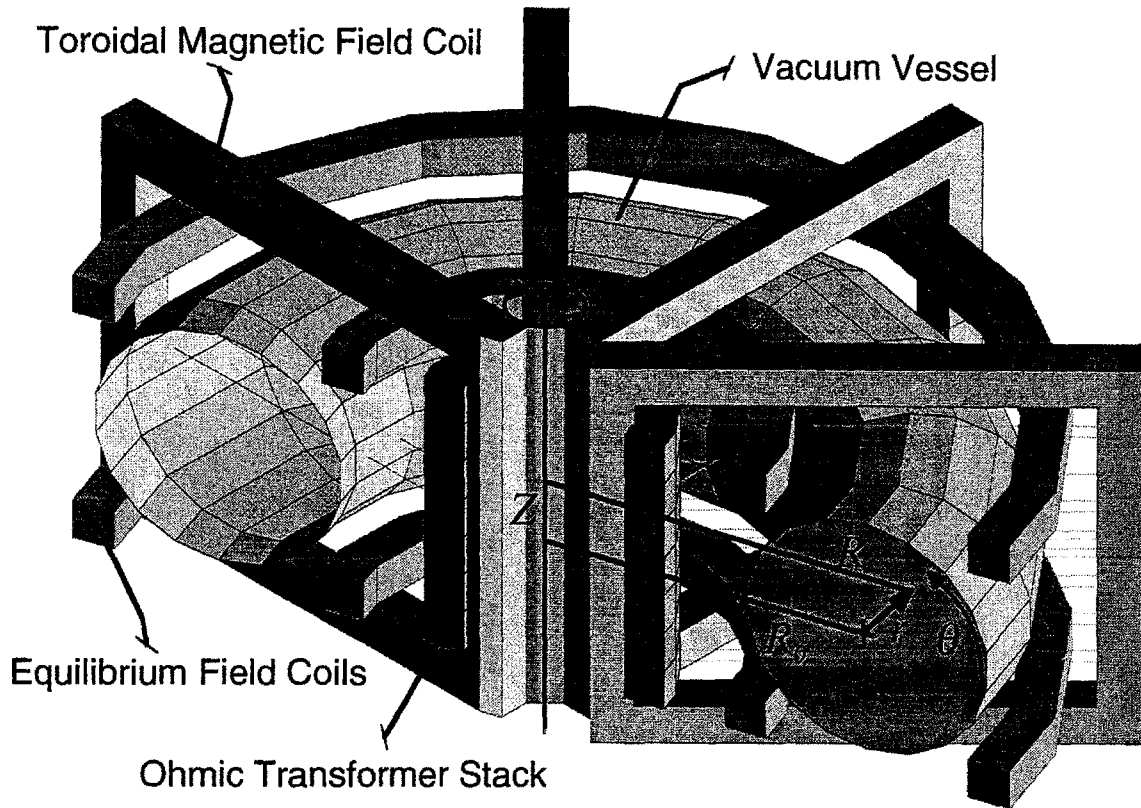


Figure 1.1: Schematic of a tokamak. (Courtesy Darren Garnier [3].)

It can be shown from both a macroscopic fluid picture, or a microscopic single particle picture, that a purely toroidal field does not provide a stable configuration for plasma confinement [4]. The addition of a poloidal field, in the θ direction as shown on figure 1.1, as well as magnetic fields in the vertical direction, will stabilize the configuration. In a tokamak, the poloidal field is created by driving a current in the plasma, which at 1 keV has the same electrical conductivity as copper. The vertical fields are provided by additional magnet coils.

The current is driven inductively by transformer action, with the plasma as the secondary, and a set of magnet coils in the center of the tokamak the primary. These coils are usually referred to as “ohmic heating coils”, as the current driven in the plasma heats it to nearly the temperatures required. The resistance of a plasma decreases with temperature, so the hotter it gets, the harder it is to resistively heat it. To reach the temperatures required for a fusion power plant, auxiliary heating methods must be employed, which heat the plasma with energetic neutral beams or resonant electromagnetic radiation.

The tokamaks of today are research devices not intended to produce commercially viable power. Their plasmas are usually purely deuterium to avoid the difficulties

of tritium handling and the increased neutron production from D-T fusion. Also, most existing tokamaks use conventional copper magnets, and none have tested a means of extracting the energy from the fusion reactions to generate electricity. A tokamak power plant of the future will need to use superconducting magnets, which require significantly less energy, and will have a “blanket” just outside of its vacuum chamber, designed to capture the neutrons emitted by the fusion reaction and convert their energy to heat in order to generate steam. Inductive current drive implies that tokamaks are inherently pulsed devices. As power plants probably need to operate continuously, future tokamaks will need to employ alternate means of sustaining the plasma current on a steady-state basis, most likely from electromagnetic current drive.

A comprehensive picture of the physics of tokamaks, their diagnosis, and operation, can be found in [5].

1.2 The Importance of Transport

Many problems remain to be solved to make fusion an economically viable power source. Several of these are technological, such as the design of the superconducting magnets or the first wall and blanket. Others, like steady-state current drive, involve more physics. Most of these problems should be solvable given sufficient ingenuity, funding, and time. There is one fundamental physics problem that may not prove tractable to conventional engineering problem solving: *transport*. In regards to tokamaks, transport refers to the flux of energy and particles out of the plasma, that is across the lines of the confining magnetic field.

From a thermodynamic viewpoint, a tokamak plasma is in a non-equilibrium state, being hotter and denser in its core than at its edge. The gradients in density and temperature are sources of free energy driving particle and heat fluxes across the magnetic field to bring the plasma to equilibrium. The rate at which this transport occurs in a large part determines the economic viability of fusion as a power source. This rate is discussed in terms of the *energy confinement time* τ_E , where the power loss P_{loss} is proportional to the amount of energy in the plasma U :

$$P_{\text{loss}} = \frac{U}{\tau_E}. \quad (1.2)$$

Power input to the plasma comes from both external sources and the fusion reactions, so that global energy balance can be written

$$\frac{dU}{dt} = -\frac{U}{\tau_E} + P_{\text{fusion}} + P_{\text{aux}}. \quad (1.3)$$

For a tokamak to be commercially viable, the fusion power must be at least equal to the externally applied power. The ideal scenario is to have P_{fusion} provide all the plasma heating, so that the external sources can be switched off. This state is known

as *ignition*, and is attained when the product of the ion density and the confinement time satisfies the *Lawson criterion*:

$$n_i \tau_E \gtrsim 1.0 \times 10^{20} / \text{m}^3 \quad (1.4)$$

with an ion temperature $T_i \sim 10$ keV [5, p.10]. The greater the transport, the lower the confinement time, and the harder it is to reach ignition.

One way to mitigate the transport losses from a plasma is to make it large, so that its surface area to volume ratio is small. Particles therefore have longer to travel through the plasma before escaping, thus increasing their probability of undergoing a fusion reaction. This idea underlies the design of the International Thermonuclear Experimental Reactor (or “ITER”), currently being studied as a potential next-generation tokamak by a consortium of the U.S., Europe, Japan, and Russia. ITER is supposed to be able to reach ignition. To get there based on our current understanding of transport, ITER must be physically very large and therefore costly; too expensive to be a practical power plant design. If tokamak transport could be accurately modelled and understood, then perhaps the size requirements of an ignition tokamak could be reduced, allowing for smaller machines and a cheaper power plant.

1.3 Alcator C-Mod

As stated in the previous section, one way to reduce the transport losses from a plasma is to make it big. This method is successfully employed by many of the tokamaks of today — JET, TFTR, JT-60U, and DIII-D — though is an undesirable long term solution, as the larger the tokamak, the more expensive it is. The other way to reduce transport is to use a strong magnetic field, essentially confining the particles more strongly. This latter approach is taken by the fusion research program at MIT.³ Alcator C-Mod [6] is the third in a series of high-field, compact tokamaks to be constructed here.

Despite its smaller size relative to the aforementioned tokamaks, Alcator C-Mod is able to produce reactor-relevant plasmas. Alcator C, its immediate predecessor, was the first tokamak to achieve the Lawson criterion, albeit at a temperature too low to reach ignition [7]. Additional transport results from the Alcator program can be found in [8–11].

Alcator C-Mod employs ICRF heating as its auxiliary heating source. This system is discussed in [10, 12–14]. Table 1.1 summarizes C-Mod’s parameters. A cross section of the machine is presented in figure 1.2.

³Its name “Alcator” is an acronym of the Italian “ALto CAmpo TORus” translating as “high field torus”.

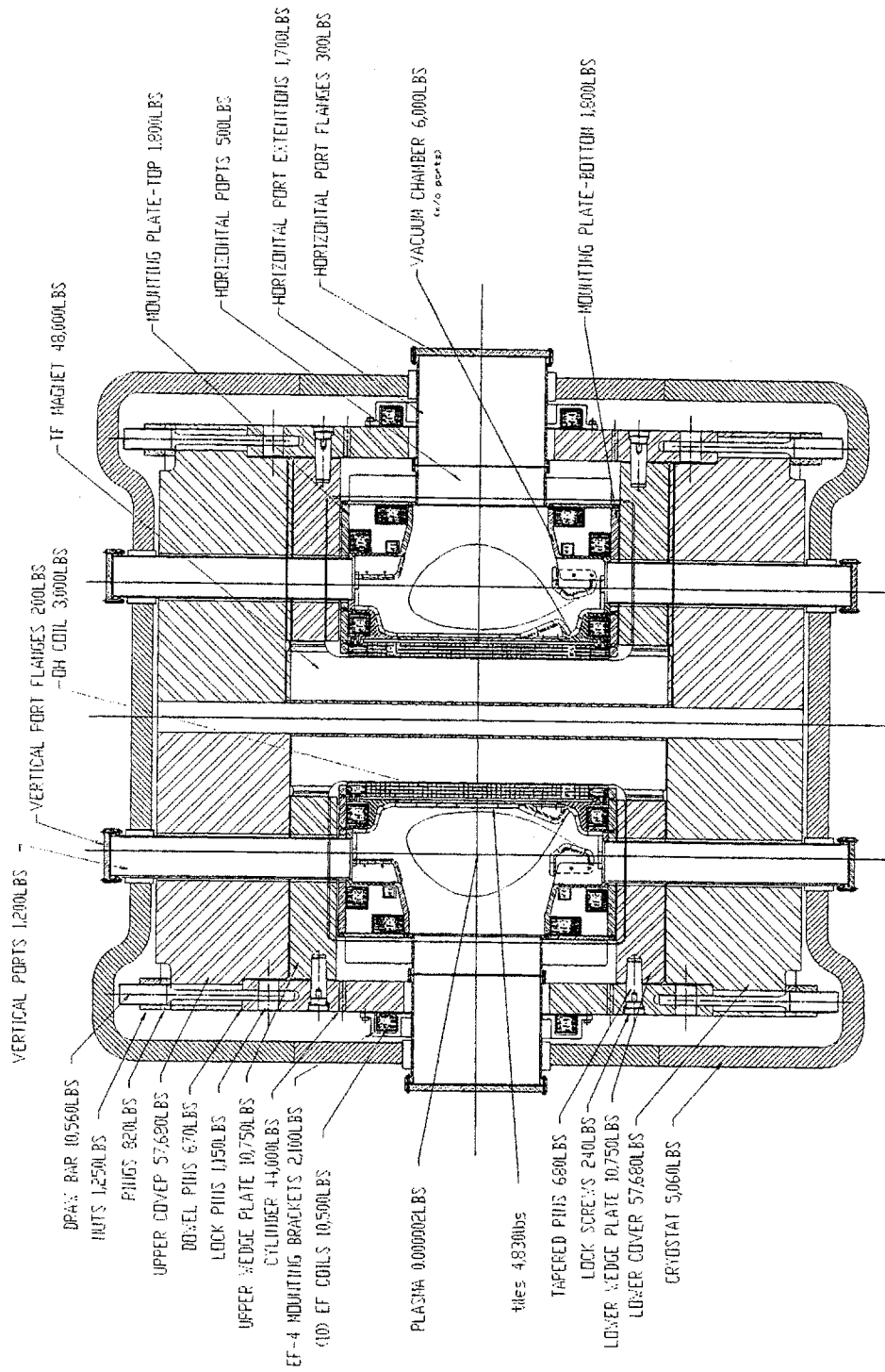


Figure 1.2: Cross section of the Alcator C-Mod tokamak

Major Radius	R_0	0.67	m
Minor Radius	a	0.21	m
Magnetic Field	B_T	7.9 (9)	T
Plasma Current	I_p	1.5 (3)	MA
Auxiliary Heating	P_{ICRF}	3.5 (8)	MW
Elongation	κ	0.95 – 1.85	
Triangularity	δ	0. – 0.4	
Electron Density	n_e	$\leq 1.1 \times 10^{21}$	m^{-3}
Electron Temperature	T_e	≤ 6	keV
Flattop time (at 8 T)		~ 1	s

Table 1.1: Alcator C-Mod parameters. Values in parentheses are for the design maximum.

1.4 Thesis Outline

The general goal of this thesis is to contribute to the understanding of tokamak transport. To this end two experiments were conducted on Alcator C-Mod. The first was an empirical study of the scaling of transport with ρ^* , the ion gyroradius normalized to the machine size. The second was a comparison of the predictions from the IFS-PPPL model of gyroBohm-scale ion temperature gradient driven transport with experimental measurements on C-Mod. The presentation of these studies is organized as follows.

Chapter 2 provides background information on both theoretical and empirical studies of tokamak transport, to place the experiments within the wider context of transport analysis, and relate them to each other as complementary methods. Most of the transport analysis of this thesis was at the “local” level, that is transport was studied as a function of position in the plasma. As explained in chapter 2, this means that the work relied heavily on a local transport analysis computer code. The code used in this thesis, TRANSP, is described in chapter 3. A presentation of the comprehensive ρ^* scaling experiments follows next, beginning with the motivation for the choice of ρ^* as the parameter of interest for scaling. The results of the tests of the IFS-PPPL model are presented in chapter 5. Finally, chapter 6 provides a summary of the results, and suggestions for further research.

Chapter 2

Physics of Transport

Tokamak transport is driven primarily by turbulence arising from the electromagnetic nature of plasma. The fact that the transport is turbulent implies it cannot be described by a closed-form analytic model, as evidenced by the lack of such a description for neutral fluids despite years of effort [15]. To include the electromagnetic properties of plasma, theoretical treatments require drastic simplifications which may render them impotent to describe plasmas relevant to power generation. § 2.2 describes the different approaches to a theoretical understanding of plasma transport, starting with the simplest “classical” transport picture and moving to the much more complex “anomalous” picture. In the process, one of the experiments presented in this thesis, the application of a particular model of anomalous transport to understanding C-Mod experimental data, is introduced.

Since transport theory is so difficult, empirical means have been used in part to predict the performance of a power plant reactor based on the results from current experiments. Basically, these involve measuring the scaling of transport with different tokamak parameters and projecting the result to the parameters of a power plant reactor. § 2.3 provides the details behind empirical transport analysis, and introduces the idea of ρ^* scaling, which constitutes the other experiment presented in this thesis.

2.1 Transport Basics

Transport theory encompasses the duality of plasma behavior. The effects of interest for experimental studies are at the fluid level: fluxes of particles and heat out of the plasma. To account for these fluxes, a theoretical examination must start at the kinetic level, as the particle collisions and wave-particle interactions responsible for them are inherently kinetic phenomena.

The starting point at the kinetic level is the Boltzmann equation for the distribu-

tion function f of a plasma species j :

$$\frac{\partial f_j}{\partial t} + \mathbf{v} \cdot \nabla f_j + \frac{q_j}{m_j} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f_j = \mathcal{C}(f_j). \quad (2.1)$$

Many texts on plasma physics provide derivations of the Boltzmann equation; see for example [16, ch.2]. Note that the interactions affecting the distribution function f are separated into those involving the plasma as a whole, and those involving short range, individual particle encounters. The former interactions are from the electric and magnetic fields in the force term $(\mathbf{E} + \mathbf{v} \times \mathbf{B})$ that are determined by the cumulative effects of all the plasma particles. The latter type of interactions is modelled by the collision operator \mathcal{C} , which usually includes only Coulomb collisions.

The first three moments of (2.1) are the fluid equations for particle, momentum, and energy conservation respectively:

$$\frac{\partial n_j}{\partial t} = -\nabla \cdot \mathbf{\Gamma}_j + S_n \quad (2.2)$$

$$m_j \frac{d\mathbf{u}_j}{dt} = q_j (\mathbf{E} + \mathbf{u}_j \times \mathbf{B}) + \frac{1}{n_j} \mathbf{R}_j - \frac{1}{n_j} \nabla \cdot \mathbf{P}_j + S_p \quad (2.3)$$

$$\frac{3}{2} n_j \frac{dT_j}{dt} = Q_j - \mathbf{P}_j : \nabla \mathbf{u}_j - \nabla \cdot \mathbf{q}_j + S_E. \quad (2.4)$$

where, with $\langle X \rangle \equiv \int d^3v X f_j$ and $\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \mathbf{u}_j \cdot \nabla$,

$$\begin{aligned} n_j &= \langle 1 \rangle && \text{is the density,} \\ \mathbf{u}_j &= \frac{1}{n_j} \langle \mathbf{v}_j \rangle && \text{is the fluid velocity,} \\ \mathbf{\Gamma}_j &= n_j \mathbf{u}_j && \text{is the particle flux} \\ \mathbf{w}_j &= \mathbf{v}_j - \mathbf{u}_j && \text{is the random velocity,} \\ \mathbf{R}_j &= m_j \sum_{\ell} \langle \mathbf{w}_j \mathcal{C}_{j\ell} \rangle && \text{is the momentum transfer due to collisions,} \\ p_j &= \frac{1}{3} m_j \langle w^2 \rangle && \equiv n_j T_j \text{ is the scalar pressure,} \\ \mathbf{P}_j &= m_j \langle \mathbf{w}_j \mathbf{w}_j \rangle && \text{is the pressure tensor,} \\ \mathbf{\Pi}_j &= \mathbf{P}_j - p_j \mathbf{1} && \text{is the viscosity tensor,} \\ Q_j &= \frac{1}{2} m_j \sum_{\ell} \langle w^2 \mathcal{C}_{j\ell} \rangle && \text{is the energy exchange due to collisions, and} \\ \mathbf{q}_j &= \frac{1}{2} m_j \langle w^2 \mathbf{w} \rangle && \text{is the random heat flux.} \end{aligned} \quad (2.5)$$

The terms S_n , S_p , and S_E are included to allow for external sources of particles, momentum, and energy, and phenomena not described by (2.1) with a Coulomb collision operator. These include externally applied heating sources like neutral beams, and atomic physics effects like radiation and charge exchange. Many standard texts on plasma physics provide a derivation of (2.2)-(2.4); a particularly clear one is found in [4]. Note that $\mathcal{C}_{j\ell}$ is the collision operator for species j colliding with species ℓ .

The terms of interest for transport analysis are the particle flux Γ and the random heat flux \mathbf{q} . Studies of the former on C-Mod would require the density profile measurements to be far more accurate than is possible given current diagnostic technology, as the particle source is restricted to within a narrow distance from the edge. Since the energy source is in general centrally peaked for C-Mod, studies of the heat flux are feasible; these will be the focus for this thesis.

(2.2)-(2.4) do not form a closed system of equations, as there are terms involving higher order moments: the viscosity Π_j , the random heat flux \mathbf{q}_j , and the collision terms \mathbf{R} and Q_j . An infinite set of moment equations would be required for a complete system, which is clearly not useful. Instead, the equations are closed by expressing the higher order terms as functions of the lower order terms n_j , \mathbf{u}_j , and T_j . This closure can be based on a theoretical treatment of the underlying kinetic equations, or can be a set of ad-hoc assumptions.

Two complimentary approaches are used for studying \mathbf{q} . Theoretical treatments attempt to calculate \mathbf{q} by returning to the kinetic picture and choosing a particular model or regime in which to solve for f . This approach only works for simplified models — a general solution for f for tokamak plasmas is not forthcoming — as discussed in § 2.2. Empirical treatments attempt to infer \mathbf{q} from plasma measurements given assumptions about closure of the fluid equations. However, as explained in § 2.3, these assumptions are not always valid, and the results are prone to large uncertainties. Both methods are therefore used to attempt to form a self-consistent and thorough understanding of energy transport.

2.2 Theoretical Transport Analysis

If the distribution function f can be determined for a given model, the heat flux \mathbf{q} can be calculated using (2.5):

$$\mathbf{q} = \langle w^2 \mathbf{w} \rangle = \int d^3v \frac{m_j}{2} w^2 \mathbf{w} f. \quad (2.6)$$

In most theoretical analyses, the distribution function f is assumed to be nearly Maxwellian, so that a perturbation technique can be applied:

$$f = f_M + f_1, \quad \text{where } f_1 \ll f_M. \quad (2.7)$$

In this case (2.6) becomes

$$\mathbf{q} = \int d^3v \frac{m_j}{2} w^2 \mathbf{w} f_1, \quad (2.8)$$

i.e. the deviation of the distribution from a Maxwellian drives the transport. Deriving the heat flux \mathbf{q} from theory then amounts to solving the Boltzmann equation for f_1 .

The perturbation f_1 can arise from two general types of plasma interactions: close range few body collisions, and longer range collective effects involving wave/particle interaction. Early theoretical treatments of transport concentrated on the former. The *classical theory* of transport solves for f_1 for a plasma in a uniform, straight magnetic field in which small angle, binary collisions are the dominant effect [17]. *Neoclassical theory* expanded this description to include the effects of the magnetic geometry, which turn out to be important for tokamak plasmas [18]. Summaries of these two theories are provided in § 2.2.1 and § 2.2.2.

Experiments on several tokamaks, including Alcator C [19], showed that the actual transport in a plasma was occurring at a much greater rate than could be accounted for by neoclassical theory. The excess transport was dubbed “anomalous” at first as its origin was a mystery. Progress in theory led to the conclusion that the longer range collective effects were responsible for the anomalous transport [20, 21]. Unfortunately, the dependence of \mathbf{E} and \mathbf{B} on f makes the resulting equation for f_1 highly nonlinear. The physical manifestation of this is the fact that the transport is turbulent and possibly even chaotic. Since a general solution for f_1 is currently not possible, simplification of the plasma into particular modes and regimes is required for a theoretical treatment. Numeric solutions are also possible, but are difficult because of the wide range of characteristic time and length scales inherent in the equations. A summary of the current state of anomalous transport theory is provided in § 2.2.3, while surveys of the current state of understanding can be found in [22–26].

2.2.1 Classical Transport

Classical transport studies the transport due to collisions in a uniform, straight magnetic field. For tokamaks, classical transport can be invoked in the “large aspect-ratio limit” $R \gg a$ when the torus can be approximated as a cylinder. The transport along and across the field is very different.

Parallel to the field, the particles are essentially free-streaming. In general, a net fluid flow exists in the parallel direction:

$$\mathbf{u}_{rel} = \mathbf{u}_e - \mathbf{u}_i. \quad (2.9)$$

Associated with this flow is a current $\mathbf{j} = -en_e\mathbf{u}_{rel}$. The most important aspect of parallel transport for tokamaks is the resistive, or “ohmic” heating resulting from this current, which is basically the thermalization of the kinetic energy of the net fluid flow \mathbf{u}_{rel} due to collisions.

Perpendicular to the field, the electrons and ions are confined to their Larmor orbits until they experience a collision. Treating the collisions as binary (two-particle) and uncorrelated, the particles will undergo a “random-walk” across the field with a

step-size Δ equal to their Larmor radius ρ :

$$\rho = \frac{v}{\omega_c}, \quad (2.10)$$

where v is the particle velocity and ω_c is the particle Larmor frequency $= ZeB/m$.

The time-step of the random walk is known as the collision time τ . For plasmas, τ is not usually taken to be the time between individual, binary collisions. The $1/r^2$ dependence of the Coulomb force means that longer-range, small angle collisions are far more prevalent than closer-range, large angle collisions. Many small angle collisions are required to scatter a particle by 90° ; τ is the time scale on which these 90° scatterings occur. Braginskii [17, p.215] has

$$\tau_e = \frac{3 m_e^{\frac{1}{2}} T_e^{\frac{3}{2}}}{4 (2\pi)^{\frac{1}{2}} e^4 n_e \log \Lambda} = 3.44 \times 10^5 \frac{T_e^{\frac{3}{2}}}{n_e \log \Lambda} \text{ s} \quad (2.11)$$

$$\tau_i = \frac{3 m_i^{\frac{1}{2}} T_i^{\frac{3}{2}}}{4\pi^{\frac{1}{2}} Z_i^4 e^4 n_i \log \Lambda} = 2.09 \times 10^7 \frac{T_i^{\frac{3}{2}}}{Z_i^4 n_i \log \Lambda} \text{ s}, \quad (2.12)$$

where $\log \Lambda$ is the ‘‘Coulomb logarithm’’ and the numeric values are for T_e, T_i in eV and n_e, n_i in cm^{-3} . Plasmas are considered magnetically confined if $\omega_c \tau \gg 1$; that is if the particles complete many Larmor orbits before colliding. Collisional transport theory applies when the macroscopic time scales satisfy $d/dt \ll 1/\tau$ and the macroscopic length scales $L_{\parallel} \gg \lambda_{\text{mfp}}, L_{\perp} \gg \rho$ where the mean free path λ_{mfp} is defined as:

$$\lambda_{\text{mfp}} = v_t \tau. \quad (2.13)$$

Braginskii [17] presents the solution of (2.1) using (2.7) for a two species plasma (electron and singly charged ion) in a uniform straight magnetic field, using a Fokker-Planck operator for \mathcal{C} . For $\omega_c \tau \gg 1$, the higher order moments in (2.2)-(2.4) impor-

tant to transport are:

$$\mathbf{R}_e = -\mathbf{R}_i = en_e(\eta_{\parallel}\mathbf{j}_{\parallel} + \eta_{\perp}\mathbf{j}_{\perp}) \quad (2.14)$$

$$\eta_{\parallel} = \frac{m_e}{e\tau_e} = 2.0\eta_{\perp} \quad (2.15)$$

$$Q_i = \frac{3m_e n_e}{m_i \tau_e}(T_e - T_i) \equiv Q_{ei} \quad (2.16)$$

$$Q_e = Q_{OH} - Q_{ei} \quad (2.17)$$

$$Q_{OH} = -\mathbf{R} \cdot \mathbf{u}_{rel} = \eta j^2 \quad (2.18)$$

$$\mathbf{q}_j = -\kappa_{j,\perp} \nabla T_j \quad (2.19)$$

$$\kappa_{j,\perp} = c_j \frac{n_j T_j}{m_j \omega_{c_j}^2 \tau_j} \quad (2.20)$$

$$c_e = 4.66 \quad (2.21)$$

$$c_i = 2 \quad (2.22)$$

The values for c_j are for T measured in eV, and all other parameters in cgs units. The thermal conductivity κ in (2.20) is usually converted to the *thermal diffusivity* χ by dividing by the thermal capacity per unit volume of the plasma for species j , $c_j = \frac{3}{2}n_j$:

$$\chi_j = c_j \frac{T_j}{m_j \omega_{c_j}^2 \tau_j}. \quad (2.23)$$

For a 2 keV, 5 T plasma with $n_e = 1 \times 10^{20}/\text{m}^3$, $\chi_e \simeq 1.0 \text{ cm}^2/\text{s}$, and $\chi_i \simeq 10 \text{ cm}^2/\text{s}$. Actual thermal diffusivities measured in tokamak plasmas are $\simeq 1 \text{ m}^2/\text{s}$, or three to four orders of magnitude larger than predicted by classical theory. Some of the terms in the power balance equation, notably the electron-ion exchange term Q_{ei} , are well described by a classical model.

2.2.2 Neoclassical Transport

Neoclassical transport theory accounts for the non-uniform magnetic field geometry in a tokamak. Two aspects of the magnetic field geometry are important: the helicity of the field resulting from the superposition of the external toroidal field and the poloidal field of the plasma current, and the drifts caused by the curvature of and gradients in the field.

Toroidal Geometry

Figure 2.1 shows the geometry of the fields in a tokamak. The externally applied toroidal field in the $\hat{\phi}$ direction adds with the poloidal field of the plasma current I_p in the $\hat{\theta}$ direction to form a net helical field. This configuration is dictated by the

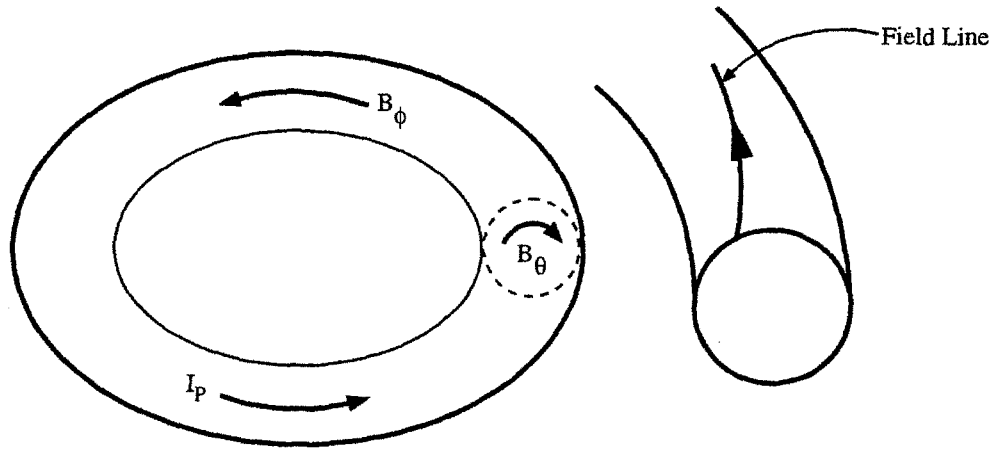


Figure 2.1: Magnetic field geometry in a tokamak. B_ϕ is the toroidal field generated by external coils, and B_θ is the poloidal field generated by the plasma current I_p , which flows predominantly in the toroidal direction. Design of figure taken from [5, p.12]

requirements of magnetohydrodynamic (MHD) stability for a toroidal plasma. As the toroidal field is stronger than the poloidal field, particles streaming along the field lines typically make one to five toroidal rotations for one poloidal rotation. The field pitch within the plasma is defined in terms of the safety factor q :

$$q(\psi) = \frac{1}{2\pi} \int_0^{2\pi} \left(\frac{r B_\phi}{R B_\theta} \right)_\psi d\theta \quad (2.24)$$

where the integral is carried out over the flux surface at ψ . In the large aspect ratio limit becomes, the tokamak approximates a cylinder, and all the quantities inside the integrand of (2.24) are constant, so

$$q(r) = \frac{r B_\phi}{R B_\theta}. \quad (2.25)$$

For irrational q values, a field line traces out a closed surface known as a “flux surface”. From MHD it can be shown [4] that the plasma pressure p is a constant on a flux surface (p is then referred to as a “flux function”). Since parallel transport is much faster than perpendicular transport, parallel gradients in temperature and density¹ are quickly smoothed out, so that they too can be considered flux functions. Only gradients across the flux surfaces, in the radial direction, therefore need to be considered.

¹Actually, if there exists significant rotation, density is no longer exactly a flux function.

Quantities which are not flux functions, such as the current density J and the magnetic field B , are flux-surface averaged for transport purposes. This average is defined as:

$$\langle A \rangle = \frac{1}{\mathcal{V}} \oint \frac{d\theta A}{\mathbf{B} \cdot \nabla \theta} \equiv \frac{1}{\mathcal{V}} \mathcal{I}(A), \quad (2.26)$$

where $\mathcal{V} \equiv \mathcal{I}(1)$.

Collisional Transport in a Tokamak

The gradients in both the toroidal and poloidal fields and the curvature of the toroidal field, combined with parallel motion, lead to particle drifts in the vertical direction [27, p.26]. The helicity of the field lines averages out the drifts so that particles remain confined [4, p.81].

On the outboard side of the torus, particles streaming along the field lines experience an increasing field as they are carried inboard. Conservation of energy and the adiabatic invariance of the particle magnetic moment $\mu = mv_{\perp}^2/2B$ imply that particles whose ratio of perpendicular and parallel velocity satisfies

$$v_{\parallel} \leq v_{\perp} \sqrt{\epsilon}, \quad (2.27)$$

will be “trapped” by the magnetic field, reflected back and forth on the outboard side by the magnetic mirror effect. The magnetic drifts combine with the mirror motion of the particles to produce “banana orbits”, so-called because of their shape in a poloidal cross section. Hazeltine and Meiss [1, p.141,342] provide a detailed discussion of this phenomenon. One way to visualize the particle orbit shape is to plot contours of constant canonical momentum $p_{\parallel} = mv_{\parallel} + eA_{\parallel}$ subject to conservation of μ and energy U . Figure 2.2 plots these contours for a circular plasma and a particular choice of μ and U ; it is meant to be schematic only. The width of the banana orbit at the midplane is known as the “banana width” Δ_b , approximately given by

$$\Delta_b \simeq \rho_p \sqrt{\epsilon} \quad (2.28)$$

where ρ_p is the *poloidal* gyroradius. The inverse of the time taken for a particle to execute one of its banana orbits is known as the “bounce frequency”, approximately given by:

$$\omega_b = \epsilon^{1/2} \frac{v}{qR}. \quad (2.29)$$

Collisions will alter the status of a particle from trapped to detrapped or vice-versa. The ratio of the collision frequency $\nu = 1/\tau$, where τ is defined by (2.11) or

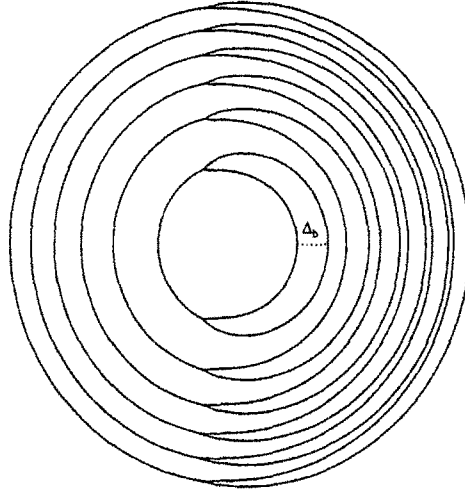


Figure 2.2: Banana orbits as shown by contours of constant p_{\parallel} .

(2.12), to the bounce frequency ω_b is known as the *collisionality*² ν^* , which for species j is given by:

$$\nu^*_{j} = \frac{\nu_j}{\omega_{bj}} = \frac{1}{\tau_j \omega_{bj}}. \quad (2.30)$$

Tokamak plasmas typically have $\nu^* < 1$, so that the trapped particles are able to execute many banana orbits before being detrapped by collisions.

Because of their banana orbits, trapped particles have different transport properties compared to untrapped, or “passing” particles. Their restricted motion parallel to the field increases their resistivity, while perpendicular to the field, collisions detrapping and retrapping them make their diffusive step-size

$$\Delta \sim \Delta_b \simeq \epsilon^{\frac{1}{2}} \rho \frac{B_t}{B_p} > \rho. \quad (2.31)$$

Using $\Delta = \Delta_b$ in the heuristic approximation of χ for the example parameters in § 2.2.1 gives³

$$\chi = \frac{\Delta_b^2}{\tau} \simeq \begin{cases} 4 \text{ cm}^2/\text{s} & \text{for electrons} \\ 100 \text{ cm}^2/\text{s} & \text{for ions.} \end{cases} \quad (2.32)$$

Neoclassical perpendicular transport of trapped particles is therefore much faster than classical predictions, but is still too slow to account for observed diffusivities — almost

²This is the collisionality relevant to tokamak transport. Ratios of the collision frequency to other characteristic frequencies are also defined, but not used here.

³See Hazeltine & Meiss [1, p.342] for a derivation of these terms.

four orders of magnitude lower for electrons and an order of magnitude lower than for ions. A more rigorous estimation of the neoclassical χ can be found in [1].

Neoclassical transport is considered the irreducible minimum transport possible for a tokamak, as if all turbulent transport could somehow be suppressed, collisions would still drive transport through neoclassical effects.

2.2.3 Anomalous Transport

Neoclassical theory is a comprehensive description of collisional transport in a tokamak. Its failure to predict the levels of transport actually observed in plasmas indicates that there are non-collisional processes involved. Collectively, these processes are referred to as *anomalous* transport. The underlying physics of anomalous transport is not completely understood.

Anomalous transport is thought to result from instabilities in the electrostatic and electromagnetic fluctuations propagating in the plasma. Their small growth rates and fine scale lengths earn them the name “microinstabilities”. As with collisional transport, the ultimate source of free energy driving the microinstabilities is the deviation of the plasma from thermodynamic equilibrium. Two types of microinstabilities are identified, based on their effects on the magnetic field structure. Electrostatic turbulence arises from electrostatic modes propagating in the plasma. The electric field perturbations from these modes are described by a scalar potential $\tilde{E} = -\nabla\tilde{\phi}$, so only weak finite β effects perturb the magnetic field structure. Electromagnetic turbulence is driven by perturbations primarily affecting the magnetic structure, which can cause field line tearing and reconnection. Since parallel transport is much more rapid than cross-field transport, a large enhancement of transport can occur if field lines are reconnected with components parallel to the density and temperature gradients. In the plasma edge, there is strong evidence suggesting that electromagnetic modes are not responsible for the observed turbulence, while in the core, where turbulence measurements are much harder to conduct, there is indirect evidence of the same [23]. There is also growing evidence specifically in favor of certain electrostatic modes as the underlying cause of anomalous transport. The following sections describe their origin and how they lead to transport. For more information on both types of instabilities and their relationship to transport, see the review article by Connor and Wilson [22].

Electrostatic Modes

There are two general mechanisms for driving electrostatic modes in a tokamak plasma: gradients in density and temperature, and field line gradients and curvature. The transport results from the $\mathbf{E} \times \mathbf{B}$ drifts associated with the perturbed electric field; this is described in the following section.

The electrostatic modes must be unstable to become finite in amplitude and drive transport. A large number of unstable electrostatic modes have been determined

from solving the Boltzmann equation (2.1), which is thought to provide a complete description of the plasma physics in the core. Expressed in a form suitable for the tokamak geometry, time and length scales, the Boltzmann equation is known as the nonlinear, electromagnetic toroidal gyrokinetic equation [28, p.4]; its sheer complexity is the reason why anomalous transport is poorly understood. The equations for each species are coupled and highly nonlinear. They are also stiff as they embody a large range of characteristic time and length scales. Analytic solutions are only possible for the linearized equation, and in many cases are obtained only in simplified geometries (slab models as opposed to being fully toroidal) for specific modes. In fact, the toroidal geometry, nonlinear behavior, and mode coupling play important roles in determining whether or not modes are stable, so these analytic solutions have so far been unable to describe experimental measurements.

Numerical solutions of the gyrokinetic equations, and the fluid equations derived from them, are proving more tractable. The “IFS-PPPL” model of ion temperature gradient (ITG) modes [29,30] and their resulting transport by Beer, Dorland, Hammett, and Kotschenreuther [31] has garnered much attention recently because of its success in predicting the observations made on a variety of tokamaks. Chapter 5 presents a study of their model’s ability to predict transport on Alcator C-Mod.

Electrostatic Turbulent Transport

The precise mechanism by which electrostatic modes drive transport is also not completely understood. The transport is thought to result from the $\mathbf{E} \times \mathbf{B}$ flow arising from the perturbing mode’s electric field. Figure 2.3 shows a contour plot of a localized $\tilde{\phi}$ perturbation. The perturbed electric field is directed down the gradient of $\tilde{\phi}$ as shown by the solid arrows. If a magnetic field is present perpendicular to $\tilde{\mathbf{E}}$, an $\mathbf{E} \times \mathbf{B}$ drift will result, circling around the perturbations and creating “eddies”.

The eddies can drive transport if their phase relative to the density or temperature perturbations is favorable. Figure 2.4 shows a schematic diagram relating the density and potential perturbations; the former are plotted with the solid lines and the latter with the dashed lines. Recall from figure 2.3 that the eddy velocity circulates around the $\tilde{\phi}$ contours. In (a) the relative phase between \tilde{n} and $\tilde{\phi}$ is such that the flux driven by the eddy velocity, shown by the arrows, is greater to the right than to the left. This is true as the system propagates in time as shown in (b). The transport will be in the opposite direction for a different relative phase between \tilde{n} and $\tilde{\phi}$ as shown in (c). For certain relative phases, there will be no net transport. The same argument holds for heat transport associated with the temperature perturbations \tilde{T} . If the eddies across a macroscopic region of the plasma are correlated, the transport will be significant. Of course, this is a highly schematic picture in an idealized geometry. Many factors are involved in determining the relative phase of the perturbations, and the ability of the eddies to drive transport. These include effects such as the field line curvature and shear, velocity shear, trapped particles, and kinetic effects like Landau damping [28].

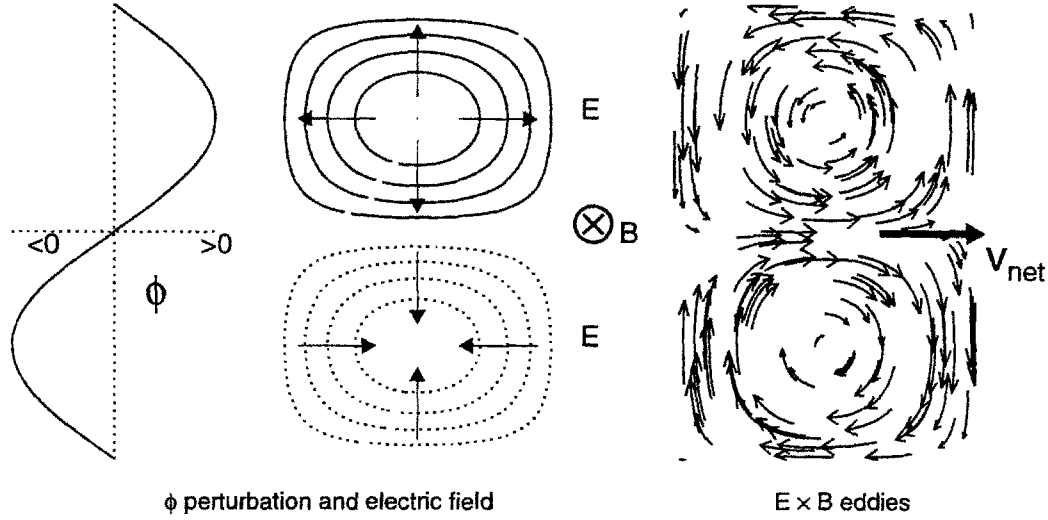


Figure 2.3: Eddies formed by perturbations in potential

The turbulent nature of the transport results from the nonlinear interaction of eddies from many modes of differing frequencies.

Numerical Estimates

A rough estimate of the transport from anomalous modes is possible by borrowing the heuristic random-walk picture from the collisional theories in § 2.2.1 and § 2.2.2, that is $\chi \sim \Delta^2/\tau$, where Δ is the characteristic step size and τ the time scale. Electrostatic modes driven by gradients in density or temperature typically have frequencies on the order of the diamagnetic frequency $\omega_* = kc_s\rho_i/L$. This can be shown by treating the electrons as adiabatic, and the ions as $\mathbf{E} \times \mathbf{B}$ drifting from a perturbed electric field $\tilde{\mathbf{E}}$ [27, p.190], [1, p.392]. The magnitude of $\tilde{\mathbf{E}}$ is $E = |\nabla\tilde{\phi}| \sim T/eL$ where it is assumed $e\tilde{\phi} \sim T$, and L is the characteristic gradient length. Returning to the heuristic picture of χ , assuming that $\tau \sim 1/\omega_*$, it is found

$$\chi = \frac{\Delta^2}{\tau} = \Delta^2 c_s \rho_i \frac{k}{L}. \quad (2.33)$$

Using a mixing length estimate of Δ [32, p.117], that is $\Delta \sim 1/k$, the heuristic form of the anomalous χ is

$$\chi \sim c_s \rho_i \frac{\Delta}{L} = \chi_B \frac{\Delta}{L}, \quad (2.34)$$

where $\chi_B \equiv c_s \rho_i = cT/eB$ is the *Bohm diffusivity*.

Theoretical treatments using the gyrokinetic equation, including the IFS-PPPL model, implicitly assume that $k\rho_i \lesssim 1$, giving $\chi \sim \chi_B \rho^*$, where $\rho^* = \rho_i/a$ is the ion

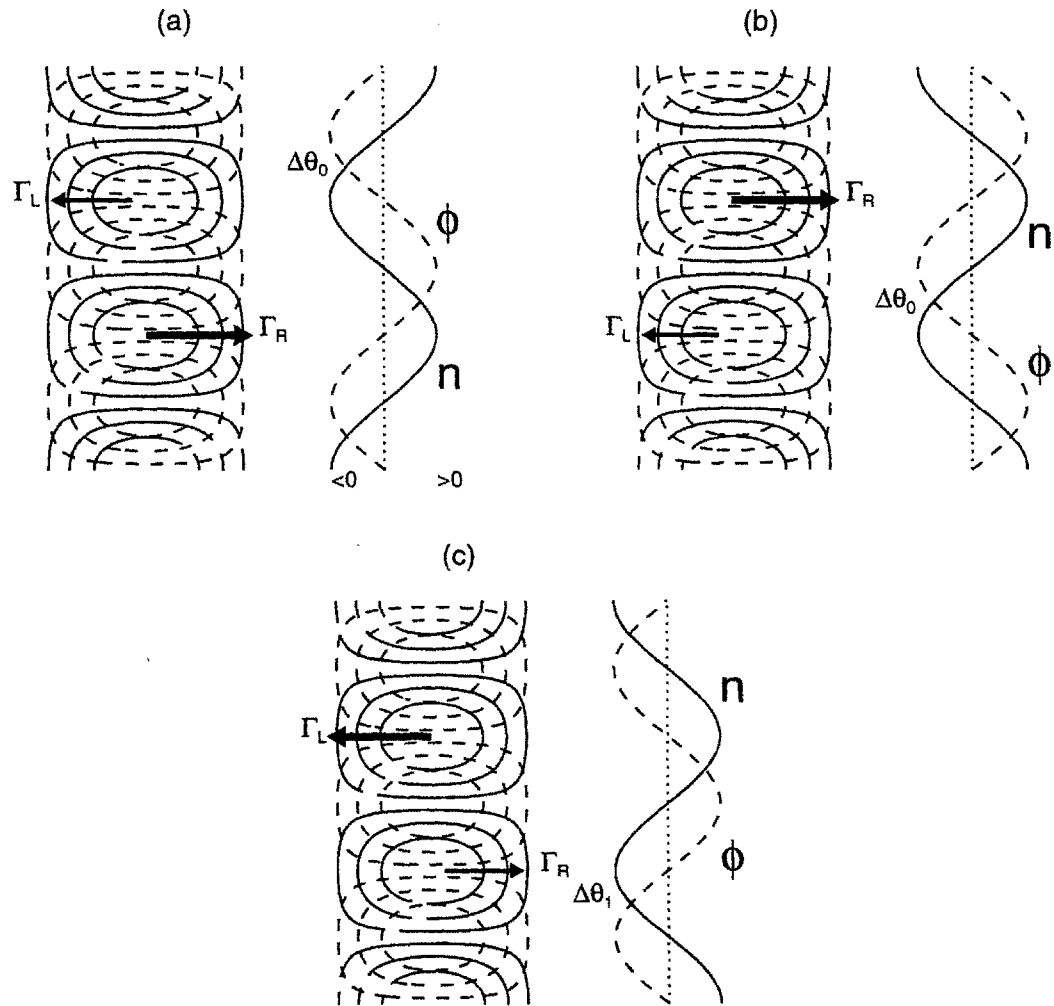


Figure 2.4: Transport driven by eddies. Density perturbations are shown with the solid lines; ϕ perturbations by the dashed lines. The arrows show the particle flux $\Gamma = nv$ driven by the net eddy velocity. The transport depends on the relative phase of $\tilde{\phi}$ and \tilde{n} . For the relative phase in (a) the region of high density is convected to the right, so $\Gamma_R > \Gamma_L$ and there is net flux to the right. Propagated in time, the same relative phase still has $\Gamma_R > \Gamma_L$. Transport can be driven in the opposite direction for a different relative phase, as shown in (c).

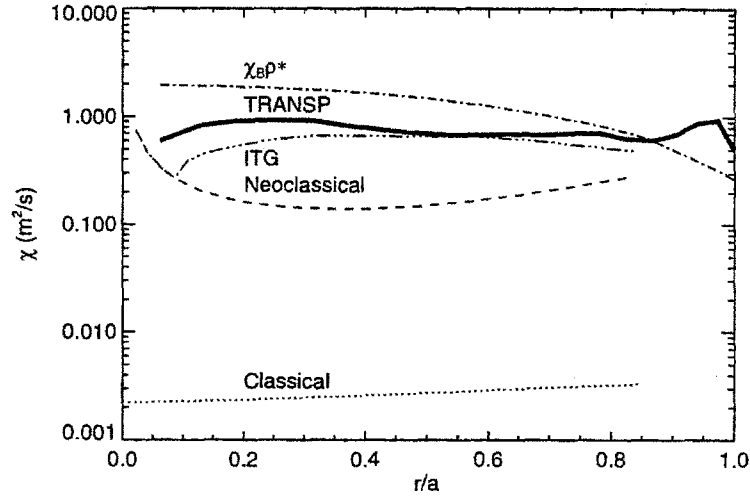


Figure 2.5: Comparison of χ_i from various models of transport to experiment

gyroradius normalized to the machine size. One type of empirical study of anomalous transport examines the scaling of χ/χ_B with the normalized ion gyroradius [33]. These ρ^* scaling experiments were conducted on Alcator C-Mod and are the subject of chapter 4.

A comparison of the χ_i profiles predicted from the various models of transport to the χ_i profile calculated from a particular shot is shown in figure 2.5. In addition to the classical and neoclassical χ_i models, the IFS-PPPL model χ_i is plotted, as well as χ_i from the heuristic picture of anomalous transport (2.34) assuming that $k\rho_i \sim 1$. The heuristic picture is close to the measured profile, and the IFS-PPPL model does very well, while the classical and neoclassical χ_i 's are too low. One plot does not a comparison make; chapters 4 and 5 are dedicated to exploring the comparison of χ with theory and its empirical scaling in more detail.

2.3 Empirical Transport Analysis

Despite the difficulties of transport theory, much progress has been made in characterizing transport from an empirical point of view. Empirical transport analysis attempts to determine relationships between the transport properties of the plasma and characteristic plasma parameters. These relationships are used primarily to predict the performance of future tokamaks, but can also suggest avenues to explore with a theoretical approach.

Empirical scaling is a useful, though potentially misleading tool for reactor design. It is precisely its most useful aspect — a method of circumscribing the difficulties with first principles theory — that makes it problematic. It is true that essentially the same scalings are observed on virtually all currently operating tokamaks, suggesting

that the underlying physics of transport is to a large degree machine independent, and that the empirical scalings determined in the late 1970s to early 1980s on the small tokamaks of the era successfully predicted the order of magnitude increase in τ_E observed on the larger tokamaks of the mid 1980s [24]. However, without a theoretical underpinning, there is no way to be certain that scalings observed on the tokamaks of today will be observed on the tokamaks of tomorrow. The techniques of empirical scaling should therefore be used in conjunction with theoretical models.

This section describes the techniques of empirical scaling, applied at the global level, studying the scaling of the global energy confinement time τ_E , and at the local level, studying the heat flux across the profile of the plasma.

2.3.1 Global Analysis

Global transport analysis is the study of the scaling of the global energy confinement time τ_E with various plasma parameters.

The scaling of τ_E is determined as a regression against a set of plasma parameters $\{p_i\}$, usually taking the form of a power law:

$$\tau_E \propto p_0^{\gamma_0} p_1^{\gamma_1} \dots p_m^{\gamma_m} \quad (2.35)$$

based on the assumption that $\log(\tau_E) \propto \{p_i\}$ in a small range of parameter space around $\{p_i\}$. This assumption is based on the scale invariance of turbulence in neutral fluids, in which turbulent fluxes show log/linear dependencies on system parameters [15].

The most straightforward choice of parameters are those that can be directly controlled during an experiment: the so-called “engineering parameters” I_p, B, R, n_e , and P_{in} . There are several well known engineering scalings for τ_E in the literature, differing in the range of plasma parameters for which they are valid or for particular plasma confinement modes. “Neo-Alcator” scaling describes low density, ohmically heated plasmas, which show a nearly linear dependence on the density [34]:

$$\tau_E \propto n^1 R^2 a^1. \quad (2.36)$$

Confinement in higher density, auxiliary heated tokamaks is well described by a generic scaling

$$\tau_E \sim I_p / P_{in}^{1/2}, \quad (2.37)$$

referred to as *Goldston* scaling after Goldston [34]. Goldston scaling appears to be very robust, as it is observed on many tokamaks over a wide range in the engineering parameters. In 1982, a new plasma mode was discovered on ASDEX [35] in which the scaling with I_p and P_{in} was similar to Goldston scaling, but the confinement time

enhanced by a factor of up to three. This new mode, dubbed “H-Mode”, “H” standing for *high*, is characterized by a large drop in the D_α light emitted by the plasma edge, and an increase in both particle and energy confinement time. The non-H-Mode state was subsequently named “L-Mode”, “L” standing for *low*. H-Modes have been observed on most diverted tokamaks since 1982. Several other enhanced confinement modes have been discovered, such as P/PEP Mode and Supershots, but none are as ubiquitous as H-Mode.

The specific form of the L-Mode or H-Mode scaling depends on the choice of the dataset to which the regressions are applied and the methods followed in the regression. The most commonly employed form for L-Mode scaling is the ITER89-P scaling relation derived by Yushmanov et al. [36] for the design of ITER using data from 13 tokamaks:

$$\tau_E = 0.048 I_p^{0.85} P_{in}^{-0.5} B^{0.2} n_e^{0.1} R^{1.2} a^{0.3} \kappa^{0.5}, \quad (2.38)$$

where I_p is measured in MA, P_{in} is the total heating in MW, B is the toroidal field in T, n_e is the line averaged density in $10^{20}/\text{m}^3$, and R and a are the major and minor radii measured in m. The complement to ITER89-P for H-Mode scaling is the ITER93-H scaling law [37, 38]:

$$\tau_E = 0.036 I_p^{1.06} P_{in}^{-0.67} B^{0.32} n_e^{0.17} R^{1.90} a^{-0.11} \kappa^{0.66}. \quad (2.39)$$

Using these scalings, and the parameters listed in table 4.3 on page 98, ITER is predicted to have a confinement time of 1.5 s in L-Mode and 3.4 s in H-Mode. This corresponds to an “H-factor” of $H = 2.3$, where

$$H \equiv \frac{\tau_E(\text{H-Mode})}{\tau_E(\text{L-Mode})}. \quad (2.40)$$

Though global scaling is very successful at describing the confinement properties of existing tokamaks, it provides little insight into the physics behind the observed confinement modes. It does not explain *why* $\tau_E \sim I_p/P_{in}^{1/2}$, nor does it suggest an origin for H-Mode. More insight can be obtained from empirical methods by turning to local analysis.

2.3.2 Local Analysis

The ultimate question of interest for the design of a tokamak is: given a set of engineering parameters, how much fusion power will be produced? The fusion power depends on the density and temperature profiles, which in turn depend on the sources of particles and energy and on the transport. Local analysis studies the the relationship between the transport fluxes and the profiles of density and temperature. The central assumption of local analysis is that the transport at any given location in the

plasma is determined only by the plasma properties *at that location*; hence the name “local”.

In general, it is not possible to measure these fluxes everywhere within the plasma [39], so typically they are determined by rearranging the particle and energy balance equations (2.2) and (2.4) to solve for them based on other measurements:

$$\nabla \cdot \mathbf{\Gamma}_j = S_n - \frac{\partial n_j}{\partial t} \quad (2.41)$$

$$\nabla \cdot \mathbf{q}_j = -\frac{3}{2}n_j \frac{dT_j}{dt} + Q_j - \mathbf{P}_j : \nabla \mathbf{u}_j + S_E. \quad (2.42)$$

Assuming all the quantities on the right hand side of (2.41) and (2.42) can be measured, knowledge of the fluxes $\mathbf{\Gamma}$ and \mathbf{q} alone is not sufficient for characterizing the transport and scaling the results to future designs; they must be related to the local plasma properties. This is done following the thermodynamical methods used for studying transport in neutral fluids, as presented for example in [40]. The fluxes are considered to arise from “thermodynamic forces” driving the plasma to equilibrium. Their relationship to the forces, which are typically gradients in the plasma parameters, is through a matrix of *transport coefficients*. The physics of the transport is contained within these transport coefficients, and is assumed to be characteristic of the plasma physics, not tokamak-specific properties.

The transport matrix relating the fluxes $\mathbf{\Gamma}$ and \mathbf{q} , as well as the parallel current density J_{\parallel} , to the thermodynamic forces is:

$$\begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{q} \\ J_{\parallel} \\ \vdots \end{bmatrix} = \begin{bmatrix} -D & k_{12} & k_{13} \\ k_{21} & -n\chi & k_{23} \\ k_{31} & k_{32} & \sigma \\ \dots & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} \nabla n \\ \nabla T \\ \mathbf{E} \\ \vdots \end{bmatrix} \quad (2.43)$$

(2.43) is not meant to be a comprehensive description of the transport matrix, rather it is meant to demonstrate several of its features. The diagonal elements of the matrix are the particle diffusivity D , the thermal diffusivity χ , and the plasma electrical conductivity σ , which combine with ∇n , ∇T and \mathbf{E} to form Fick’s law of diffusion, Fourier’s law of heat conduction, and Ohm’s law respectively.

Identifying the physics behind the off-diagonal elements k_{ij} is not as straightforward. Example off-diagonal terms from neoclassical theory are *bootstrap* current [41] and the *Ware pinch* [42]. Bootstrap current is parallel current driven by a radial pressure gradient in the plasma, which would enter (2.43) through the coefficients k_{31} and k_{32} . The Ware pinch is an inward radial particle flux driven by the electric field, which would enter through the coefficient k_{13} . Onsager [43, 44] showed that the off-diagonal elements of the transport matrix for a neutral fluid are related through the so-called “Onsager symmetries”. This is also true for collisional transport in a plasma. The application of Onsager symmetry to turbulent transport is much more

complex [39].

While theory can provide forms of the transport coefficients in (2.43), in empirical transport analysis the coefficients are inferred from plasma measurements. Even using Onsager symmetry, the fact that there are more transport coefficients than equations in (2.43) means that not all the transport coefficients can be determined uniquely. Many of the off-diagonal terms are therefore ignored. The transport coefficients that are determined will then represent the cumulative effect of several thermodynamic forces. It is also important to note that for plasma physics, a description of the form (2.43) is ad hoc. Since the transport is driven predominantly by turbulence, it is not clear which thermodynamic forces are appropriate. For example, the heat flux may depend on n and T directly in addition to containing a diffusive part depending on ∇T . The transport coefficients also depend on the thermodynamic forces, so the system is nonlinear.

The Thermal Diffusivity, χ

For empirical analysis of local heat transport, the parameter of interest is the thermal diffusivity χ . Already introduced in the discussion of collisional transport, where it characterizes diffusive flow for length scales longer than the mean free path, χ is generically used to relate the gradient in the temperature profile to the heat flux crossing it. There is conflicting evidence about the diffusive nature of anomalous heat transport [45,46]. Despite this, χ is used as a matter of convention, primarily as a means of comparing local transport for intra- and inter-device studies. The comparisons take the form of empirical scalings of χ at particular radii, or examinations of the entire χ profile, with the basic understanding that the larger the value of χ , the stronger the transport. The calculation of χ from plasma measurements and the power balance equation is performed by an analysis code. The code used for this thesis, TRANSP [47], is described in the next chapter.

Local measurements of χ can be related to global measurements of τ_E assuming that (i) the conducted heat flux \mathbf{q} is the only power loss in the energy balance equation for each species, and (ii) \mathbf{q} is entirely diffusive, so that $\mathbf{q} = -n\chi\nabla T$. If these two assumptions are valid, then

$$\tau_E = \frac{a^2}{\langle\chi\rangle} \gamma_p, \quad (2.44)$$

where $\langle\chi\rangle$ is an effective diffusivity from every plasma particle species averaged over the plasma profile, and γ_p is a dimensionless quantity depending on the shapes of the density, χ , and input power density profiles. In situations where there are other significant sinks of power such as radiation, or if the conducted heat flux is not entirely diffusive, the relationship between χ and τ_E is more complicated than (2.44).

It has been observed on several tokamaks that the shape of the electron temperature profile appears to be independent of the input power density profile, depending

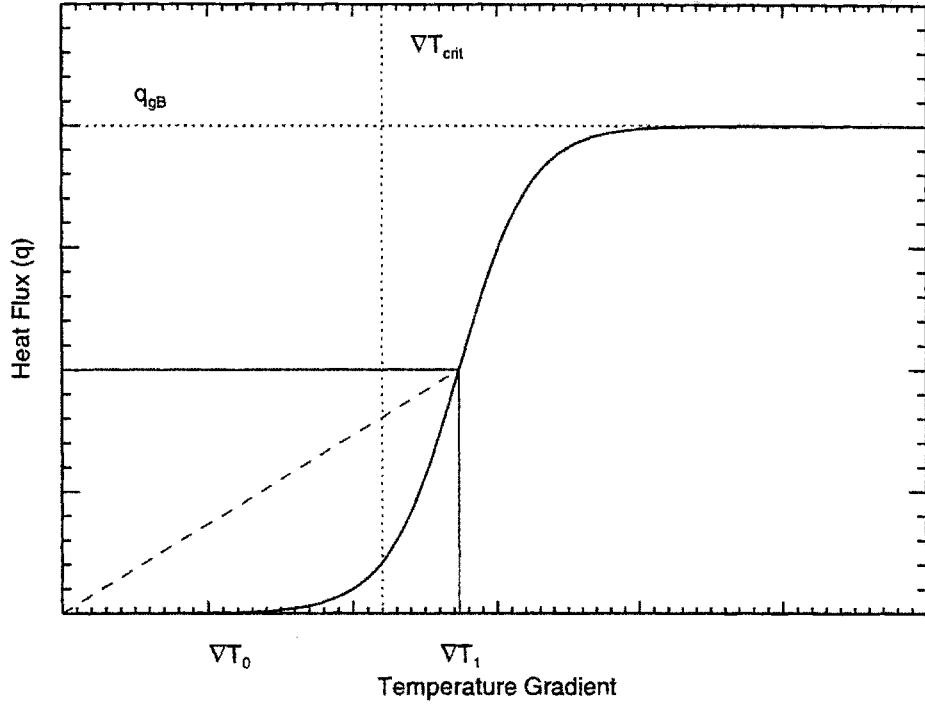


Figure 2.6: Marginal stability model of transport. At $\nabla T = \nabla T_{\text{crit}}$, q rises sharply. For a marginally stable plasma with $\nabla T = \nabla T_1 > \nabla T_{\text{crit}}$, the transport is better described by ∇T_{crit} and the slope $dq/d(\nabla T)$ than by χ , which is the slope of the dashed line.

only on the total amount of power deposited in the plasma [23]. This phenomenon, referred to as “profile resilience” [23, 48], calls into question the usefulness of χ as a parameter for comparison. The diffusive heat flux assumption gives

$$\nabla T = -\frac{\mathbf{q}}{n\chi} = -\frac{\int_0^r Q_{\text{in}}(r)dV}{n\chi}, \quad (2.45)$$

where Q_{in} is the input power density. If T_e is independent of the shape of Q_{in} , then either χ_e is a strong nonlinear function of ∇T or the transport is better described by a *marginal stability* model [23, 49]. In either case, χ_e is extremely sensitive to ∇T , so to the accuracy that it can be determined the uncertainty in χ_e is very large. Given this, it would be more fruitful in local analysis to study the scaling of ∇T and the temperature profile scale length $L_T \equiv T/\nabla T$ instead of χ .

Marginal stability in the context of plasma transport is explained as follows, with reference to figure 2.6, which plots the heat flux q versus ∇T for a marginal stability model. The description begins with a plasma whose energy balance is in equilibrium, so that the input power matches the power losses. The temperature profile in this

equilibrium is characterized by $\nabla T = \nabla T_0$, with a fixed edge temperature. Assume for simplicity that the power loss is entirely diffusive through \mathbf{q} . If additional input power is applied to the plasma, the heat flux changes only slightly, so that most of the power goes to increasing the plasma stored energy and thus ∇T . At some critical value, $\nabla T = \nabla T_{\text{crit}}$, a dramatic increase in the transport is triggered, which quickly removes energy from the plasma and keeps ∇T constant as the input power is increased further; this is the marginally stable point. The turbulence eventually saturates at its nonlinear level, able to conduct no more power than \mathbf{q}_{sat} . When this point is reached, the input power again goes to increase the plasma stored energy, and ∇T can increase.

For the marginal stability model in figure 2.6, the temperature profile resulting from a given input power is determined solely by the critical gradient ∇T_{crit} (and the edge temperature) for powers in the range $(\mathbf{q}_0, \mathbf{q}_1)$. $\chi = \mathbf{q}/n\nabla T$ varies over a large range while ∇T hardly changes, so it is not instructive as a parameterization of transport.

Chapter 3

TRANSP and Local Transport Analysis

TRANSP is the transport analysis code used in this thesis for local transport analysis. Its purpose is to calculate the local thermal transport coefficients χ_e , χ_i , and χ_{eff} as functions of space and time, given profile and equilibrium measurements from the plasma. In the process, TRANSP must model some phenomena that impact the calculation of χ but cannot be measured. This chapter provides the details behind these calculations.

TRANSP models the plasma as toroidally symmetric, taking the density, temperature, and current density as flux surface quantities. This reduces the geometry of the situation to the radial dimension only. The mapping of flux surface to spatial coordinates is done using a built-in fixed boundary equilibrium solver. Once this mapping is calculated, TRANSP proceeds to calculate the terms input to the energy balance equation, and then solves the energy balance equation for χ . Figure 3.1 (taken from [47]) presents a schematic diagram of the calculations TRANSP executes for each time step. Note that the calculation of χ rests on many other models and calculations; it is therefore subject to particularly large errors. We will begin by describing the energy balance equation, afterwards covering the various preliminary calculations on which it depends. As Alcator C-Mod is ICRF-heated, the SPRUCE and FPPRF codes are discussed in some detail. Also, as T_i is both crucial to energy balance calculations, and yet not well-measured on C-Mod, some discussion on how T_i was handled is warranted.

3.1 Calculating χ from Energy Balance

TRANSP assumes that the conducted heat flux \mathbf{q}_j in (2.4) is entirely diffusive, so that it can be written $\mathbf{q} = n\chi\nabla T$. The energy balance equation (2.4) is rearranged to

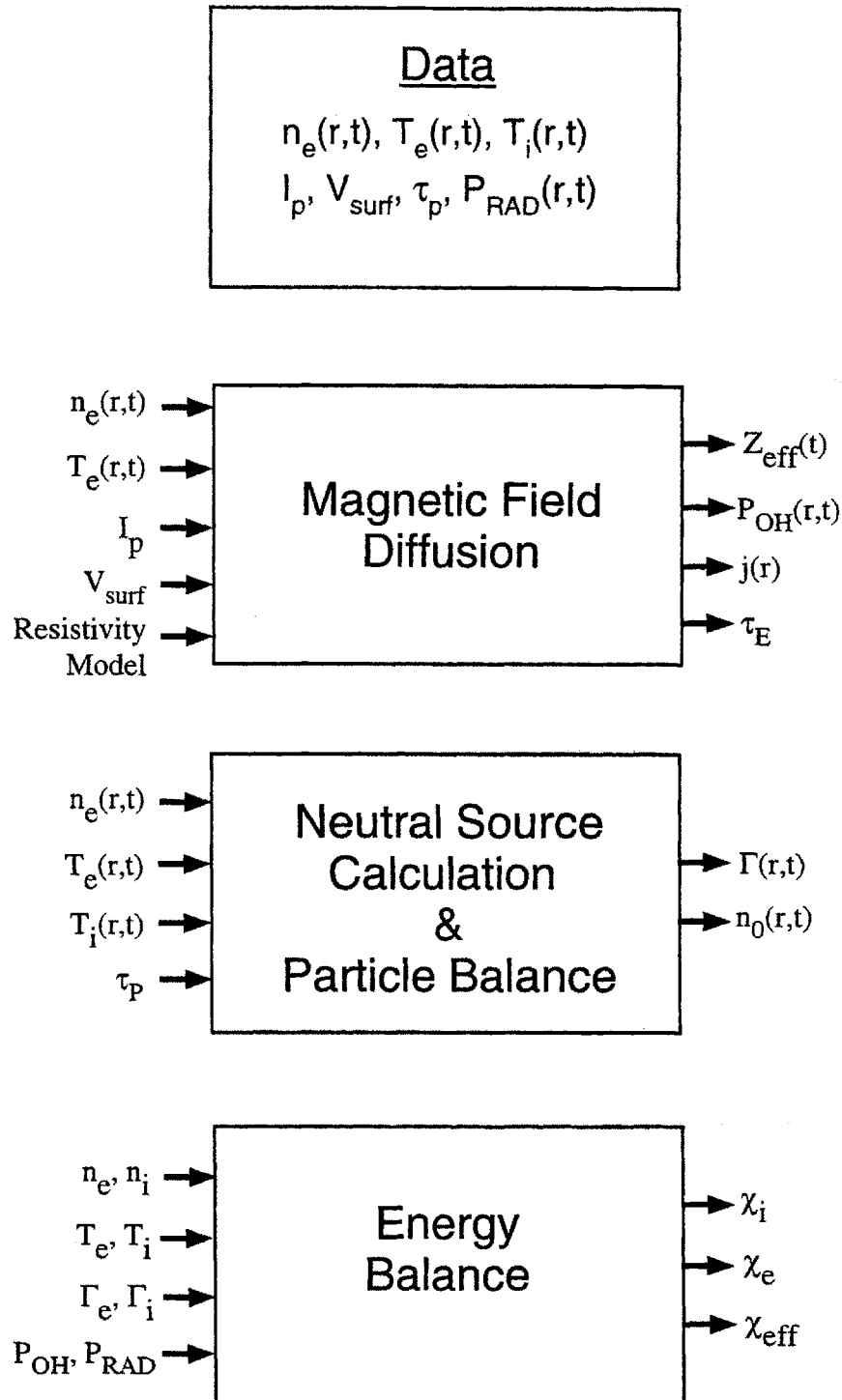


Figure 3.1: [Models, inputs, and outputs in TRANSP. TRANSP first solves the magnetic diffusion equation and particle balance before solving the energy balance equations for χ . Parameters on left are inputs to models; parameters on right are outputs.

solve for $\nabla \cdot \mathbf{q}$ as a function of the other terms:

$$\nabla \cdot \mathbf{q} = -\frac{3}{2}n_j \frac{dT_j}{dt} + Q_j - \mathbf{P}_j : \nabla \mathbf{u}_j + S_E, \quad (3.1)$$

and then χ_j is given by

$$\chi_j(r) = \frac{\int_0^r dV \nabla \cdot \mathbf{q}_j}{A_{\text{surf}} n_j \nabla T_j} \equiv \frac{P_{\text{cond}j}}{A_{\text{surf}} n_j \nabla T_j}. \quad (3.2)$$

TRANSP must therefore calculate or model each of the terms on the right hand side of (3.1) to calculate χ . How this is done is described in this section. First, note that the following convention is used for representing power and energy densities and their volume integrals. Energy densities, that is energies per unit volume, are denoted by W_j for species j , while power densities are denoted by Q_j . The volume integrals of these quantities are represented by U_j and P_j respectively, so that

$$U_j(r) = \int_0^r dV W_j \quad P_j(r) = \int_0^r dV Q_j. \quad (3.3)$$

The first term on the right hand side of (3.1) is rewritten to present the time rate of change of stored energy density explicitly:

$$\frac{3}{2}n_j \frac{dT_j}{dt} = \frac{\partial W_j}{\partial t} + \nabla \cdot (\mathbf{u}_j W_j). \quad (3.4)$$

The convected power density is identified in this term as $Q_{\text{conv}} \equiv \nabla \cdot (\mathbf{u}W)$. The second term, Q_j , is assumed to be classical, given by the flux surface averages of (2.16) and (2.17):

$$Q_j = \begin{cases} Q_{\text{OH}} - Q_{ei} & \text{for electrons,} \\ Q_{ei} & \text{for ions.} \end{cases} \quad (3.5)$$

The viscosity $\mathbf{\Pi}$ in the third term is neglected, so that

$$\mathbf{P}_j : \nabla \mathbf{u}_j = p_j \nabla \cdot \mathbf{u}_j. \quad (3.6)$$

This is identified as the compressive heating $Q_{\text{comp}j} \equiv -p_j \nabla \cdot \mathbf{u}_j$. The fourth term is the source term S_E , which actually includes both energy source and loss terms not covered by (2.1) using a Coulomb collision operator. For C-Mod, this means:

$$S_E = \begin{cases} Q_{\text{RF}} - Q_{\text{rad}} - Q_{\text{ion}} & \text{for electrons,} \\ Q_{\text{RF}} - Q_{cx} + Q_{i,s} & \text{for ions,} \end{cases} \quad (3.7)$$

Term	Source
U_j	$= \int_0^r dV W_j = \int_0^r dV \frac{3}{2} n_j T_j$
\mathbf{u}_j	from particle balance (see § 3.2)
$P_{\text{conv}j}$	uses \mathbf{u}_j from particle balance
P_{comp}	uses \mathbf{u}_j and flux reconstruction
P_{ei}	classical, using (2.16)
$P_{\text{RF}j}$	from SPRUCE and FPPRF (see § 3.5)
P_{oh}	from magnetic diffusion (since no q profile measurement)
P_{rad}	input from measurement
$P_{\text{ion}}, P_{\text{cx}}, P_{i,s}$	from cross section data and neutral density profile

Table 3.1: Terms in TRANSP energy balance calculation

where Q_{ion} is the power lost to ionization of neutrals, Q_{cx} is the charge exchange loss, and $Q_{i,s}$ is the net power gain from the ion particle source.

Putting (3.4)-(3.7) into (3.1) gives the form of the energy balance equation TRANSP solves to get the conducted power density for each species:¹

$$\nabla \cdot \mathbf{q}_e = -\frac{\partial W_e}{\partial t} - Q_{\text{conve}} - Q_{\text{comp}_e} - Q_{ei} + Q_{\text{RF}_e} + Q_{\text{OH}} - Q_{\text{rad}} - Q_{\text{ion}} \quad (3.8)$$

$$\nabla \cdot \mathbf{q}_i = -\frac{\partial W_i}{\partial t} - Q_{\text{conv}_i} - Q_{\text{comp}_i} + Q_{ei} + Q_{\text{RF}_i} - Q_{\text{cx}} + Q_{i,s}. \quad (3.9)$$

Volume integrating these terms gives the conducted powers used in (3.2) to calculate χ_e and χ_i :

$$P_{\text{conde}} = -\frac{\partial U_e}{\partial t} - P_{\text{conve}} + P_{\text{comp}_e} - P_{ei} + P_{\text{RF}_e} + P_{\text{oh}} - P_{\text{rad}} - P_{\text{ion}} \quad (3.10)$$

$$P_{\text{cond}_i} = -\frac{\partial U_i}{\partial t} - P_{\text{conv}_i} + P_{\text{comp}_i} + P_{ei} + P_{\text{RF}_i} - P_{\text{cx}} + P_{i,s}. \quad (3.11)$$

The sources for the terms in these equations are listed in table 3.1.

It is important to note that this calculation of χ is based on several assumptions. First, only the energy balance terms in table 3.1 are used to find the conducted power. If there are in fact other energy source or loss mechanisms in the plasma not represented in these equations, they will be counted as conducted power. Second, even if (3.10) and (3.11) were complete descriptions of power balance, so that the difference between the source and sink terms accounted for by TRANSP is in fact

¹This represents the best attempt of the author to derive the terms in the energy balance equation in TRANSP. TRANSP's treatment of the convected and compressive power terms Q_{conv} and Q_{comp} is not entirely clear. For more information, see [47,50]. Fortunately, these two terms are very small for C-Mod, so the fact that their derivation is not well understood is not consequential to the calculation of χ .

the conducted power density $\nabla \cdot \mathbf{q}$, it is not necessarily true that $\nabla \cdot \mathbf{q}$ is entirely diffusive. As discussed in § 2.3.2, it is possible that $\nabla \cdot \mathbf{q}$ also depends on ∇n , or involves additional flow-like terms $nT\mathbf{v}$ where \mathbf{v} has units of velocity.

3.1.1 Error Analysis

The uncertainty in TRANSP's calculation of χ is affected by many plasma measurements, as evidenced by the number of terms in (3.10) and (3.11). Treating the error in each of these terms as well as the other terms in (3.2) as independent, and for simplicity keeping only the terms that are expected to be large, gives the error in χ as:

$$\left(\frac{\delta\chi_j}{\chi_j}\right)^2 = \left(\frac{\delta P_j}{P_j}\right)^2 + \left(\frac{\delta n_j}{n_j}\right)^2 + \left(\frac{\delta \nabla T_j}{\nabla T_j}\right)^2, \quad (3.12)$$

where

$$(\delta P_j)^2 = \begin{cases} (\delta P_{\text{oh}})^2 + (\delta P_{\text{RF}})^2 + (\delta P_{\text{rad}})^2 + (\delta P_{ei})^2 & \text{for } j = e, \\ (\delta P_{\text{RF}})^2 + (\delta P_{ei})^2 & \text{for } j = i. \end{cases} \quad (3.13)$$

In fact, since the power densities in (3.10) and (3.11) depend on n_j and T_j , the parameters in (3.2) are not independent, so (3.12) is not a complete treatment of the errors in χ .

A thorough analytic calculation of $\delta\chi$ that takes into account the covariances in the parameters would be very difficult, as the power densities Q_{OH} , Q_{RF} , and Q_{ei} are calculated numerically by TRANSP. Since TRANSP does not have the capability of propagating errors through its calculations, the most straightforward way to estimate $\delta\chi$ is to compare the χ values obtained from scaling the input profiles by their error ranges to the nominal values. For example, if the errors in T_e and T_i are 10%, the profiles can be scaled up and down by this amount and fed to TRANSP in every combination. The resulting range in the calculated χ 's can then be taken as the error. Figures 3.2 and 3.3 plot χ_e and χ_i calculated for three situations: (1) the nominal situation, (2) $T_e \rightarrow 1.1T_e, T_i \rightarrow 0.9T_i$, and (3) $T_e \rightarrow 0.9T_e, T_i \rightarrow 1.1T_i$. [p] Over most of the profile, the difference in the two extreme calculations of χ_e is $\simeq 1.0$ m²/s. The error can be taken as half this range, or ± 0.5 m²/s, though the range is not symmetric about the nominal value. A similar analysis of χ_i gives $\delta\chi_i = 0.5$ m²/s as well.

Even this method is approximate, as the profile shapes were not adjusted, and only T_e and T_i were scaled. Scaling all these parameters would require many TRANSP runs, making error estimation tedious and time consuming. Since the calculated variation in χ depends on plasma parameters that differ from shot to shot, $\delta\chi$ must be calculated for every shot analyzed, providing all the more reason to limit the number of runs required. The method used for the analysis in this thesis concentrated on four inputs,

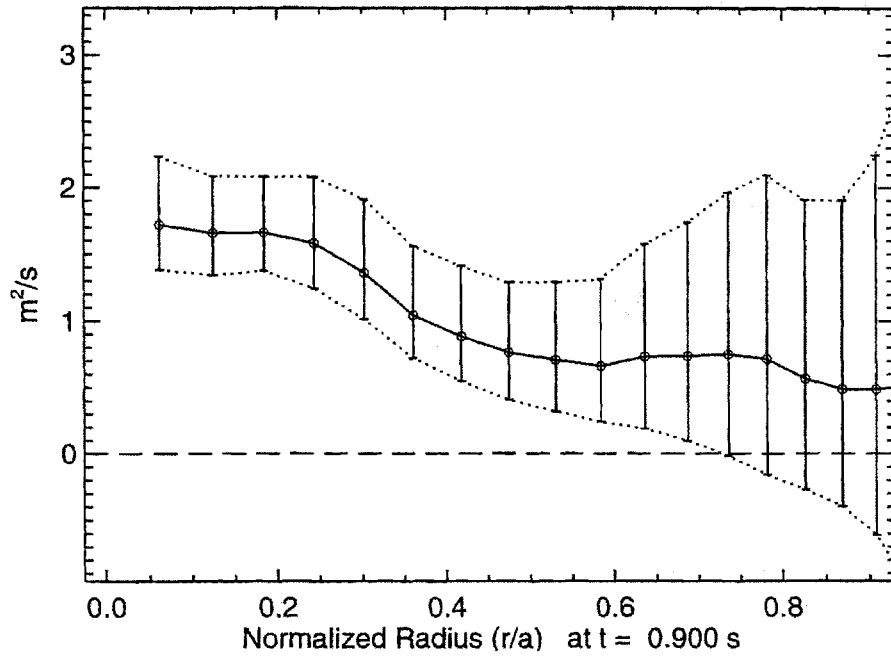


Figure 3.2: Error in χ_e from errors in T_e and T_i

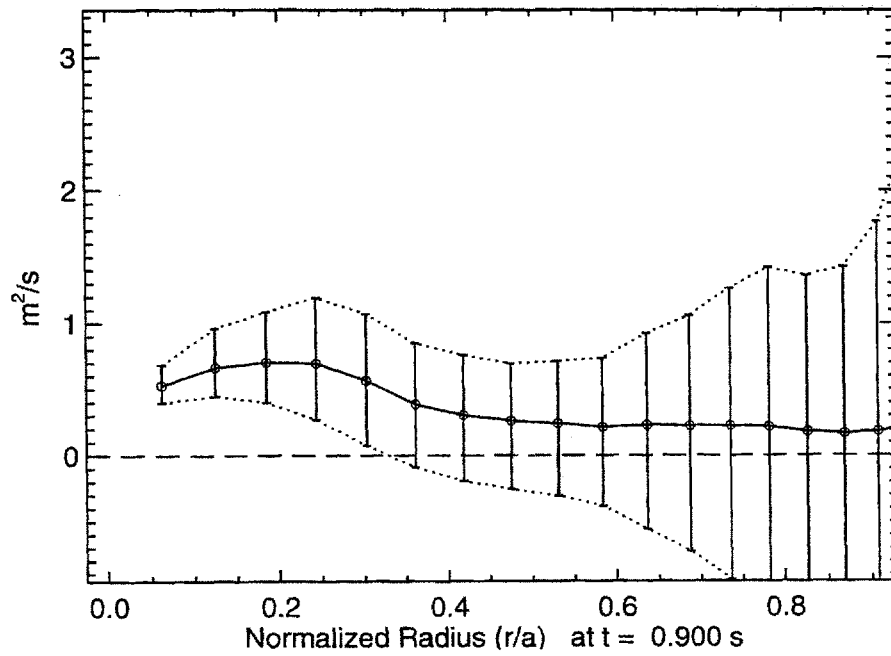


Figure 3.3: Errors in χ_i from errors in T_e and T_i

and only scaled their magnitudes, not their profile shapes. The four parameters scaled were T_e , T_i , n_e , and P_{rad} . T_e and T_i were scaled together in one run as the most significant contribution to $\delta\chi$ from δT_e and δT_i is through Q_{ei} (see § 3.1.2). n_e and P_{rad} were scaled individually, so that the total error in χ_j from the errors in the input profiles was taken to be:

$$\left(\frac{\delta\chi_j}{\chi_j}\right)^2 = \left(\frac{\delta\chi}{\chi}\right)_{T_e, T_i}^2 + \left(\frac{\delta\chi}{\chi}\right)_{n_e}^2 + \left(\frac{\delta\chi}{\chi}\right)_{P_{\text{rad}}}^2. \quad (3.14)$$

No attempt was made to determine the error in χ from varying the profile shapes, though a qualitative picture of the profile shape error is possible. Towards the center of the plasma, $\nabla T \rightarrow 0$, so χ will diverge. However, in the center of the plasma, the effects of sawteeth dominate transport. For the runs in this thesis, these effects were averaged by smoothing the data over a window longer than the sawtooth period. To study transport in the core as distinct from the sawteeth effects, careful analysis of the profiles between sawtooth crashes would be required. Near the plasma edge, the errors in the n_e and T_e profile measurements are especially large. The focus of local transport analysis is then in the *confinement* region, which is taken to be $r/a \in [0.25, 0.75]$ for the work in this thesis.

(3.14) is not the whole story. TRANSP will give the errors in χ_j from variations in its inputs based on the models it uses to calculate Q_{RF} and Q_{OH} . The accuracy of the models will also affect the error in χ_j . However, this accuracy cannot be judged, as the requisite measurements of the power deposition profiles for Q_{RF} , or the current and resistivity profiles for Q_{OH} , are not available. The calculations of Q_{OH} and Q_{RF} are discussed in more detail in § 3.3 and § 3.5 respectively.

3.1.2 Exchange Power Q_{ei}

At first glance it may be surprising that a $\pm 10\%$ change in the temperatures gives the large errors in χ_e and χ_i shown in figures 3.2 and 3.3. Closer inspection reveals that the large errors in χ_e and χ_i are the result of the errors in the electron-ion exchange power Q_{ei} . The contribution to $\delta\chi_j$ from δQ_{ei} is very large for C-Mod's high density plasmas, as shown by the following calculations.

The exchange power density is given by (2.16):

$$Q_{ei} = \frac{c_0 n^2}{T_e^{1/2}} \left(1 - \frac{T_i}{T_e}\right) \equiv Q_x (1 - \Theta), \quad (3.15)$$

where $\Theta \equiv T_i/T_e$. Its error is given by the usual rules of error propagation:

$$(\delta Q_{ei})^2 = 4Q_{ei}^2 \left(\frac{\delta n}{n}\right)^2 + \frac{Q_{ei}^2}{4} \left(\frac{\delta T_e}{T_e}\right)^2 + Q_x^2 (\delta\Theta)^2. \quad (3.16)$$

If n_e is sufficiently high, Q_x will be large even if Q_{ei} is small, so δQ_{ei} will still be large. The error in χ_j from Q_{ei} in this case is:

$$\delta\chi_j \sim \frac{\delta\Theta \int \delta Q_{ei} dV}{An_e \nabla T_e} = \frac{P_x \delta\Theta}{An_e \nabla T_e}. \quad (3.17)$$

To quantify this at $r/a = 0.5$, take $Z_{\text{eff}} = 1$ and $\Lambda = 15$. Then

$$Q_x = 1.9 \frac{n_{e20}^2}{T_{e\text{keV}}^{0.5}} \text{ MW/m}^3. \quad (3.18)$$

To estimate $P_x = \int Q_x dV$, replace n_e and T_e in (3.18) with their volume averages, and multiply the result by the volume at $r/a = 0.5$ ($\sim 0.25 \text{ m}^3$). With $\langle n_e \rangle = 2.0 \times 10^{20} / \text{m}^3$, $\langle T_e \rangle = 0.5 \text{ keV}$, this gives $P_x \simeq 2.7 \text{ MW}$. Taking $\nabla T_e \sim T_{e0}/a = 9 \text{ keV/m}$ gives

$$\delta\chi_j = \frac{2.7 \text{ MW}}{(3 \text{ m}^2)(2 \times 10^{20} / \text{m}^3)(9 \text{ keV/m})} \delta\Theta = 3 \delta\Theta \text{ m}^2/\text{s}. \quad (3.19)$$

If $\delta\Theta = 20\%$, the contribution to $\delta\chi$ from Q_{ei} is $0.6 \text{ m}^2/\text{s}$. This is close to the error observed by varying the temperature profiles and rerunning TRANSP.

3.1.3 One-Fluid Effective Diffusivity χ_{eff}

The one-fluid effective thermal diffusivity χ_{eff} is calculated by TRANSP as:

$$\chi_{\text{eff}} = \frac{n_e \nabla T_e \chi_e + n_i \nabla T_i \chi_i}{(n_e \nabla T_e + n_i \nabla T_i)} = \frac{P_{\text{cond}}}{A(n_e \nabla T_e + n_i \nabla T_i)}, \quad (3.20)$$

where

$$P_{\text{cond}} \equiv P_e + P_i = \int_0^r (Q_e + Q_i) dV \quad (3.21)$$

is the total conducted power within r . Since Q_{ei} cancels out in the calculation of P_{cond} , its error does not affect χ_{eff} . Also, for ICRF heated shots, P_{cond} depends on the total deposited power, so the systematic error in the species power distribution do not affect χ_{eff} . A systematic error in the total power deposition profile shape will affect χ_{eff} , but as long as the profile is centrally peaked χ_{eff} should be insensitive to these errors in the confinement region. The error in χ_{eff} is determined using (3.14), where the error in the " T_e, T_i " term is now much lower.

The fact that χ_{eff} is much more precise than χ_e or χ_i makes it a useful figure of merit for the net transport. However, it is important to remember that χ_{eff} represents the net transport from both species. Each individual species can in general have different transport properties.

3.2 Particle Balance

For the purposes of the research in this thesis, the only function of the particle balance calculation in TRANSP is to provide the species radial velocity profiles \mathbf{u}_j for use in the calculation of P_{conv} for energy balance. The full capability of the model, which allows multiple species mixing and pellets, is therefore not exploited.

TRANSP solves (2.4) for the radial component of $\mathbf{u} = \Gamma/n$:

$$u_{r,j}(r) = \frac{1}{rn_j} \int_0^r \left(S_{n,j} - \frac{\partial n_j}{\partial t} \right) r dr. \quad (3.22)$$

For a single ion species plasma, the inputs $n_e(r, t)$ and $Z_{\text{eff}}(r, t)$ are sufficient to determine $n_i(r, t)$ from quasineutrality. The source terms for the two species $S_{n,e}$ and $S_{n,i}$ are not measured; TRANSP calculates them using a neutral penetration model (FRANTIC [51]), with the total source rate determined by the particle confinement time τ_p , defined by

$$\int_0^a \nabla \cdot \Gamma dV = \oint \Gamma \cdot d\mathbf{S} \equiv \frac{N}{\tau_p}, \quad (3.23)$$

where N is the total particle inventory in the plasma. The total source rate for species j is then

$$S \equiv \int_0^a S_j dV = \frac{\partial N_j}{\partial t} + \frac{N_j}{\tau_{pj}}. \quad (3.24)$$

If τ_p is measured as a function of time, TRANSP allows it as an input. However, in most cases it is not, so TRANSP uses a fixed value input through the namelist. This means that the values of \mathbf{u} that it calculates are no more believable than the estimate of τ_p .

As C-Mod does not have neutral beams, the only particle sources in the core of the plasma are injected pellets and neutrals penetrating from the edge. Pellet injection was not used in the discharges studied for this thesis, and the neutral penetration is weak, so the source term is very small. In steady state, with $dn/dt = 0$, the radial component of \mathbf{u} is also small, so that the convected power in (3.10) and (3.11) is insignificant compared to the input power, and the inability to measure τ_p does not affect the uncertainty in χ . Figure 3.4 plots the electron convected power $P_{\text{conv},e}(r) = \int_0^r dV Q_{\text{conv},e}$ for three values of τ_p : 3 ms, 30 ms, and 300 ms. Compared to the power input to the electrons, P_{conv} is very small, except for near the edge where the neutral penetration is stronger, so χ is insensitive to P_{conv} over the region of interest $r/a \in (0.25, 0.75)$. The radial velocity profiles for these values of τ_p are plotted in figure 3.5.

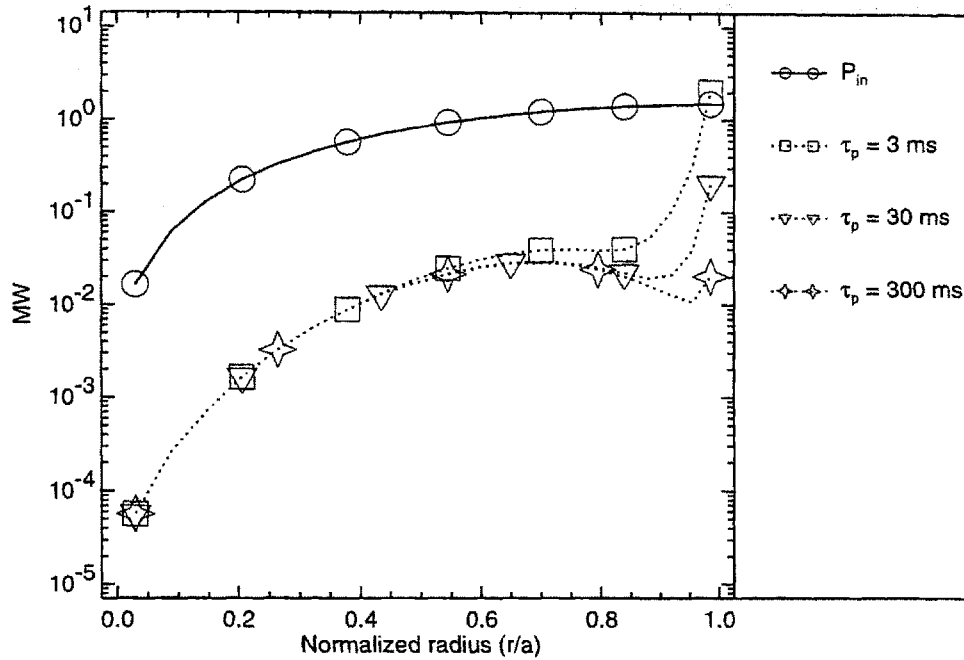


Figure 3.4: Electron convected power for three values of τ_p . P_{conv} is only significant at the edge. (C-Mod shot 950127015, $t = 0.9$ s)

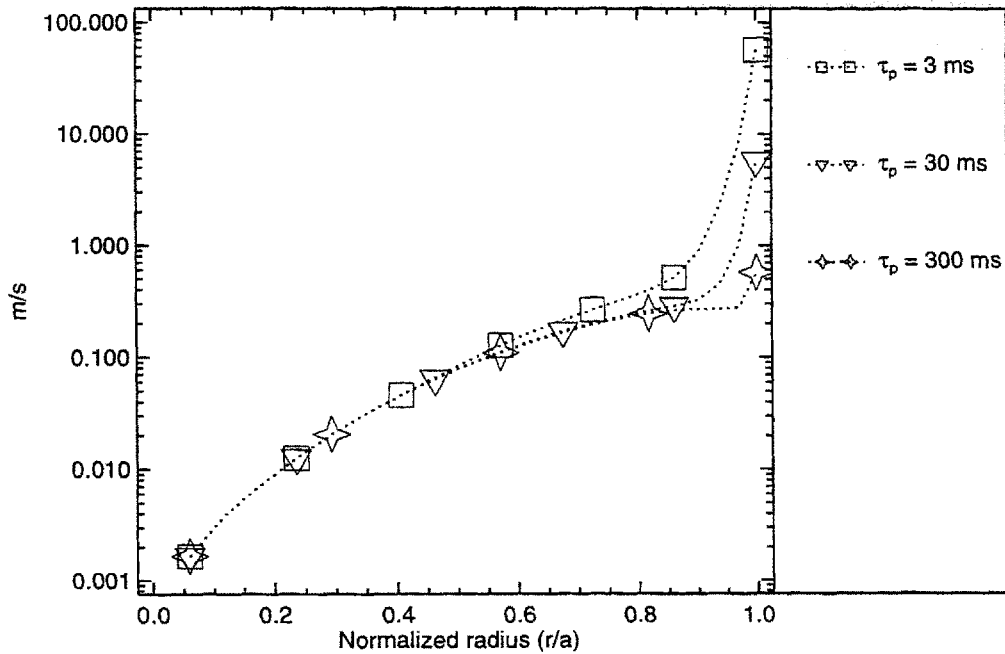


Figure 3.5: Electron radial velocity for three values of τ_p . (C-Mod shot 950127015, $t = 0.9$ s)

3.3 Magnetic Diffusion

The ohmic heating power density profile Q_{OH} is one of the terms in the energy balance equation (3.10). $Q_{\text{OH}} = \eta_{\parallel} J_{\parallel}^2$, so knowledge of the parallel resistivity η_{\parallel} and the parallel current density J_{\parallel} is required to calculate χ . q profile measurements well resolved in space and time can be supplied to TRANSP to calculate both η_{\parallel} and J_{\parallel} directly from measurement. Alternatively, when the q profile is not available as is the case for C-Mod, TRANSP solves the poloidal field diffusion equation for J_{\parallel} given a choice of model for η_{\parallel} [50]. The poloidal field diffusion equation is derived from Maxwell's equations in the low frequency limit [18, 50, 52]:²

$$\frac{\partial}{\partial t} \frac{\partial \psi}{\partial \rho} = \frac{\partial}{\partial \rho} \left(\frac{2\pi \langle \mathbf{E} \cdot \mathbf{B} \rangle}{R B_t \langle R^{-2} \rangle} \right), \quad (3.25)$$

where ψ is the poloidal flux within a flux surface, R is the local major radius, \mathbf{E} and \mathbf{B} are the electric and magnetic fields, B_t is the toroidal magnetic field, and ρ is the radial flux surface coordinate. The flux surface average of a quantity A is evaluated as

$$\langle A \rangle = \frac{\partial \rho}{\partial V} \oint d\mathbf{S} A \frac{1}{|\nabla \rho|}, \quad (3.26)$$

where the integral is over the flux surface, and V is the volume enclosed by the surface. The parallel electric field and parallel current density are related by

$$\langle \mathbf{E} \cdot \mathbf{B} \rangle = \eta_{\parallel} \langle (\mathbf{J} - \mathbf{J}_{DR}) \cdot \mathbf{B} \rangle, \quad (3.27)$$

where \mathbf{J}_{DR} is the total current density not driven by the electric field. The boundary conditions for the solution are:

$$I_p = 2\pi \int_0^a J_{\parallel}(r) r dr \quad (3.28)$$

$$V_{\text{surf}} = \left. \frac{2\pi \langle \mathbf{E} \cdot \mathbf{B} \rangle}{R B_t \langle R^{-2} \rangle} \right|_a. \quad (3.29)$$

TRANSP provides two choices for the resistivity model: Spitzer [53] and neoclassical [54]. These models will be discussed in more detail in § 3.3.1; for the moment it is sufficient to state that both are functions of n_e , T_e , and Z_{eff} . Recall that on C-Mod, the Z_{eff} profile is not well measured, so it is assumed to be flat for the purposes of conducting TRANSP runs. Since neither resistivity model is necessarily an accurate description of the plasma, and the Z_{eff} profile is in general not flat, it is not usually possible to satisfy both (3.28) and (3.29) simultaneously. In TRANSP the boundary

²TRANSP actually solves for the magnetic winding number $\iota = 1/q$ for an axisymmetric plasma, but this can be shown to be equivalent to (3.25) [50].

condition for I_p (3.28) is always met. For the surface voltage boundary condition (3.29), there are two options:

- treat Z_{eff} in η_{\parallel} as a free parameter adjusted to satisfy (3.29), or
- use the measured Z_{eff} in η_{\parallel} to calculate V_{surf} unconstrained by (3.29).

In the former case, the calculated Z_{eff} , referred to as the *magnetic* Z_{eff} , can be compared to the measured Z_{eff} to judge the accuracy of the model choice; in the latter, the calculated V_{surf} can be compared to the measured V_{surf} .

The typical course at C-Mod is to use the V_{surf} boundary condition (3.29) and treat Z_{eff} as a free parameter. This is because for C-Mod the calculation of V_{surf} from the free boundary equilibrium code EFIT [55–58] is subject to less uncertainty than the Z_{eff} measurement. Also, for global scaling it is important to get the total ohmic power as accurately calculated as possible; which implies that the best equation to use is $P_{\text{oh}} = I_p \times V_{\text{surf}}$, since the resistivity profile is not measured. The choice of resistivity model is addressed in the following section.

Note that when running with the neoclassical model, TRANSP's model of the bootstrap current was also turned on for consistency, as it is also a neoclassical effect. It was observed to be small compared to the ohmic current except near the edge in H-Mode where ∇p was large.

3.3.1 Resistivity model

Spitzer resistivity is given by [53]:

$$\eta_{\parallel}^{\text{SP}} = \frac{m_e Z_{\text{eff}} \nu_{ee}}{n_e e^2} \left(\frac{1 + 1.198 Z_{\text{eff}} + 0.222 Z_{\text{eff}}^2}{1 + 2.966 Z_{\text{eff}} + 0.753 Z_{\text{eff}}^2} \right), \quad (3.30)$$

where ν_{ee} is the electron-electron collision frequency [17] $\nu_{ee} \simeq 2/\tau_e$ and τ_e is the electron collision time defined by (2.11).

Banana trapping of the electrons reduces the plasma conductivity, so the neoclassical resistivity [54] is higher:

$$\eta_{\parallel}^{\text{NC}} = \eta_{\parallel}^{\text{SP}} \left(1 - \frac{f_t}{1 + \xi \nu_{*e}} \right)^{-1} \left(1 - \frac{c_R f_t}{1 + \xi \nu_{*e}} \right)^{-1}, \quad (3.31)$$

where f_t is the trapped particle fraction, ν_{*e} is the electron collisionality (2.30) defined in § 2.2.2, and

$$c_R = \frac{0.56 \ 3.0 - Z_{\text{eff}}}{Z_{\text{eff}} \ 3.0 + Z_{\text{eff}}} \quad (3.32)$$

$$\xi = 0.58 + 0.20 Z_{\text{eff}}. \quad (3.33)$$

Note that for a highly collisional plasma ($\nu^*e \gg 1$), $\eta_{\parallel}^{\text{NC}} \rightarrow \eta_{\parallel}^{\text{SP}}$, while for a collisionless plasma ($\nu^*e \ll 1$), $\eta_{\parallel}^{\text{NC}} \simeq \eta_{\parallel}^{\text{SP}}(1 - f_t)^{-2}$.

Estimating the volume average ν^*e for C-Mod using (2.30) gives $\nu^* \sim 0.5$. With $Z_{\text{eff}} = 1.5$, $\xi \simeq 1$ and $c_R \simeq .1$. In the large aspect ratio limit, $f_t = \sqrt{2\epsilon} = 0.8$, giving $\eta_{\parallel}^{\text{NC}} \simeq 2\eta_{\parallel}^{\text{SP}}$. There should therefore be a discernable difference between the magnetic Z_{eff} calculated from the two models.

The two models were tested on two shots by fixing V_{surf} and comparing the magnetic Z_{eff} to the measured³ Z_{eff} . Both shots were RF-heated H-Modes with $B_t = 5.3$ T and $I_p = 1.0$ MA.

The first shot, 960116024, has $\langle n_e \rangle$ ranging from 1.0 to $3.0 \times 10^{20}/\text{m}^3$. Figure 3.6(a) compares the magnetic Z_{eff} calculated from each of the two models to the measured value. While there is not perfect agreement, it is clear that the neoclassical resistivity model does a better job matching Z_{eff} . The q profiles calculated using the two models are also different, as evidenced by the time history of q_0 plotted in figure 3.6(b). Here there is no measurement to compare with, so it is not clear which model is more accurate. On the other hand, q_0 drops to 1 sooner for the neoclassical model at $t \simeq 0.3$ s, versus $t \simeq 0.55$ s for the Spitzer model. The fact that sawteeth were observed as early as $t = 0.2$ s is one more observation in favor of the neoclassical model for this shot, although this time is during the current rampup when MHD effects can drive anomalous current penetration, so the sawtooth onset time is not a definitive measure of the resistivity model's success. Notice that q_0 calculated from the neoclassical model drops as low as 0.5, as the sawtooth model in TRANSP was not used for the calculations shown here. Using the sawtooth model would keep q_0 closer to 1.0, but does not change the calculated magnetic Z_{eff} values.

Regardless of the differences in q and Z_{eff} , figure 3.6(c) shows that the total ohmic power is the same for both models, as expected since V_{surf} is fixed. What is really of concern is how the different models affect the calculation of χ ; this is shown in figure 3.6(d), which plots $\chi_{\text{eff}}(r/a = 0.5)$ as a function of time. There are slight differences in χ_{eff} after $t=0.6$ s, when the RF heating turns on. This can be explained by the dependence of the RF power deposition profile on the magnetic flux surface geometry. The differences are small enough to conclude that the choice of resistivity model does not affect χ_{eff} significantly for this shot.

For the second shot examined, 960130028, the comparison of the magnetic Z_{eff} calculations to the measured value in figure 3.7(a) indicates that the resistivity is better described by the Spitzer model. On the other hand, the neoclassical model appears better from the point of view of the sawtooth onset time. q_0 calculated from the Spitzer model does not drop below 1.0 until $t \simeq 0.5$ s, while the shot began sawtoothing at $t \simeq 0.2$ s. Again, even though the magnetic Z_{eff} and the calculated q profiles are different, the calculated total ohmic power P_{oh} is the same for both models as V_{surf} was fixed, and $\chi_{\text{eff}}(r/a = 0.5)$ was mostly unaffected by the choice of

³The Z_{eff} measurement used for comparison was taken from the node \Z67_YAG.TCI.

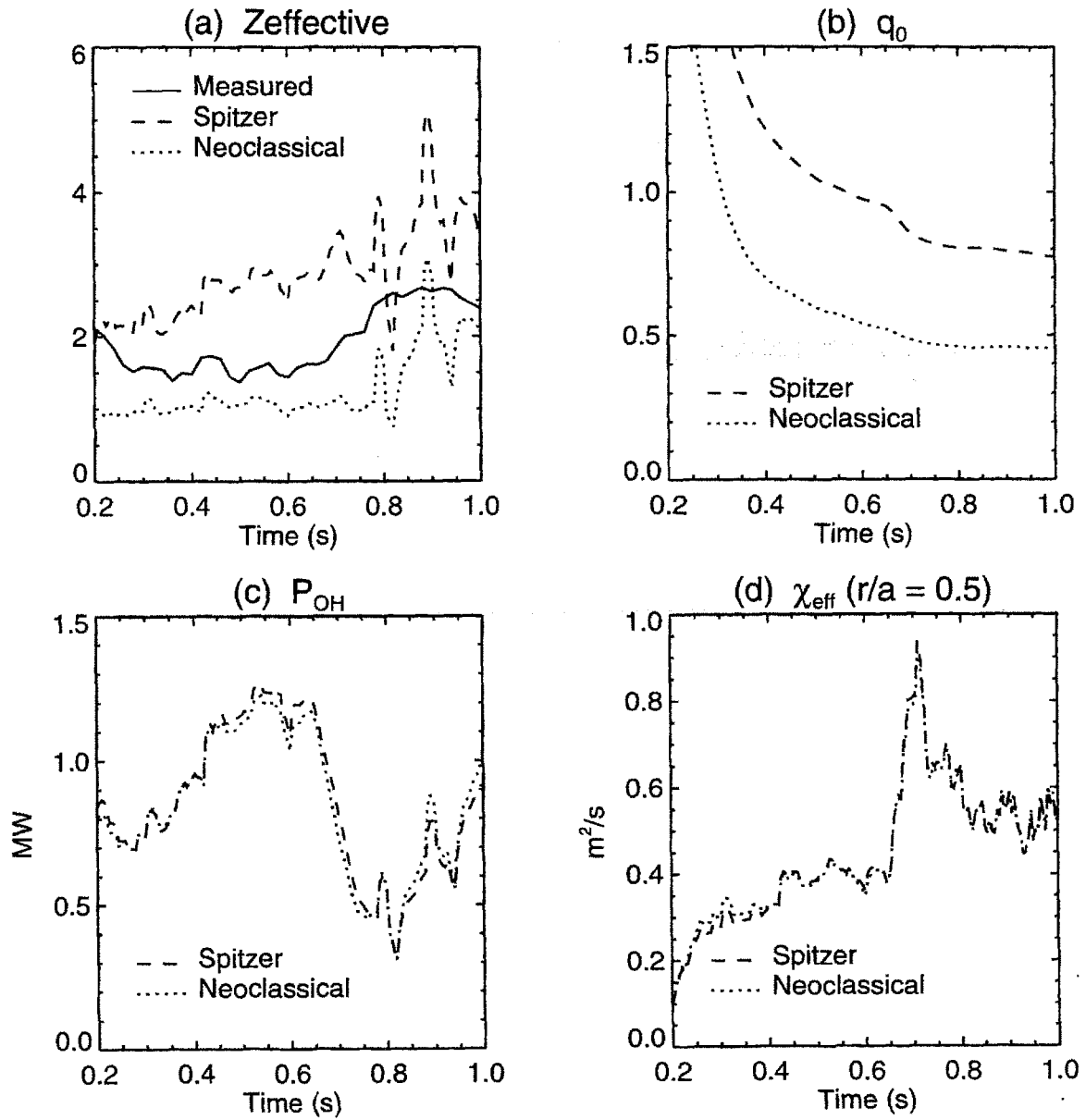


Figure 3.6: Comparison of Spitzer and neoclassical resistivity models for shot 960116024.

resistivity model as shown in figures 3.7(c) and 3.7(d) respectively.

Other than shifts in the RF power deposition profile caused by differing q profiles, χ_{eff} was unaffected by the choice of resistivity model for the two shots studied as long as the V_{surf} boundary condition (3.29) was met. Determining the resistivity model which best describes C-Mod plasmas would require better Z_{eff} profile measurements and a systematic study as done for TFTR [52]. Measurements of the q profile using the new diagnostic neutral beam would be invaluable in this endeavor. The neoclassical model was used in the TRANSP runs conducted for this thesis.

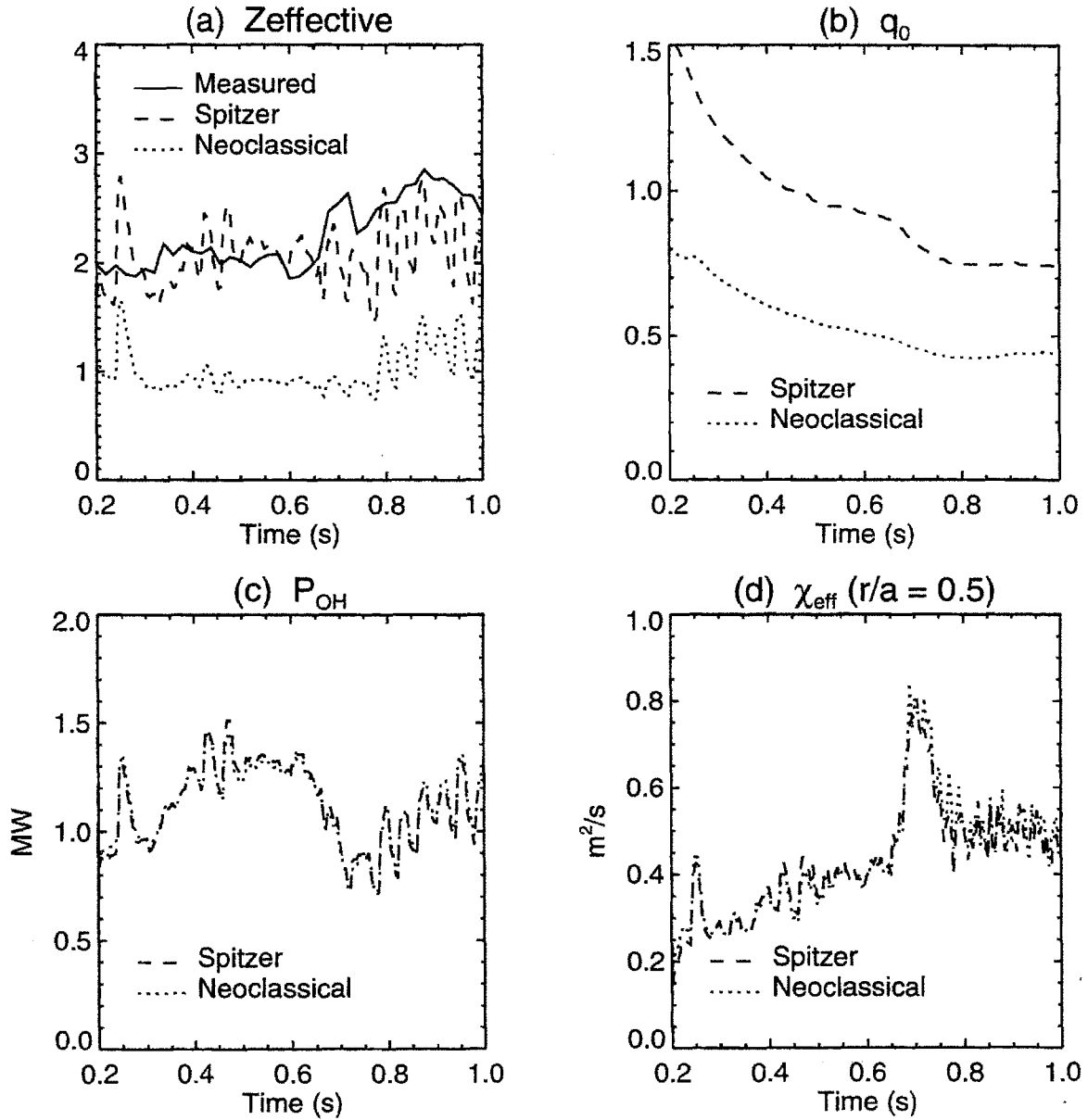


Figure 3.7: Comparison of Spitzer and neoclassical resistivity models for shot 960130028.

3.4 MHD Equilibrium

TRANSP uses the fixed boundary equilibrium code VMEC [59] to determine the magnetic geometry on which it solves the energy balance equation. The fixed boundary surface input to VMEC is taken from EFIT [55–58] for C-Mod shots. The two equilibrium codes complement each other, as EFIT is a free boundary code, but is usually run with the free functions determined by a least squares fit to the external magnetics measurements,⁴ whereas VMEC uses the profile measurements. Figure 3.8 plots the flux surface positions on the midplane for reconstructions from the two codes.

VMEC and TRANSP both use a fourier moments representation of the flux surfaces. § A.4.2 presents some of the details behind generating these moments. One factor worth noting here is that although the official release of TRANSP is designed to handle up-down asymmetric plasmas in all situations, the version currently used at C-Mod employs the old ICRF model that can only handle up-down symmetric representations.

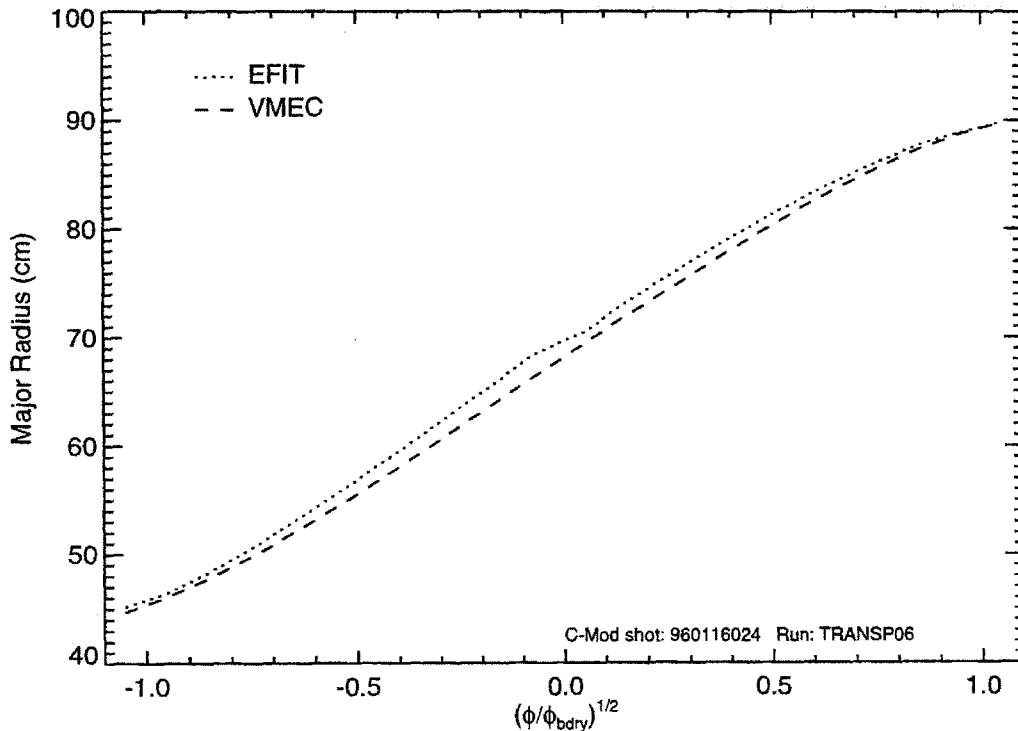


Figure 3.8: Flux surface mappings from EFIT and VMEC. Mapping is from $\sqrt{\hat{\phi}}$ to major radius at midplane. Negative $\hat{\phi}$ refers to inboard side of magnetic axis. The two mappings are within 2 cm of each other.

⁴EFIT can be configured to use the measured profile shapes, but this is not done for the EFIT run occurring automatically after every shot.

3.5 ICRF Power Deposition

Alcator C-Mod relies on ICRF heating⁵ as its only auxiliary heating method. Local transport analysis of C-Mod plasmas therefore requires an accurate estimate of the RF power deposition profiles for each species to solve the power balance equations for χ 's. Measurement of the direct electron RF heating is possible by studying the sawtooth reheat rate using the grating polychromator diagnostic in shots where mode conversion heating is strong [60]. It is also possible to determine the minority transfer to the electrons from the GPC measurements, though the slower time scale of this process makes untangling the RF heating from transport difficult. Measurement of the ion heating is currently not possible on C-Mod, as the current T_i diagnostics cannot resolve sawteeth to determine a reheat rate.

Since measurements of the power deposition profiles are not generally available, the profiles must be calculated using an RF modelling code. TRANSP includes the codes SPRUCE [61–63] and FPPRF [64] for this purpose. The former code calculates the wave propagation and damping on the plasma, while the latter models the high energy tail and its slowing down on the background species.⁶ These codes currently provide the best possible estimate of the RF power deposition profiles on C-Mod. Any systematic errors in the models contained within them will manifest themselves as errors in the calculation of χ .

There are two systematic errors of concern here. The first is the error in the relative heating of each species. This affects the χ_e and χ_i calculations; if TRANSP has too much power into the electrons, for example, the χ_e values will be overestimated, while the χ_i values underestimated. χ_{eff} is not affected by this error, as it depends only on the total power. Both χ_{eff} and χ_{e,χ_i} are affected by the second systematic error: an error in the power deposition profile shape. Measurements of the electron heating profile [60] suggest that the SPRUCE/FPPRF results are too peaked. This would cause an overestimation of χ , but only near the center, as the profiles are volume-integrated for the calculation of χ .

3.5.1 Code Physics

A complete description of the physics in FPPRF and SPRUCE is beyond the scope of this thesis. This section provides a summary of the code physics, and highlights features of interest for subsequent analysis; the interested reader is referred to [61–64] for more information.

SPRUCE solves the dispersion relation for ion cyclotron waves in the plasma given the wave frequency, k_{\parallel} , and total launched power, the plasma density, temperature, and shape, minority density, and the antenna and vacuum vessel locations. From

⁵For simpler notation, I will refer to ICRF heating as “RF” from now on.

⁶There is a standalone version of SPRUCE/FPPRF as well that is basically the same code as the TRANSP version.

the solution of the dispersion relation, the direct heating of the electrons, plasma ions, and minority ions can be calculated. SPRUCE assumes perfectly conducting walls, so that if single-pass absorption is low, the waves reflect off the walls and pass back through the plasma until they are absorbed. Rather than solving three coupled equations for \mathbf{E} , it solves a scalar equation for H_{\parallel} , and uses it to calculate \mathbf{E} . This vastly reduces the amount of CPU time required for a solution. The tradeoff for this reduction in CPU time is the potential for unresolvable numeric problems that can contaminate the results, owing to the fact that H_{\parallel} is singular at the resonance layer. See § 3.5.2 for more information on this problem and how it affects analysis of C-Mod plasmas.

Once the direct power absorption is determined, FPPRF solves the Fokker-Planck equation for the minority ion distribution function. The direct heating of the minority creates a high energy tail on its distribution function that slows down and heats the background plasma. The division of the slowing down power between the electrons and background ions is determined by the average energy of the minority tail relative to their temperatures; qualitatively, if $E_{\text{tail}} \gtrsim 10T_e$ more power will go to the electrons, while if $E_{\text{tail}} \lesssim 10T_e$ more will go to the ions.

The hotter the minority ions get, the more easily they absorb additional power. Sharp central peaking of the calculated power deposition profile is therefore possible, which is likely unphysical as sawteeth tend to flatten the minority ion temperature profile within the mixing radius. FPPRF includes an ad-hoc sawtooth mixing model for the ICRF minority ion tail for this reason.

3.5.2 Singularity Problem in SPRUCE

H_{\parallel} is numerically undefined at the resonance layer (*i.e.* it becomes 0/0). Though SPRUCE contains a patch that attempts to correct for this behavior close to the layer, if the geometry is such that there is poor resolution in this region, pathologically high field magnitudes can be calculated near it. The singularity occurs off-axis on the resonance layer, usually towards the top of the plasma. When mapped to the midplane, this manifests as a peak in the heating profile off-axis, usually at $r/a = 0.5$ – 0.6 or very close to the edge. The problem seems to be exacerbated by small, elongated plasmas; indeed it was first observed on C-Mod. In TFTR's circular geometry the problem is inconsequential, while for JET's large plasmas it has less of an effect [65].

In 1995 [63], SPRUCE was modified to allow up-down asymmetric geometries, and to include a complete formulation of the Bessel functions, replacing the small $k_{\perp}\rho_i$ expansion. In the process, the singularity problems were made much worse, so much so that the new code would not work on the vast majority of C-Mod plasmas. The problems experienced with the code on C-Mod led to a re-evaluation of the implementation of the RF physics modules within TRANSP. The pre-1995 version of SPRUCE, subsequently referred to as the “old” version, was reintroduced to the code, so that users now can choose which version of SPRUCE is used in their calculations.

In the process, the interface between TRANSP and the RF physics modules was standardized, to facilitate smooth integration of other full wave codes that could be run instead of SPRUCE. Plans are currently underway to integrate the code TORIC [66,67], which performs a full solution of the vector \mathbf{E} equation and therefore does not experience the singularity problem. Until a better full wave code is included, C-Mod will use the old version SPRUCE, and will make do with the singularity problem when it occurs.⁷

Shot 960130020 presents a “worst-case” scenario for the calculation of the field magnitudes and the direct absorption. Figure 3.9(a) plots the RF power density profiles for the different species using the old SPRUCE. Since the volume integrated power density is what enters into the calculation of χ , it is plotted in figure 3.9(b). Clearly visible in figure 3.9(a) are peaks at $r/a \simeq 0.5$ and at the edge. These peaks are evidence of the singularity problem. Even though the power density of these off-axis peaks appears low compared with the central peak, roughly 35% of the total power is calculated as being deposited at $r/a > 0.5$ because of the larger volume, as shown in figure 3.9(b). In reality, the amount of power deposited for $r/a > 0.5$ should be small [68], so the singularity problem introduces a systematic error in the volume-integrated power deposition profiles, which in turn affects χ . In this case, the error in P_{RF} is assessed as 35% for $r/a \leq 0.7$. Figures 3.9(c) and 3.9(d) present the same plots at the same time slice from the new version of SPRUCE. Here, the 100% of the power is calculated as being deposited at the edge, which is certainly not what occurs in reality. The resulting χ profiles are then completely wrong.

The singularity problem appears to be very sensitive to plasma conditions, fluctuating wildly in time as shown in figure 3.10. This plot shows the old code’s calculation of the power deposited within $r/a = 0.5$ as a function of time, that is $P_{\text{RF}}(r/a = 0.5) = \int_0^{0.5} Q_{\text{RF}} dV$. At one time slice, the power deposition profile can appear entirely reasonable, while at the next time step, which may be only a few milliseconds after, the profile can be completely wrong. Every RF-heated shot run through TRANSP must then be checked for this pathology. Fortunately, with the old code, the problem was never as severe as shown in figure 3.9(c), though if the off-axis heating was found to be significant, a larger error was assessed in the power deposition. The sawtooth mixing model in FPPRF was observed to aggravate the singularity problem, for reasons that were not readily apparent, so it was not used for the analysis in this thesis.

When the singularity problem was not severe so that at least some power was centrally deposited (see for example the profile in figure 3.9 obtained using the old SPRUCE), the spurious off-axis power was redistributed within $r/a = x_0$ to obtain a

⁷At time of writing, C-Mod had not yet received the latest version of TRANSP with the reintroduced old version of SPRUCE as a supported analysis option. Instead, a “kludged” version was being used, in which the old SPRUCE replaced the new version. The latest version of TRANSP will be obtained as soon as possible.

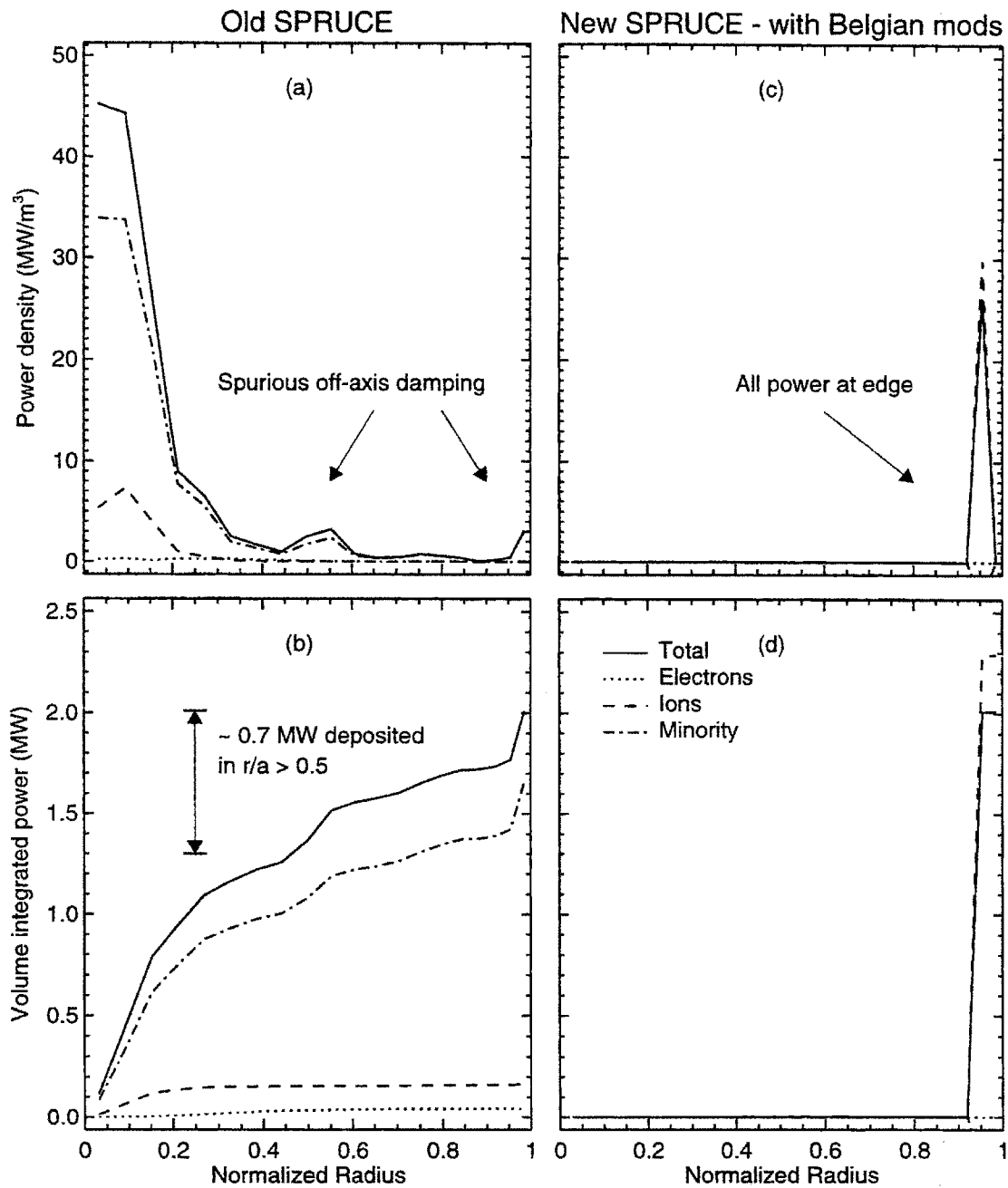


Figure 3.9: ICRF power deposition from old and new SPRUCE for shot 960130020 ($t = 0.95$ s). For old version, off-axis peaks in power deposition are the result of the singularity problem. About 0.7 MW of power is contained within the spurious peaks. The new version has all of the power deposited at the edge.

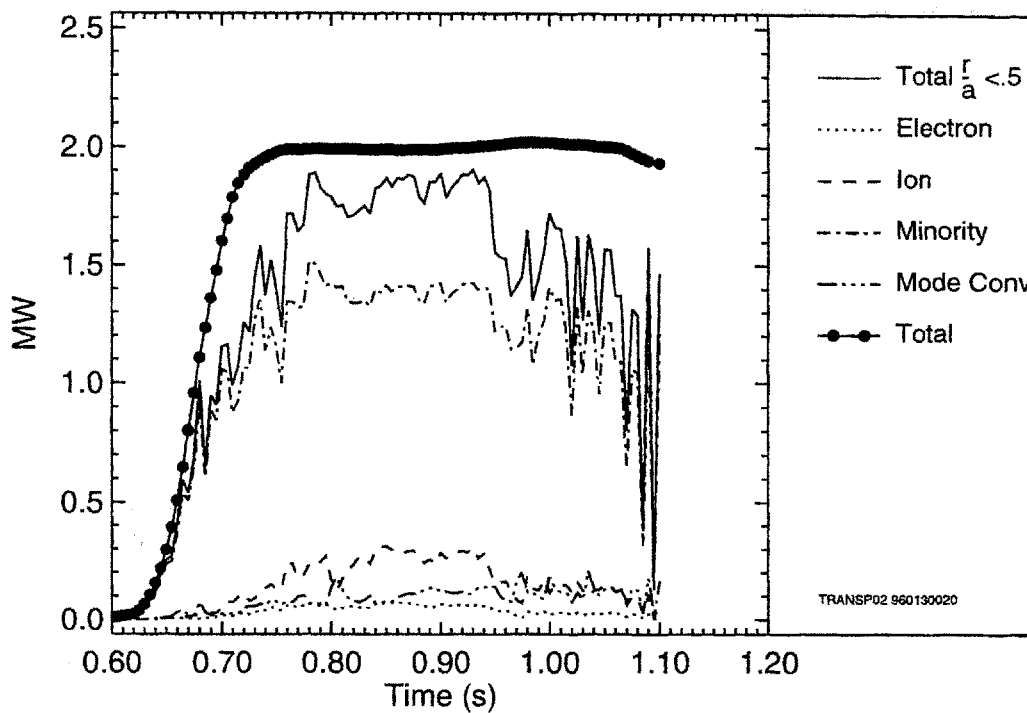


Figure 3.10: Time history of the RF power deposition within $r/a = 0.5$ as calculated by the old version of SPRUCE. The wild fluctuations after $t = 0.95$ s indicate that the off-axis damping problem is numerically unstable when time-evolved.

centrally peaked power deposition Q_{RF}' using the following algorithm:⁸

$$P_{\text{in}} = \int_0^{x_0} Q_{\text{RF}} dV \quad P_{\text{out}} = \int_{x_0}^1 Q_{\text{RF}} dV. \quad (3.34)$$

$$Q_{\text{RF}}' = \begin{cases} Q_{\text{RF}} \left(1 + \frac{P_{\text{out}}}{P_{\text{in}}}\right) & \text{for } r/a < x_0 \\ 0 & \text{for } r/a > x_0. \end{cases} \quad (3.35)$$

3.5.3 Total RF Power Absorption

Some of the power launched from the antenna is given up to parasitic losses in the antenna, the wall, or the plasma edge, so the amount of power absorbed by the plasma is in general less than the amount launched. Since SPRUCE does not model these losses, the absorbed power fraction must be measured externally and the power fed to SPRUCE corrected accordingly. This fraction is defined as:

$$f_{P_{\text{RF}}} = \frac{\text{Power absorbed}}{\text{Power launched}}. \quad (3.36)$$

$f_{P_{\text{RF}}}$ for these shots is measured using a “break-in-slope” technique on the stored energy. The measurements indicate that single pass absorption for fundamental H minority heating is large, so $f_{P_{\text{RF}}} \simeq 0.9 \pm 0.1$. For C-Mod’s other heating schemes, fundamental ^3He and 2nd harmonic H, the single pass absorption is much lower and more variable.

The break-in-slope technique takes the change in the time derivative of the stored energy after a sharp transition in the launched ICRF power as equal to the change in the absorbed power. In order for this assumption to be reasonable, the RF power transition must be sufficiently rapid so that transport effects with time scales on the order of τ_E are ignorable. Also, other plasma parameters should be constant over the transition period; in particular, the plasma should remain in the same confinement mode (either L or H-Mode), and the current and density should not change.

The stored energy measurement must have sufficient time resolution to allow determination of the change in its time derivative. Usually, only the diamagnetic stored energy U_{dia} is suitable for this technique, although EFIT can be run with a finer time grid around the transition time if necessary. Unfortunately, the U_{dia} measurement was not always properly calibrated, or gave unphysical energies.

⁸TRANSP users: Since TRANSP does not allow direct input of the ICRF power deposition, it was run once with the SPRUCE/FPPRF models turned on to calculate the (problematic) ICRF power deposition profile from which Q_{RF}' was determined. Another run was then conducted with the SPRUCE/FPPRF models off and the ICRF power input as ion Bernstein power so that it could be included in the energy balance calculation. Note that the energy stored in the fast ion tail was therefore not accounted for in the latter run.

Figure 3.11 presents the calculation of $f_{P_{RF}}$ for shot 960226035, a 2.6 T D₂ plasma with 2nd harmonic H minority heating. At the time of the power transition the plasma was in H-Mode, and remained in H-Mode for about 6 ms after the transition, long enough to get a measurement of the slope. U_{dia} was falling slightly before the transition, with a slope of -0.1 MJ/s. After the transition, U_{dia} decreased at a rate of -0.9 MJ/s. The change in slope at the transition was then

$$\Delta \frac{dU}{dt} = -0.9 - (-0.1) = -0.8 \text{ MJ/s}; \quad (3.37)$$

this is assumed to be the actual ICRF power absorption. The change in launched power was -1.2 MW, so the fractional power absorption was

$$f_{P_{RF}} = \frac{-0.8 \text{ MW}}{-1.2 \text{ MW}} = 0.66. \quad (3.38)$$

The error in $f_{P_{RF}}$ was assessed at ± 0.10 based on the uncertainty in determining dU/dt .

An example calculation of $f_{P_{RF}}$ for ³He minority heating at 8 T (shot 960213019) is presented in figure 3.12. This time the plasma was in L-Mode at the time of the transition. The result of the calculation is $f_{P_{RF}} = 0.4 \pm 0.1$.

If the calculation of $f_{P_{RF}}$ was not possible for a given shot, either because there was no clean transition in P_{RF} , or because the time resolution of the stored energy was too low, $f_{P_{RF}}$ was assumed to be the value in table 3.2 appropriate for the heating scheme, with an uncertainty of ± 0.1 . The assumed values are based on the calculations from several shots for each scheme [68]. In general it is observed that 2nd harmonic H is more efficient than ³He, though $f_{P_{RF}}$ for ³He varies over a wider range.

Field (T)	Minority Heating Scheme	$f_{P_{RF}}$
2.5	2 nd harmonic H	0.70
5.3	fundamental H	0.90
8.0	fundamental ³ He	0.50

Table 3.2: Assumed values of $f_{P_{RF}}$ for different heating schemes when measurement is not possible

3.5.4 Minority Fraction

The ICRF power deposition depends on the fraction of the plasma ions that are minority ions, the so-called ‘‘minority fraction’’. Assuming a negligible impurity density,

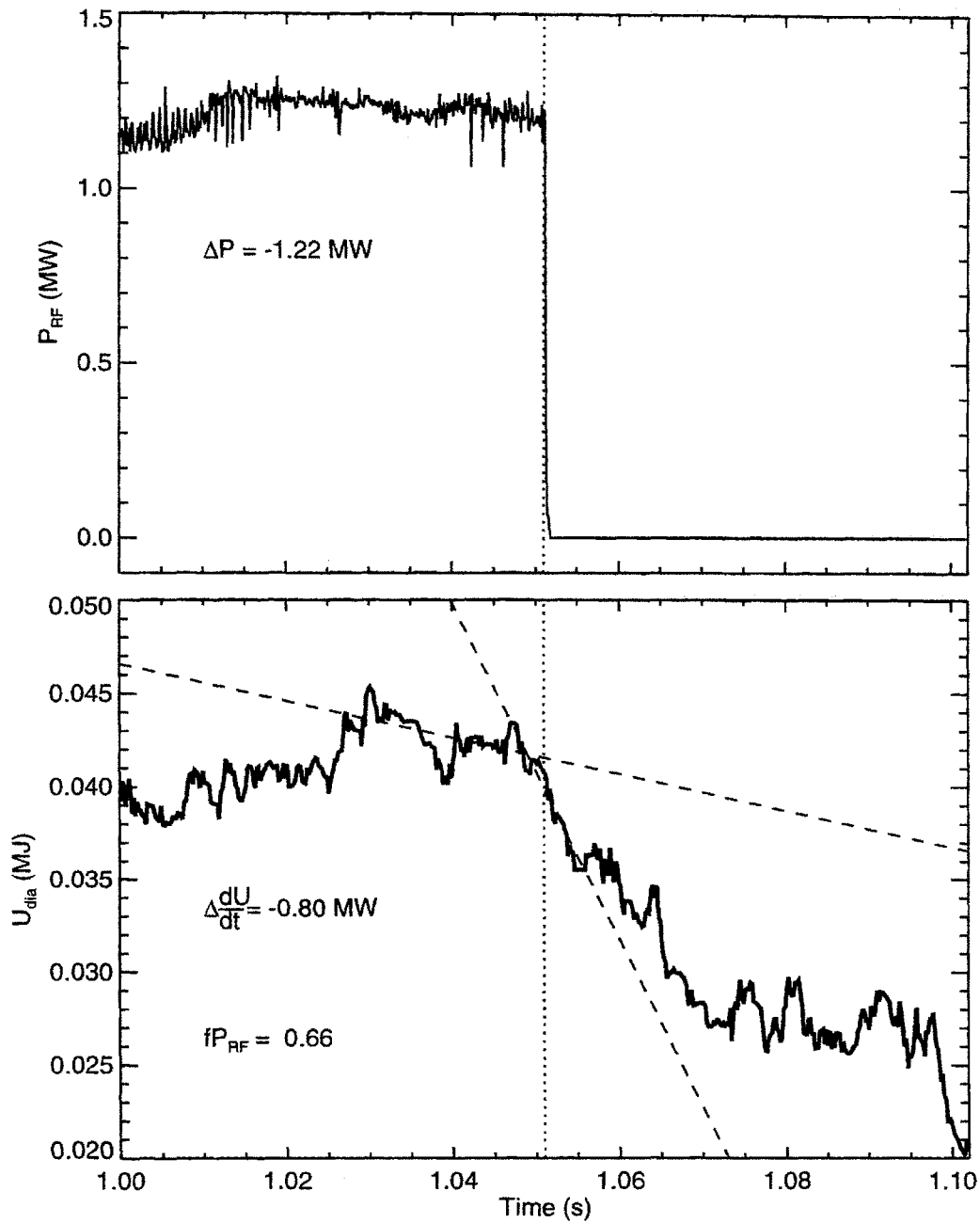


Figure 3.11: Fractional RF power absorption for 2nd harmonic H minority heating, shot 960226035. Top plot shows launched ICRF power vs. time; bottom plot shows corrected diamagnetic stored energy vs. time. Dashed lines show dU/dt before and after transition.

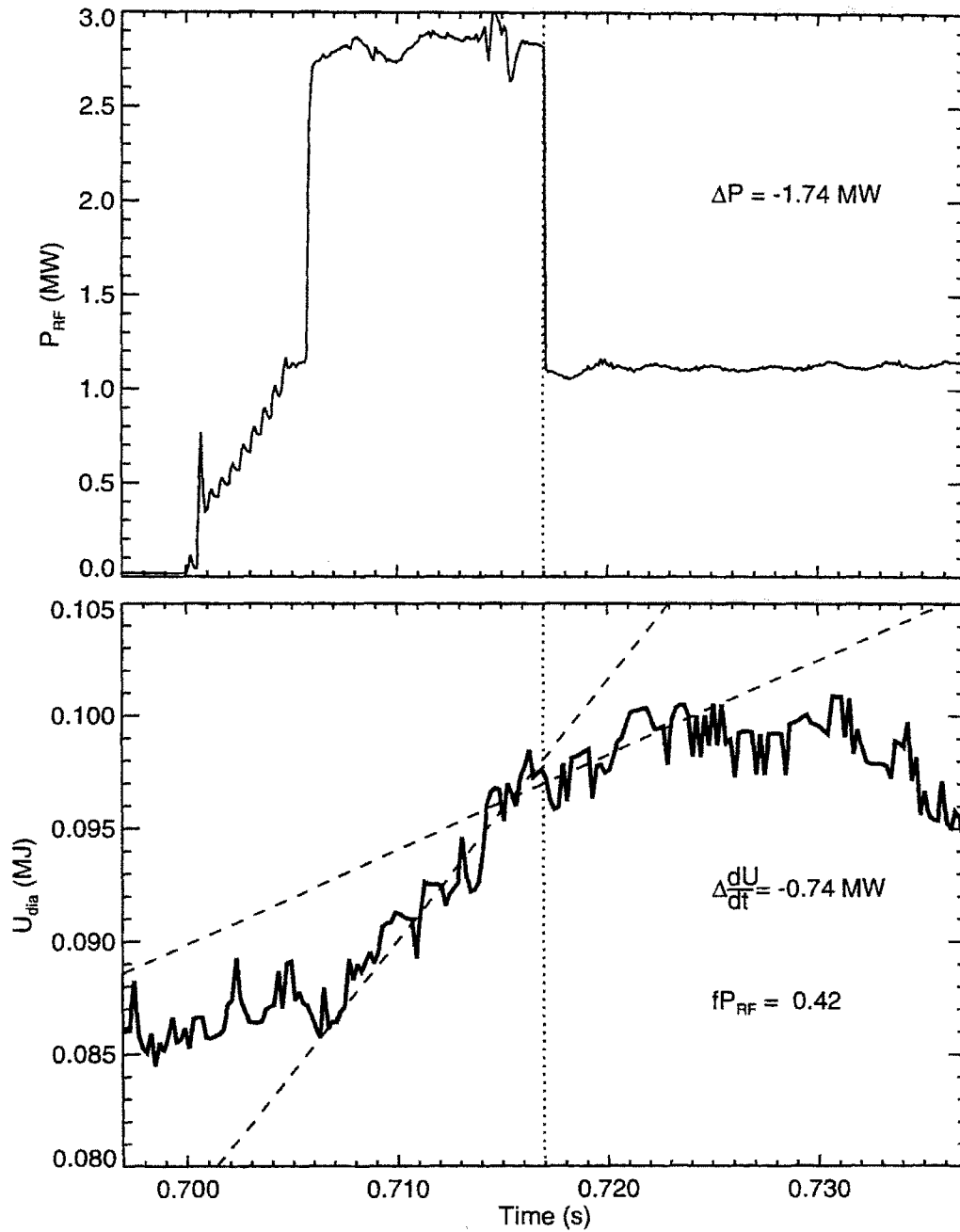


Figure 3.12: Fractional RF power absorption for ^3He minority heating, shot 960213019. Top plot shows launched ICRF power vs. time; bottom plot shows corrected diamagnetic stored energy vs. time. Dashed lines show dU/dt before and after transition.

this fraction is given by quasineutrality as

$$f_{\min} = \frac{n_{\min}}{n_{\min} + n_D}, \quad (3.39)$$

where n_D is the deuterium density and n_{\min} is the minority density. On C-Mod, f_{\min} for H minority heating is measured by charge exchange [69]. For ^3He it is estimated from the density rise due to ^3He gas puffing, spectroscopic measurements of He radiation, and Z_{eff} [13].

Charge exchange measurement of f_{\min} is not possible when the ICRF is on, so it is usually conducted early in the shot. There is no way of telling with certainty whether f_{\min} changes after the measurement is made as the shot evolves. Between the uncertainty in f_{\min} and the fact that measurements are not available for every shot, the sensitivity of the power deposition to f_{\min} was determined to estimate the error in χ from the uncertainty in f_{\min} . For each of two RF-heated shots, SPRUCE/FPPRF was run for five values of the minority fraction f_{\min} : 0.01, 0.02, 0.05, 0.075, and 0.10. The fraction of the total power f_{tot} deposited within $r/a = 0.5$, as well as the relative heating of the two species f_{rel} was examined as a function of f_{\min} . These were evaluated as follows:

$$f_{\text{tot}} = \frac{\int_0^{0.5a} dV (Q_{\text{RF},e} + Q_{\text{RF},i})}{\int_0^a dV (Q_{\text{RF},e} + Q_{\text{RF},i})} \quad f_{\text{rel},j} = \frac{\int_0^{0.5a} dV Q_{\text{RF},j}}{\int_0^{0.5a} dV (Q_{\text{RF},e} + Q_{\text{RF},i})}. \quad (3.40)$$

Note that Q_{RF} in (3.40) is the *total* heating of the species, which includes direct heating and minority slowing down. For the electrons, this also includes mode conversion power.

The first shot, 960214017, was a high density H-Mode ($\langle n_e \rangle \simeq 3.0 \times 10^{20}/\text{m}^3$). At this density a low minority tail energy is obtained over most of the profile, as shown in figure 3.13, so the heating of the electrons and ions should be roughly comparable. This is borne out by figure 3.14(a), which plots f_{tot} and f_{rel} from equation (3.40) versus f_{\min} for this shot. With the exception of $f_{\min} = 0.075$, the fraction of the total power deposited within $r/a = 0.5$ is between 0.80 and 0.90, so the error in χ_{eff} from the variation of Q_{RF} with $f_{\min} = 0.05 \pm 0.04$ is $\sim \pm 5\%$. The singularity problem in SPRUCE was particularly severe for $f_{\min} = 0.075$ for some reason, resulting in an anomalously low value of f_{tot} .

f_{rel} for each species ranges between 0.4 and 0.6, with the total relative heating of the electrons increasing with increasing f_{\min} . The fraction of the electron heating that is from mode conversion is shown with the dashed line; this amounts to 15-20% of the total electron heating. The error in χ_e and χ_i from the variation in f_{rel} with $f_{\min} = 0.05 \pm 0.04$ is therefore estimated to be $\pm 10\%$.

The second shot, 960126007, was a low density L-Mode ($\langle n_e \rangle \simeq 0.88 \times 10^{20}/\text{m}^3$). A much larger tail energy is obtained at low density as shown in figure 3.13, so the minority slows down predominantly on the electrons. The relative total power depo-

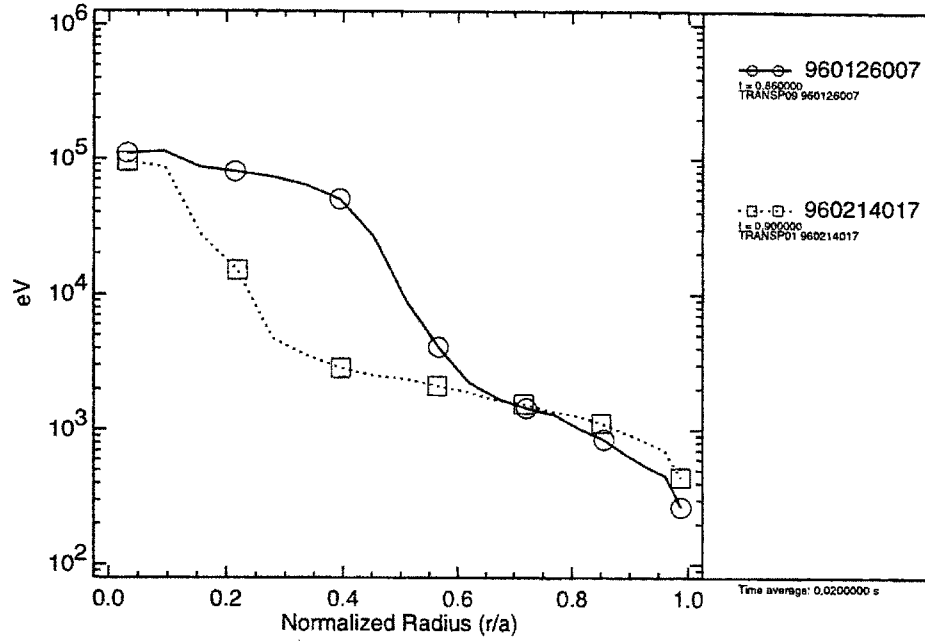


Figure 3.13: Tail energies for two shots used in f_{\min} scan

sition within $r/a = 0.5$ varies from 0.60 to 0.80 for the f_{\min} values studied, suggesting a $\pm 10\%$ uncertainty in χ_{eff} at $r/a = 0.5$. There is much less variation in f_{rel} for the two species, so the error in χ_e and χ_i from an uncertainty in f_{\min} is small here.

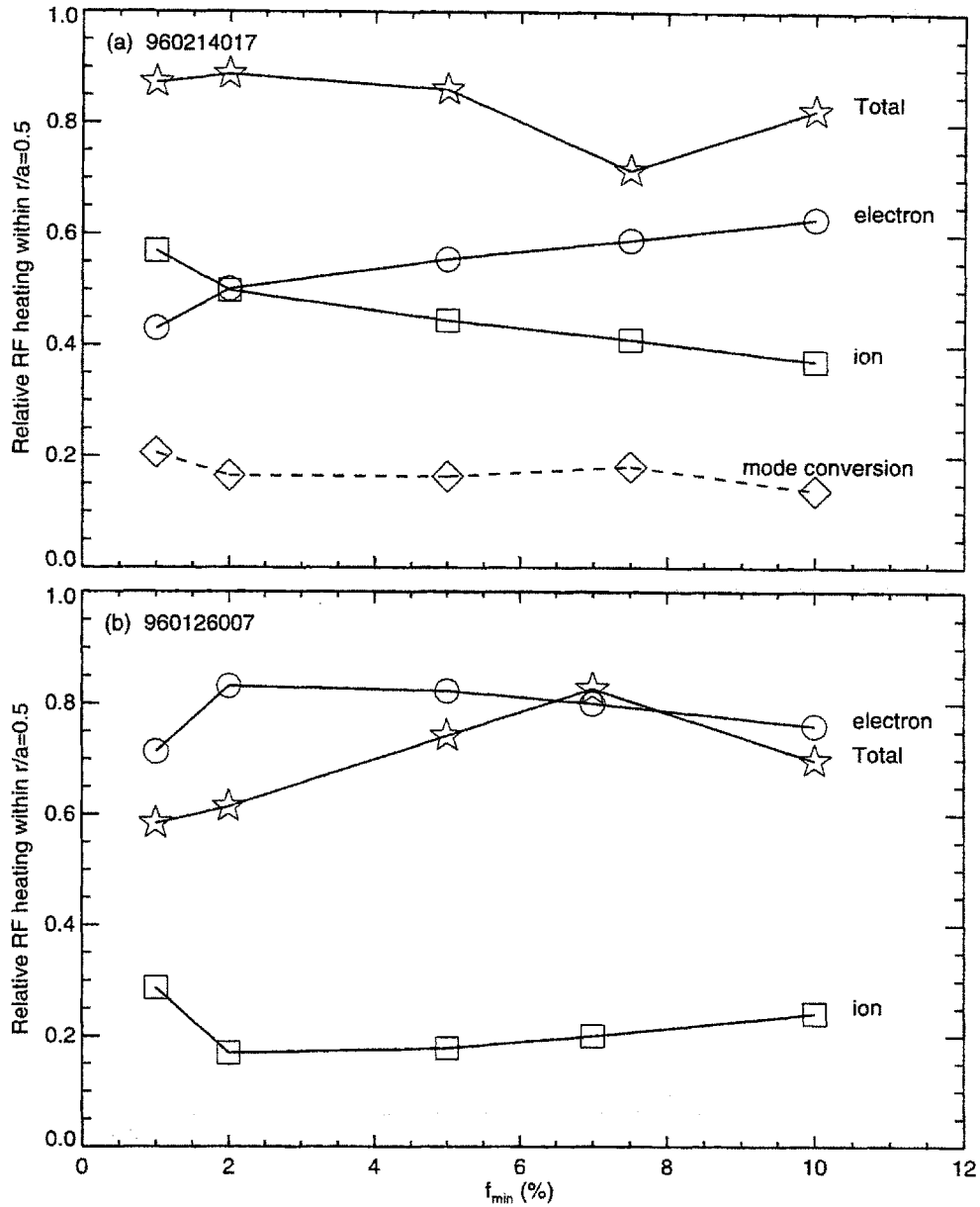


Figure 3.14: ICRF power absorption and relative heating as a function of minority concentration. Plotted are the ratio of the total heating within $r/a = 0.5$ to the total heating over the whole plasma (f_{tot}) and the relative heating of the two species (f_{rel}), as a function of f_{min} . Also shown for the high density shot (960214017) is the fraction of the electron heating from mode conversion. See (3.40) for the forms of f_{tot} and f_{min} .

3.6 Ion Temperature

3.6.1 Introduction

Power balance analysis in TRANSP requires temperature profile data for both electrons and ions. Profiles of the former are usually easy to come by, as the Michelson interferometer [70] and the Grating Polychromator [60] provide excellent profile data. Ion temperature profiles are harder to measure on Alcator C-Mod.

At the time the research for this thesis was conducted, the only available T_i profile diagnostic was a high resolution X-ray spectrometer system, known as “Hirex” [71]. Hirex measures the X-ray radiation from injected argon impurities. The doppler broadening of the various lines allows calculation of the impurity temperature, which is assumed to equal the temperature of the background ion species because the thermal equilibration time between the (helium-like) argon and deuterium ions is $\sim 25 \mu\text{s}$ in C-Mod’s high density plasmas. Each of its five spectrometers measures the emission along a chord through the plasma. To provide a local measurement for a profile, the emission is assumed to come from the innermost flux surface intersecting the chord, where the plasma is hottest. This assumption depends on which charge state of argon is being examined, and is not always valid. All of the measurements used for this thesis were conducted on He-like argon. If T_e is sufficiently high, this state will be fully stripped in the hottest regions of the plasma, so the peak of the measured emissivity profile will not be located at the closest approach of the chord to the magnetic axis.

There are several limitations of the Hirex diagnostic that make local transport analysis more difficult. First, the fact that the argon injected into the plasma for the Hirex measurements sometimes adversely affects the ability to obtain current ramp up (because of radiated power) has prevented its injection on many shots. Another problem is the diagnostic’s limited spatial coverage of the plasma. A maximum of five radial locations can be measured during a shot. A more detailed profile can be obtained by scanning the Hirex chords over several shots; however many run plans cannot afford, or are not afforded, the luxury of repeated shots. A final limitation of the Hirex diagnostic is its time resolution. Spectra are collected over a 50 ms period on average, but the collection time can be much longer for lower argon concentrations.

Given that T_i profiles from Hirex are not always available, and even when they are they may not be suitable for local transport analysis, an alternate means of providing a profile to TRANSP must be determined. There are two possibilities: (1) determining the profile using a model for the ion thermal diffusivity χ_i and a measurement on which to feedback the ion power balance; and (2) determining the profile from the electron temperature profile directly given certain assumptions about the transport. § 3.6.2 and § 3.6.3 describe the details behind these methods, and § 3.6.4 provides examples and comparisons of the T_i profiles. The latter method was used for the analysis in this thesis.

It should be stated up front that the profile resulting from this modelling and

feedback is not to be considered “truth” by any means. In particular, nothing about local ion transport, which depends on the profile shape, can be inferred. However, to accomplish any local transport analysis at all, some approximation of T_i is required. If done properly, the approximation can allow a reasonable estimate of the one-fluid effective thermal diffusivity χ_{eff} . To ensure that the profile is not unreasonable, the stored energy calculated from it and the T_e profile should be close to U_{mhd} , and the neutron rate calculated by TRANSP from the profile should be close to the measured rate.

C-Mod’s high density helps in this situation, as it shortens the electron-ion equilibration time τ_{ei} relative to the energy confinement time τ_E . The equilibration time is the characteristic time scale of temperature equilibration between the species based on classical collisional transport. For electrons and deuterons with $T_i \sim T_e$, this time is given by Braginskii [17] as:

$$\tau_{ei} = 2.1 \times 10^{-7} \frac{T^{\frac{3}{2}}}{n_{20}} \quad (3.41)$$

where the Coulomb logarithm is $\Lambda = 15$, T is measured in eV and n_{20} in $10^{20}/\text{m}^3$. The power exchanged between the electrons and ions is P_{ei} , which roughly scales as $P_{ei} \sim n \Delta T / \tau_{ei}$, where ΔT is the difference between the temperatures of the two species. This is compared to the input power, which roughly scales as $P_{\text{in}} \sim nT / \tau_E$. If τ_{ei} is small compared to τ_E , ΔT must be small compared to T to keep $P_{ei} < P_{\text{in}}$ and satisfy energy conservation. In the core of a C-Mod plasma with $n_{20} = 3.0$ and $T = 3$, $\tau_{ei} \simeq 10$ ms. Since the density profiles observed on C-Mod are quite flat, towards the edge, the $T^{\frac{3}{2}}$ dependence makes τ_{ei} much smaller. With $n_{20} = 2.0$ and $T = 2$ keV, $\tau_{ei} \simeq 9$ ms. Since the confinement times observed on C-Mod are typically $\tau_E \geq 20$ ms, τ_{ei} is usually less than τ_E over most of the profile.

3.6.2 Modelling χ_i

First coded in the early 1970’s, before T_i diagnostics were common, TRANSP provides several alternatives in cases where T_i profile data cannot be supplied. All these alternatives involve modelling the ion thermal diffusivity to calculate T_i from the ion power balance equation. With a model for χ_i used in the conducted heat flux $\mathbf{q}_i = n_i \chi_i \nabla T_i$, (3.9) is solved for T_i instead of χ_i .

χ_i is modeled as a scalar multiple of either χ_e or a neoclassical thermal diffusivity $\chi_{i,\text{nc}}$; that is:

$$\chi_i = c_\chi \times \begin{cases} \chi_e \\ \chi_{i,\text{nc}} \end{cases} \quad (3.42)$$

Feedback against either a measurement of the central ion temperature T_{i0} , or the measured neutron rate R_{DD} , is used to adjust the multiplier c_χ to obtain a self-

consistent solution.

The usual course for Alcator C-Mod is to choose χ_e to model χ_i , as it guarantees that $T_i \rightarrow T_e$ as $r \rightarrow a$ which would be expected given that T_e is dropping as $r \rightarrow a$, and the equilibration time $\tau_{ei} \propto 1/T_e^{3/2}$. Neoclassical models can result in a pathologically large difference between T_e and T_i near the edge, so the choice $\chi_i \propto \chi_{i,nc}$ is usually made only when testing the neoclassical models by comparing the modelled T_i to a measured profile.

On the other hand, if the T_e profile is flat or hollow, χ_e will not be well determined, and the feedback for T_i with $\chi_i \propto \chi_e$ can run amok. A flat profile gives a singular χ_e while hollow profiles can have $\chi_e < 0$. Both these results would break the feedback loop for numeric stability reasons, so TRANSP rails the multiplier c_χ at a maximum or minimum value. Smoothing and forcing a monotonic T_e profile will reduce this problem, but will not eliminate it.

3.6.3 T_i from T_e

The modelling techniques in the previous section break down if $T_i > T_e$. For these cases, and when problems with χ_e interfere with modelling χ_i , an alternative approach was developed to provide plausible T_i profiles directly from the T_e profile measurements and the neutron measurement of T_{i0} . The approach is based on the assumption that the electrons and ions are well equilibrated as $r \rightarrow a$, and relies on the definition of the local ion energy confinement time $\tau_{E,i}$:

$$\frac{dW_i(r)}{dt} = Q_{in_i}(r) - \frac{W_i(r)}{\tau_{E,i}(r)}, \quad (3.43)$$

where $W_i(r) = k_B n_i T_i$ is the ion stored energy density and $Q_{in_i}(r)$ is the power density input to the ions at radius r . It is assumed that $\tau_{E,i}$ so defined is a weak function of position in the plasma, so that it can be approximated as constant with radius.

The ion input power density is $Q_{in_i} = Q_{RF_i} + Q_{ei}$, where Q_{RF_i} is the ICRF power density and Q_{ei} is the exchange power density from the electrons. For the time being, assume Q_{RF_i} is known. Also, it is assumed that dW/dt is small. The central ion confinement time can then be calculated:

$$\tau_{E,i}(0) = \frac{k_B n_{i0} T_{i0}}{Q_{RF_{i0}} + Q_{ei0}} \quad (3.44)$$

where Q_{ei} is given by (2.16):

$$Q_{ei} = \frac{Q_x}{T_e} (T_e - T_i). \quad (3.45)$$

$\tau_{E,i}$ is assumed to be independent of r , so that $T_i(r)$ can be calculated from (3.43).

The result is:

$$T_i(r) = \frac{T_e(1 + Q_{RF_i}/Q_x)}{1 + \frac{p_T^{3/2}}{p_n f_0} \left(1 - f_0 + \frac{Q_{RF_{i0}}}{Q_{x0}}\right)} \quad (3.46)$$

where $p_T = T_e(r)/T_{e0}$, $p_n = n_i(r)/n_{i0}$ and $f_0 = T_{i0}/T_{e0}$.

Q_{RF_i} is actually calculated by TRANSP using T_i , and not measured directly, so using (3.46) for T_i would require iteration, substantially increasing the complexity of preparing a TRANSP run. However, if $Q_{RF_i} \ll Q_x$, which is true for most C-Mod plasmas because of their high density, Q_{RF_i} can be dropped from (3.46), giving

$$T_i(r) = \frac{T_e}{1 + \xi} \quad (3.47)$$

where

$$\xi = \frac{p_T^{3/2}}{p_n} \left(\frac{T_{e0}}{T_{i0}} - 1 \right) \quad (3.48)$$

Figure 3.15 compares the results of using (3.47) for calculating T_i to (3.46). Q_{RF_i} for the latter was calculated by TRANSP using T_i from the former. Clearly, the difference between the two profiles is smaller than the error in the measurement of T_e , and insignificant given the fact that the actual T_i profile is not known. The volume integrals of the two profiles are almost identical. Equation (3.47) should therefore provide a reliable and simple method of generating T_i to allow calculation of χ_{eff} .

3.6.4 Testing T_i Models

This section presents the results of a comparison of the T_i profiles obtained from the methods described in the previous section to the T_i profile measured by Hirex. In the process, an attempt is made to estimate the errors introduced when the profile shape is not measured. The two methods used to generate the T_i profile were: (1) modelling $\chi_i \propto \chi_e$ and feeding back on T_{i0} (§ 3.6.2), and (2) scaling the T_i profile shape from T_e (§ 3.6.3).

Run 950317 - low density

The run 950317 presented an ideal opportunity to obtain a detailed T_i profile from Hirex, as for the entire day the same shot was run repeatedly. Figure 3.16 shows the time history of one shot from the run day (shot 23). Most plasma parameters are constant during the period $t = 0.4 - 0.6$ over which the Hirex data was averaged.

From scanning the Hirex chords from shot to shot, the profile shown in figure 3.17

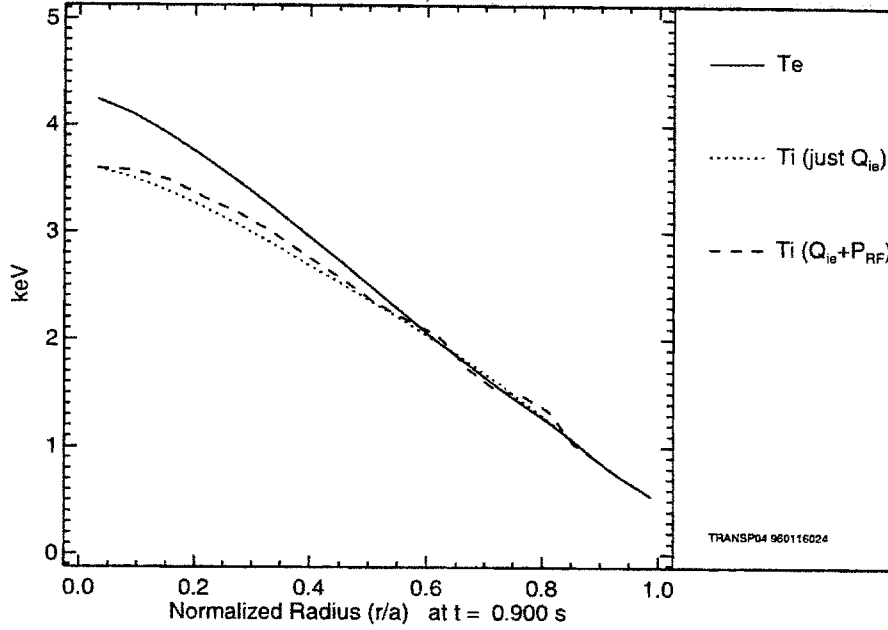


Figure 3.15: Calculation of T_i from T_e using two different methods

was constructed. Also shown in figure 3.17, with the solid line, is a least squares fit to the Hirex data of the form

$$T_i = T_{i0}[1 - (r/a)^2]^n. \quad (3.49)$$

This fit provided the input profile to TRANSP for local analysis. TRANSP runs were also conducted for methods (1) and (2), with the resulting T_i profiles plotted in figure 3.17 as the dashed and dot-dashed lines respectively. T_e is plotted for comparison. Note that no problems were encountered in the feedback of method (1) – the measured T_{i0} was closely followed without any pathologies arising in the profile.

Over the inner half of the plasma, the generated profiles agree with the measured profile within 20%. For $R > 80$ cm, the difference between the invented and measured profiles is much greater than the error. Both generated profiles have $T_i \rightarrow T_e$ as $r/a \rightarrow 1$ since they assume that $\tau_{ei} \ll \tau_E$. Calculating τ_{ei} using (3.41) at $R = 85$ cm, where $T_e \simeq 500$ eV and $n_e \simeq 0.8 \times 10^{20}/\text{m}^3$, confirms this assumption: $\tau_{ei} \simeq 3$ ms $\ll \tau_E$. The Hirex profile on the other hand gives $T_i \ll T_e$, which to be sustained would require an enormous power transfer Q_{ei} from the electrons to the ions. Since χ_e plotted in figure 3.18 is > 0 over most of the profile, electron power balance can in fact sustain this Q_{ei} , except perhaps at the edge where the temperatures are not well measured anyway.

The question is then whether the ΔT obtained from the Hirex T_i profile is really large for $R > 80$ cm, or whether T_i is systematically too low or T_e too high. Towards the edge of the plasma, the T_e measurement from ECE is subject to large errors, so it

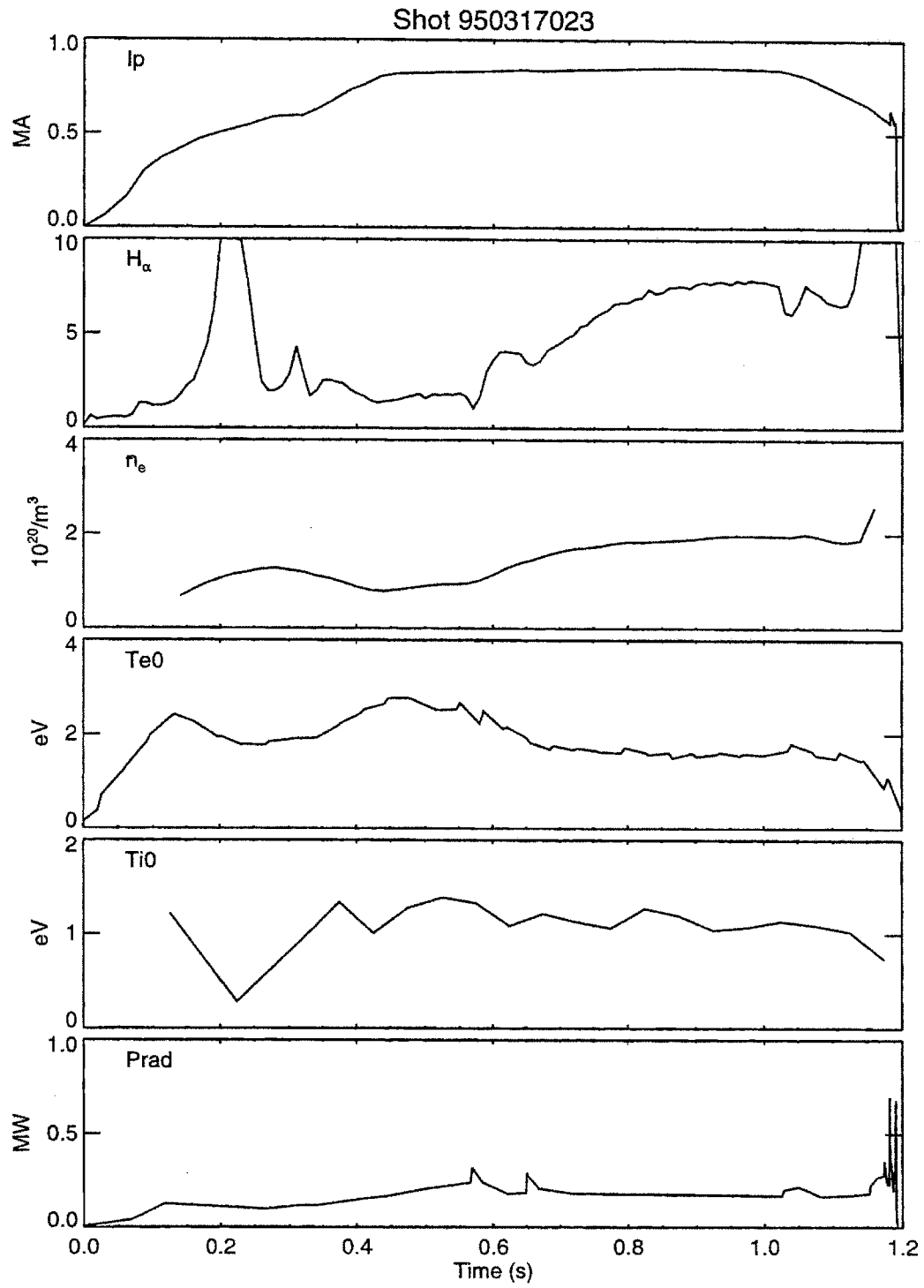


Figure 3.16: Time history of shot 950317023

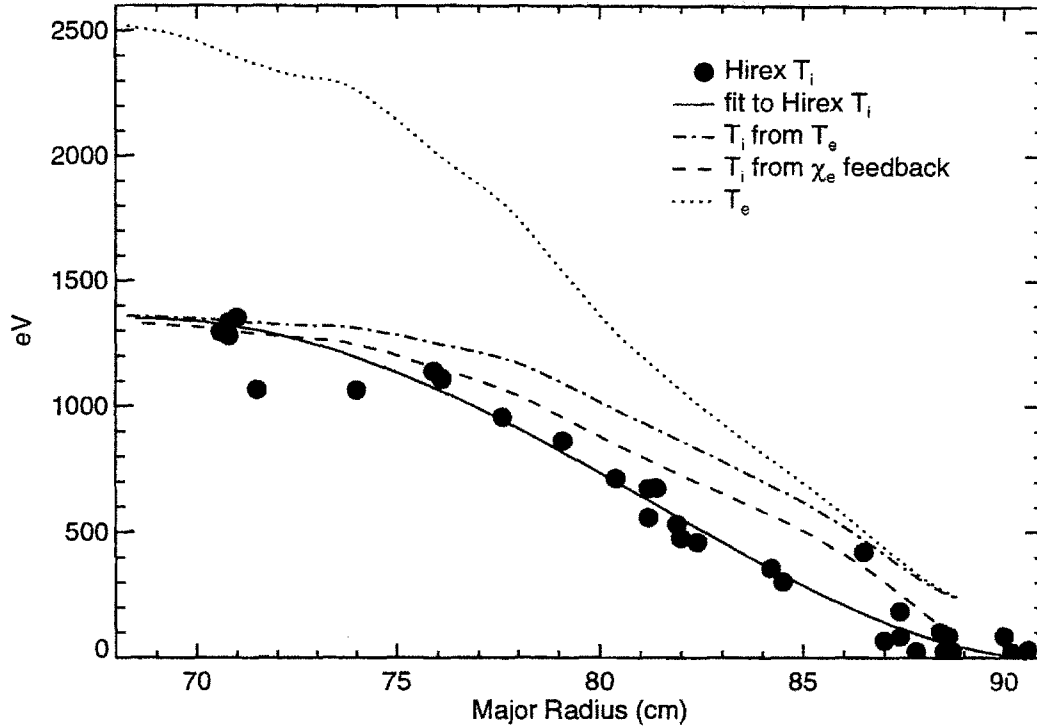


Figure 3.17: Calculated T_i profile compared to measured profile

is possible that T_e is too high. It is also possible that T_i from Hirex is under-estimated near the edge, though this would imply that the instrument resolution is better than thought [72]. Comparing the stored energies calculated from the profiles to U_{mhd} does not shed much light on the source of the discrepancy, as the differences between U_{mhd} and U_{kin} are within their uncertainty range. At $t = 0.5$ s, $U_{mhd} = 45 \pm 12$ kJ. U_{kin} calculated from the Hirex T_i is 29 kJ, while U_{kin} from the invented T_i profile of method (2) is 34 kJ. There were no neutron rate measurements available from this shot for comparison, but they would not be sensitive to T_i near the edge so would not help resolve the issue either.

A plot of the χ_i profiles calculated from T_i for each method shows that over the range $r/a \in (0.4, 0.7)$, χ_i from either method is within 0.25 m^2/s of χ_i calculated from the Hirex T_i ; and at $r/a = 0.5$ they are within 0.06 m^2/s . Since the error in χ_i was estimated in § 3.1.1 as $\delta\chi \sim 1$ m^2/s , this can be considered good agreement. The shapes of the χ_i profiles are quite different, with χ_i from the invented profiles diverging on axis, while χ_i from Hirex diverges near the edge. The divergence of χ_i near the axis calculated the generated profiles is likely due to raggedness in the T_i profile, as figure 3.19 shows that the ion conducted power $P_{cond_i}(r) = \int_0^r \nabla \cdot \mathbf{q}_i dV$ is smooth in this range, with the values for the three methods nearly identical. Since this is an ohmic shot, $P_{ei} \simeq P_{cond_i}$. For the Hirex case, P_{ei} gets very large near the edge as mentioned previously.

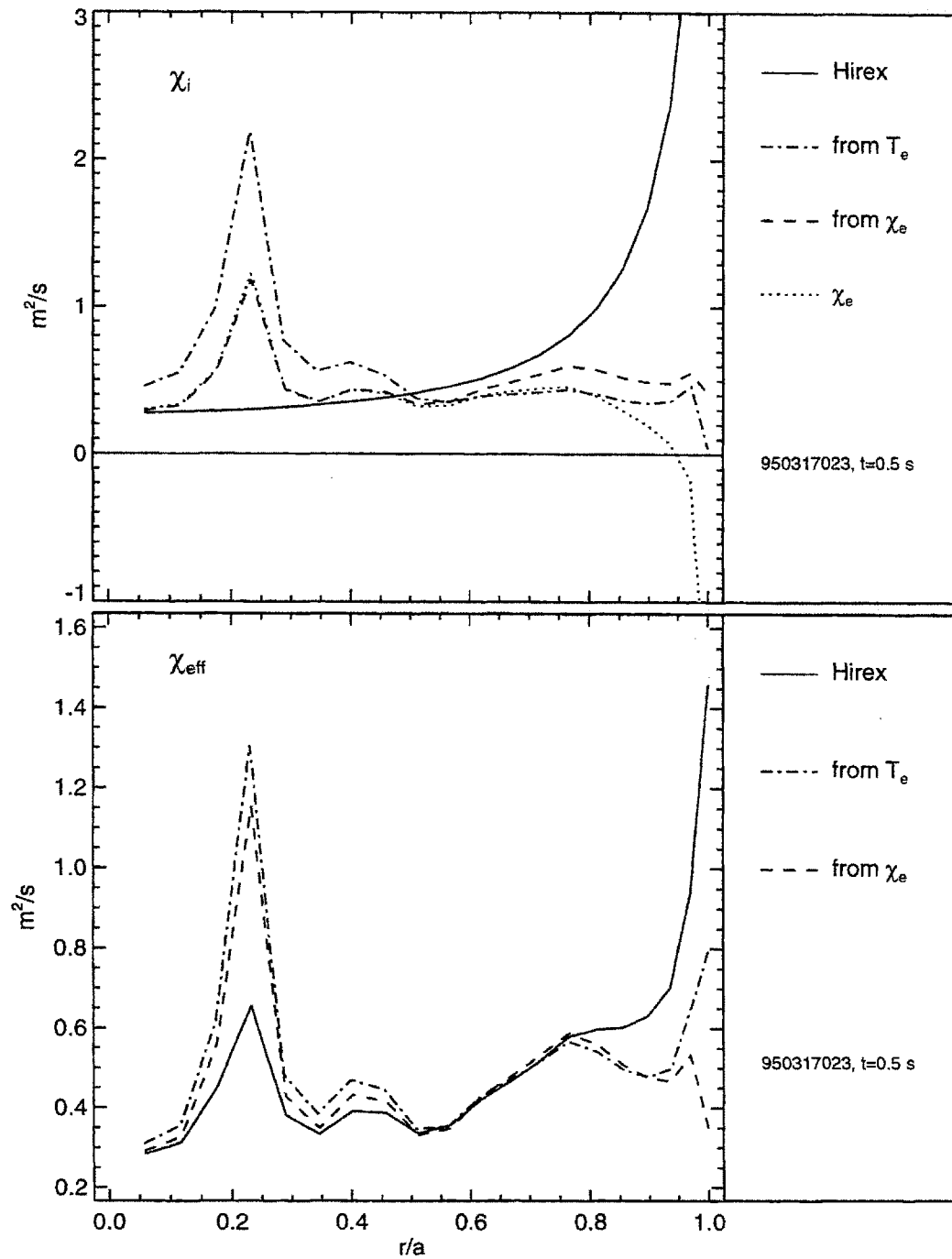


Figure 3.18: χ_i and χ_{eff} from different T_i profiles, run 950317

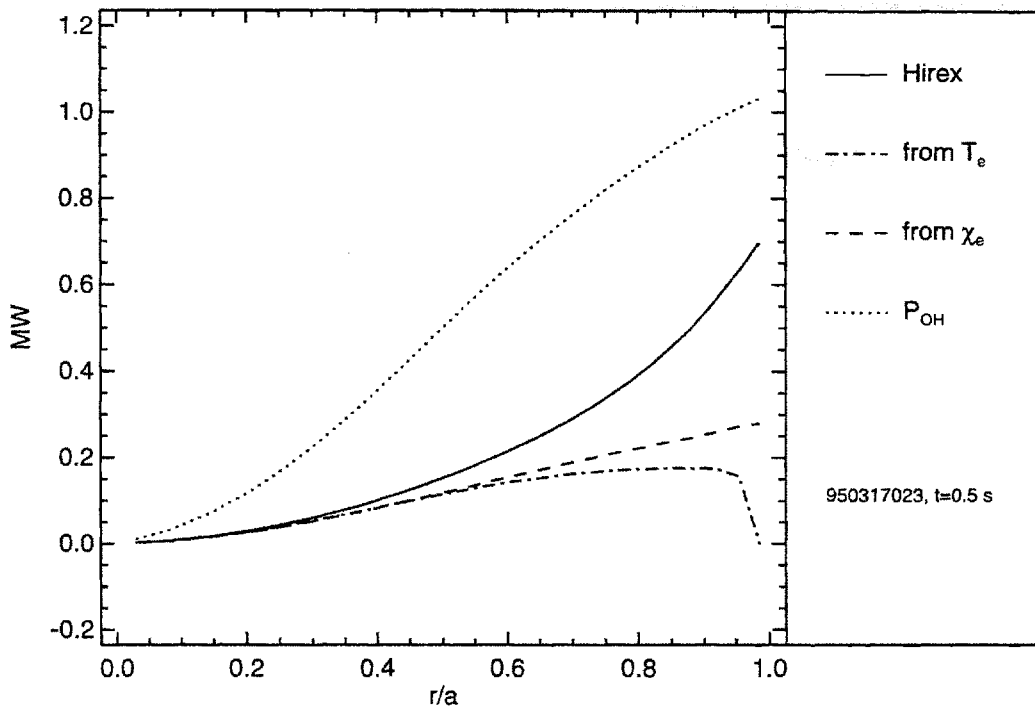


Figure 3.19: Volume-integrated conducted power density $P_{\text{cond}}(r) = \int_0^r dV \nabla \cdot \mathbf{q}$ calculated from different T_i profiles, run 950317

The χ_{eff} profiles shown in figure 3.18 also agree very well over the range $r/a \in (0.3, 0.8)$. They are virtually identical over most of this range, with a maximum difference of $0.1 \text{ m}^2/\text{s}$, well within the uncertainty.

Run 960229 - higher density

Three shots on 960229 also presented an opportunity to obtain a detailed Hirex T_i profile, this time at a higher density ($\langle n_e \rangle = 1.8 \times 10^{20}/\text{m}^3$). However, T_i was scanned over only three shots, so fewer positions were studied than on 950317. Figure 3.20 plots the measured Hirex T_i profile with circles, with least-squares fit of the form (3.49) plotted with the solid line. The profiles generated from methods (1) and (2) are again plotted with the dashed and dot-dashed lines respectively. T_e is shown for comparison.

Though the Hirex data does not reach as far out in radius, it is clear that here $T_i \rightarrow T_e$ as $r \rightarrow a$. The generated profiles both agree very well with most of the Hirex data. The two points where the agreement is not as good tend to widen the polynomial fit to the Hirex data, causing the neutron rate calculated from the fit to overestimate the neutron rate as shown in figure 3.21. On the other hand, the generated profiles do a better job of estimating the measured neutron rate. This suggests that the T_i profile is not as broad as the polynomial fit would indicate.

Figure 3.22 plots the χ_i profiles from the three T_i profiles. This time the χ_i profiles are somewhat different. Though the gradients in the T_i profiles are close, since the density is higher Q_x is higher, causing χ_i to be more sensitive to $|T_e - T_i|$. Also, these shots are ICRF heated, so χ_i is affected by the sensitivity of the singularity problem to T_i . The maximum range in χ_i at $r/a = 0.5$ is $0.25 \text{ m}^2/\text{s}$, which is within the uncertainty of χ_i .

χ_{eff} calculated from the invented profiles is also different than from the Hirex profile. This is likely due to the sensitivity of the RF model to the T_i profile. The maximum range in χ_{eff} at $r/a = 0.5$ is $0.18 \text{ m}^2/\text{s}$, which is within the uncertainty.

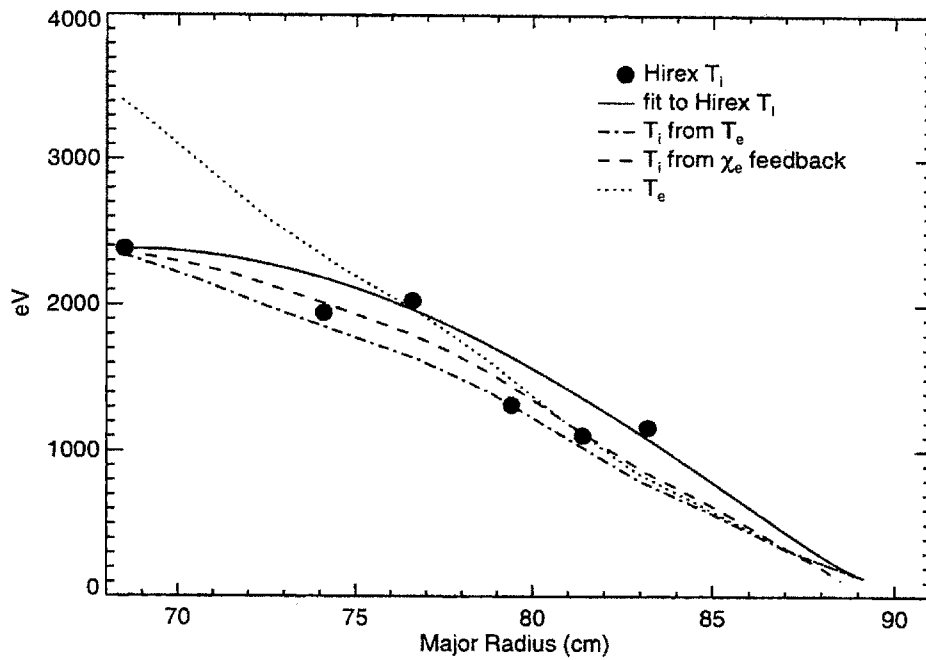


Figure 3.20: Measured and calculated T_i profiles for shot 960229042

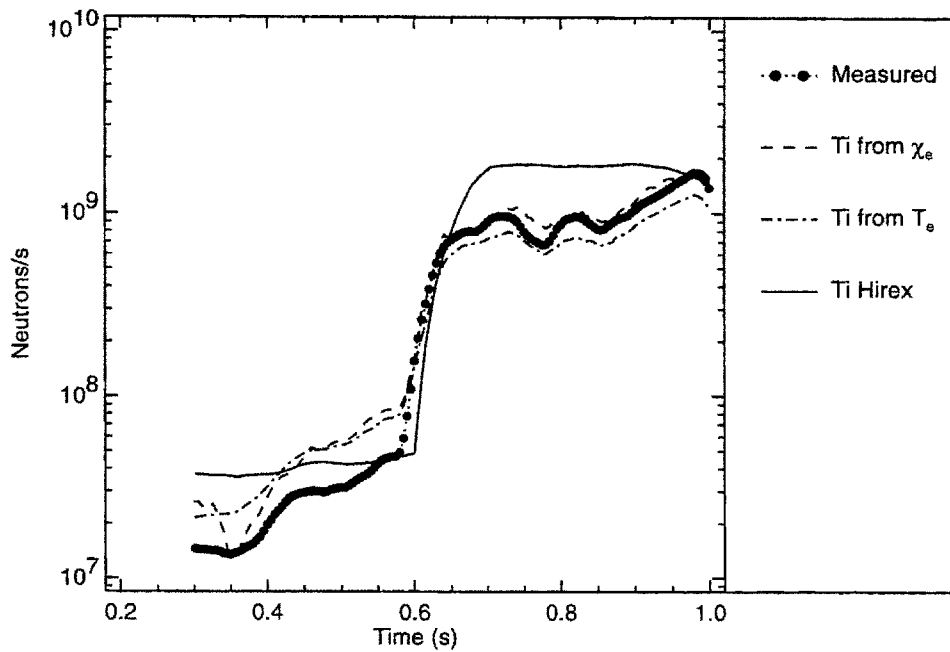


Figure 3.21: Neutron rates calculated for T_i profiles from figure 3.20

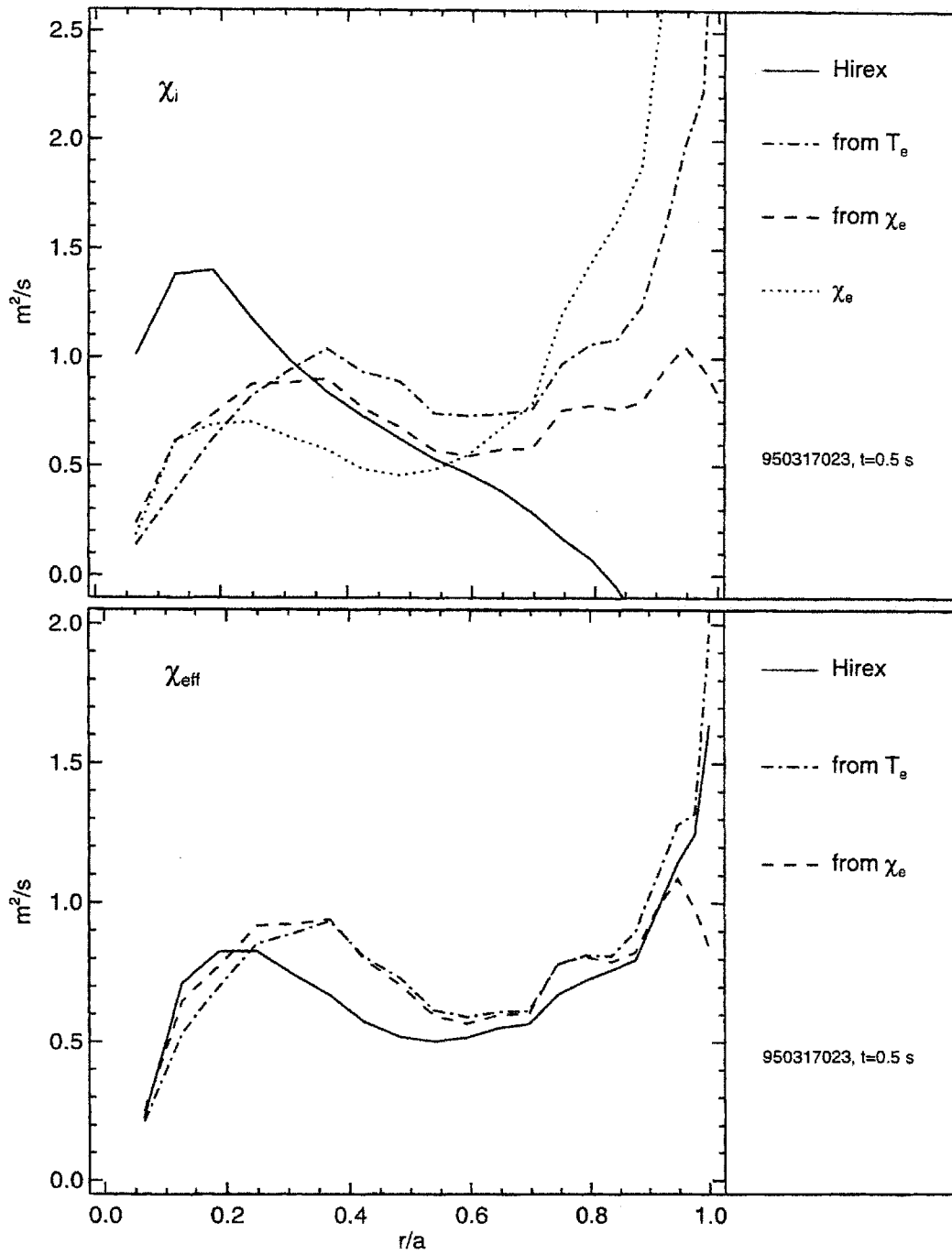


Figure 3.22: χ_i and χ_{eff} profiles for T_i profiles from figure 3.20

Chapter 4

ρ^* Scaling

This chapter presents the results of the application of local analysis techniques to empirical transport analysis. The question addressed is the scaling of transport with ρ^* , the ion gyroradius normalized to the machine size:

$$\rho^* = \frac{\rho_i}{a} = \frac{\sqrt{2T_i m_i}}{eBa}. \quad (4.1)$$

The outline of this chapter is as follows: § 4.1 provides the motivation for and background of ρ^* scaling. The physics behind ρ^* scaling experiments, and a general description of how they are conducted, are presented in § 4.2 and § 4.3 respectively. The specific methods followed for the experiments on C-Mod are described in § 4.4, with the results from the experiments in § 4.5. ν^* scaling is conceptually similar to ρ^* scaling; a presentation of ν^* scaling analysis is given in § 4.6. § 4.7 summarizes the results and places them in the wider context of fusion research.

4.1 Background and Motivation

ρ^* scaling is a type of empirical analysis in which the scaling of transport with the normalized ion gyroradius ρ^* is determined. The choice of ρ^* , a dimensionless plasma parameter, as the parameter for the analysis is based on the application of empirical scaling to the prediction of an ignition scale tokamak like ITER, and the limitations of an engineering scaling approach.

4.1.1 Limitations of Dimensional Scaling

As discussed in § 2.3.1, most if not all currently operating tokamaks observe a generic Goldston global confinement scaling, that is $\tau_E \sim I_p P_{in}^{-\frac{1}{2}} R^x$, over a wide variety of plasma parameters. The ITER89-P (2.38) and ITER93-H (2.39) scaling relations for L- and H-Mode respectively are Goldston-like scalings determined from a multi-

machine database. These scalings are being applied to the design of ITER, a tokamak intended to reach ignition [73]. Their application to this problem is questionable, given the difficulty in measuring the scaling with machine size, and given the large difference between ITER's parameters and those of current tokamaks.

Scaling with Size

While the scaling of τ_E with I_p and P_{in} is well characterized, the scaling of τ_E with the size of the tokamak R is harder to determine. A given tokamak can only accommodate plasmas within a limited range of R and a . Tokamaks with divertors, or with ICRF heating, have even more limited ranges available, as only a specific shape will allow proper coupling from the antenna, or fitting the divertor geometry. Regressions against a parameter with very limited range result in large errors. Intuitively, a scaling with size cannot be determined from a dataset in which the size does not vary appreciably beyond the error in measurement, as there is essentially only one value for the independent variable [74, 75].

Determining the size scalings using data from different tokamaks does increase the range of size covered, but is also subject to large errors. Differences in tokamak design (like divertor shape), or machine operation (boronization, for example), as well as uncontrolled or unknown variables, can complicate the extraction of the dependence of τ_E on size alone [36, 74–77].

Extrapolating to Future Tokamaks

The dimensional parameters of ignition tokamaks are very different than those for current machines. For example, ITER is designed to have $I_p = 21$ MA, $P_{in} = 50$ –100 MW, and $R = 8.1$ m. Using the ITER89-P and ITER93-H relations to predict ITER's performance requires a large extrapolation from the range of parameters over which they were tested, and there is no a priori reason to expect that the log-linear assumption on which the scaling is based will hold over such a wide range.

Even assuming the plasma current and input power scalings are valid for ITER, it is hard to justify extrapolation of the size scaling. ITER will be over twice as large as JET ($R = 3$ m), currently the largest tokamak ever constructed, and over an order of magnitude larger than C-Mod ($R = 0.67$ m). Given the problems in determining the scalings for existing machines, an extrapolation over this range of sizes is questionable at best. The fact that the size scalings in (2.38) and (2.39) are strong further exacerbates the problem.

Fortunately, it is not necessary to rely on dimensional scalings alone. Dimensionless scaling provides a more powerful, and more plausible, means of scaling up to ITER's performance. Dimensionless scaling has its origins in the mathematics of differential equations, which will be discussed in the next section.

4.1.2 Dimensional Analysis and Scale Invariance

Even though it is not possible to solve the full Boltzmann equation (2.1) for tokamak plasmas, the equation itself is still thought to be a valid model for the plasma behavior. Applied to this model, the techniques of dimensional analysis and scale invariance provide constraints to empirical scaling which can assist in determining valid scaling laws, and lead to the idea of dimensionless scaling.

These two techniques are familiar methods in solving differential equations for scientific applications. The basic idea is that symmetries in the equation allow the use of group theory to simplify the system, either by reducing the order of the equations or by allowing a transformation from partial to ordinary equations.¹

Connor and Taylor [79] applied these methods to empirical scaling analysis for tokamak plasmas. They assumed that any empirical scaling for a plasma property thought to be governed by (2.1) must exhibit the same symmetries as the equation itself. In other words, if the governing equations are invariant under a certain group of transformations, then so must be any scaling laws, and thus their form is restricted. For a collisional high β model, they found that the confinement time must scale as:

$$B\tau_E = F_1(na^2, Ta^{1/2}, Ba^{5/4}) \quad (4.2)$$

where F_1 is a function of only the three variables, known as *Connor-Taylor* variables, and $B\tau_E$ is a dimensionless form of the confinement time: $\tau_E\Omega_c = \frac{e}{m}B\tau_E$. This result assumes that the phenomena of interest have scale lengths much longer than the Debye length λ_{db} , so that λ_{db} is not an important parameter.

Note that (4.2) determines the size scaling of normalized transport. Two machines with different minor radii are predicted to have the same confinement if their densities, temperatures, and fields scale as:

$$\begin{aligned} n &\sim a^{-2} \\ T &\sim a^{-1/2} \\ B &\sim a^{-5/4} \end{aligned} \quad (4.3)$$

Various similarity experiments have validated these scalings [80–83].

The scaling of τ_E in (4.2) connects directly with the work in dimensional analysis first published by Kadomstev [84]. It can be shown that if n , T , and B satisfy (4.3), then (4.2) can be rewritten as

$$B\tau_E = F_2(\rho^*, \nu^*, \beta). \quad (4.4)$$

The requirements of scale invariance imply that τ_E must be a function only of the dimensionless parameters of the plasma!

¹The curious reader is referred to [78] for more information on the general methods of scale invariance and dimensional analysis.

If a diffusive model is assumed for the turbulent transport, with the thermal diffusivity χ determined by local plasma parameters, then χ must meet the requirements of scale invariance as well, so that it too is a function only of the dimensionless parameters:

$$\chi = G_1(\rho^*, \nu^*, \beta). \quad (4.5)$$

The symmetries that determine the important dimensionless parameters obviously depend on the model chosen to describe the plasma. If there are effects not included in (2.1), the constraints of Connor-Taylor invariance will not apply. For example, using a Coulomb collision operator in (2.1) does not include any atomic physics effects, so if radiation or charge exchange play an important role in determining transport, scaling with ρ^* , ν^* , and β can be misleading.

Assuming that atomic physics effects are not important, so that (4.4) and (4.5) are valid, two implications can be drawn. First, any dimensional scaling law of the form (2.35) must satisfy (4.4) in order to meet the requirements of scale invariance. Presumably, an empirical law that does not cannot be valid. The dimensionless form of the ITER89-P scaling law (2.38) is:

$$B\tau_E \propto \rho^{*-2.0} \nu^{*-0.28} \beta^{-0.52} q^{-1.7}. \quad (4.6)$$

With a residual of 0.04 on the R coefficient, or 3%, this form is not quite dimensionally correct. It can be made dimensionally correct by adjusting the density scaling from $n^{0.10}$ to $n^{0.08}$, which is within the uncertainty of the n coefficient. This changes the β dependence in (4.6) to $\beta^{-0.56}$, but otherwise leaves it the same. The dimensionless form of ITER93-H (2.39) is also close to being dimensionally correct:

$$B\tau_E \propto \rho^{*-2.7} \nu^{*-0.28} \beta^{-1.2} q^{-3.2}, \quad (4.7)$$

with a residual on R of 0.04 or 2%. Adjusting the density scaling from $n^{0.17}$ to $n^{0.19}$, again within the uncertainty, drops the residual to less than 1%. The dimensionally correct scaling is

$$B\tau_E \propto \rho^{*-2.8} \nu^{*-0.28} \beta^{-1.2} q^{-3.2}. \quad (4.8)$$

(4.4) and (4.5) also imply that the scalings of $B\tau_E$ and χ with the dimensionless parameters can be measured directly in dimensionless scaling experiments. These experiments are the subject of the next section.

4.1.3 Dimensionless Scaling

Equation (4.4) seems to provide a solution to the problems with dimensional scaling discussed in § 4.1.1. If existing tokamaks can attain the ρ^* , ν^* , and β of an ignition

machine, then they can predict the latter's performance without extrapolation. This type of "wind-tunnel" scaling, first introduced to plasma physics by Waltz, DeBoo, and Rosenbluth [33], is common in fluid mechanics [85, ch.6], where laboratory tests are conducted on small scale models at the same dimensionless parameters as the full scale system. In most applications for fluid dynamics, the only dimensionless parameter of importance to the system is the Reynolds number. Though plasmas are much more complicated than neutral fluids, as exemplified by the large list of dimensionless parameters in table 4.1, this approach is still valuable.

Symbol	Parameter	Definition/form	Notes
ρ^*	normalized gyroradius	ρ_i/a	$\rho_i =$ ion larmor radius
β	relative pressure	$(\sum_j n_j T_j)/B^2$	taking sum over species j
ν^*	collisionality	ν_e/ω_b	$\omega_b =$ bounce frequency
	normalized Debye length	λ_{db}/a	usually $\ll 1$
$q_{\psi 95}$	safety factor	$\sim rB/RB_\theta$	
Θ	Temperature ratio	T_i/T_e	(Θ is not universal)
Z_{eff}	Z effective	$\sum_i (n_i Z_i^2)/n_e \sum_i (Z_i)$	
ϵ	inverse aspect ratio	a/R	

Table 4.1: Some tokamak plasma dimensionless parameters

For most of the dimensionless parameters listed in table 4.1, existing tokamaks are able to achieve values close to those of ignition tokamaks. Not so for ρ^* , the ion gyroradius normalized to the plasma minor radius, which for ITER is almost an order of magnitude lower than for existing machines. Extrapolation of transport scaling with ρ^* is therefore necessary to predict ITER's performance, substantially weakening the power of the dimensionless scaling approach. This negative effect is compounded by the fact that ρ^* is a strong function of two of the most important design parameters for a tokamak: the magnetic field B and the minor radius a .

If dimensionless scaling is to provide even a remotely plausible prediction for ITER, it is crucial that the ρ^* scaling of existing machines be carefully tested to ensure the validity of the approach, and to provide the best possible means for extrapolation to ITER.

4.2 Physics of ρ^* Scaling

The physics behind ρ^* scaling is more profound than simple empirical dimensionless scaling. Assuming a local diffusive model in which the transport can be characterized by a local thermal diffusivity χ satisfying (4.5), it is possible to relate the ρ^* scaling of χ to the nature of the diffusive processes underlying the transport.

For empirical scaling purposes, the log-linear assumption is invoked to express the ρ^* dependence of χ in a power law form:

$$\chi \sim \chi_B \rho^{*\alpha} F(\nu^*, \beta) \quad (4.9)$$

where χ_B is the Bohm diffusion coefficient

$$\chi_B = \frac{cT}{eB} = c_s \rho_i. \quad (4.10)$$

The choice of χ_B as the normalization constant in (4.9) is based on the assumption that electrostatic instabilities are behind the turbulence driving the transport; this has already been discussed in § 2.2.3, and can also be shown with dimensionless scaling arguments [86]. (2.34) in this section presented an heuristic form for the diffusion coefficient from electrostatic turbulence:

$$\chi = \frac{\Delta^2}{\tau} = \chi_B \frac{\Delta}{a}. \quad (4.11)$$

Comparing (4.11) to (4.9), with the gradient scale length $L \sim a$, a relationship between the ρ^* scaling of transport and the characteristic step size Δ of the diffusion is evident. If the step size of the diffusion is on the order of the machine size, that is $\Delta \sim a$, then $\chi \sim \chi_B$, implying $\alpha = 0$, so-called *Bohm* scaling. As discussed in § 2.2.3, most theoretical treatments of anomalous transport determine that the wavelengths of the turbulent modes are on the order of ρ_i , not a . If the characteristic step size $\Delta \sim \rho_i$, then $\chi \sim \chi_B \rho^*$, and $\alpha = 1$. Since $\rho^* \ll 1$, the transport is significantly reduced over Bohm scaling, and hence $\alpha = 1$ scaling is named *gyro-reduced Bohm* (or simply *gyroBohm*) scaling. The standard L-Mode dependence $\tau_E \sim I_p^{1.0}/P_{in}^{0.5}$ is referred to as *Goldston* scaling [34]. Showing no dependence on the magnetic field B , if $\rho^* \sim B^{-2/3}$, it should therefore have $B\tau_E \sim \rho^{*-3/2}$, or $\alpha = -1/2$. If the magnetic field is completely ergodic, filling the plasma volume, it provides no assistance in confinement, and $\Delta \gg a$. The resulting scaling should then show no dependence on B , suggesting $\alpha = -1$ — *stochastic* scaling — as a worst-case scenario for magnetic confinement.

Experiments on other tokamaks have shown Bohm-like scaling of χ in certain circumstances, seemingly contradicting the predictions of theory. One possible theoretical explanation for the observation of Bohm scaling despite eddy sizes on the order of ρ^* is the correlation of eddies over a finite distance in the tokamak. Garbet and Waltz [87] discuss this in the context of ITG turbulence, while Newman et al. [88] use the idea of “self-organized criticality” as a non-mode specific description of Bohm scaling.

Global ρ^* scaling can be derived from the local scaling, given the assumptions relating χ and τ_E discussed in § 2.3.2 leading up to (2.44), that is that the radiated power and other loss terms are negligible, and the heat flux is entirely diffusive. Then

$\tau_E \sim a^2/\chi$. Since the physics of electron and ion heat transport is different, the ρ^* scaling of the two species can be different. The scaling of the effective diffusivity $\langle\chi\rangle$, and thus τ_E , would then be determined by the relative heating of the two species and the amount of coupling between them. This observation, first made by Petty and Luce [89], explains the variety of global ρ^* scalings measured for different tokamaks. It is therefore preferable to individually determine the scalings for electrons and ions whenever possible.

4.3 ρ^* Scaling Experiments

The goal of ρ^* scaling experiments is to determine the scaling of transport with ρ^* given the assumptions behind dimensionless scaling. Assuming the power-law form of (4.9) is correct, α can be calculated by measuring χ during a systematic variation of ρ^* with all other dimensionless parameters kept fixed. In this case, $F(\nu^*, \beta)$ is constant, so that

$$\chi \sim \chi_B \rho^{*\alpha}. \quad (4.12)$$

For global scaling, assuming the profile shapes remain fixed as ρ^* is varied so that γ_p in (2.44) remains constant,

$$\tau_E \sim \frac{a^2}{\langle\chi\rangle} \sim \frac{a^2}{\chi_B} \rho^{*-\alpha}. \quad (4.13)$$

Putting (4.13) into dimensionless form gives

$$B\tau_E \sim \rho^{*-2-\alpha}. \quad (4.14)$$

Since $\rho^* \sim T_e^{1/2}/Ba$, the largest variation in ρ^* can be had by varying the magnetic field B . To keep the dimensionless parameters fixed as B is scaled, it is necessary to scale other engineering parameters concurrently. Keeping q constant as B is scaled requires

$$I_p \sim B. \quad (4.15)$$

The other two dimensionless parameters that depend on B are β and ν^* . With a fixed plasma size and shape, these two parameters are functions of three variables: n , T , and B , so keeping them fixed as B is scaled requires

$$n \sim B^{4/3} \quad (4.16)$$

$$T \sim B^{2/3} \quad (4.17)$$

so that $\rho^* \sim B^{-2/3}$.

Thus a $2\times$ scaling of B gives a $1.6\times$ scaling of ρ^* . Any identically shaped plasmas at different magnetic fields whose plasma currents, densities, and temperatures satisfy the relations (4.15)-(4.17) should have matching dimensionless parameters and thus be suitable for ρ^* scaling analysis. These shots are referred to as *dimensionally similar*.

α in (4.12) or (4.14) can be determined directly by regressing χ/χ_B or $B\tau_E$ versus ρ^* for a set of dimensionally similar shots. Table 4.2 summarizes the values of α for the aforementioned scalings.

Name	χ scaling $\chi \sim \chi_B \rho^{*\alpha}$	$B\tau_E$ scaling $B\tau_E \sim \rho^{*-2-\alpha}$	P_{in} scaling [†] $P_{in} \sim B^\gamma$
gyroBohm	1	-3	1
Bohm	0	-2	5/3
Goldston [†]	-1/2	-3/2	2
Stochastic	-1	-1	7/3

[†] Scaling defined by $\tau_E \sim B^0$, so scaling versus ρ^* depends on T_i scaling.

[‡] Scaling of input power also depends on scaling of T_i vs. B .

In both cases, values shown are for $T_i \sim B^{2/3}$.

Table 4.2: Named ρ^* scalings.

In trying to obtain suitable plasmas, note that while it is possible to directly control B and I_p , controlling n and T is not straightforward. Feedback control on the gas puffing is necessary to control the density, but if the wall source is large enough, it is not sufficient. Some type of feedback-controlled pumping, especially in the divertor, could be required as well. This is especially true for H-Mode plasmas on C-Mod, when the increase in particle confinement time often causes an uncontrollable density rise.

Controlling the temperature is more difficult as there is no engineering knob for it. Instead, the input power is adjusted, and the temperature is determined by it and the resulting change in transport. Auxiliary heating is desirable for ρ^* experiments, as the input power can be scanned to obtain the proper temperature. Since the only knobs on the input power for ohmic heating, I_p and n_e , are already fixed by (4.15) and (4.16), it may not be possible to meet (4.17).

Actually, the scaling for ohmically heated plasmas should be completely determined if the coupling between the electrons and ions is weak. The input power should scale as

$$P_{OH} = \int \eta J^2 dV \sim \frac{I_p^2}{T_e^{3/2}} \sim B \quad (4.18)$$

implying a global gyroBohm scaling:

$$\tau_E \sim \frac{U}{P_{OH}} \sim B. \quad (4.19)$$

In fact, as will be discussed in the following sections, electrons are observed to have gyroBohm scaling over a wide range of plasma conditions, even when strongly heated by an auxiliary source. In high density plasmas with strong electron-ion coupling, the global scaling will be determined by both species. For auxiliary heated plasmas, if $T_i \sim B^{2/3}$, and $P_{\text{rad}}/P_{\text{in}}$ is low, the input power should scale as

$$P_{\text{in}} \sim B^{\frac{1}{3}(5-2\alpha)}. \quad (4.20)$$

The scaling of P_{in} with B is listed in table 4.2 as well.

4.3.1 ρ^* Scaling on Alcator C-Mod

For ρ^* scaling to be a valid tool, the same relations must hold for many tokamaks over a wide variety of plasma conditions. Testing these scaling on Alcator C-Mod's unique parameter range of high density and high field plasmas is therefore particularly important.

Alcator C-Mod can operate over a field range of 2.6 T–8.0 T. With ICRF auxiliary heating at a fixed frequency of 80 MHz, the choice of fields at which heating is possible is limited. C-Mod's nominal field is 5.3 T, corresponding to fundamental minority Hydrogen heating. Other possible schemes are second harmonic minority H at 2.6 T, and fundamental ^3He at 8.0 T.

Table 4.3 lists the latest set of parameters for ITER [90]. Ideally, the ρ^* scaling experiments on C-Mod would be conducted at ITER-like values of β , ν^* , and q . There are plans for a set of "ITER demonstration" plasmas on C-Mod, though at present to reach the high value of β_p they either require more heating power than is currently available on C-Mod, or a different RF system allowing good heating at fields lower than 5.3 T. Two sets of parameters for C-Mod ITER demonstration plasmas are listed in table 4.3, where the input power required was calculated assuming an H factor of 2 over Goldston scaling ($\tau \sim I_p/P_{\text{in}}^{1/2}$). Table 4.3 also lists the parameters from the highest β_t H-Mode on C-Mod, shot 960116024.

Since it was not possible to run at the parameters expected for ITER, the ρ^* scaling experiments conducted for this thesis covered the range of C-Mod's current set of parameters. If the assumptions behind ρ^* scaling are valid, as long as β_p and ν^* are fixed, the results should be independent of which values were used. The following sections present the methods used in conducting these experiments and their results.

Parameter	ITER nominal	C-Mod ITER demo	C-Mod ITER demo	C-Mod highest β_t	Units
I_p	21	1.3	0.86	1.0	MA
B_t	5.7	5.3	3.5	5.3	T
P_{in}	100	9.5	4.7	2.6	MW
$\langle n_e \rangle$	1.1	2.3	1.3	2.4	$\times 10^{20}/\text{m}^3$
$\langle T_e \rangle$	11	4.6	3.5	1.6	keV
R	8.1	0.67	0.67	0.67	m
a	2.8	0.22	0.22	0.22	m
$q_{\psi 95}$	3	3	3	3.7	
β_t	3.0	3.0	3.0	1.4	%
ν^*	0.01	0.01	0.01	0.10	
ρ^*	0.95	8.3	8.3	8.0	$\times 10^{-3}$
κ	1.6	1.6	1.6	1.6	

Table 4.3: ITER and Alcator C-Mod parameters

4.4 Methods

The basic plan for the ρ^* scaling experiments performed on C-Mod was as follows. Dimensionally similar shots suitable for analysis were identified using the C-Mod relational database *LOGBOOK*. Several dedicated runs were conducted to fill in ranges of parameters not already covered in the database; most of these were at 2.6 T and 3.5 T, lower than the nominal C-Mod field. Attempts at obtaining explicitly matched shots during these dedicated runs turned out not to work. Fortunately there were enough shots in the database to provide a sufficient number of matching pairs. Once a set of suitable shots was identified, regression analysis was applied to it to determine the global scaling with ρ^* . A pair of shots at two different fields with matching dimensionless parameter profiles was then selected from the set for local analysis using the TRANSP code.

Careful attention was paid to error analysis. In selecting shots, dimensionless parameters were taken as matching if they lay within each other's error range. Proper error analysis was also necessary to determine the ability to differentiate between the ρ^* scalings, and to determine the sensitivity to mismatches in the dimensionless parameters. The following sections provide the details of this analysis.

4.4.1 Plasma Measurements

Table 4.4 provides a summary of the diagnostics that measured the plasma parameters used in this analysis and the assessed errors in these measurements. There are several books on plasma diagnostics, for example [91], that provide background on the physics

involved in each. A summary of the diagnostics on C-Mod can be found in [92].

Of particular importance to ρ^* scaling is the ability of the diagnostics to work over a range of magnetic fields. The ECE michelson interferometer and grating polychromator are subject to large error for fields < 3 T [100] and so could not be used at $B_i = 2.6$ T to measure T_e . The only T_e measurements at this field were therefore from the Thomson scattering system (also known as “YAG”), which had at most 5 spatial measurements at a given time. A fit of the form $T_e(r) = T_{e0} [1 - (r/a)^2]^n$ was used to provide a profile to TRANSP. Note that at 2.6 T the YAG T_e measurement was also used to get Z_{eff} from the VB emissivity profiles.

4.4.2 Dimensionless Parameters

The calculations of the dimensionless parameters and their errors based on the measurements discussed in § 4.4.1 are presented here. Two values for a dimensionless parameter are considered to match for ρ^* scaling analysis if their difference lies within the error range of the parameter. Requiring the difference to be much less than the precision of the measurements is not meaningful. The error ranges stated in the following sections are one standard-deviation confidence intervals.

This section does not discuss the effects of slight mismatches of a dimensionless parameter p_i on ρ^* scaling, as the effects depend on the scaling with p_i . A strong dependence of τ or χ on p_i would require a better match of the p_i values than a weak dependence to avoid spurious variation. Attempts were made to measure these effects, the results of which are presented as part of § 4.5.

Normalized Ion Larmor Radius ρ^*

The parametric dependence of ρ^* is given by (4.1): $\rho^* \sim T_i^{1/2} a B$. The errors in a and B are taken to be small, so that $\sigma_{\rho^*}/\rho^* \sim 1/2 \sigma_{T_e}/T_e$. The error in T_i from the methods described in § 3.6 is 20%, so the error in ρ^* is 10%. For global analysis ρ^* is treated as a scalar, with T_i replaced by its volume average $\langle T_i \rangle$.

β

There are two ways of calculating β . The first is from the magnetic reconstruction from EFIT [55–58]. EFIT solves the grad-Shafranov equation for the poloidal flux function $\psi(r, t)$, given magnetics measurements from the various flux loops, Rogowskii coils, and B_p loops installed in C-Mod, and a parameterization of the free functions p' and FF' as power series in normalized ψ . From the solution EFIT calculates the pressure profile $p(r)$, which is then volume-averaged and used to calculate β_p :

$$\beta_p = \frac{2\mu_0 \langle p \rangle}{B_p(a)^2}, \quad (4.21)$$

Diagnostic	Measures	Error	References	Comments
ECE Michelson Interferometer	$T_e(r, t)$	10%	[70]	Good spatial resolution. Only good for $B_t > 3$ T
ECE Grating Polychromator (GPC)	$T_e(r, t)$	10%	[60]	Good temporal resolution. $B_t > 3$ T.
Two-color Interferometer (TCI)	$n_e(r, t)$	15%	[93]	Vertical chords, so inversion is subject to large error from divertor density
Thomson Scattering (YAG)	$T_e(r, t)$	20%	[94, 95]	Max. 5 spatial locations. Used polynomial fit for profile. Density measurement not good.
X-Ray spectrometer (HIREX)	$T_i(r, t)$	10%	[71]	Max. 5 spatial locations. Requires argon injection. See § 3.6
Neutron detectors	R_{DD}, T_{i0}	10%	[96]	T_{i0} measurement depends on n_e .
Bolometer array	$P_{\text{rad}}(r, t)$	15%	[97]	16 channel array, foil detector
Visible Bremsstrahlung Spectrometer	Z_{eff}	30%	[98]	Used for central value only. Profile very uncertain due to strong dependence on n_e .
Magnetics and EFIT	$I_p, B_t, V_{\text{surf}}, \text{bdry.}$	< 5%	[55–58, 99]	EFIT is a free boundary equilibrium code that uses the magnetics measurements.

Table 4.4: Diagnostics and plasma parameters for ρ^* scaling analysis

where $B_p(a) = \mu_0 I_p / l_p$ and l_p is the perimeter of the last closed flux surface. Based on calculations performed by Steve Wolfe [101] for test cases with $q_0 \in [0.75, 1.0]$, the uncertainty in β_p is found to be $\sigma_{\beta_p} = 0.05$. For low β_p cases this can amount to a large percentage uncertainty.

The other method of calculating β is by integrating the profile data:

$$\beta = \frac{2\mu_0}{B^2} \frac{1}{V} \int_0^a (n_e T_e + n_i T_i) dV. \quad (4.22)$$

The uncertainty in the profile shape is dominated by the errors in the profile measurements, as B and V are known to within 5%. Rewriting (4.22) as

$$\beta = b_0 \int_0^a n_e T_e (1 + Z\Theta) dV \quad (4.23)$$

gives the error as

$$\left(\frac{\delta\beta}{\beta}\right)^2 = \left(\frac{\delta n_e}{n_e}\right)^2 + \left(\frac{\delta T_e}{T_e}\right)^2 + \left(\frac{\delta\Theta}{\Theta}\right)^2. \quad (4.24)$$

For T_e measured by the ECE or GPC diagnostics, this gives $\sigma_{\beta}/\beta = 20\%$, while for T_e measured by the YAG system, $\sigma_{\beta}/\beta = 30\%$. Note that (4.22) works for both β_p and β_t , depending on which field is used as B . The relative error in each is therefore the same.

For ICRF-heated shots with a high energy minority tail, the MHD calculation of β will include the tail energy, while the kinetic calculation will not, as the temperature profiles are assumed thermal. The MHD calculation will therefore result in a higher value of β in these cases.

EFIT's calculation of β was used for global scaling analysis, as this is the value stored in the core table of the *LOGBOOK* database. Local ρ^* scaling analysis required the β profile calculated by TRANSP.

Collisionality ν^*

The collisionality relevant to tokamak transport processes is defined by (2.30) as the ratio of the collision frequency to the bounce frequency. For the electrons, this is:

$$\nu^*_{e} = \frac{\nu_e}{\omega_{b,e}} = \frac{104 R q n_e Z_{\text{eff}}}{\epsilon^{3/2} T_e^2}, \quad (4.25)$$

Parameter	Error (%)
j	< 5
T_{e0}	10 (15)
f_{T_e}	10 (15)
n_{e0}	15
f_{n_e}	15

Table 4.5: Errors in parameters making up ν^* . Larger errors in T_e when measured by Thomson Scattering.

where R is measured in cm, n_e in $10^{20}/\text{m}^3$, and T_e in eV. To form a scalar parameter for global analysis, insert the volume averages of the density and temperature:²

$$\nu^*_{e} \propto \frac{\langle n_e \rangle Z_{\text{eff}}}{\langle T_e \rangle^2}. \quad (4.26)$$

Z_{eff} is itself a function of n_e and T_e [91, 98]

$$Z_{\text{eff}} \propto \frac{T_{e0}^{0.37} J}{n_{e0}^2} \quad (4.27)$$

where J here is a function of the visible bremsstrahlung emissivity. Substituting (4.27) into (4.26) gives the form of ν^* suitable for error analysis:

$$\nu^* \propto \frac{J f_{T_e}^2}{T_{e0}^{1.63} f_{n_e} n_{e0}}, \quad (4.28)$$

where $f_{T_e} \equiv T_{e0}/\langle T_e \rangle$ and $f_{n_e} \equiv n_{e0}/\langle n_e \rangle$. Table 4.5 summarizes the errors for each of the parameters in (4.28), which when combined using the normal rules of error propagation, [102] give $\sigma_{\nu^*}/\nu^* = 27\%$ for nominal C-Mod shots, and $\sigma_{\nu^*}/\nu^* = 32\%$ for shots in which Thomson Scattering is the only available diagnostic for T_e .

Safety Factor q

As discussed in § 3.3, the q profile is not measured on Alcator C-Mod. TRANSP calculates it from the magnetic field diffusion equation, while EFIT calculates it from the flux reconstruction given parameterizations for the free functions and an assumption about the behavior of the central value q_0 . Since neither profile is necessarily “truth”, the q profile was not matched explicitly in the process of finding dimensionally similar shots. Instead, the value of q at the boundary was used for comparison. The q profile

²Recall Z_{eff} is treated as a scalar for C-Mod data purposes.

is calculated by EFIT as:

$$q = \frac{F(\psi)}{2\pi} \oint \frac{dl_p}{R^2 B_p}, \quad (4.29)$$

where $F(\psi)$ is one of the free functions, and the integral is over the perimeter of the flux surface ψ . For a circular, large aspect ratio, low β plasma, the value of q on the boundary is given approximately by

$$q_{\text{bdry}} = \frac{aB_t}{RB_p} = \frac{5a^2 B_t}{RI_p}, \quad (4.30)$$

where I_p is measured in MA, and the other quantities in S.I. units.

For a diverted plasma with a separatrix, q is undefined at the boundary, as $B_p \rightarrow 0$ at the x-point. The convention in the tokamak community is then to use the value of q at the 95% poloidal flux surface, referred to as $q_{\psi 95}$, as the scalar value for comparison. Since $q_{\psi 95}$ depends mostly on global quantities that are measured to better than five percent, its uncertainty is small.

4.4.3 Calculation of τ_E

This section discusses the calculation of and uncertainty in the global energy confinement time τ_E . TRANSP was not used for global analysis; a discussion of the uncertainties in the TRANSP calculations relevant for ρ^* scaling analysis is presented in § 4.4.4.

The global energy confinement time is calculated by:

$$\tau_E = \frac{U}{P_{\text{in}} - \frac{dU}{dt}} \quad (4.31)$$

where U is the plasma stored energy and P_{in} is the total input power (ohmic + ICRF). For most shots, the ρ^* scaling will be measured when the plasma is in steady state, so assume $\frac{dU}{dt} = 0$ to simplify the error analysis. The calculations of U and P_{in} will be discussed in turn, followed by the combination of their errors into the error in τ_E .

Stored Energy

In the same manner that there are two calculations of β , two calculations providing U are available. The ‘‘MHD’’ calculation of U is from EFIT’s calculation of the pressure profile $p(r)$:

$$U_{\text{mhd}} = \int_0^a \frac{3}{2} p(r) dr. \quad (4.32)$$

The error in U_{mhd} was calculated by Wolfe [101] using the same methods as for β_p described in § 4.4.2. The result was $\delta U_{mhd} \sim 10$ kJ.

The “kinetic” calculation³ of U is from integrating the profile nT for each plasma species j :

$$U_{kin} = \sum_j \int_0^a \frac{3}{2} n_j T_j dV. \quad (4.33)$$

The error in U_{kin} is from the error in the profile measurements n_j and T_j for each species. However, these errors do not combine linearly.

The line integral density measured along any of the TCI interferometer chords is subject to far less uncertainty than the density profile, as it depends only on the total phase shift measured by the interferometer and the size and shape of the plasma determined by EFIT, both of which are well determined.⁴ The uncertainty in U_{kin} from the uncertainty in the profile shape is therefore mitigated by the constraint of keeping the line-integrated density constant. This is illustrated by the two graphs in figure 4.1. The solid line plots a nominal density profile taken from an H-Mode shot. The other profiles all result in the same line-integral density. The stored energy calculated from each density profile and the nominal T_e profile are listed on the graph. Clearly, the density profile must vary by an amount much larger than its uncertainty of 15% to give an appreciable difference in U_{kin} .

The error in the kinetic calculation of the stored energy was therefore taken to be the combination of the uncertainty in the temperature profile and the uncertainty in the line averaged density ($\simeq 5\%$). For the electrons, this gives $\sigma_{U_{kin_e}}/U_{kin_e} = 11\%$ if the ECE or GPC measurements are available, and $\sigma_{U_{kin_e}}/U_{kin_e} = 21\%$ if T_e is taken from the YAG measurement. For the ions, the error in U_{kin_i} is 21%. Assuming the errors in the species stored energies are independent gives an upper bound to the total error in U_{kin} as $\sigma_{U_{kin}}/U_{kin} = 25\%$ (30% using the YAG T_e).

For L-Mode shots at low field (2.6 or 3.5 T) where the total stored energy is $\simeq 10-15$ kJ, the relative uncertainty in U_{mhd} is very large, so U_{kin} was used whenever possible. Note that as for β , RF-heated shots with significant high energy minority tails will have U_{mhd} systematically larger than U_{kin} .

³The kinetic calculations used in the database were conducted by the homegrown IDL code “WCONFINE”. See [103] for more information.

⁴This is provided the plasma in the divertor area is not dense.

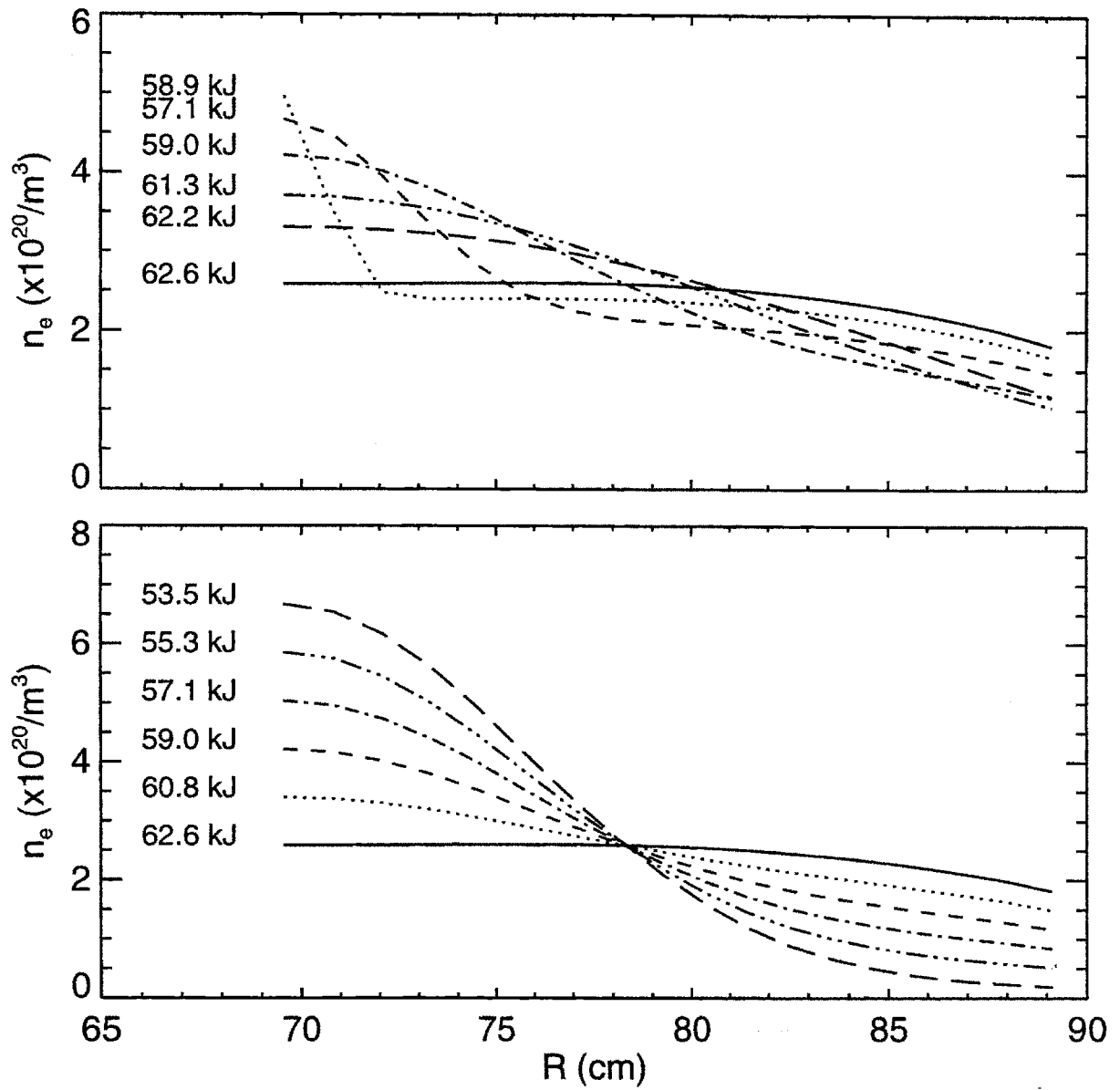


Figure 4.1: Variations in density profile shape with line-integral fixed, and stored energies calculated from each.

Input Power

The input power consists of the total ohmic heating and the total ICRF heating power. For global analysis, the former is calculated by EFIT as:

$$P_{\text{oh}} = I_p \times V_{\text{surf}} - \frac{dU_{\text{mag}}}{dt}, \quad (4.34)$$

where $V_{\text{surf}} = -\frac{d\Psi_b}{dt}$ is the surface voltage. U_{mag} is the magnetic stored energy:

$$U_{\text{mag}} = \mu_0 \frac{\ell_i}{4} I_p^2 V \frac{1}{l_p^2}, \quad (4.35)$$

where ℓ_i is the plasma inductance, l_p is the perimeter of the last closed flux surface, and V is the plasma volume. The error in P_{oh} is then dominated by the uncertainty in U_{mag} , as I_p and V_{surf} are determined very accurately. The uncertainty in U_{mag} in turn is dominated by the uncertainty in ℓ_i . If the plasma is in steady state, so that $\frac{dU_{\text{mag}}}{dt} = 0$, the uncertainty in P_{oh} calculated by EFIT is reduced.

The calculation of the total ICRF heating power is broken up into the power launched by the antenna, and the fraction of that power absorbed by the plasma, after the discussion in § 3.5.3:

$$P_{\text{RF}}(\text{absorb}) = P_{\text{RF}}(\text{launch}) f_{P_{\text{RF}}}. \quad (4.36)$$

The launched power is measured by a diode attached to the RF transmission line with a directional coupler. Its uncertainty is typically 5% [104]. For fundamental H minority heating at 5.3 T, the uncertainty in $f_{P_{\text{RF}}}$ is small as most of the power is single-pass absorbed. At this field, the uncertainty in P_{RF} was then taken to be $\pm 10\%$. At 2.6 T, $f_{P_{\text{RF}}}$ was typically 0.7 ± 0.1 , which amounts to a 15% uncertainty, while at 8 T, where $f_{P_{\text{RF}}}$ was typically 0.5 ± 0.1 , the uncertainty was $\simeq 20\%$.

Errors

To calculate the overall uncertainty in the global energy confinement time, rewrite (4.31) to explicitly show its dependencies:

$$\tau_E = \frac{U}{P_{\text{oh}} + P_{\text{RF}}}, \quad (4.37)$$

where U is given by either the MHD or kinetic calculation. The error in τ_E is determined by the usual rules of error propagation, as the quantities in (4.38) are all

independent:

$$\left(\frac{\delta\tau_E}{\tau_E}\right)^2 = \left(\frac{\delta U}{U}\right)^2 + \frac{1}{P_{\text{in}}} (\sigma(P_{\text{oh}})^2 + \sigma(P_{\text{RF}})^2). \quad (4.38)$$

Since the uncertainties in the power terms enter absolutely, not relatively, σ_{τ_E}/τ_E must be calculated for each shot in question. Assuming $P_{\text{RF}} = 2P_{\text{oh}}$, and $U_{\text{mhd}} \simeq 50$ kJ, gives

$$\sigma_{\tau_E}/\tau_E = \left[(0.2)^2 + \left(\frac{1}{3} \cdot .1\right)^2 + \left(\frac{2}{3} \cdot .1\right)^2 \right]^{\frac{1}{2}} \simeq 20\%. \quad (4.39)$$

For an RF-heated shot at 2.6 T, once again assuming $P_{\text{RF}} = 2P_{\text{oh}}$,

$$\sigma_{\tau_E}/\tau_E = \left[(0.3)^2 + \left(\frac{1}{3} \cdot .1\right)^2 + \left(\frac{2}{3} \cdot .2\right)^2 \right]^{\frac{1}{2}} \simeq 35\%. \quad (4.40)$$

4.4.4 Calculation of χ

§ 3.1 discusses TRANSP's calculation of χ . Since $\chi/\chi_B = \rho^{*\alpha}$, the ratio of the χ values for a pair of shots, one at high field and one at low field, is:

$$\frac{(\chi/\chi_B)_{\text{hi}}}{(\chi/\chi_B)_{\text{lo}}} = \left(\frac{\rho^{*}_{\text{hi}}}{\rho^{*}_{\text{lo}}}\right)^{\alpha}, \quad (4.41)$$

or for brevity of notation,

$$\mathcal{R}(\chi/\chi_B) = \mathcal{R}(\rho^{*})^{\alpha}. \quad (4.42)$$

Local ρ^{*} scaling was conducted by comparing the profile of $\mathcal{R}(\chi/\chi_B)$ to the profile of $\mathcal{R}(\rho^{*})$.

The error in $\mathcal{R}(\chi/\chi_B)$ is large. Using the methods described in § 3.1.1, the error in χ_{eff} was typically 30% for 5.3 T shots and 40% for shots at other fields with less precise temperature profile or RF power deposition measurements. Treating the error in χ_{eff} for each shot as independent, the error in $\mathcal{R}(\chi_{\text{eff}}/\chi_B)$ for a pair of shots at 5.3 T and 2.6 T is 50%. Since the errors in χ_e and χ_i can be 100%, the error in their ratios will be enormous; too large to unambiguously determine the scaling with ρ^{*} . If the energy balance calculations for each species appear reasonable, so that the χ_e and χ_i profiles are not pathological, the scaling of the center of their uncertainty ranges will be examined, bearing in mind that the error is very large.

4.4.5 Regression Analysis

Global ρ^* scaling was conducted by regressing $B\tau_E$ versus ρ^* from sets of dimensionally similar shots. A weighted linear least-squares method [102, 105, 106] was applied to the logarithms of $B\tau_E$ and ρ^* to obtain a scaling:

$$\log B\tau_E = a_0 + a_1 \log \rho^*. \quad (4.43)$$

Comparing (4.43) to the logarithm of (4.14) gives:

$$\alpha = -2 - a_1. \quad (4.44)$$

Implicit in this technique is the assumption that the uncertainty in the independent variable is small. Since the error in ρ^* can be as large as 10% from its dependence on T_i , this assumption is not valid. To include the effects of the uncertainty in ρ^* on the regression, it was added in quadrature to the uncertainty in the dependent variables, following [102]. If the regression were performed against B_t , which is technically the independent variable, this would not be necessary as its error is very small, but then to infer the ρ^* scaling, the scaling of ρ^* (or T_i) with B_t would also need to be measured.

There are other possibly invalid assumptions behind the least-squares technique that are not as easily mitigated. The estimation of the uncertainties in the coefficients obtained from the regressions relies on the data being independent with normally distributed errors. Both these factors are not necessarily true for the sets analyzed for ρ^* scaling.

The assumption of independence relates to the degrees of freedom n_{dof} available for the fit. The independent variable for the ρ^* scaling experiments, B , was restricted to the five values covered by the data in the core table of *LOGBOOK*: 2.5, 3.5, 5.3, 6.5, and 8.0 T. The sets of dimensionally similar shots sample at most three of these fields, so instead of being distributed uniformly over the independent variable range, the data is clustered at each field. Furthermore, the data within a cluster at a given field may contain several time slices from the same shot. Both these factors act to reduce the degrees of freedom, increasing the uncertainty in the regression coefficients [77, 105–107].

Rather than delving into complicated statistical theory, the following heuristic method, which will subsequently be referred to as the *field clustering method*, was used to attempt a better estimate of the regression errors. A error-weighted mean $\overline{B\tau_{E,k}}$ was determined from the N_k slices in each field cluster k by [102]

$$\overline{B\tau_{E,k}} = \frac{1}{S_k} \sum_{i=1}^{N_k} \frac{B_i \tau_{Ei}}{\sigma_{\tau_{Ei}}^2} \quad (4.45)$$

where $\sigma_{B,\tau E_i}^2$ is the error in the i th value of $B\tau_E$, and

$$S_k = \sum_{i=1}^{N_k} \frac{1}{\sigma_{\tau E_i}^2}. \quad (4.46)$$

The weighted error in $\overline{B\tau_{E,k}}$ is $\sigma(\overline{B\tau_{E,k}}) = 1/\sqrt{S_k}$.

The mean values $\overline{B\tau_{E,k}}$ were used to determine the coefficients a_0 and a_1 of the regression, which is essentially what happens if all points are treated independently. The difference between the techniques is in the calculation of the error in the coefficients. The error in a_1 was determined from the range of a_1 values that would fit the $\overline{B\tau_{E,k}}$ values within their errors $\sigma(\overline{B\tau_{E,k}})$.

Aside from the question of the number of degrees of freedom, there are other potential problems with these regressions. The error calculations are based on the assumption that the errors in each point of the dataset are random, with a normal distribution [102,105]. It is possible that the scatter of the residuals is the result of dependence on uncontrolled, possibly hidden variables, so that they are not randomly distributed, but instead correlated. The meaning of the uncertainty in the fit coefficients a_0 and a_1 would then be clouded as they would no longer refer to standard confidence intervals. The residuals were not thoroughly examined for correlations, as the number of plasma parameters associated with each shot in the set is large. Instead, they were simply checked for dependence on ν^* and β_p in the process of measuring the sensitivity of the regression to mismatches in those parameters.

4.4.6 Field Ratio and Sensitivity

To minimize the uncertainty in determining the ρ^* scaling of χ or $B\tau_E$, the field range used should be as wide as possible, increasing the ratio of the χ or $B\tau_E$ values relative to its uncertainty.

For local analysis, the ρ^* scaling was determined using only one shot at each of the high and low fields. Referring to (4.41),

$$\alpha = \frac{\log(\mathcal{R}(\chi/\chi_B))}{\log(\mathcal{R}(\rho^*))}. \quad (4.47)$$

Assuming no uncertainty in ρ^* , the uncertainty in α is:

$$\sigma_\alpha = \frac{1}{\mathcal{R}(\rho^*)} \frac{\sigma(\mathcal{R}(\chi/\chi_B))}{\mathcal{R}(\chi/\chi_B)}. \quad (4.48)$$

In § 4.4.4 it was shown that the error in the ratio of χ/χ_B at high field to that at low field is typically 50%. To keep $\sigma_\alpha < 0.5$ requires $\mathcal{R}(\rho^*) > 2.7$, or since $\rho^* \sim B^{-2/3}$, the field ratio should be $B_{hi}/B_{lo} > 4.5$. A field ratio this large cannot be achieved on Alcator C-Mod.

Global ρ^* scaling was determined from regressions on datasets with more than two shots, so a similar calculation for the required field ratio is more complicated. It can be shown that the error in α from the regression is in fact reduced by increasing B_{hi}/B_{lo} .

$B_{hi}/B_{lo} = 2$ was selected as the minimum ratio allowable, leading to the following possible field pairs: $B_1 = 2.5$ T, $B_2 \geq 5.3$ T and $B_1 = 3.3$ T, $B_2 \geq 6.5$ T. Since most C-Mod shots are at 5.3 T, most of the pairs selected for analysis will be at (2.5/5.3).

4.5 Results

The results of the ρ^* scaling analysis on Alcator C-Mod are presented in this section. For each of L-Mode and H-Mode, global scaling results are presented first followed by local scaling for each pair of shots examined. § 4.7 provides a summary.

4.5.1 L-Mode Global Analysis

The L-Mode shots selected from the core table of *LOGBOOK* for analysis were all single-null lower diverted D₂ plasmas with $\kappa \in [1.5, 1.7]$. Shots run earlier than 1995 were excluded.⁵ This set of shots will be referred to as \mathcal{L} .

Before identifying sets of similar shots within \mathcal{L} for ρ^* scaling analysis, a regression of the entire data set was performed to get an overall picture of the scaling. Figure 4.2 plots a regression of $\tau_{E,kin}$ against I_p , n_e , and P_{in} . The other two standard engineering parameters, B_t and R , were not included in the regression as their range of values in the database was very limited. The resulting scaling is

$$\tau_E = 0.046 I_p^{0.86 \pm 0.05} n_e^{0.26 \pm 0.04} P_{in}^{-0.55 \pm 0.03} \quad (4.49)$$

with an RMS error of 18% and a linear correlation coefficient $r = 0.79$. The scaling is nearly identical to the ITER89-P scaling (2.38). Figure 4.3 plots a regression of $B\tau_{E,kin}$ against the dimensionless parameters ρ^* , ν^* , β_p , and $q_{\psi 95}$. The result is

$$B\tau_E = 1.8 \rho^{*-1.7 \pm 0.07} \nu^{*-0.27 \pm 0.02} \beta_p^{0.13 \pm 0.07} q_{\psi 95}^{0.15 \pm 0.09} \quad (4.50)$$

with an RMS error of 25% and a linear correlation coefficient of 0.91. Referring to table 4.2, $B\tau_E \sim \rho^{*-1.7}$ is consistent with a Bohm-like global scaling. The larger errors on β_p and $q_{\psi 95}$ reflect the fact that the range of those parameters covered by the database is quite small. Converting (4.49) to dimensionless form gives:

$$B\tau_E \sim \rho^{*-1.8 \pm 0.2} \nu^{*-0.11 \pm 0.55} \beta_p^{-0.56 \pm 0.11} q_{\psi 95}^{-1.9 \pm 0.2}, \quad (4.51)$$

⁵The complete SQL query for selection was: (topic='rhostar' or topic containing 'confinement') and (gas_maj = 'D2') and (topology='SNB') and (kappa between 1.5 and 1.7) and (shot between 950000000 and 960399999).

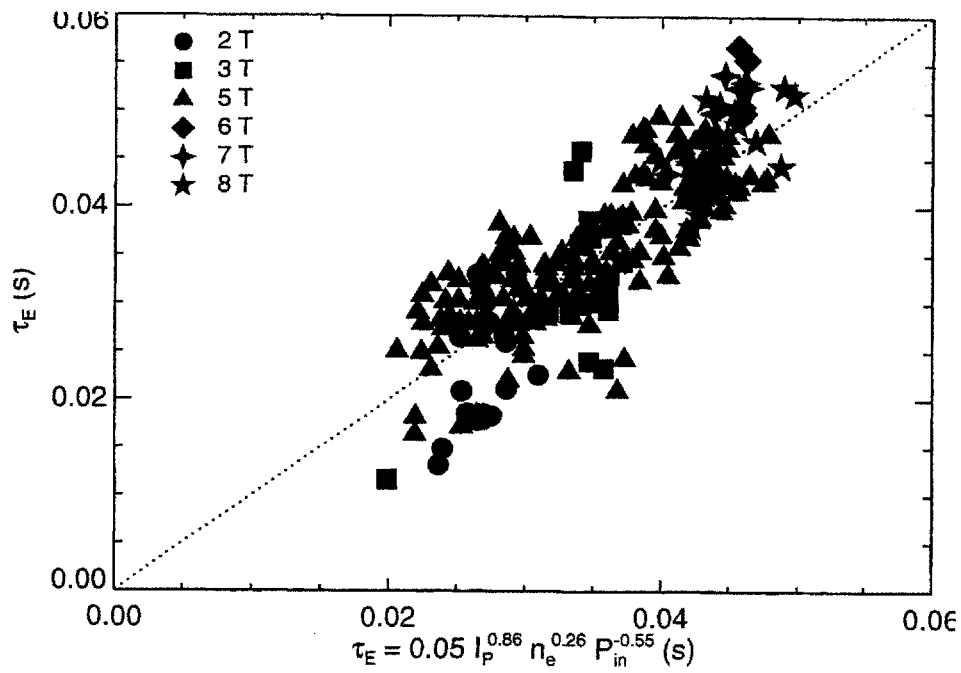


Figure 4.2: Regression of τ_E against engineering parameters for L-Mode

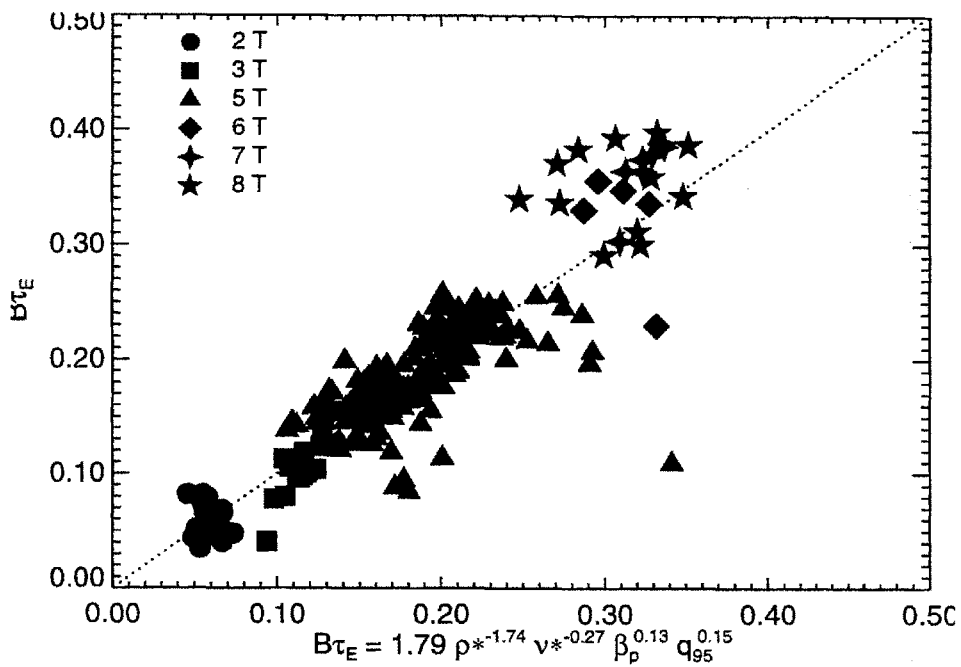


Figure 4.3: Regression of τ_E against dimensionless parameters for L-Mode

Group	$q_{\psi 95}$		β_p		ν^*	
	average	s.d.	average	s.d.	average	s.d.
La	4.18	0.02	0.26	0.00	0.54	0.03
Lb	5.14	0.04	0.28	0.01	0.61	0.04
Lc	5.04	0.06	0.24	0.01	0.21	0.02
Le	4.09	0.03	0.26	0.01	0.31	0.02

Table 4.6: L-Mode shot groups for global scaling

which has a similar ρ^* and ν^* dependence but very different dependencies on β_p and $q_{\psi 95}$. (4.50).

To isolate the ρ^* dependence of B_{TE} , groups of similar shots were selected from \mathcal{L} . The set was first broken down into three ranges of $q_{\psi 95}$. Scatter plots of β_p versus ν^* for these ranges of $q_{\psi 95}$ are given in figure 4.4. Note that the different symbol shapes in figure 4.4 are used to identify the different B_t ranges as shown in the legend. This convention will be used throughout the remainder of this section whenever it is desired to identify the magnetic field in a scatter plot.

Since the vast majority of C-Mod shots are at 5.3 T, the process of identifying suitable groups began with locating groupings of shots at different fields; four of these groupings are identified in figure 4.4(a), two in figure 4.4(b), and two in figure 4.4(c). A box covering the error ranges in β_p and ν^* was plotted around one shot from each of the identified groupings, and the other shots lying within were further examined for suitability. Shots were considered suitable if their parameters satisfied:

$$|\beta_p - \beta_{p0}| < 0.05 \quad (4.52)$$

$$|\nu^* - \nu^*_0|/\nu^*_0 < 30\% \quad (4.53)$$

$$|q_{\psi 95} - q_{\psi 95_0}|/q_{\psi 95_0} < 5\% \quad (4.54)$$

as well as having matched Z_{eff} , κ , R , and a . Boundary shapes were not matched for global analysis. The data in \mathcal{L} covered the range $\kappa \in (1.5, 1.7)$, $\delta_l \in (0.4, 0.6)$, $\delta_u \in (0.2, 0.4)$; the range of these parameters within identified groups of similar shots was usually narrower. Whenever possible, the fractional ICRF power absorption f_{RF} was measured explicitly for RF-heated shots at 2.6 and 8 T. If the measurement was not possible, f_{RF} was taken to be 70% for 2.6 T shots and 50% for 8 T shots, as discussed in § 3.5.3.

Of the eight regions of $(\beta_p, \nu^*, q_{\psi 95})$ space identified for further investigation, four contained dimensionally similar shots suitable for ρ^* scaling analysis. Parameters for these groups are summarized in table 4.6. The following sections present the results from each group in turn.

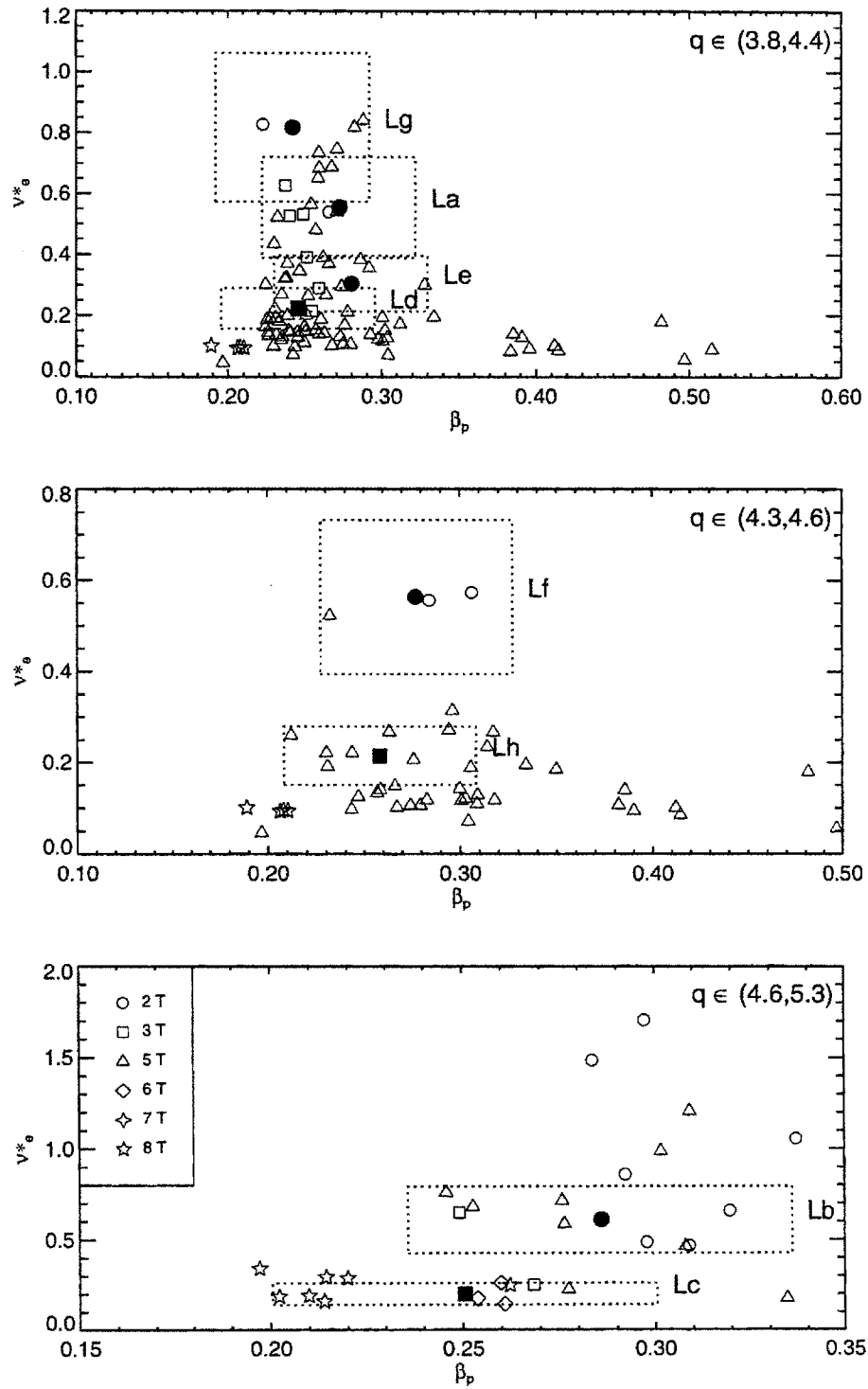


Figure 4.4: $q_{\psi 95}$ and ν^* vs. β_p for L-Mode shots. Regions examined for dimensionally similar shots marked by boxed regions.

Group *a*

All shots in this group have $q_{\psi 95} \in [3.8, 4.4]$. Figure 4.5(a) plots ν^* versus β_p for them.

The first order of business is to compare the two methods of calculating τ_E (discussed in § 4.4.3) to identify any pathological shots. As shown in figure 4.5(b), the two methods agree within the error for most of the shots in the group. No obvious differences can be identified between these shots and the two 5.3 T shots for which the difference in the two calculations is larger than the error.

The next step is to check that the scaling of the group is consistent with L-Mode. Figure 4.5(c) plots the calculation of τ_E using the ITER89-P L-Mode scaling expression versus the measured kinetic τ_E . It is evident that the shots do indeed show an L-Mode scaling within the estimated error of measurement.

Figure 4.6 plots the result of a regression of $B\tau_E$ versus ρ^* . The result is

$$B\tau_E = 3.1\rho^{*-2.0\pm 0.4} \text{ T s}, \quad (4.55)$$

with a reduced $\chi^2 = 0.3$, reflecting the fact that the measurement errors are very large compared to the average residual. The linear correlation coefficient is $r = -0.95$ indicating a strong inverse correlation between $B\tau_E$ and ρ^* . The dotted lines in figure 4.6 show the extremes of the error ranges δa_0 and δa_1 of the parameters in the fit (4.43).⁶

$$\begin{aligned} \tau_{E_{max}} &= e^{\delta a_0} a_0 \rho^{*a_1 - \delta a_1} \\ \tau_{E_{min}} &= e^{-\delta a_0} a_0 \rho^{*a_1 + \delta a_1}. \end{aligned}$$

These errors are the standard deviations of the parameters determined as part of the regression algorithm. Using (4.44) with $a_1 = -2.0 \pm 0.4$ gives $\alpha = 0.0 \pm 0.4$, which upon reference to table 4.2 indicates Bohm scaling, though the error range admits the possibility of Goldston scaling. Applying the field clustering technique from § 4.4.5 to better estimate the uncertainty in the regression coefficients gives $B\tau_E \sim \rho^{*-2.0\pm 0.6}$, an increase of 50% in the error. Using this error in the calculation of α gives $\alpha = 0.0 \pm 0.6$, which still excludes gyroBohm scaling.

Figures 4.7(a) and (b) address the dependence of the residuals on ν^* and β_p issue as well as the sensitivity of the scaling result to the matches in them.

For the former figure, the constraint (4.53) was relaxed, introducing shots with similar β_p and $q_{\psi 95}$, but with any ν^* value. $B\tau_{E_{fit}}$ was calculated for these shots using the result (4.55). Figure 4.7(a) plots the residual $B\tau_E/B\tau_{E_{fit}}$ versus the ratio of ν^* to the average value of ν^* of the dimensionally similar shots. For the dimensionally similar shots (solid points), the residuals show no obvious dependence with ν^* over the $\pm 30\%$ range in ν^* covered. For ν^* values substantially different from that used

⁶The same method of displaying the range of uncertainty is used for all regression plots.

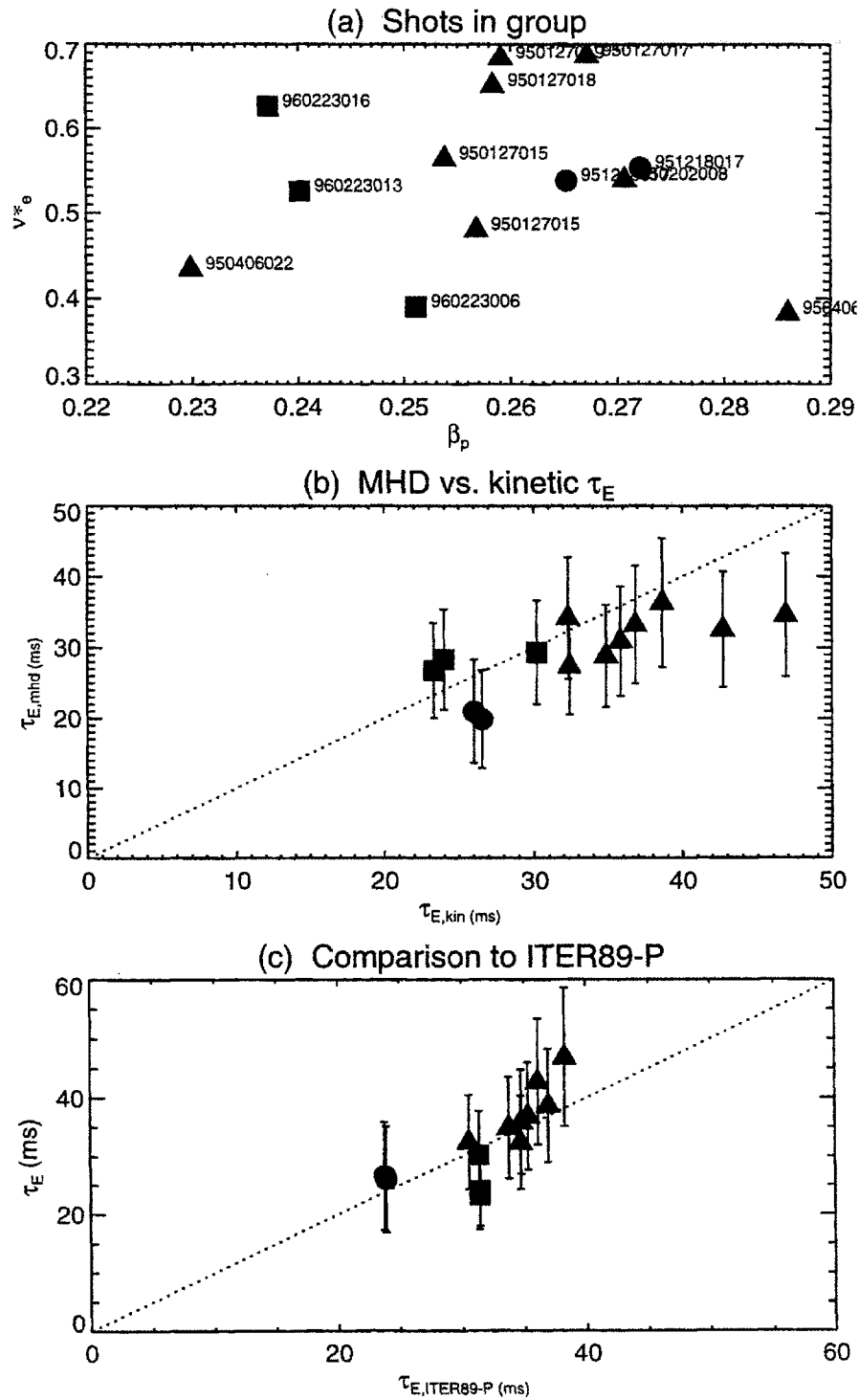


Figure 4.5: Preliminary checks for global scaling, L-Mode group *a*

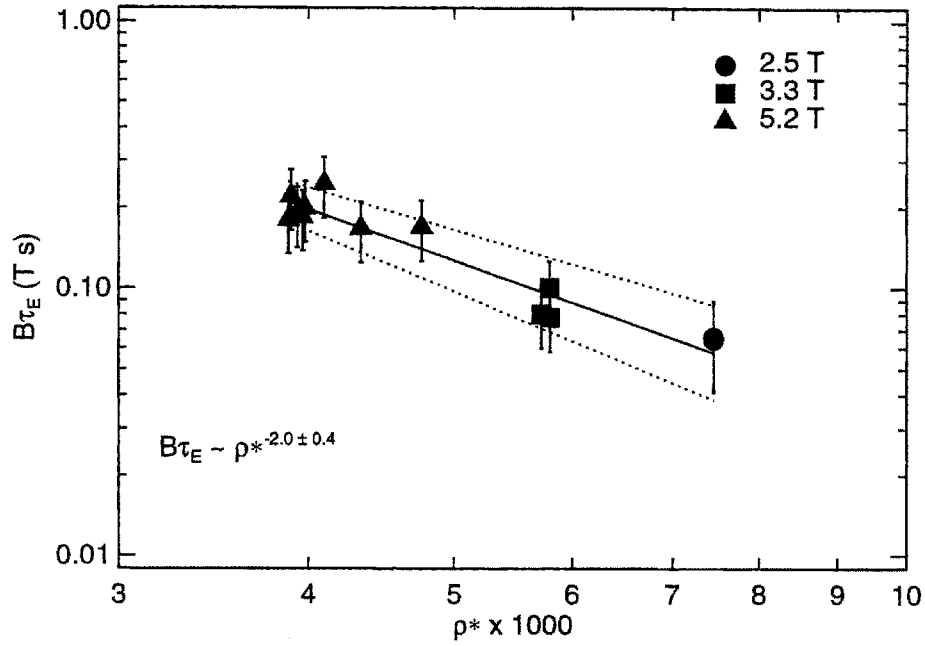


Figure 4.6: $B\tau_E$ versus ρ^* for L-Mode group a

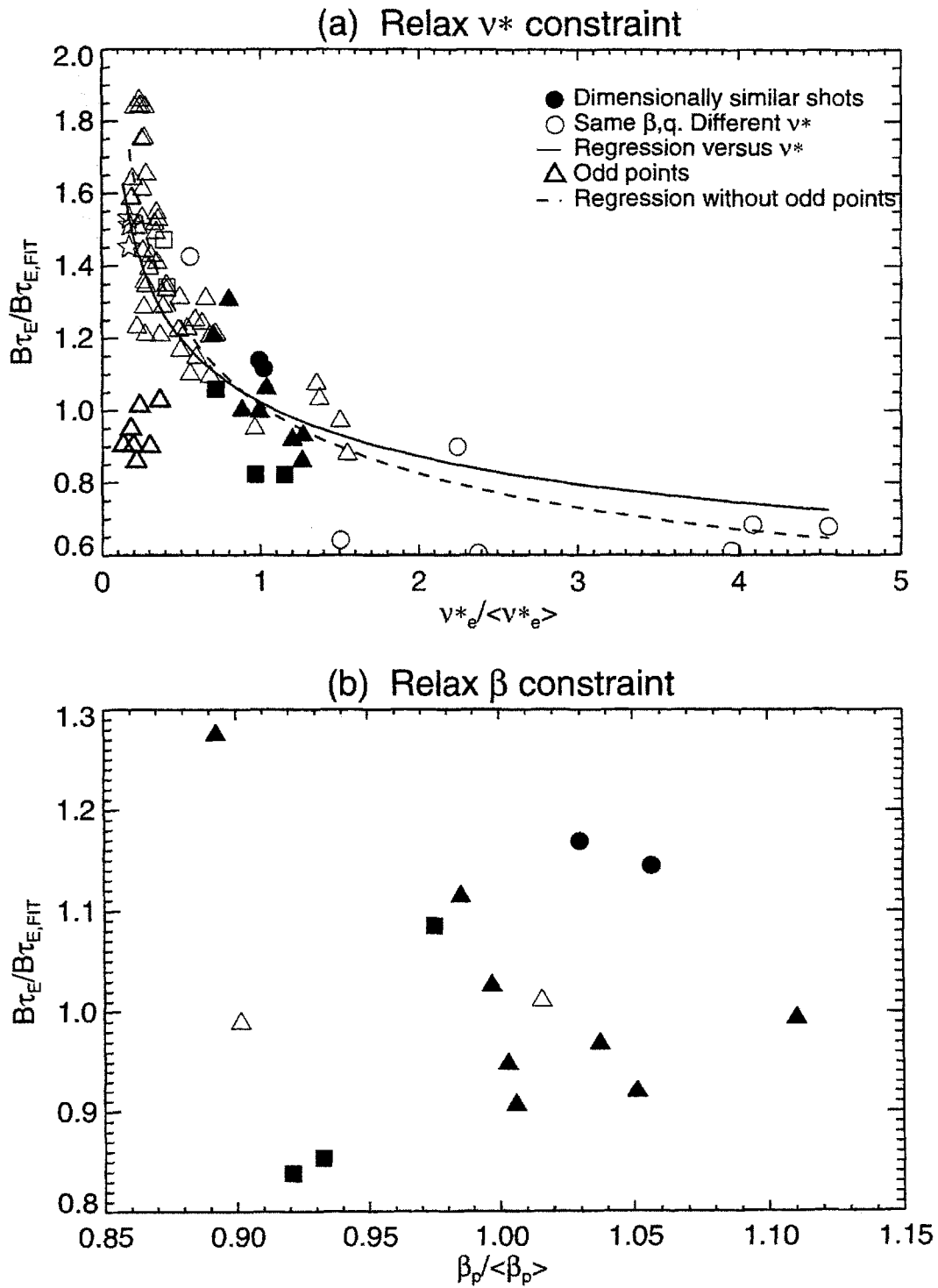
in the fit, the residuals do show an inverse ν^* scaling, as evidenced by the solid line, plotting the regression versus ν^* :

$$\frac{B\tau_E}{B\tau_{E_{\text{fit}}}} \sim \nu^{*-0.23 \pm 0.04}. \quad (4.56)$$

For the 30% error range in ν^* , a scaling of $\nu^{*-0.23}$ gives a 6% variation in $B\tau_E$, well within the error, so matching ν^* within 30% should be sufficient for ρ^* scaling. Several shots marked on figure 4.7(a) do not fit the residual ν^* scaling. Most of these are from one run day (960127) and are RF-heated. The kinetic calculation of the stored energy for these shots is up to 40% lower than the MHD calculation. Though nothing else could be found to distinguish these shots from the rest, the regression was repeated without them, with the result shown by the dashed line:

$$\frac{B\tau_E}{B\tau_{E_{\text{fit}}}} \sim \nu^{*-0.30 \pm 0.04}. \quad (4.57)$$

The same is attempted for β_p sensitivity, with the results shown in figure 4.7(b). The range of β_p available is much narrower. No β_p dependence is evident in the residuals, and there are not enough points to allow a proper regression against β_p .

Figure 4.7: Sensitivity of fit to ν^* and β_p , Group *a*

Group *b*

The next group of shots examined have $q_{\psi 95} \in [4.9, 5.3]$. Figure 4.8(a) plots ν^* versus β_p for them.

A comparison of the two methods of calculating τ_E is plotted in figure 4.8(b). Here, $\tau_{E,\text{mhd}}$ is systematically lower than $\tau_{E,\text{kin}}$. Though most of the 5.3 T shots have $\tau_{E,\text{mhd}} \approx \tau_{E,\text{kin}}$ within error, the difference between the two methods is as much as 60% (= 12 ms) for the low field shots. The systematic difference could be the result of EFIT overestimating ℓ_i . Wolfe explains [101] that β_p is often negative at the start of a shot, suggesting that ℓ_i is too high. Removing the constraint on q_0 returns an even higher ℓ_i , so it is possible that the constraint is not strict enough.

As shown in figure 4.8(c), the $\tau_{E,\text{kin}}$ values are consistent with the ITER89-P scaling, so a Goldston-like scaling with ρ^* is expected. Figure 4.9 plots the regression of $B\tau_E$ versus ρ^* ,

$$B\tau_E = 0.9\rho^{*-1.2\pm 0.3}. \quad (4.58)$$

The goodness of fit parameters are: χ^2 (reduced) = 0.18, $r = -0.94$, indicating once again that the average uncertainty is much larger than the average residual, and that there is a strong inverse correlation between $B\tau_E$ and ρ^* . Calculating α from (4.58) gives $\alpha = -0.8 \pm 0.3$, which suggests a Goldston-like scaling upon reference to table 4.2. Regressing τ_E versus B_t , as shown in figure 4.10, confirms that the scaling is Goldston-like:

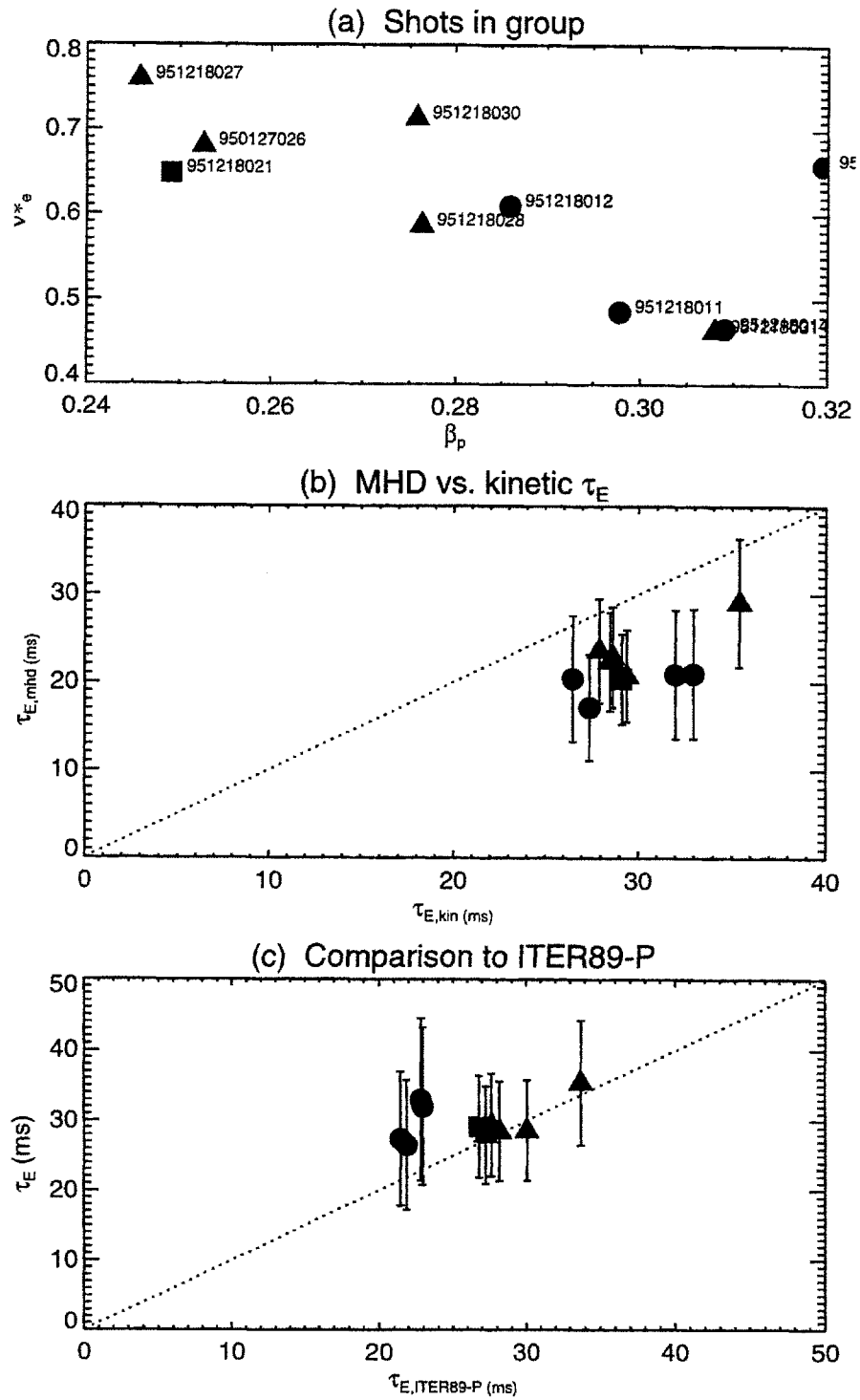
$$\tau_E = (29 \pm 5 \text{ ms/T}) B_t^{0.0\pm 0.3}. \quad (4.59)$$

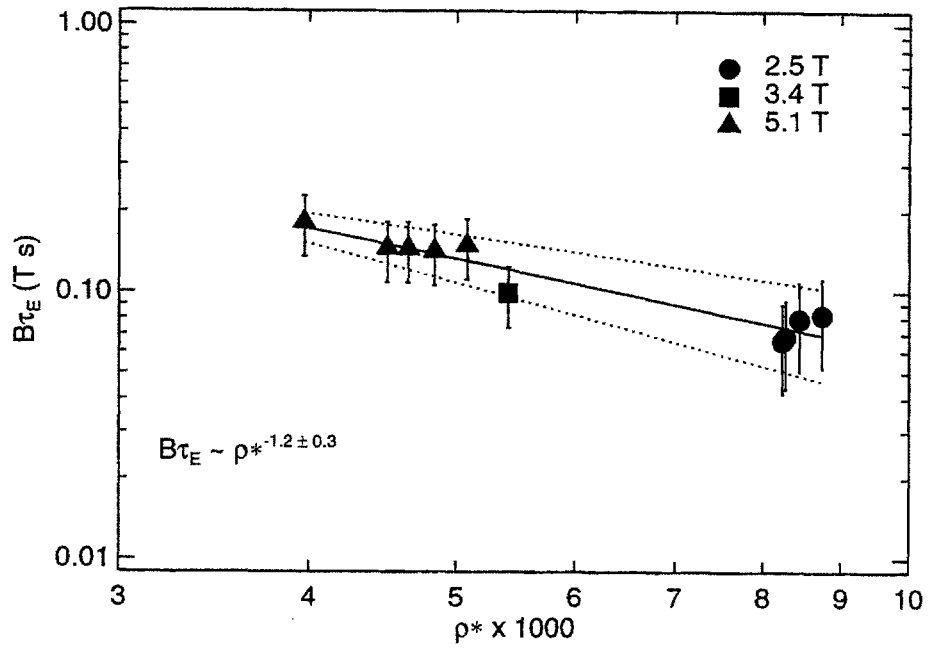
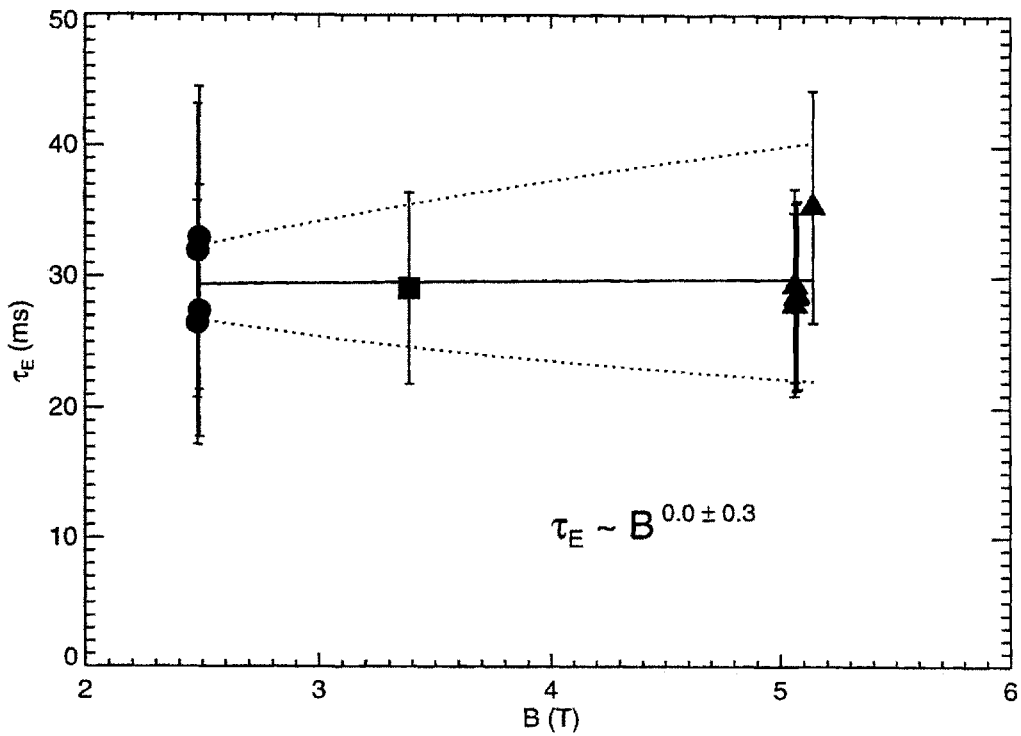
The aforementioned regressions were conducted assuming independent measurements. Applying the field clustering technique described for group *a*, the estimation of the error including the reduction in the degrees of freedom is ± 0.5 , which still indicates Goldston-like scaling unambiguously.

Sensitivity to variations in ν^* and β_p are examined in figures 4.11(a) and (b). While the residuals of the shots in group *b* show no systematic dependence on either parameter, relaxing the constraint on ν^* shows that there is once again a weak inverse scaling with ν^* . Regressing the residual versus ν^* for this dataset gives

$$\frac{B\tau_E}{B\tau_{E\text{fit}}} \sim \nu^{*-0.34\pm 0.07}. \quad (4.60)$$

The points at the lowest ν^* values seem to be clustered into two groups, though no obvious differences could be identified between them. These shots actually form part of group *c*, and will be examined in the next section. Not enough shots were added by relaxing the constraint on β_p to permit a sensitivity study.

Figure 4.8: Preliminary checks for global scaling, L-Mode group *b*

Figure 4.9: $B\tau_E$ versus ρ^* for L-Mode group b Figure 4.10: τ_E versus B_t for L-Mode group b

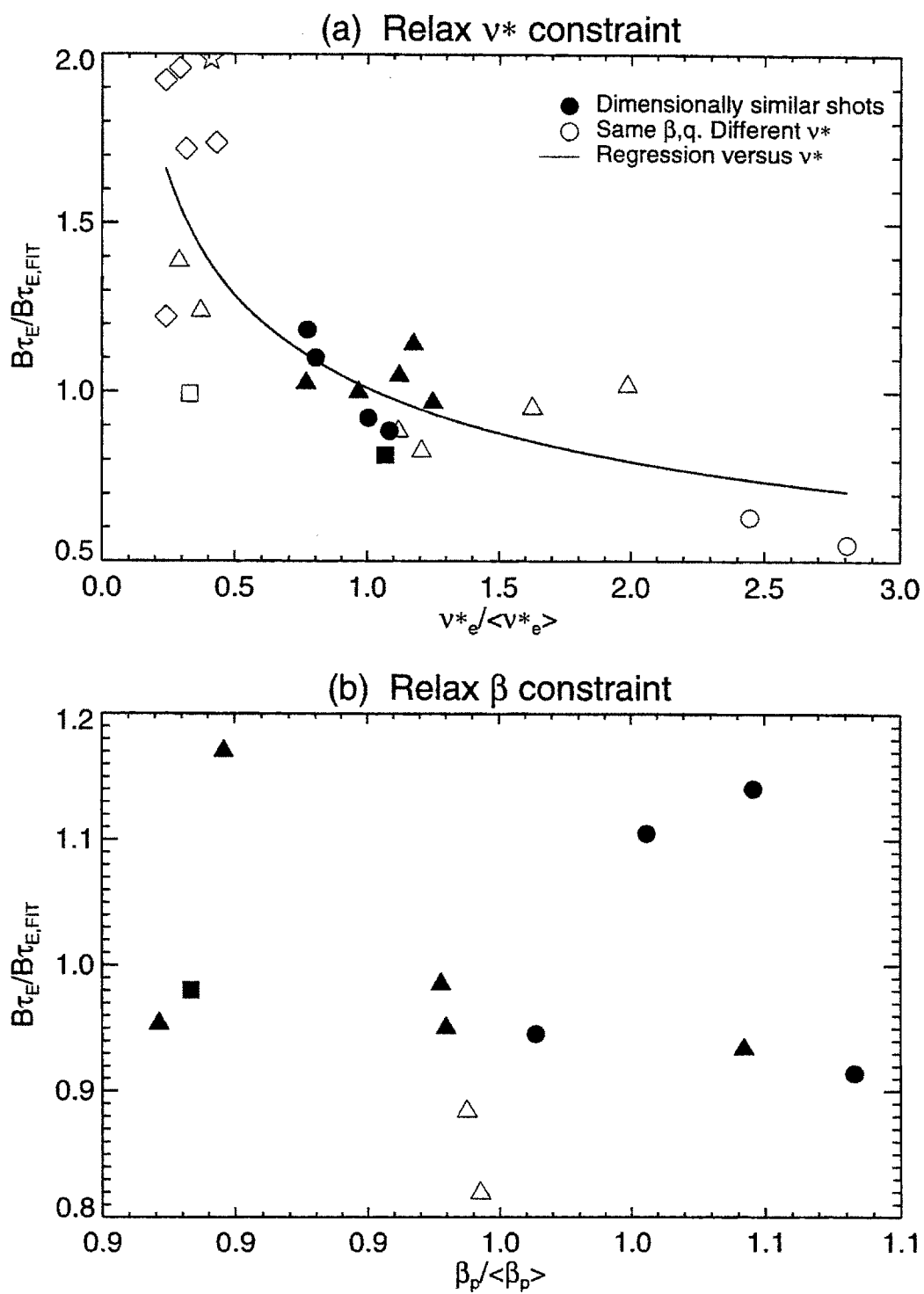


Figure 4.11: Sensitivity of fit to ν^* and β_p , Group *b*

Group c

The same six plots are presented for group *c* in figure 4.12. This group, with $q_{\psi 95} \in [4.9, 5.3]$, is the only one analyzed containing 8 Tesla shots. All shots in this group show good agreement between the two methods of calculating τ_E , and between τ_E and $\tau_{EITER89-P}$. A regression of $B\tau_E$ versus ρ^* treating all points as independent gives

$$B\tau_E = 6\rho^{*-2.2\pm 0.4}, \quad (4.61)$$

or $\alpha = 0.2 \pm 0.4$, indicating Bohm-like scaling within the uncertainty. The field clustering method increases the uncertainty to ± 0.7 , so that gyroBohm scaling is possible. Once again the residuals suggest a weak inverse dependence with ν^* :

$$\frac{B\tau_E}{B\tau_{E_{fit}}} \sim \nu^{*-0.24\pm 0.08}. \quad (4.62)$$

They may also show a weak dependence on β_p , though there are too few points to be confident about this.

Group e

Figure 4.13 shows the same plots for group *e*, with shots having $q_{\psi 95} \in [3.8, 4.4]$. Nothing unusual is notable about the comparison of the τ_E calculations methods. Once again the shots are consistent with an ITER89-P scaling. The regression gives

$$B\tau_E = 2\rho^{*-1.7\pm 0.8}, \quad (4.63)$$

where the uncertainty quoted has been calculated using the field clustering method. Note that there is only one 2.5 T shot for this group, so this uncertainty is particularly high. Many shots were added by relaxing the ν^* constraint as shown in figure 4.13(e), so the scaling of the residual with ν^* can be more confidently stated as:

$$\frac{B\tau_E}{B\tau_{E_{fit}}} \sim \nu^{*-0.31\pm 0.04}. \quad (4.64)$$

This scaling gives an 8% difference in $B\tau_E$ from a 30% variation in ν^* . Nothing enlightening is had from the plot of the residuals versus β_p .

Summary

Table 4.7 summarizes the results of the regressions of $B\tau_E$ versus ρ^* , and the residual dependence on ν^* , for the four groups studied. The errors in α are calculated using the field clustering method.

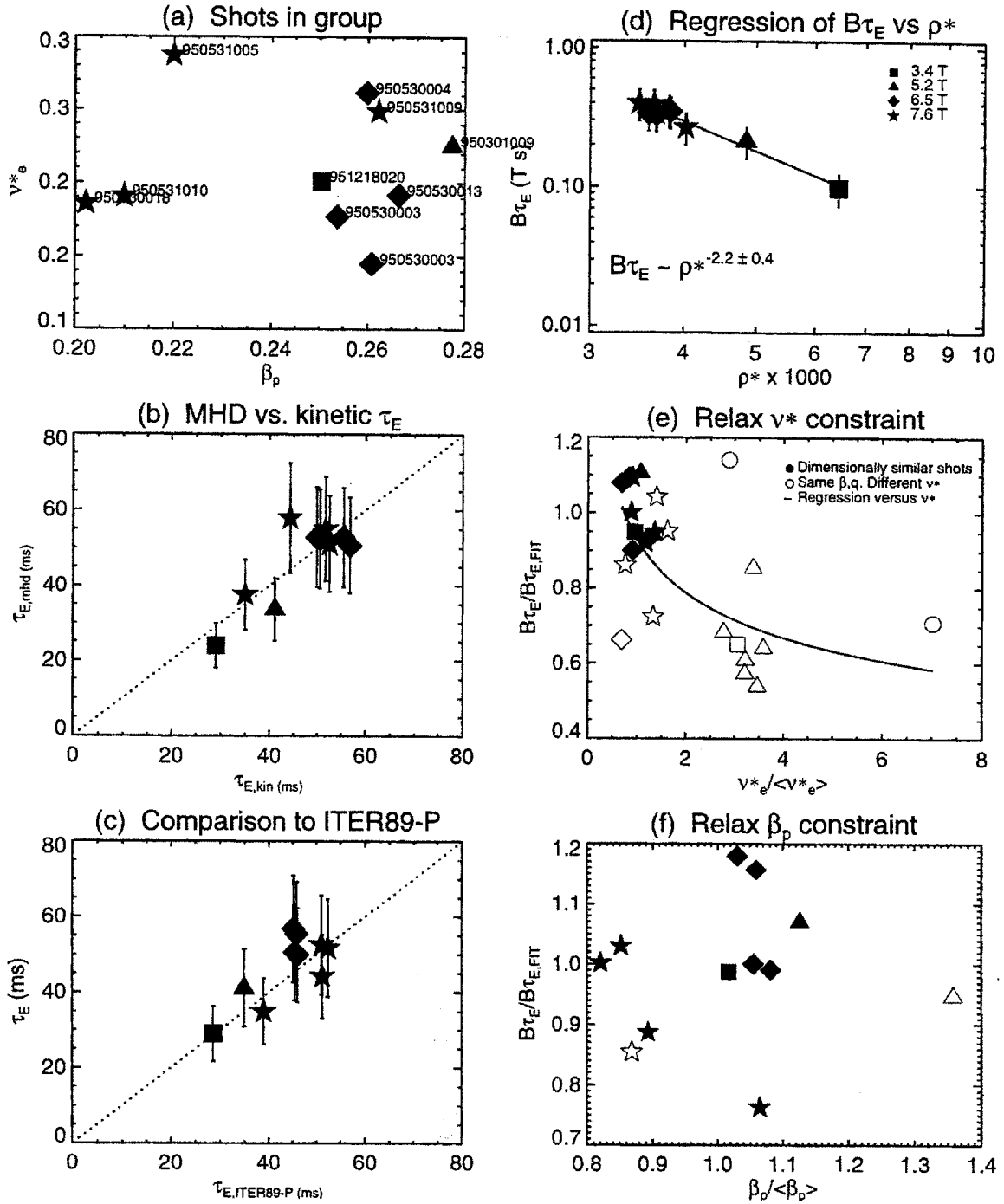


Figure 4.12: Global Analysis for L-Mode group c

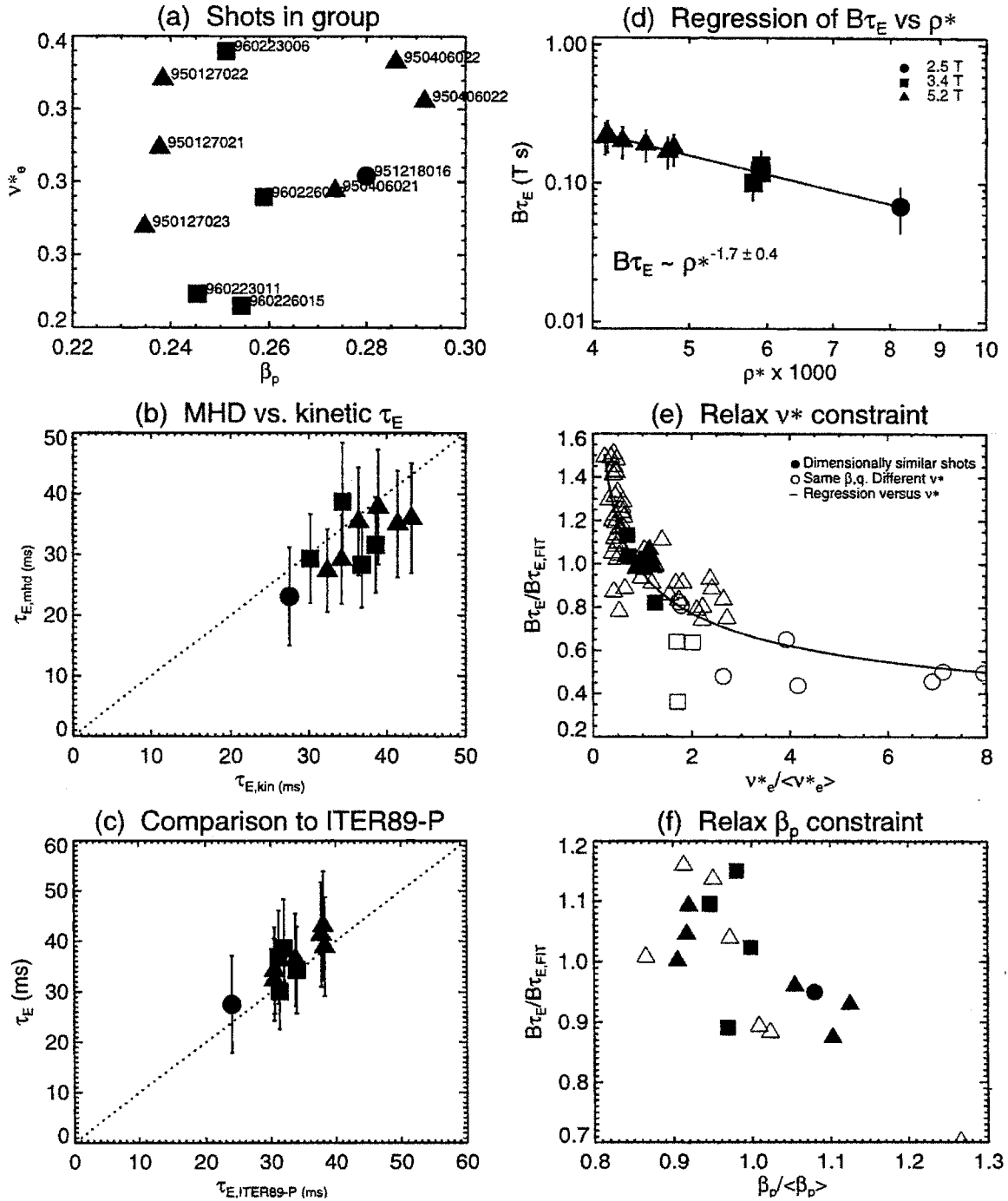


Figure 4.13: Global Analysis for L-Mode group e

Group	$B\tau_E \sim \rho^* a_1$	$\alpha = -2 - a_1$	Scaling	Residual $\sim \nu^* b_1$
La	-2.0 ± 0.6	0.0 ± 0.6	Bohm	-0.30 ± 0.04
Lb	-1.2 ± 0.5	-0.8 ± 0.5	Goldston	-0.34 ± 0.07
Lc	-2.2 ± 0.7	0.2 ± 0.7	Bohm	-0.24 ± 0.08
Le	-1.7 ± 0.8	-0.3 ± 0.8	Goldston-Bohm	-0.31 ± 0.04

Table 4.7: Global scaling for L-Mode groups studied

4.5.2 L-Mode Local Analysis

Ideally, χ data would be available for all the shots in the core table, so that the regressions performed for global analysis could also be performed for local analysis. Alas, preparing TRANSP runs for that many shots would be a monumental task. Furthermore, in local ρ^* scaling analysis, the dimensionless parameter *profiles* must match within error for shots to be considered similar, so not all of the shots used for global analysis would qualify for local analysis.

Because of these difficulties, local ρ^* analysis was conducted on only one pair of shots — one at low field and one at high field — taken from each of the groups identified for global analysis. The results are presented in the following sections.

Group *a*

The two shots chosen from group *a* for local analysis are 951218017 and 950127015. Both shots, at 2.6 and 5.3 T respectively, are ohmically heated. The top half of table 4.8 provides a summary of their engineering parameters. Note that the scalings of the total stored energy and the volume average electron density are close to what should be obtained for ρ^* scaling, that is $U \sim B^2$ and $n \sim B^{4/3}$. Figure 4.14 shows time histories of important plasma parameters for the two shots, with the region averaged over for local analysis shaded. The time slices t_0 chosen for local analysis are well into the flattop for both shots. 951218017 was in steady state for several hundred milliseconds around t_0 , while the density for shot 950127015 was ramping slightly.

The bottom half of table 4.8 lists the scalar dimensionless parameters for the two shot, which except for $P_{\text{rad}}/P_{\text{in}}$ match within error. For local ρ^* analysis to be valid, β_t and ν^* must match over the entire profile, which they do as shown in figure 4.15.

The local ρ^* scaling is addressed in figures 4.16(a)-(d). Before describing each of the plots, note first the convention followed for identifying the traces in them: the electrons are identified by the diamonds \diamond , ions by the squares \square , and total or effective quantities by the circles \circ . In plot (c), which shows the χ profiles from each shot, filled symbols represent the low field shot (951218017 here) and open symbols represent the high field shot (950127015).

Figure 4.16(a) plots the ratio of the volume-integrated input power profile at high

Shot	951218017	950127015
t_0 (s)	0.85	0.90
B_t (T)	2.5	5.3
I_p (MA)	0.4	0.8
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	0.89	2.89
P_{RF} (MW)	0.00	0.00
U/B^2	2.24	2.01
$\langle n_e \rangle / B^{4/3}$	0.25	0.31
β_t	0.55	0.54
β_p	0.40	0.41
$q_{\psi 95}$	4.15	4.02
ν^*_e	0.56	0.87
T_i/T_e	0.74	0.81
κ	1.54	1.60
Z_{eff}	1.45	1.05
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.32	0.12
$\rho^* \times 1000$	4.78	2.73
$B\tau_E$ (T s)	0.07	0.20
TRANSP run	TRANSP01	TRANSP02

Table 4.8: Parameters for L-Mode, group a

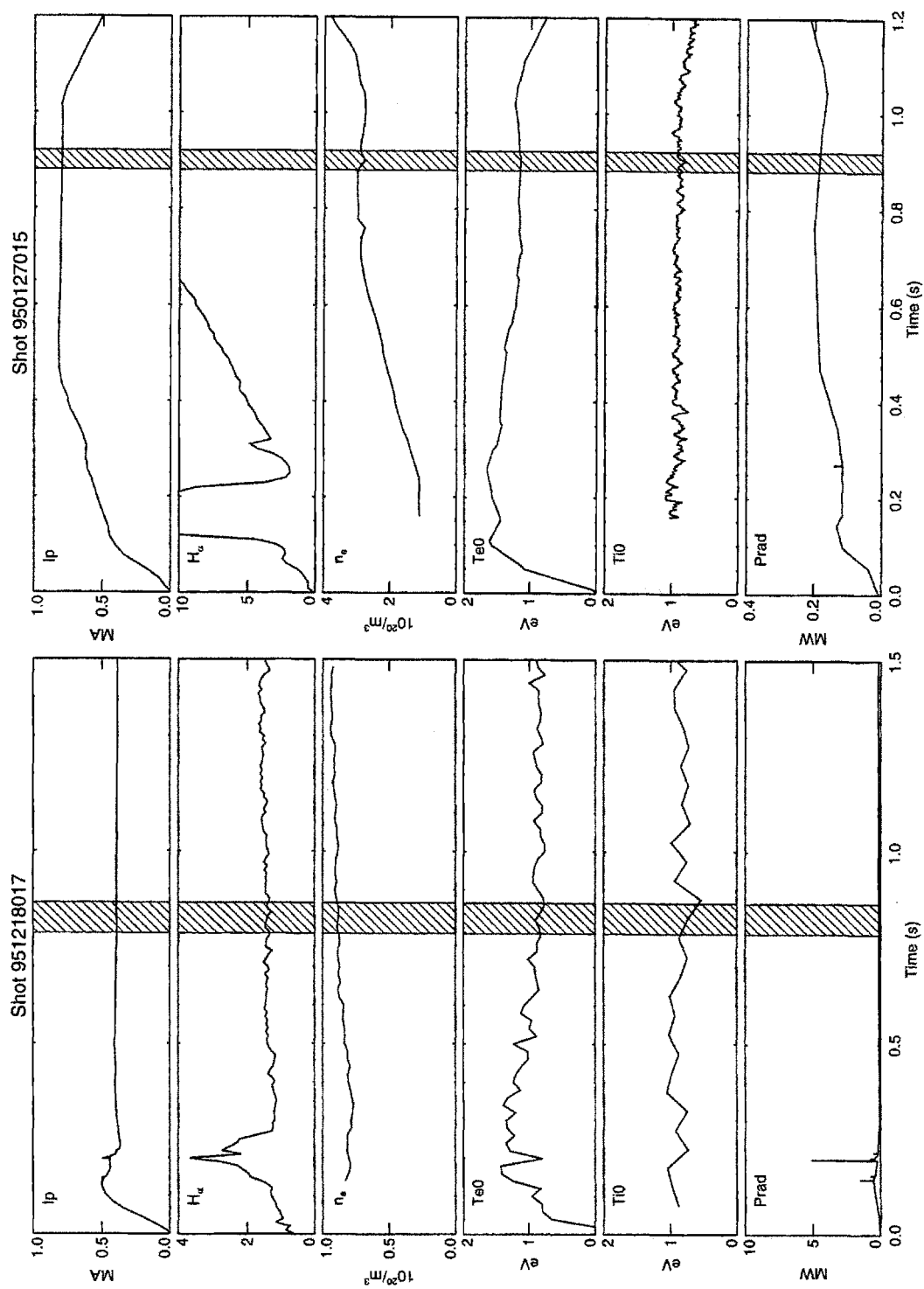


Figure 4.14: Time history of shots in L-Mode group *a*. Shot 951218017: $B_t = 2.5$ T. Shot 950127015: $B_t = 5.3$ T.

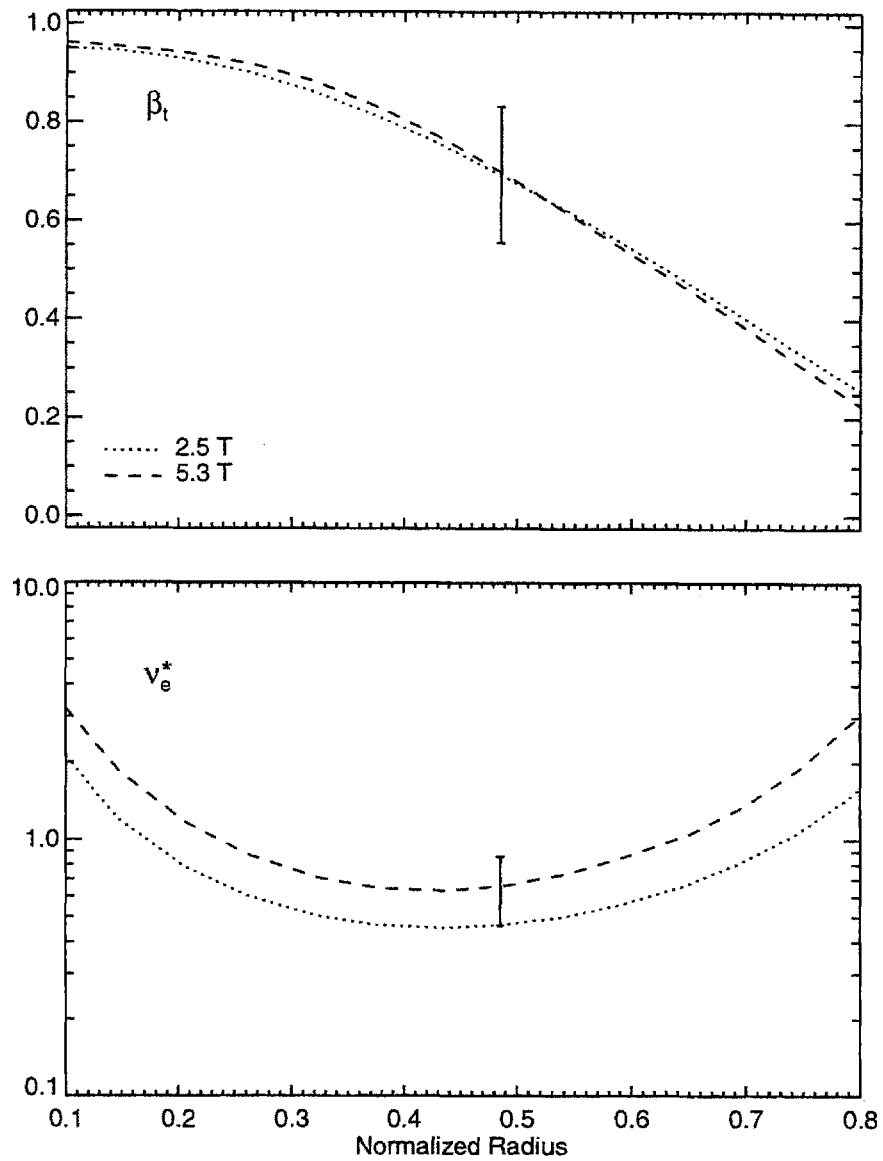


Figure 4.15: Dimensionless parameter profiles for group *a* shots

field to that at low field, that is

$$\mathcal{R}(P_{\text{in}}) = \frac{\int_0^r Q_{\text{in,hi}} dV}{\int_0^r Q_{\text{in,lo}} dV}. \quad (4.65)$$

Recall that the input power densities for each species are:

$$Q_{\text{in}} = \begin{cases} Q_{\text{OH}} - Q_{ei} + Q_{\text{RF}} & \text{for electrons} \\ Q_{ei} + Q_{\text{RF}} & \text{for ions.} \end{cases} \quad (4.66)$$

Referring to (4.20),

$$\mathcal{R}(P_{\text{in}}) \sim \mathcal{R}(B)^\gamma, \quad (4.67)$$

where $\gamma = \frac{1}{3}(5 - 2\alpha)$. Table 4.2 on page 96 lists the expected values of γ for each of the named scalings. These are shown on the plot with the dotted lines. The total input power shows a scaling somewhere between Bohm and gyroBohm. These two shots are ohmic, and since the exchange power transferred from the electrons to the ions increases with density, the net power input to the ions is greater at high field, while the net power into the electrons is lower.

Figure 4.16(b) plots the ratio of the volume-integrated *conducted* power at high field to that at low field, that is

$$\mathcal{R}(P_{\text{cond}}) = \frac{\int_0^r \nabla \cdot \mathbf{q}_{\text{hi}} dV}{\int_0^r \nabla \cdot \mathbf{q}_{\text{lo}} dV}. \quad (4.68)$$

If the radiated power fraction is small, the scaling of $\mathcal{R}(P_{\text{cond}})$ with B should be the same as the scaling of $\mathcal{R}(P_{\text{in}})$. Evidently this is not the case for these two shots, as the total conducted power is $\sim 4\times$ greater for the high field shot while the total input power is only $\sim 2.5\times$ greater. This implies that the radiated power is proportionally higher for the 2.5 T shot. Referring to table 4.8, f_{rad} at $r/a = 0.5$ is about $3\times$ greater for the low field shot. P_{rad} is a loss term for the electrons, so the electron conducted power is similarly affected, while the scaling of the ion conducted power is roughly the same as the ion input power.

Note that $\mathcal{R}(P_{\text{cond}})$ for the electrons actually goes negative for $r > 0.6a$, implying that the magnitude of P_{ei} is greater than the ohmic input power P_{oh} . This of course violates conservation of energy; the problem arises because of the large error in Q_{ei} as discussed in § 3.1.2. Scaling the estimate of $\delta\chi_j$ from (3.19) to the density of 950127015 gives $\delta\chi_j = \pm 0.8 \text{ m}^2/\text{s}$ for a 20% uncertainty in the relative temperatures. (Remember that this is *only* the error from Q_{ei} . The total error in χ_j will be larger.) As shown in figure 4.16(c), which plots the χ_{eff} , χ_e , and χ_i profiles for each shot, $\delta\chi_e \gg \chi_e$, $\delta\chi_i \simeq \chi_i$ for the high field shot, so that scalings for χ_e and χ_i will be meaningless.

χ_{eff} is unaffected by the large errors in Q_{ei} . Being larger for the low field shot at $r/a = 0.5$, there appears to be an improvement in the overall confinement with increasing field. Actually, factoring in the scaling of χ_B , $\chi_B \sim B^{-0.7}$, the scaling with ρ^* is Goldston-like. This is shown in figure 4.16(d), which plots the ratio of χ/χ_B at high field to that at low field as explained in § 4.4.4. The dotted lines in figure 4.16(d) show the profile of $\mathcal{R}(\rho^*)^\alpha$ for the scalings listed in table 4.2. The $B\tau_E$ values for the two shots listed in table 4.8 show a Bohm-like scaling

$$B\tau_E \sim \rho^{*-1.95 \pm 0.6}, \quad (4.69)$$

consistent with the value for group *a* as a whole in § 4.5.1. The scaling of χ_{eff} is consistent with the scaling of $B\tau_E$ within the large uncertainty of $\mathcal{R}(\chi_{\text{eff}}/\chi_B)$. On the other hand, the effects of the large difference in $f_{\text{rad}} = P_{\text{rad}}/P_{\text{in}}$ on the scaling of the conducted power suggests that ρ^* scaling assumptions may not be valid for this pair.

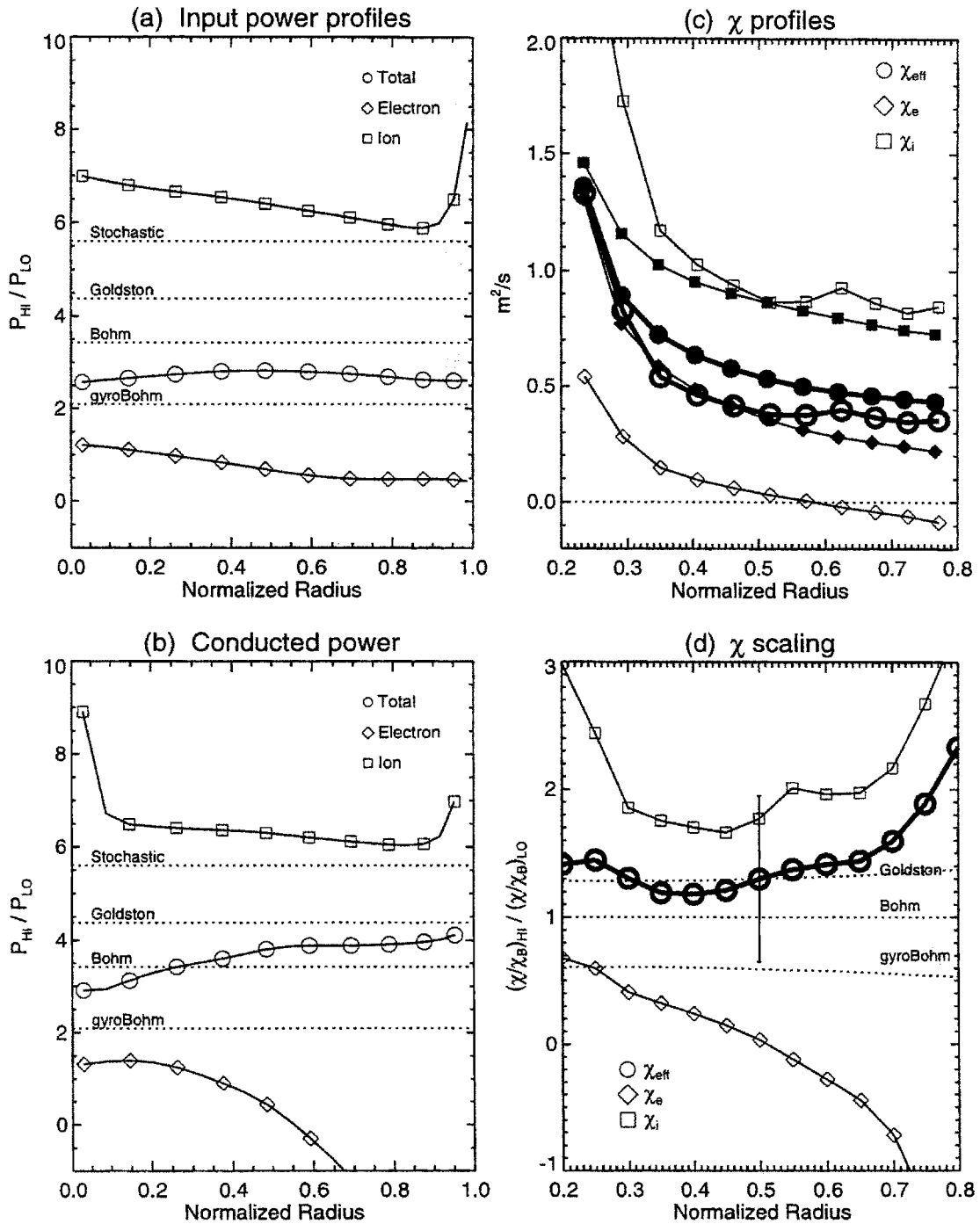


Figure 4.16: Local analysis plots for ρ^* scaling L-Mode group *a*. Shown are the ratios of P_{in} and P_{cond} (high field/low field), the χ profiles for both shots (filled = low, open = hi), and the ratio $\mathcal{R}(\chi/\chi_B)$ for each species.

Shot	951218014	951218031
t_0 (s)	1.32	0.76
B_t (T)	2.5	5.2
I_p (MA)	0.3	0.6
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	0.63	1.60
P_{RF} (MW)	0.00	0.79
U/B^2	1.32	1.40
$\langle n_e \rangle/B^{4/3}$	0.18	0.18
β_t	0.34	0.33
β_p	0.45	0.48
$q\psi_{95}$	4.95	5.17
ν^*_{e}	0.51	0.61
T_i/T_e	0.60	0.71
κ	1.53	1.59
Z_{eff}	1.22	1.26
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.32	0.15
$\rho^* \times 1000$	4.53	3.01
$B\tau_E$ (T s)	0.06	0.16
TRANSP run	TRANSP07	TRANSP05

Table 4.9: Parameters for L-Mode, group b

Group b

The low field shot of group b is 951218014, ohmically heated at 2.5 T. The higher field shot, 951218031, is RF-heated. The run day 951218 was dedicated to ρ^* scaling, with this pair the only one that was properly matched. Table 4.9 lists the engineering and dimensionless parameters for these shots. Again the all dimensionless parameters agree well within their uncertainties, except for the radiated power fraction at $r/a = 0.5$. Figure 4.17 plots the β_t and ν^* profiles, which also agree within their uncertainties. Time histories of the two shots are shown in figure 4.18.

Local analysis is addressed by the four plots in figure 4.19. The profiles of the ratios $\mathcal{R}(P_{\text{in}})$ and $\mathcal{R}(P_{\text{cond}})$ are both peaked on axis, where they are $\gg 1$, because of the RF power input to the higher field shot. The scaling of the total input power is Goldston-like for $r \geq 0.5a$. The radiated power fraction f_{rad} is significant in the low field shot, and almost twice as large as f_{rad} from the higher field shot, so that the ratio of the conducted powers P_{cond} scales more strongly with B than P_{in} . The strong scaling of P_{cond} with B implies that $\chi_{\text{eff}}(5.3 \text{ T}) > \chi_{\text{eff}}(2.6 \text{ T})$. This is borne out by the plots of χ_{eff} , χ_e , and χ_i in figure 4.19(c). All three χ 's are larger at the higher field. Their ρ^* scalings, shown in figure 4.19(d) by comparing $\mathcal{R}(\chi/\chi_B)$ to $\mathcal{R}(\rho^*)$, reflect this fact. This is in contrast with the global scaling of the shots, which from table 4.9 is $B\tau_E \sim \rho^{*-2.3 \pm 0.9}$, between Bohm and gyroBohm-like.

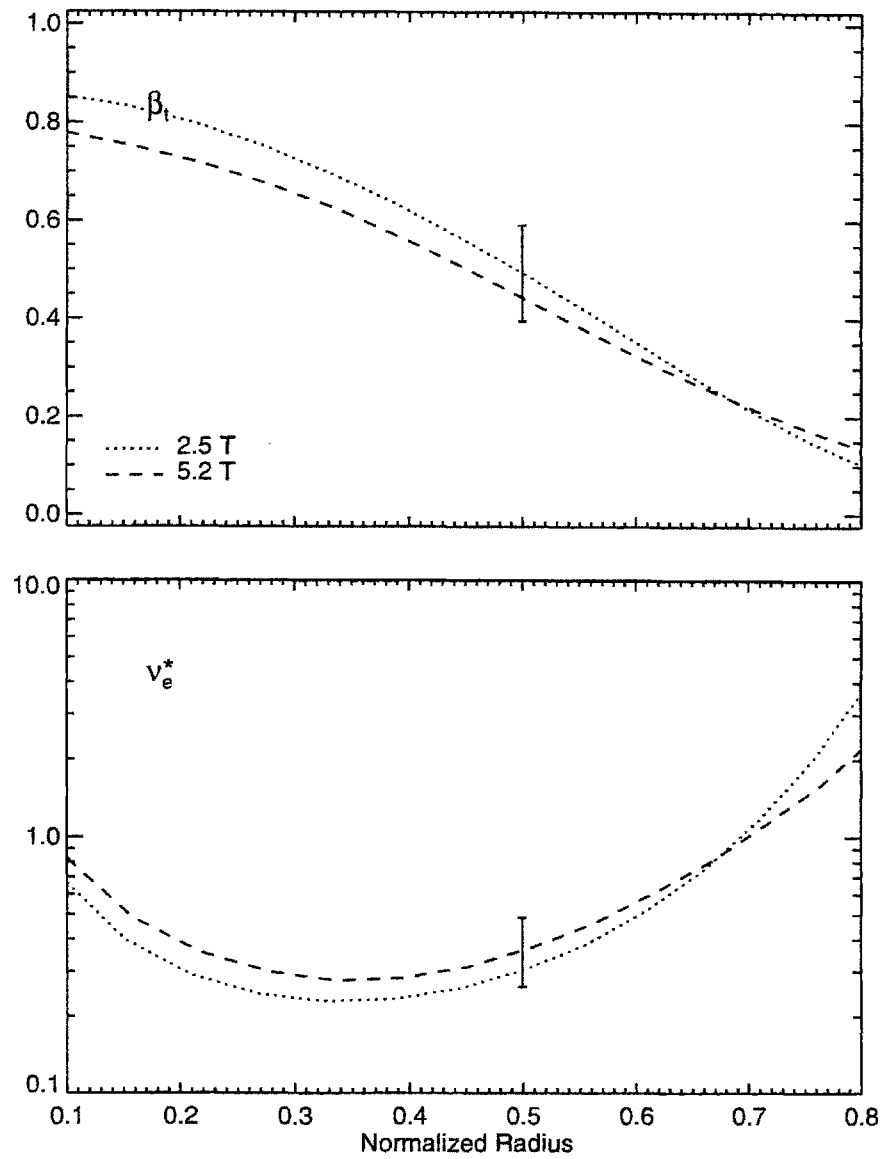


Figure 4.17: Dimensionless parameter profiles for group *b* shots

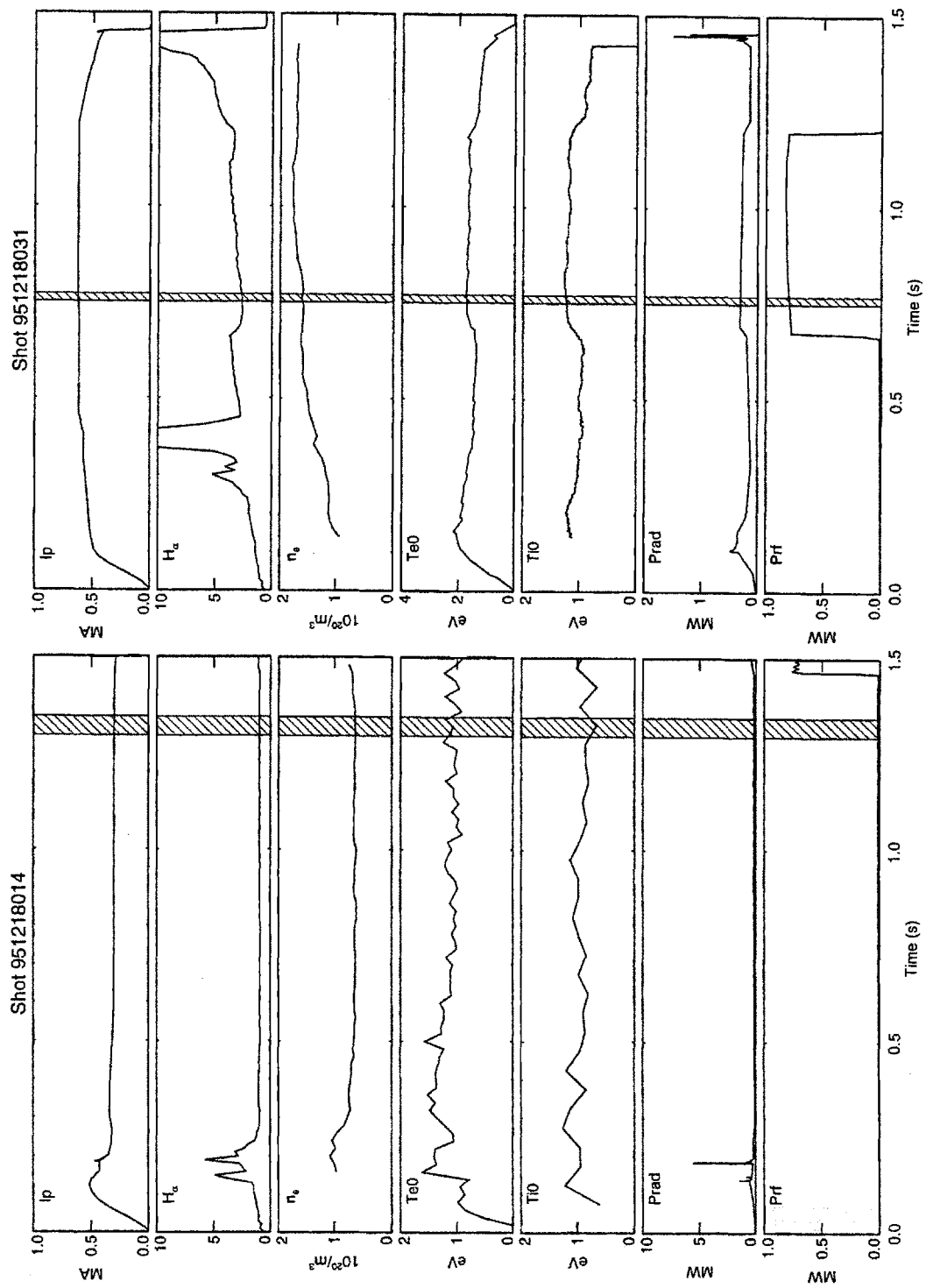


Figure 4.18: Time history of shots in L-Mode group *b*. Shot 951218014: $B_t = 2.5$ T. Shot 951218031: $B_t = 5.2$ T.

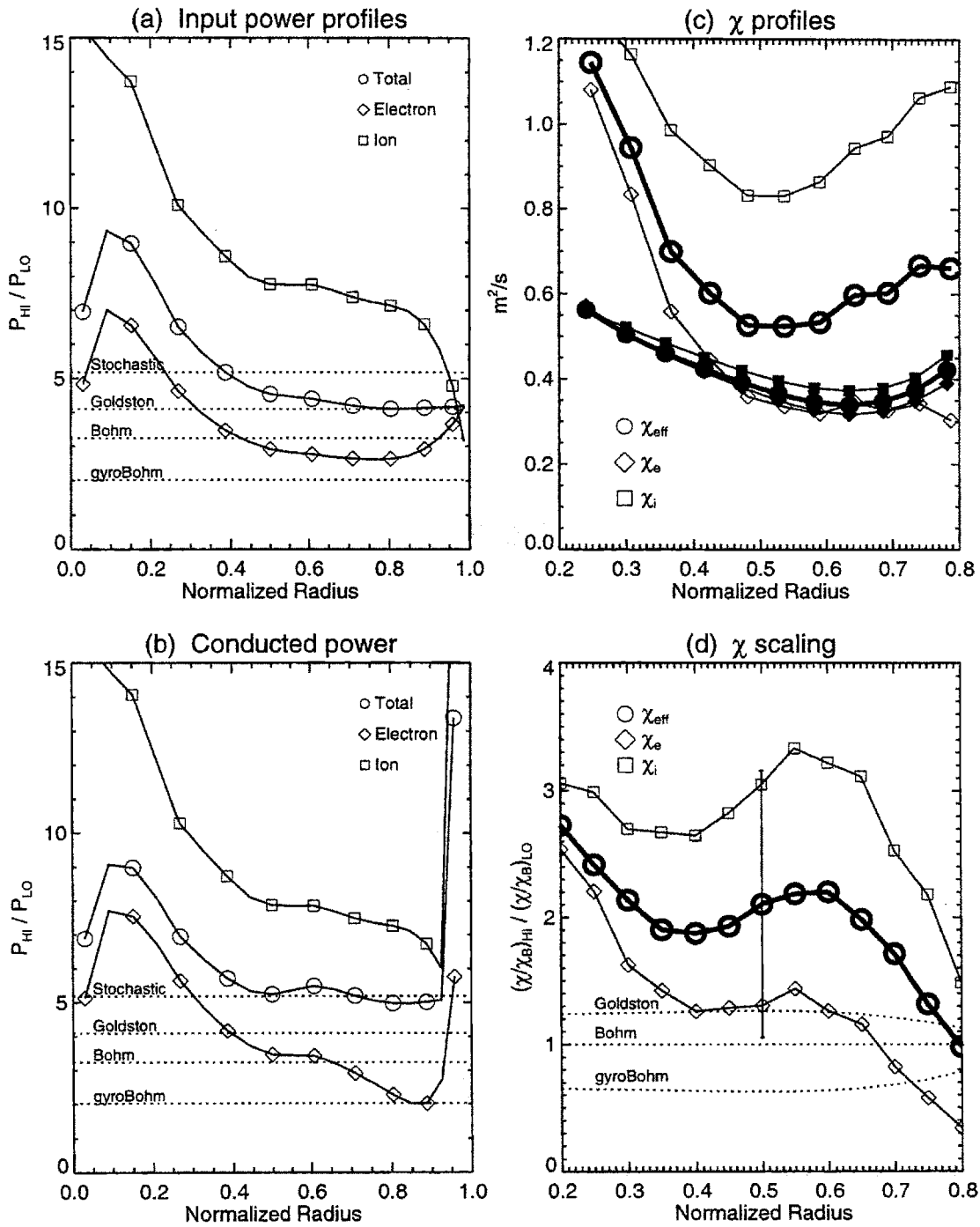


Figure 4.19: Local analysis plots for ρ^* scaling L-Mode group *b*. Shown are the ratios of P_{in} and P_{cond} (high field/low field), the χ profiles for both shots (filled = low, open = hi), and the ratio $\mathcal{R}(\chi/\chi_B)$ for each species.

Shot	951218020	950531009
t_0 (s)	0.90	0.96
B_t (T)	3.5	7.9
I_p (MA)	0.4	1.0
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	0.98	1.57
P_{RF} (MW)	0.00	0.00
U/B^2	1.23	1.02
$\langle n_e \rangle / B^{4/3}$	0.19	0.10
β_t	0.27	0.23
β_p	0.38	0.30
$q_{\psi 95}$	5.16	5.10
$\nu^* e$	0.98	0.38
T_i/T_e	0.69	0.79
κ	1.57	1.67
Z_{eff}	1.26	2.91
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.23	0.21
$\rho^* \times 1000$	3.49	2.62
$B\tau_E$ (T s)	0.10	0.23
TRANSP run	TRANSP03	TRANSP03

Table 4.10: Parameters for L-Mode, group *c*

Group *c*

Group *c* is the only group studied to have shots with $B_t > 6$ T. While the global dimensionless parameters of the two shots, 951218020 and 950531009, match within their uncertainties (except for ν^* and $P_{\text{rad}}/P_{\text{in}}$) as shown in table 4.10, their dimensionless parameter profiles plotted in figure 4.20 do not agree as well. As with the rest of the 8 T shots in this group, β_p is too low compared to the 3.5 T shots. Time histories of the two shots are plotted in figure 4.21. 950531009 is RF-heated. Its fractional RF power absorption could not be measured, and so was assumed to be 50% based on the discussion in § 3.5.3.

The plots for local analysis are presented in figure 4.22. The ratios $\mathcal{R}(P_{\text{in}})$ of the total volume-integrated input power and the power input to each species are peaked on axis because of the RF heating in the higher field shot. All approach a Bohm-like scaling as $r \rightarrow a$. The ratios $\mathcal{R}(P_{\text{cond}})$ of the total and electron conducted powers decrease rapidly as $r \rightarrow a$ because of the radiated power. While f_{rad} at $r/a = 0.5$ from the two shots is well matched and low (both $\simeq 0.2$), it is much larger outside $r = 0.7a$ (where $f_{\text{rad}} = 0.6 - 0.8$), suggesting that the assumptions behind ρ^* scaling are not valid in this region. Confining the examination of the scaling of χ to the region $r/a \in (0.4, 0.6)$, the one fluid χ_{eff} shows a Bohm-like scaling, though the uncertainty in $\mathcal{R}(\chi_{\text{eff}})$ is very large. The scaling of χ_e with ρ^* is stronger than the scaling of χ_i .

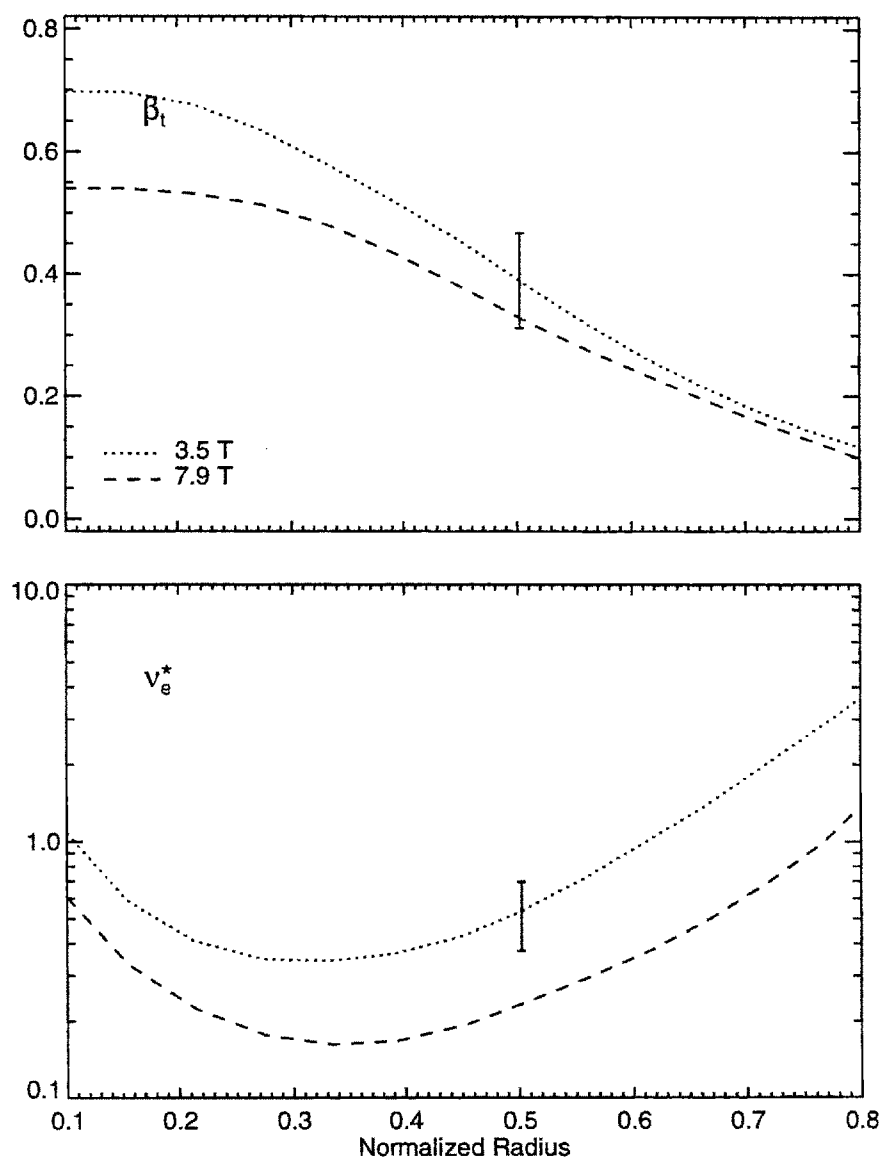


Figure 4.20: Dimensionless parameter profiles for group *c* shots

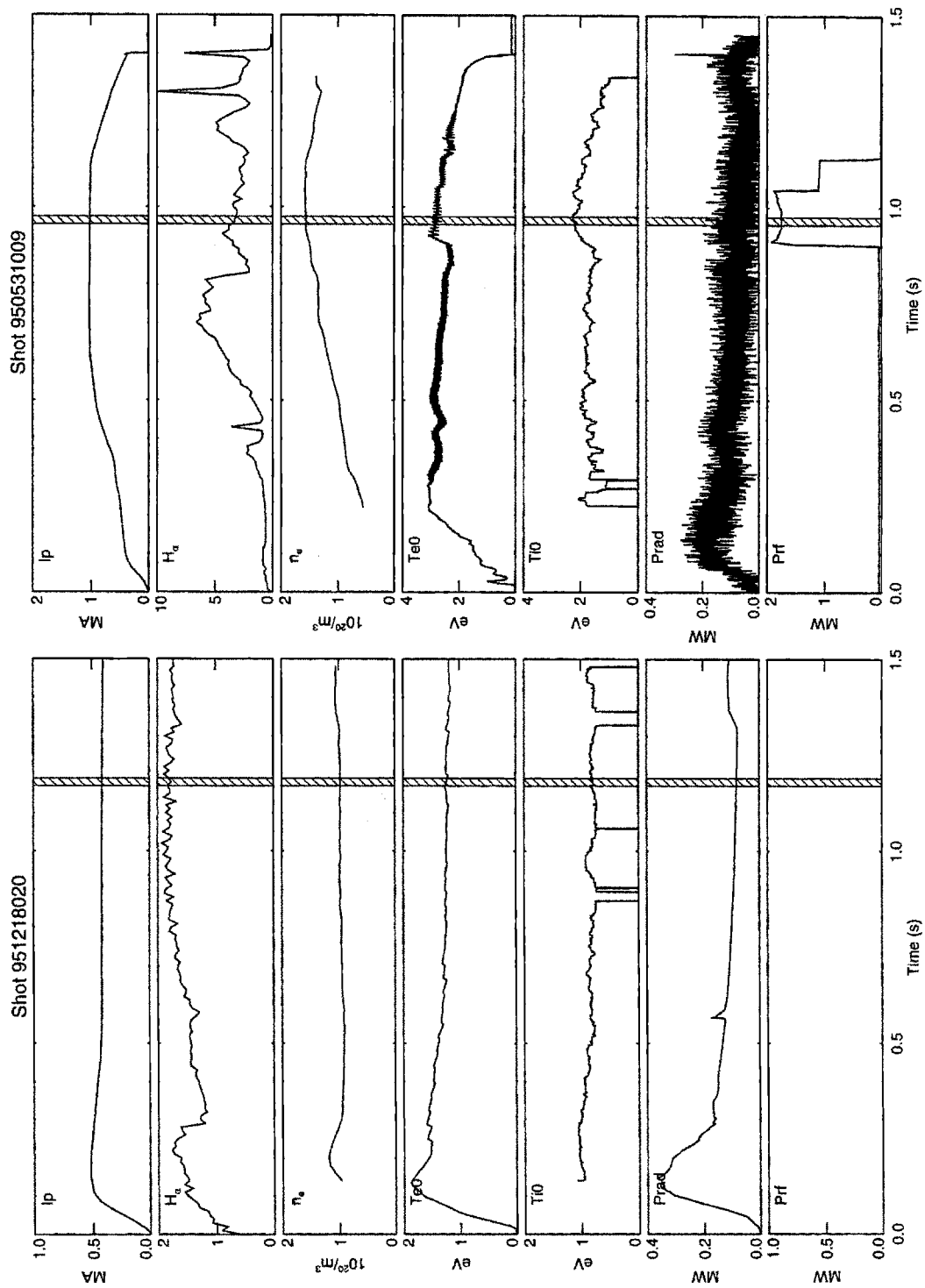


Figure 4.21: Time history of shots in L-Mode group c. Shot 951218020: $B_t = 3.5$ T. Shot 950531009: $B_t = 7.9$ T.

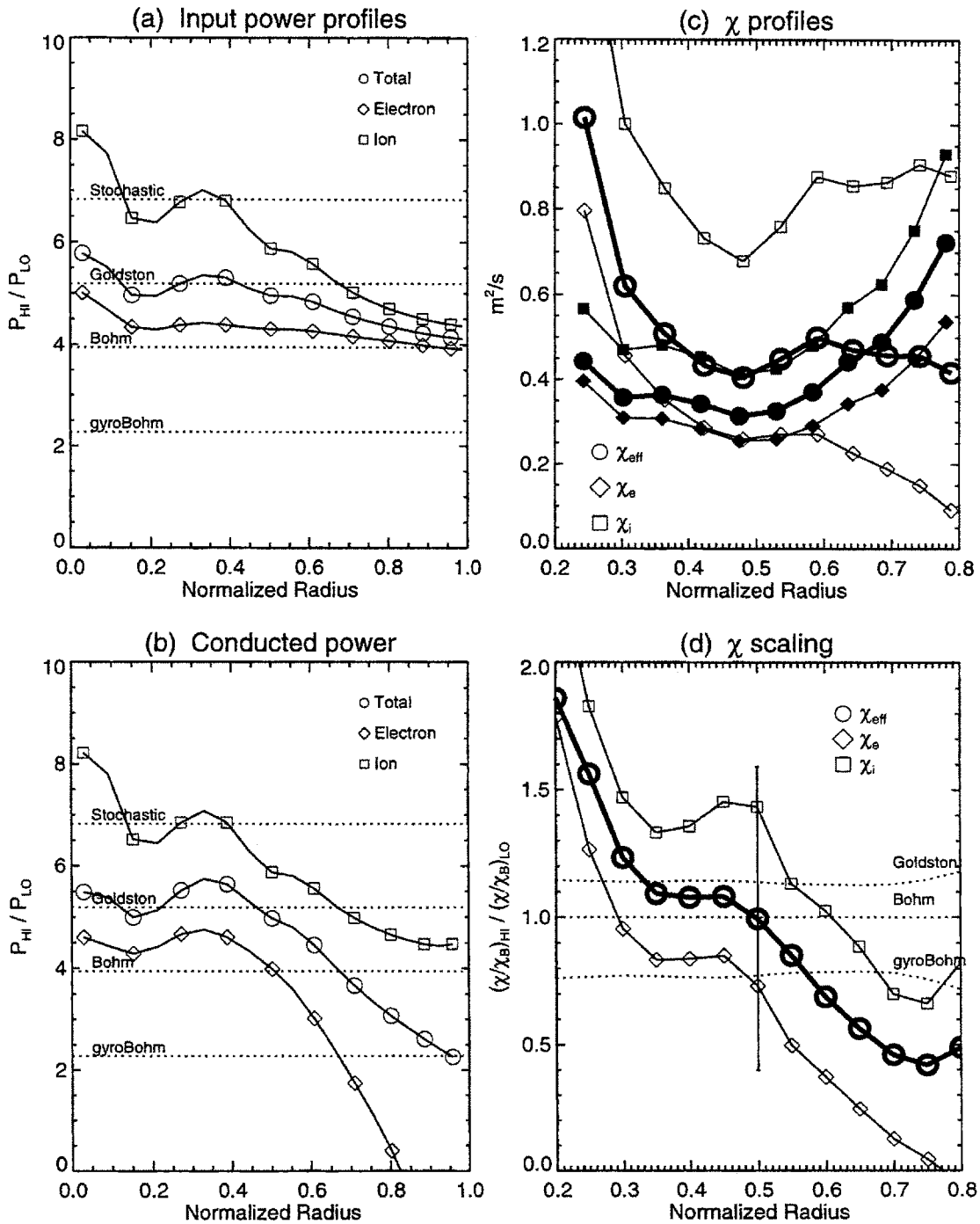


Figure 4.22: Local analysis plots for ρ^* scaling L-Mode group c. Shown are the ratios of P_{in} and P_{cond} (high field/low field), the χ profiles for both shots (filled = low, open = hi), and the ratio $\mathcal{R}(\chi/\chi_B)$ for each species.

Shot	951218016	950406022
t_0 (s)	0.96	0.71
B_t (T)	2.6	5.4
I_p (MA)	0.4	0.8
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	0.65	1.91
P_{RF} (MW)	0.00	0.92
U/B^2	2.04	2.06
$\langle n_e \rangle/B^{4/3}$	0.18	0.20
β_t	0.52	0.49
β_p	0.37	0.42
$q_{\psi 95}$	3.96	4.18
$\nu^* e$	0.21	0.28
T_i/T_e	0.84	0.69
κ	1.56	1.62
Z_{eff}	1.05	1.05
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.36	0.11
$\rho^* \times 1000$	5.64	3.40
$B\tau_E$ (T s)	0.06	0.18
TRANSP run	TRANSP01	TRANSP01

Table 4.11: Parameters for L-Mode, group e

Group e

Group e has 951218016 as the low field shot and 950406022 as the high field shot. Time histories of the shots are shown in figure 4.23.

The latter shot is RF-heated at the time of interest. The parameters are listed in table 4.11, and the dimensionless profiles plotted in figure 4.24. The shots are well matched for all dimensionless parameters except $P_{\text{rad}}/P_{\text{in}}$, and for the B scaling of the total stored energy and volume averaged density.

The mismatch in the radiated power fractions once again complicates the ρ^* scaling analysis. The problem is evident by comparing the ratios of the total input power P_{in} and the conducted power P_{cond} at $r/a = 0.5$, shown in figures 4.25(a) and 4.25(b) respectively. While $\mathcal{R}(P_{\text{in}}) \sim 4.5$, the ratio conducted power is much larger: $\mathcal{R}(P_{\text{cond}}) \sim 7$. This increased scaling is the result of the higher radiated power fraction for shot 951218016. Since the scaling of P_{cond} is so strong, it is not surprising that the ρ^* scaling of χ_{eff} shown in figure 4.25(d) is worse than Goldston like. This is in contrast to the global scaling for the two shots: $B\tau_E \sim \rho^{*-2.2 \pm 0.7}$, which is Bohm-like. The effects of P_{rad} here too suggest that the assumptions behind ρ^* scaling may not be valid.

Since the density for 951218016 is low ($\langle n_e \rangle = 0.65 \times 10^{20}/\text{m}^3$), the coupling between the electrons and ions is weak. Most of the ohmic input power stays with

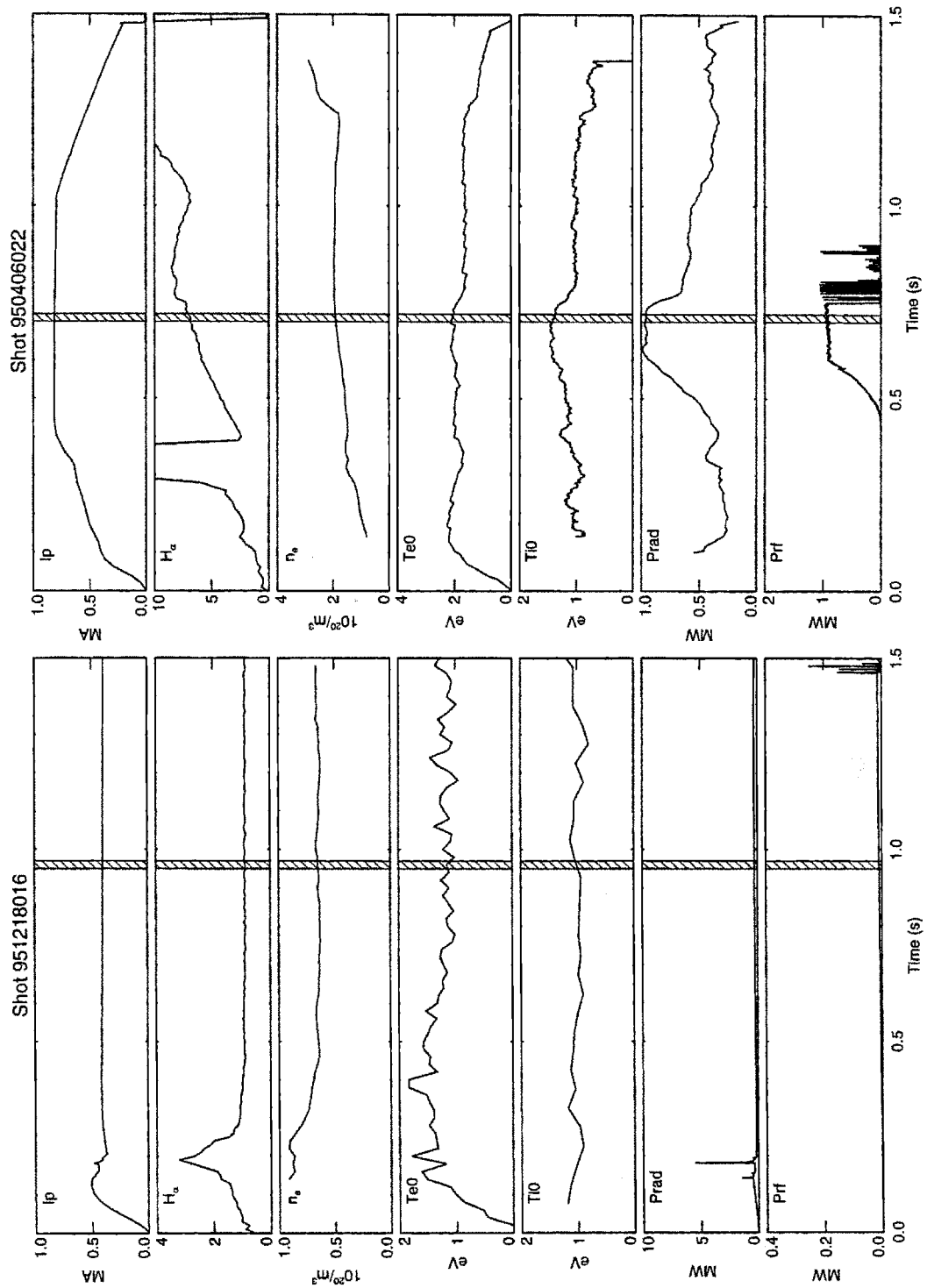


Figure 4.23: Time history of shots in L-Mode group *e*. Shot 951218016: $B_t = 2.6$ T. Shot 950406022: $B_t = 5.4$ T.

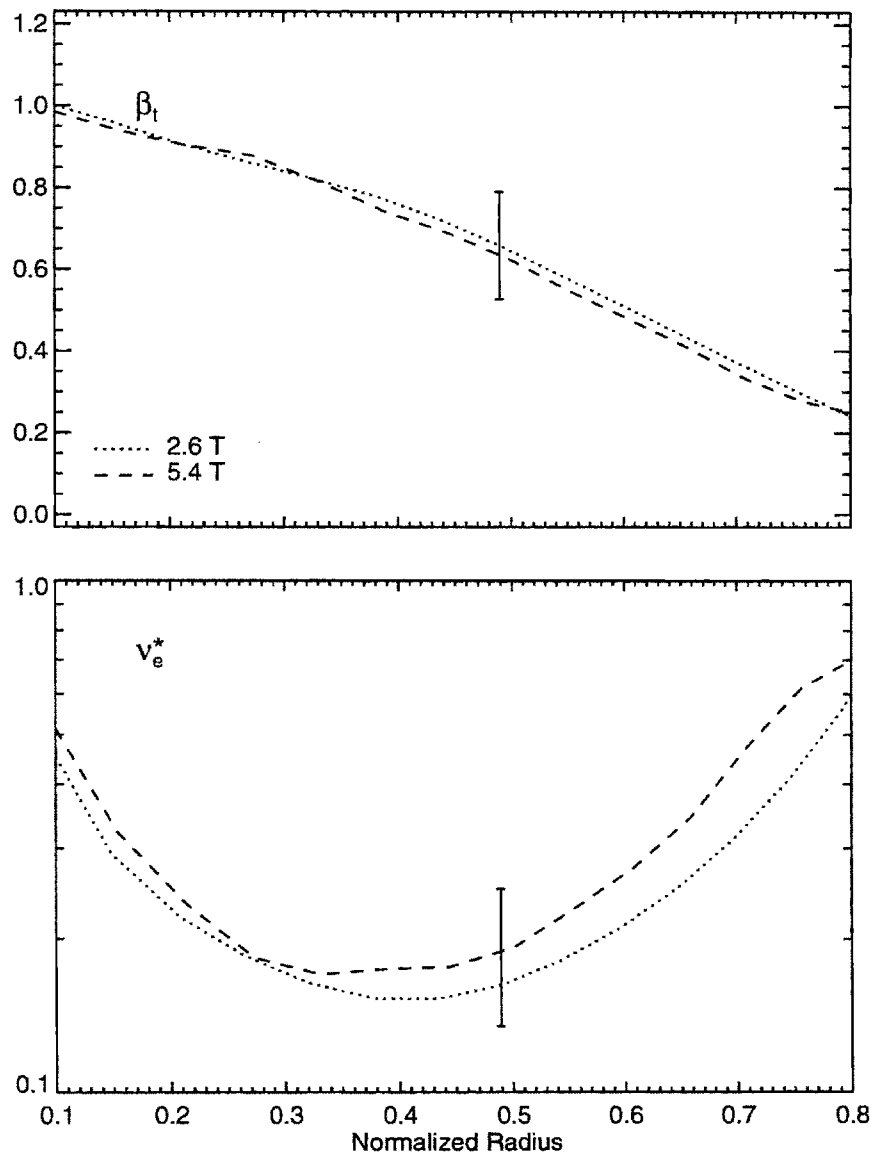


Figure 4.24: Dimensionless parameter profiles for group e shots

Group	Global scaling, group	Local scaling, pair	$\mathcal{R}(f_{rad})$
La	0.0 ± 0.6	-0.5 ± 1.0	0.4
Lb	-0.8 ± 0.5	-1.6 ± 1.1	0.5
Lc	0.2 ± 0.7	0.0 ± 2.3	0.9
Le	-0.3 ± 0.8	-1.3 ± 0.9	0.3

Table 4.12: Summary of ρ^* scaling in L-Mode. Global scaling is value of α from $B\tau_E \propto \rho^{*-2-\alpha}$. Local scaling is value of α from $\chi \propto \chi_B \rho^{*\alpha}$. Also listed is the ratio of the radiated power fraction f_{rad} at high field to that at low field.

the electrons, so that for the RF-heated 950406022, P_{in_i} is over 20 times larger than P_{in_i} from 951218016. This dramatic difference in the input power suggests a problem in comparing low-density ohmic shots with RF-heated shots for ρ^* scaling. The uncertainty in the low field χ_i from Q_{ei} is ± 0.2 m²/s, which is $\simeq \chi_i$ as shown in figure 4.25(c). Even with this uncertainty, the scaling of χ_i is still much worse than Goldston-like. The ratio $\mathcal{R}(\chi_i)$ is so large it is not plotted on figure 4.25(d), so that it is possible to examine χ_e and χ_{eff} . The scaling of χ_e is better behaved. At $r/a = 0.5$, it lies somewhere between gyroBohm and Bohm-like, although the uncertainty is large.

Summary

Table 4.12 provides a summary of the χ_{eff} scalings observed for the pair of shots from each L-Mode group, which are compared to the scalings of $B\tau_E$ for the pairs and for the whole group. Figure 4.26 plots these values of α versus the ratio of the radiated power fractions from the pair of shots used in local analysis: $\mathcal{R}(f_{rad}) = f_{rad}(hi)/f_{rad}(lo)$. While the global and local α values do show the same trend, both the scatter in the global values, and their uncertainties, are less than those of the local values.

The values of α obtained from local scaling are systematically lower than those from global scaling, reflecting the fact that the radiated power in the low field shot was proportionally larger than in the high field shot. The conducted power in the low field shots was therefore proportionally smaller, implying that the transport was reduced at low field. Whether or not the transport was really reduced at low field is unclear. However, the fact that the radiated power played an important role in determining the scaling of χ_{eff} suggests that the assumptions inherent in ρ^* scaling, that is that the Boltzmann equation completely describes the plasma physics, are not valid here. Definite conclusions on ρ^* scaling in L-Mode are therefore not possible based on the shots analyzed.

The fact that the global scalings show less scatter in figure 4.26 imply that they were less sensitive to the radiated power fraction f_{rad} . This is consistent with the observation that the H-factor of C-Mod L-Mode shots appears to be independent of

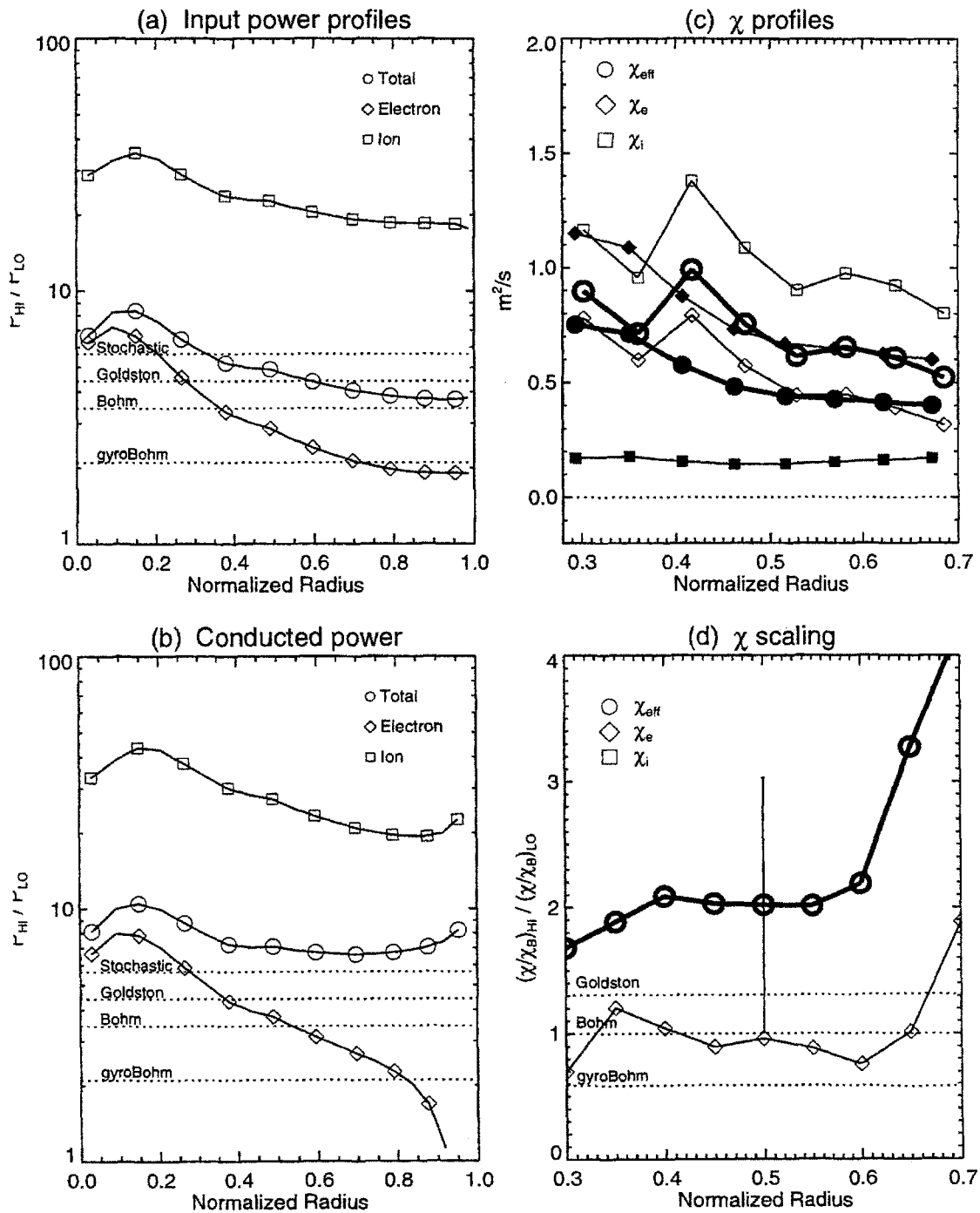


Figure 4.25: Local analysis plots for ρ^* scaling L-Mode group e. Shown are the ratios of P_{in} and P_{cond} (high field/low field), the χ profiles for both shots (filled = low, open = hi), and the ratio $\mathcal{R}(\chi/\chi_B)$ for each species.

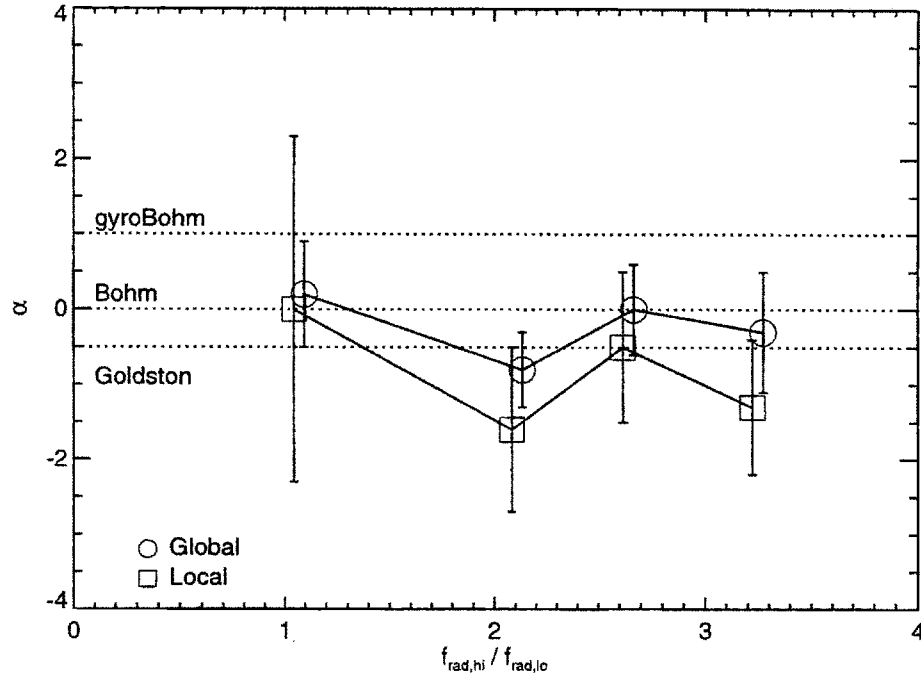


Figure 4.26: Values of α obtained from global and local scaling in L-Mode. The ordinates of the local values have been shifted slightly so that both values of α could be distinguished.

f_{rad} [11]. H-factor s of 1 are seen even with $f_{rad} \sim 1.0$, while χ_{eff} would be zero, suggesting that χ is not a useful parameter in L-Mode. This idea is explored further in chapter 5, in which marginal stability is examined as an alternative to local scaling, but first for completeness the ρ^* scaling of C-Mod H-Modes is examined.

4.5.3 H-Mode

H-Mode shots selected from the core table of *LOGBOOK* for analysis were single-null lower diverted D₂ plasmas with $\kappa \in [1.5, 1.7]$. Only elm-free or enhanced D α [10] H-Modes from the 1996 run campaign were considered.⁷ This set of shots will be referred to as \mathcal{H} .

As done for L-Mode, regressions of τ_E against both engineering and dimensionless parameters were performed on \mathcal{H} as a whole before dimensionally similar shots were selected for ρ^* scaling analysis. Figure 4.27 plots the regression of τ_E versus I_p , n_e , and P_{in} . The result is:

$$\tau_E = 0.06 I_p^{1.4 \pm 0.1} n_e^{0.17 \pm 0.10} P_{in}^{-0.36 \pm 0.08}. \quad (4.70)$$

The scattering of the data for $B > 2.5$ T results from the fact that all types of H-Modes were included in \mathcal{H} , with a variety of H-factor s, ELM conditions, and edge temperatures. Figure 4.28 plots the H-factor, defined as

$$H \equiv \frac{\tau_E}{\tau_{E,ITER89-P}}. \quad (4.71)$$

Also, as shown in figure 4.29 the I_p and P_{RF} values in \mathcal{H} are strongly clustered, which reduces the degrees of freedom available to the fit.

The dimensionless scaling of the group is addressed in figure 4.30. The result is

$$B\tau_E = 210 \rho^*^{-2.9 \pm 0.1} \nu^*^{-0.48 \pm 0.03} \beta_p^{0.89 \pm 0.08} q_{\psi 95}^{-1.2 \pm 0.2}. \quad (4.72)$$

Using (4.44), this gives $\alpha = 0.9 \pm 0.7$, a gyroBohm scaling. Converting (4.70) to dimensionless form gives:

$$B\tau_E \sim \rho^*^{-3.7 \pm 0.5} \nu^*^{-0.5 \pm 0.7} \beta_p^{0.2 \pm 0.2} q_{\psi 95}^{-2.2 \pm 0.3}. \quad (4.73)$$

As with L-Mode, the ρ^* and ν^* scalings of (4.72) and (4.70) are similar, but the β_p and $q_{\psi 95}$ scalings are different.

There are fewer H-Mode slices in the core table, so no breakdown in $q_{\psi 95}$ is necessary to facilitate identification of groups of dimensionally similar shots suitable for ρ^* scaling. Figures 4.31(a) and (b) plot $q_{\psi 95}$ and ν^* versus β_p for the shots in \mathcal{H} . Most of the 2.5 T shots have ν^* values much higher than those of the 5.3 T shots. This is because their temperatures, scaled up to 5.3 T using (4.17), were too cold. T_e was low because the efficiency of the RF-heating at 2.5 T was low, as 2nd harmonic minority H damping was the heating scheme. Only two groups of 2.5/5.3 T shots were found to satisfy (4.52)-(4.54) and therefore be suitable for ρ^* scaling analysis.

⁷The complete SQL query for selection was: (topic='h rhostar' or topic='h boron' or topic='h hi-rad') and (gas_maj = 'D2') and (topology='SNB') and (kappa between 1.5 and 1.7) and (shot between 960000000 and 960399999).

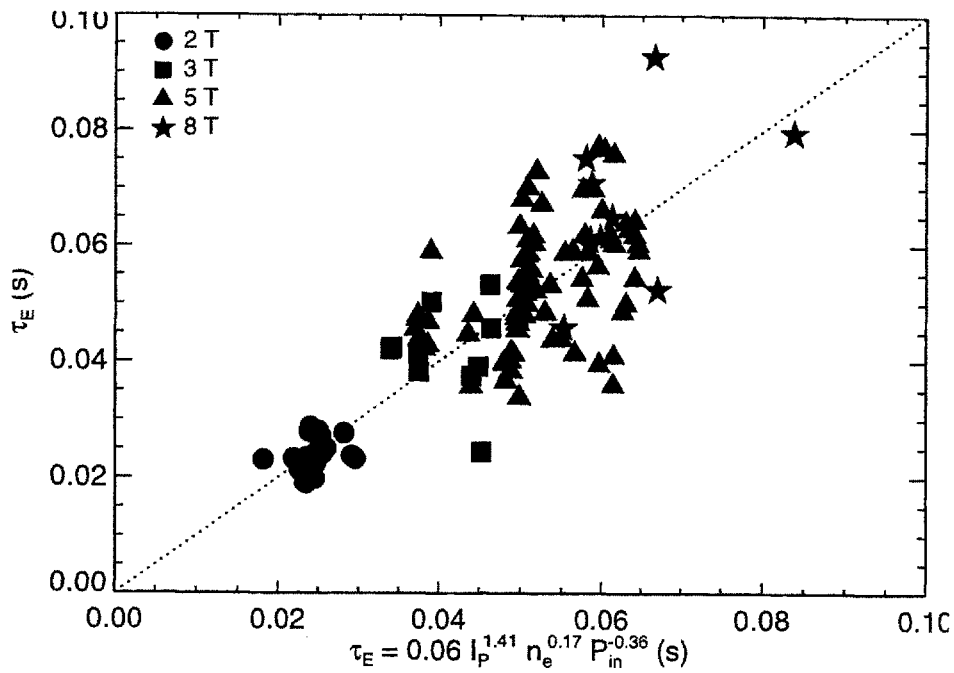


Figure 4.27: Regression of τ_E against engineering parameters for H-Mode

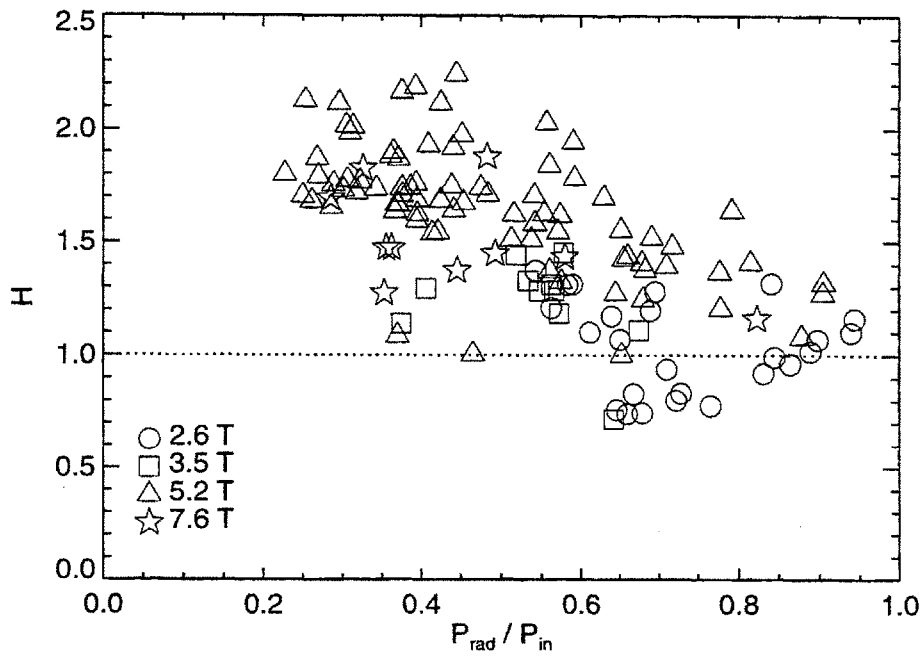


Figure 4.28: H-factor s of shots in H-Mode dataset. $H \equiv \tau_E / \tau_{E,ITER89-P}$.

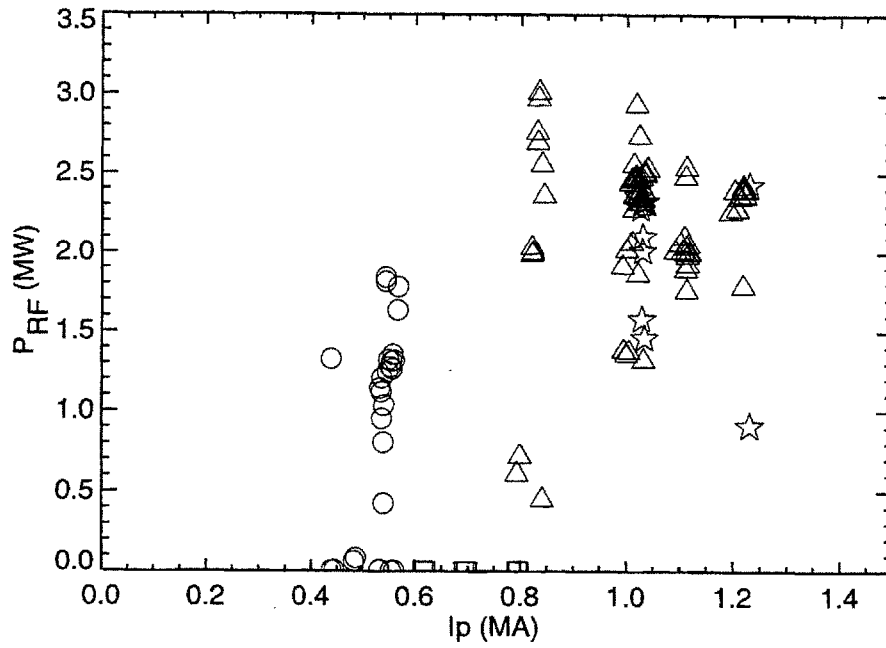


Figure 4.29: Range of P_{RF} and I_p covered by \mathcal{H} .

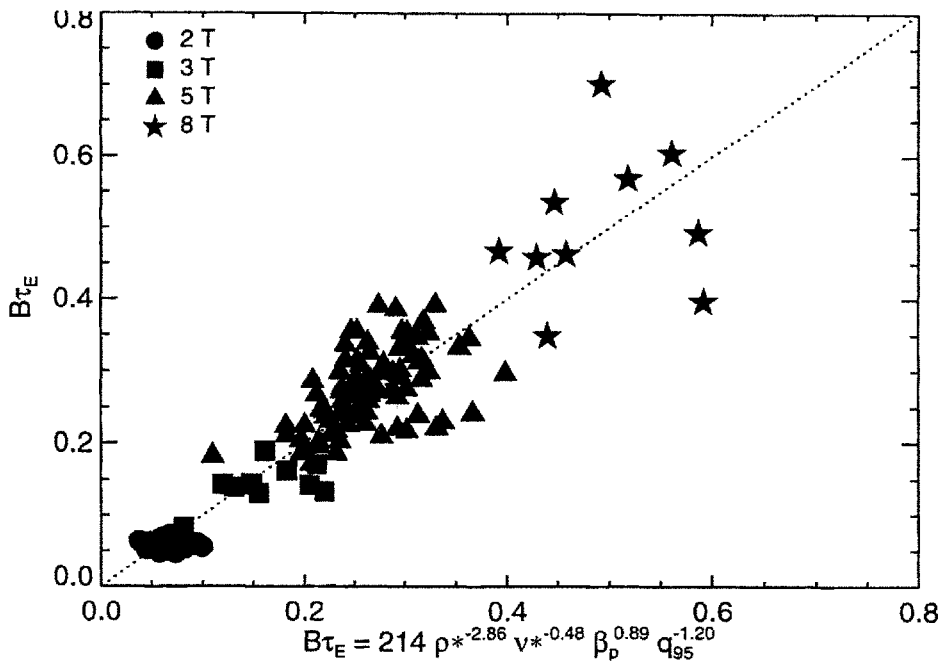


Figure 4.30: Regression of τ_E against dimensionless parameters for H-Mode

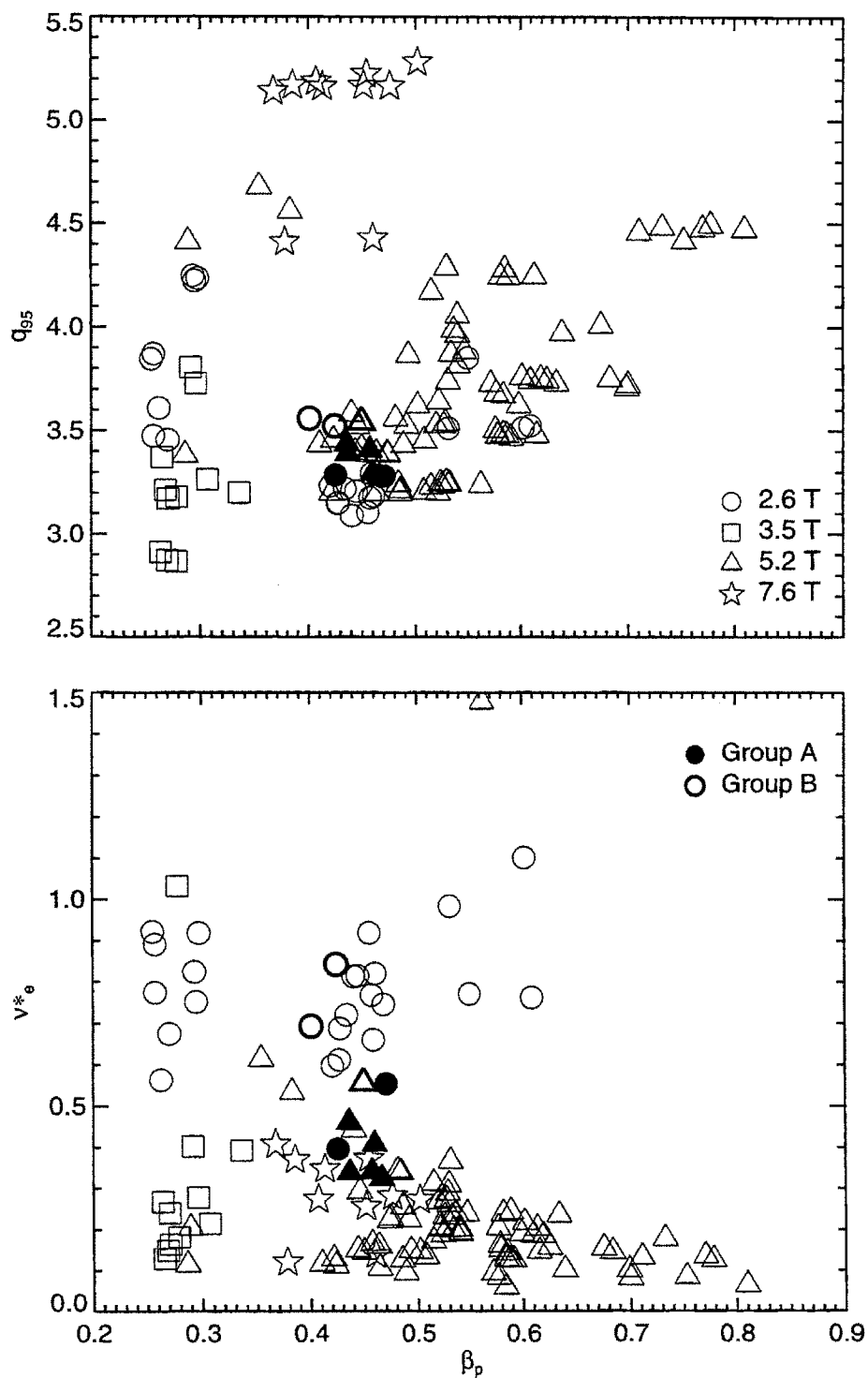


Figure 4.31: Distribution of H-Mode shots. Two groups identified for further study marked by filled/heavy symbols.

Group	$q_{\psi 95}$		β_p		ν^*	
	average	s.d.	average	s.d.	average	s.d.
Ha	3.34	0.03	0.45	0.01	0.40	0.03
Hb	3.54	0.01	0.42	0.02	0.70	0.10

Table 4.13: H-Mode shot groups for global scaling

These are plotted by the filled and thick-bordered symbols respectively.

No H-Mode shots with fields higher than 5.3 T could be matched to lower field shots for ρ^* scaling. All of the 8 T shots in \mathcal{H} had $q_{\psi 95} \geq 4.5$, with the majority having $q_{\psi 95} = 5.2$, corresponding to $I_p = 1.0$ MA. Obtaining H-Mode at 3.5 T required high I_p , and therefore lower $q_{\psi 95}$, as no RF-heating was possible. The few 2.6 T with sufficiently low currents did not match β_p or ν^* of the 8 T shots.

Each of the two 2.5/5.3 T groups will be examined in turn in the following sections. Their parameters are summarized in table 4.13.

Group *a*

Group *a* is comprised of a total of 7 slices from 6 shots. Plots for global analysis are presented in figure 4.32.

Figure 4.32(a) plots β_p versus ν^* for these shots. The two calculations of τ_E agree very well for all shots in the group, as shown in figure 4.32(b). Since β_p is higher for these shots than for L-Mode, the MHD calculation should be more accurate. Figure 4.32(c) plots their H-factor *s*, defined by $H = \tau_E / \tau_{E, ITER89-P}$, which are between 1.2 and 1.5, lower than the best H-Modes from C-Mod [11] because the H-Modes at lower field were not strong. The scaling of $B\tau_E$ with ρ^* is addressed in figure 4.32(d). Treating all points as independent gives $B\tau_E = 61 \rho^{*-3.1 \pm 0.6}$, with a reduced χ^2 of 0.36 and a linear correlation coefficient $r = -0.96$. Using the field clustering method previously described to account for the dependencies in the data gives an error of ± 0.7 in the exponent, which translates to $\alpha = 1.1 \pm 0.7$ using (4.44). Within the uncertainty, this is unambiguously gyroBohm scaling.

As with L-Mode, the residuals show an inverse dependence with ν^* . The regression, shown in figure 4.32(e), gives $\frac{B\tau_E}{B\tau_{E_{fit}}} \sim \nu^{*-0.62 \pm 0.07}$, which is stronger than found in L-Mode. No obvious dependence is found in plotting the residuals versus β_p as shown in figure 4.32(f).

The two shots from group *a* chosen for local scaling are 960118016 (2.6 T) and 960130020 (5.3 T). Time histories of the two shots are plotted in figure 4.33 with the time range of interest shaded.

Table 4.14 summarizes their scalar parameters. With the exception of Z_{eff} , all dimensionless parameters including $P_{\text{rad}}/P_{\text{in}}$ match well within their uncertainties. Their β_p and ν^* profiles also match, as shown in figure 4.34. The fractional RF

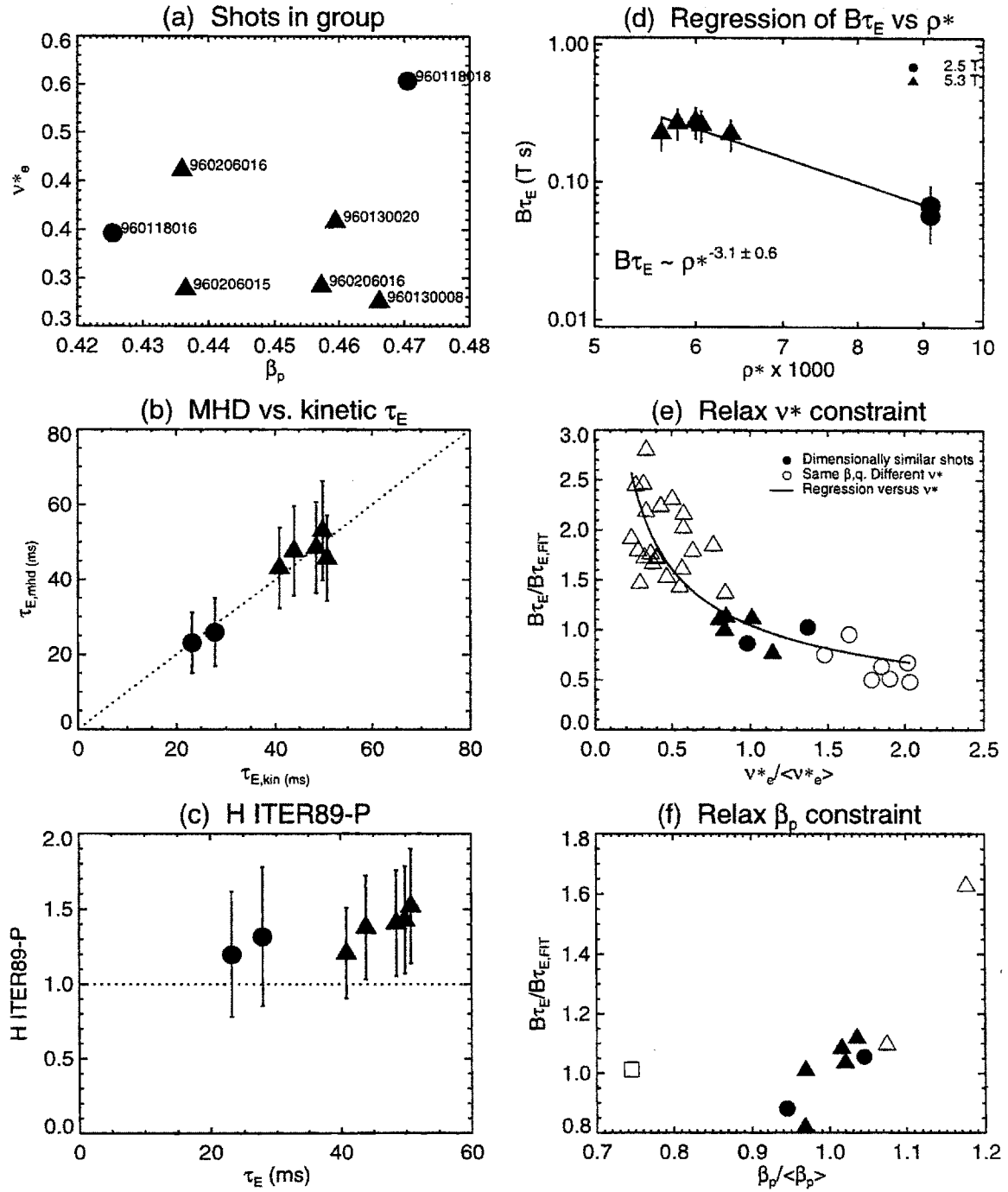


Figure 4.32: Global Analysis for H-Mode group a

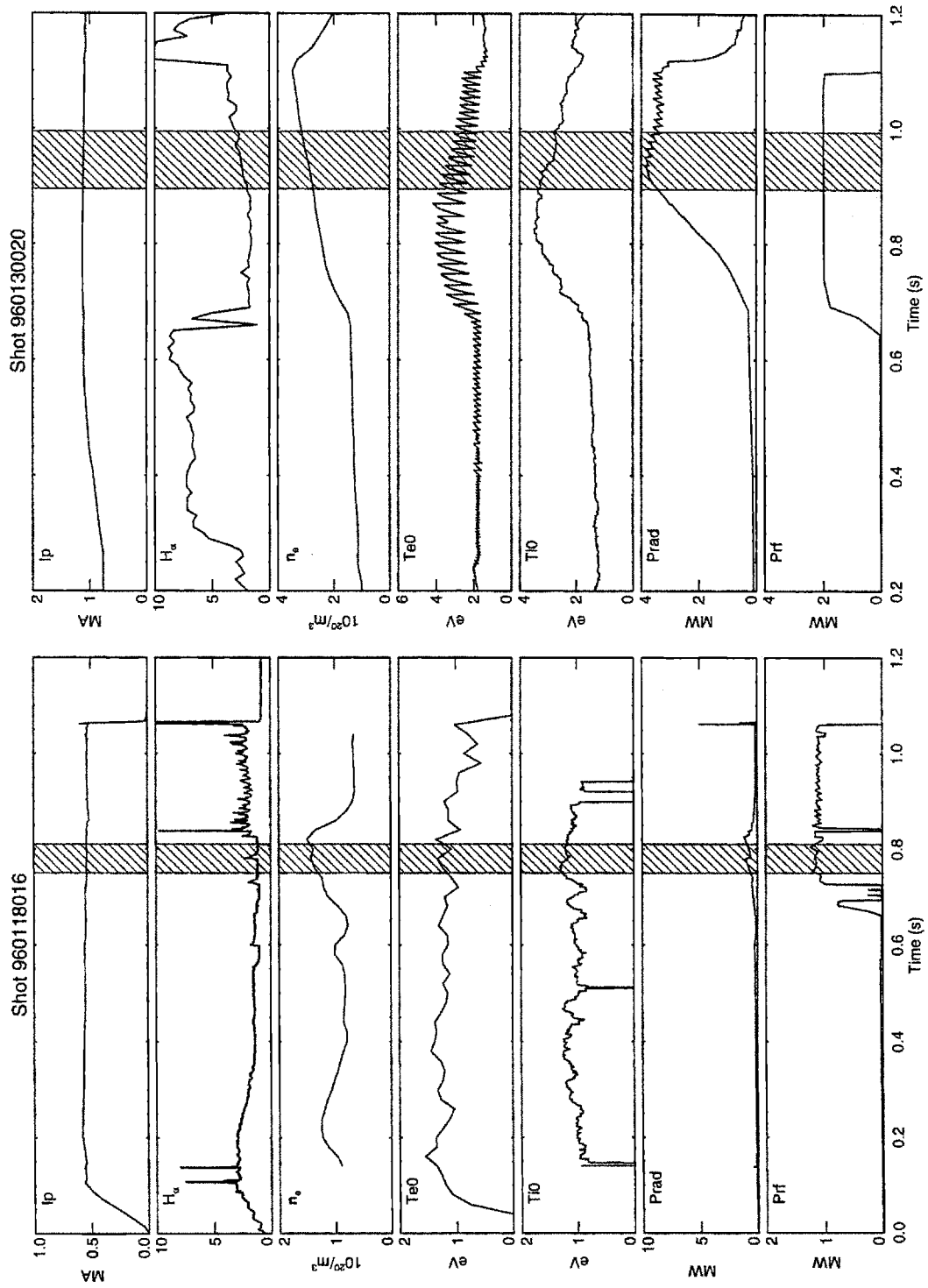


Figure 4.33: Time history of shots in H-Mode group *a*. Shot 960118016: $B_t = 2.5$ T. Shot 960130020: $B_t = 5.4$ T.

Shot	960118016	960130020
t_0 (s)	0.78	0.94
B_t (T)	2.5	5.4
I_p (MA)	0.5	1.1
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	1.38	2.90
P_{RF} (MW)	0.92	2.01
U/B^2	4.82	4.40
$\langle n_e \rangle / B^{4/3}$	0.40	0.31
β_t	1.10	1.03
β_p	0.47	0.45
$q_{\psi 95}$	3.26	3.23
ν^*_e	0.54	0.40
T_i/T_e	0.87	1.01
κ	1.60	1.59
Z_{eff}	1.64	2.48
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.18	0.13
$\rho^* \times 1000$	6.24	4.21
$B\tau_E$ (T s)	0.06	0.22
TRANSP run	TRANSP02	TRANSP02

Table 4.14: Parameters for H-Mode, group a

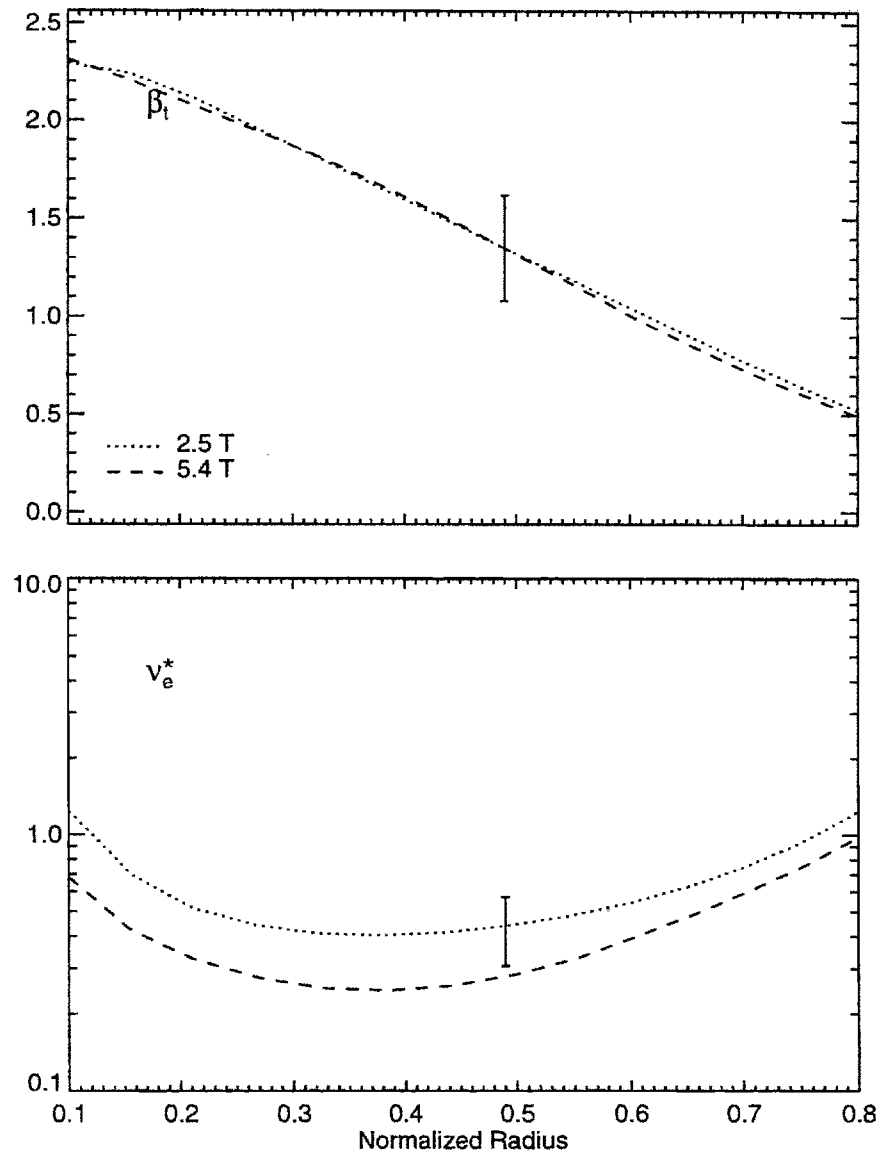


Figure 4.34: Dimensionless parameter profiles for H-Mode group a shots

power absorption for 960118016 was measured to be $f_{P_{RF}} = 0.80 \pm 0.10$.

Local analysis is addressed by the four plots in figure 4.35. The total input and conducted powers in figures 4.35(a) and (b) show a gyroBohm-like scaling ($\sim B^1$) over most of the profile. Radiated power is not an issue for these shots, as P_{rad}/P_{in} is low for both. The scaling of χ_{eff} is also gyroBohm-like within the uncertainty range, consistent with the scaling of $B\tau_E$ for the two shots, $B\tau_E \sim B^{-3.4 \pm 0.9}$, and with the scaling of the whole group.

The scalings of the individual channels are again subject to large uncertainties. P_{ini} for 960118016 is very low compared to 960130020, so the ratio $\mathcal{R}(P_{ini})$ is very large as shown in figure 4.35(a). As shown in figures 4.36 and 4.37, almost all of the ICRF power deposition is into the minority, which then slows down predominantly on the electrons. The heating for 960130020 appears to be more evenly distributed between the species.

Because of the large discrepancy in input power, the scaling of χ_i is between Bohm and Goldston. Since the electrons receive most of the power at low field, they show a better than gyroBohm scaling. Both scalings appear to be dominated by the large difference in input power to the ions between the two shots. This suggests that the relative heating of the two species, P_{ine}/P_{in} and $P_{ini}/P_{in} = 1 - P_{ine}/P_{in}$, is another dimensionless parameter that should be kept constant when measuring their ρ^* scaling.

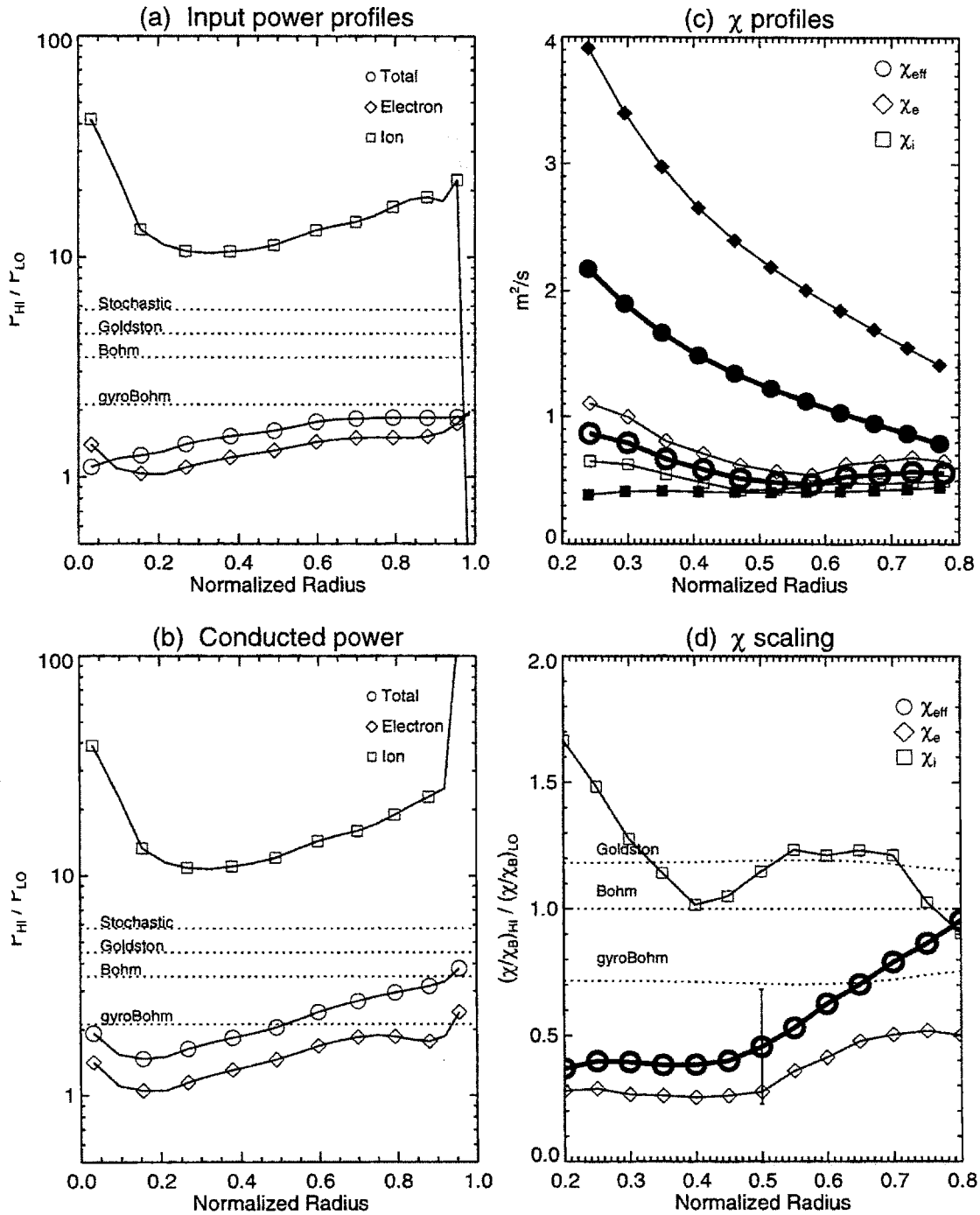


Figure 4.35: Local analysis plots for ρ^* scaling H-Mode group *a*. Shown are the ratios of P_{in} and P_{cond} (high field/low field), the χ profiles for both shots (filled = low, open = hi), and the ratio $\mathcal{R}(\chi/\chi_B)$ for each species.

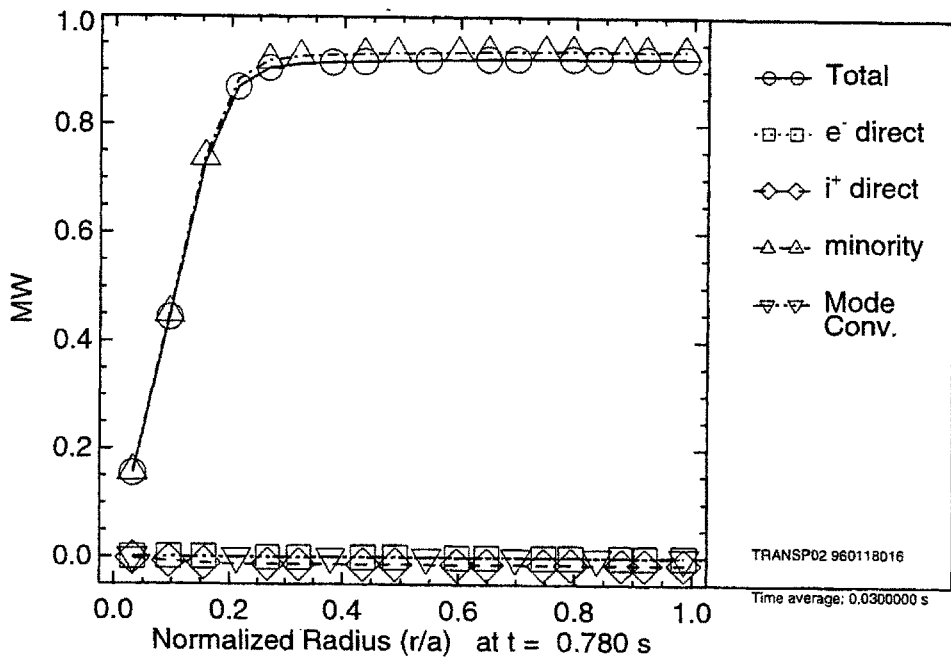


Figure 4.36: Volume-integrated ICRF power deposition at $t=0.78$ s for shot 960118016 in H-Mode, group a

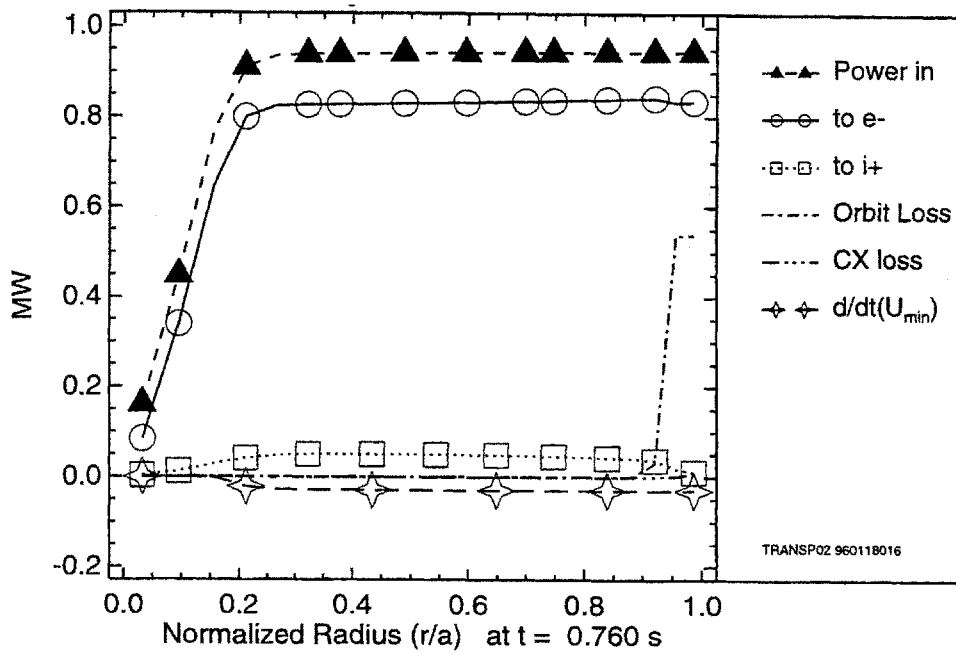


Figure 4.37: Minority slowing down at $t=0.78$ s for shot 960118016 in H-Mode, group a

Shot	960226027	960130028
t_0 (s)	0.98	0.91
B_t (T)	2.8	5.3
I_p (MA)	0.5	1.0
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	1.28	2.96
P_{RF} (MW)	0.73	1.36
U/B^2	3.56	3.49
$\langle n_e \rangle / B^{4/3}$	0.33	0.32
β_t	0.91	0.83
β_p	0.42	0.49
$q_{\psi 95}$	3.51	3.54
ν^*_{e}	0.53	0.61
T_i/T_e	0.90	1.06
κ	1.64	1.60
Z_{eff}	2.10	2.25
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.16	0.11
$\rho^* \times 1000$	5.39	3.76
$B\tau_E$ (T s)	0.06	0.24
TRANSP run	TRANSP03	TRANSP01

Table 4.15: Parameters for H-Mode, group b

Group b

There are only a total of three slices from two shots making up group *b*: two from 960226027 (2.6 T) and one from 960130028 (5.3 T). The ICRF heating in the former shot was tripping as the plasma went in and out of H-Mode. Over the time range used for analysis the RF power was constant. Its fractional absorption was measured as 0.8 ± 0.1 . Figure 4.38 plots the time histories of the two shots, while their parameters are listed in 4.15.

Since there are only two shots, global ρ^* analysis is limited to determining the slope of the line $B\tau_E$ versus ρ^* formed by the two points, as shown in figure 4.39: $B\tau_E = 210\rho^{*-3.8 \pm 1.5}$. Determining α using (4.44) gives $\alpha = 1.8 \pm 1.5$, which is consistent with gyroBohm scaling within the uncertainty. Of the two slices from 960226027, the one at $t = 0.975$ s was chosen for local analysis. The dimensionless parameter profiles, shown in figure 4.40 match very well.

The scaling of the total input power P_{in} , shown in figure 4.41(a), is gyroBohm over most of the profile. Since the radiated power fractions are low, the scaling of χ_{eff} is also gyroBohm. This is consistent with the scaling of $B\tau_E$. As seen for group *a*, most of the ICRF heating is deposited to the electrons for the low field shot, although the ratio $\mathcal{R}(P_{\text{in}i})$ is not as large for these shots. The scalings of χ_e and χ_i are consistent with the scalings of their conducted powers, that is gyroBohm and Bohm/Goldston

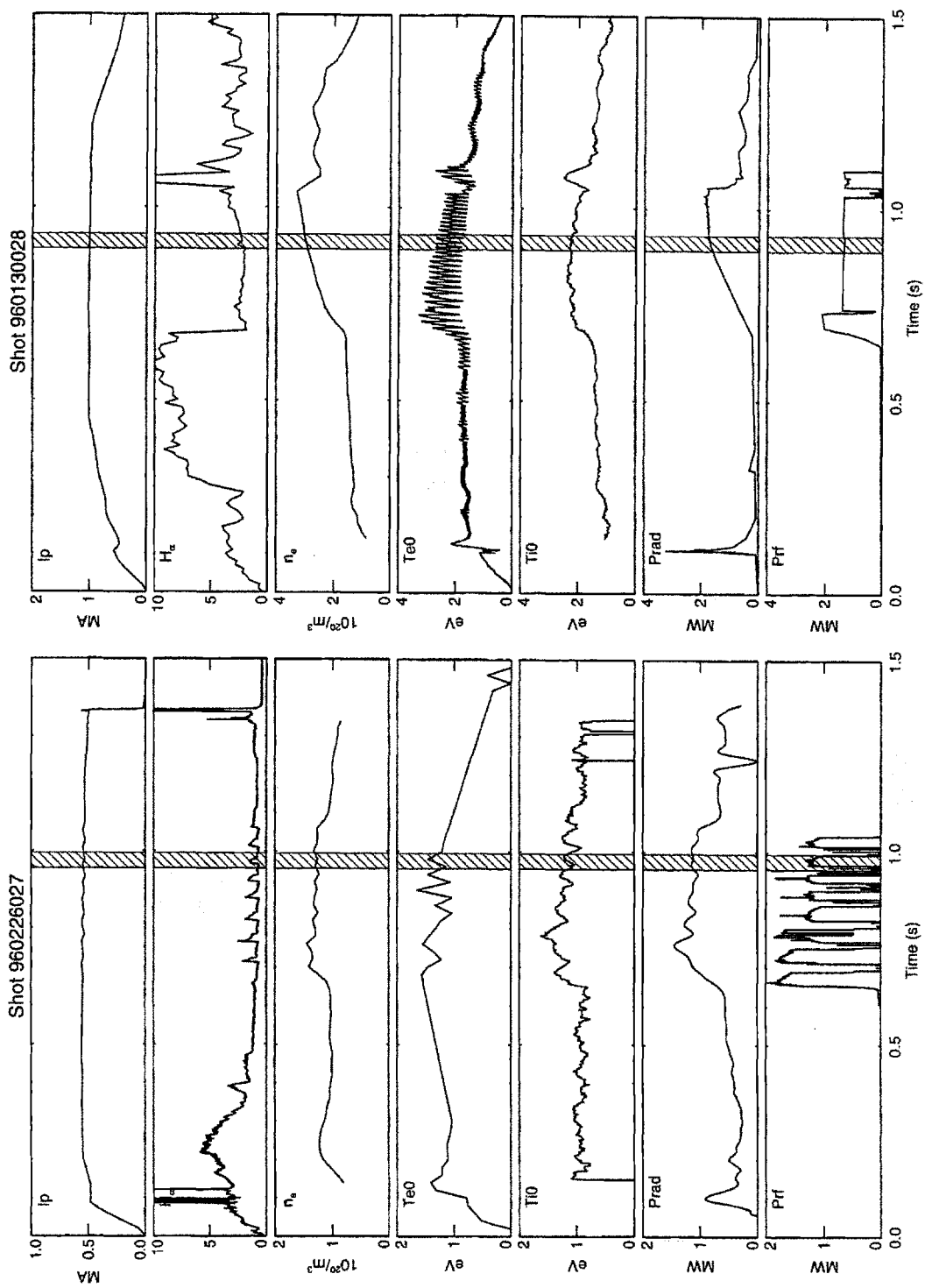


Figure 4.38: Time history of shots in H-Mode group *b*. Shot 960226027: $B_t = 2.8$ T. Shot 960130028: $B_t = 5.3$ T.

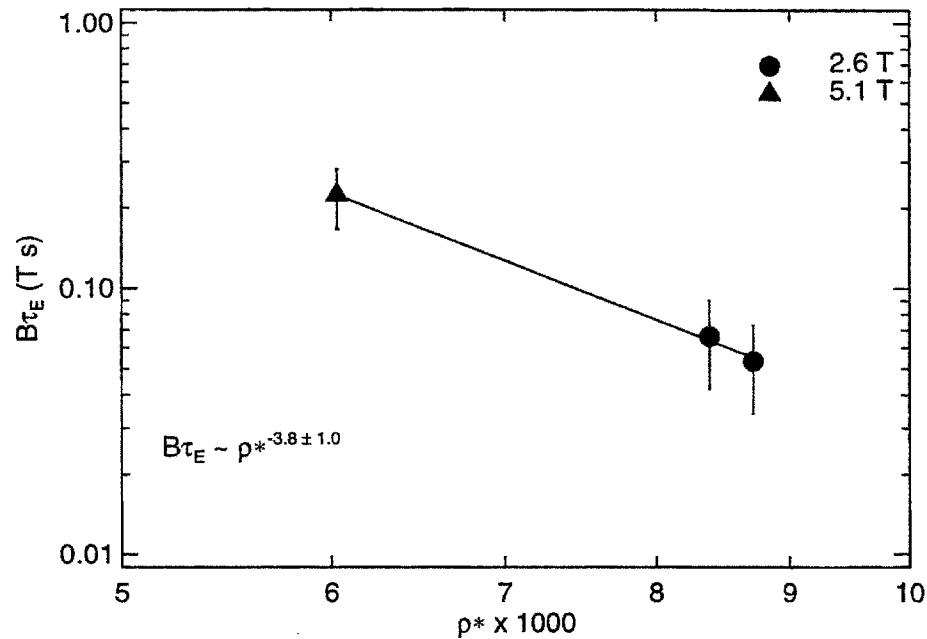


Figure 4.39: $B\tau_E$ versus ρ^* for H-Mode group b

Group	Global scaling, group	Local scaling, pair	$\mathcal{R}(f_{rad})$
Ha	1.1 ± 0.9	2.3 ± 1.4	0.7
Hb	1.8 ± 1.5	1.0 ± 1.3	0.7

Table 4.16: Summary of ρ^* scaling in H-Mode. Global scaling is value of α from $B\tau_E \propto \rho^{*-2-\alpha}$. Local scaling is value of α from $\chi \propto \chi_B \rho^{*\alpha}$. Also listed is the ratio of the radiated power fraction f_{rad} at high field to that at low field.

scaling respectively.

Summary

Global scaling for the two H-Mode groups is observed to be gyroBohm-like, that is $\alpha \simeq 1$, although the large error range (± 1) does not preclude Bohm scaling. Local scaling of χ_{eff} is consistent with global scaling, that is $\chi_{eff}/\chi_B \sim \rho^*$. Given that there are very large uncertainties, and the relative heating of each species is very different for the two fields, the observed scalings for χ_e and χ_i — gyroBohm and Bohm to Goldston like respectively — are not to be taken as definitive. Table 4.16 provides a summary of the $B\tau_E$ and χ_{eff} scalings.

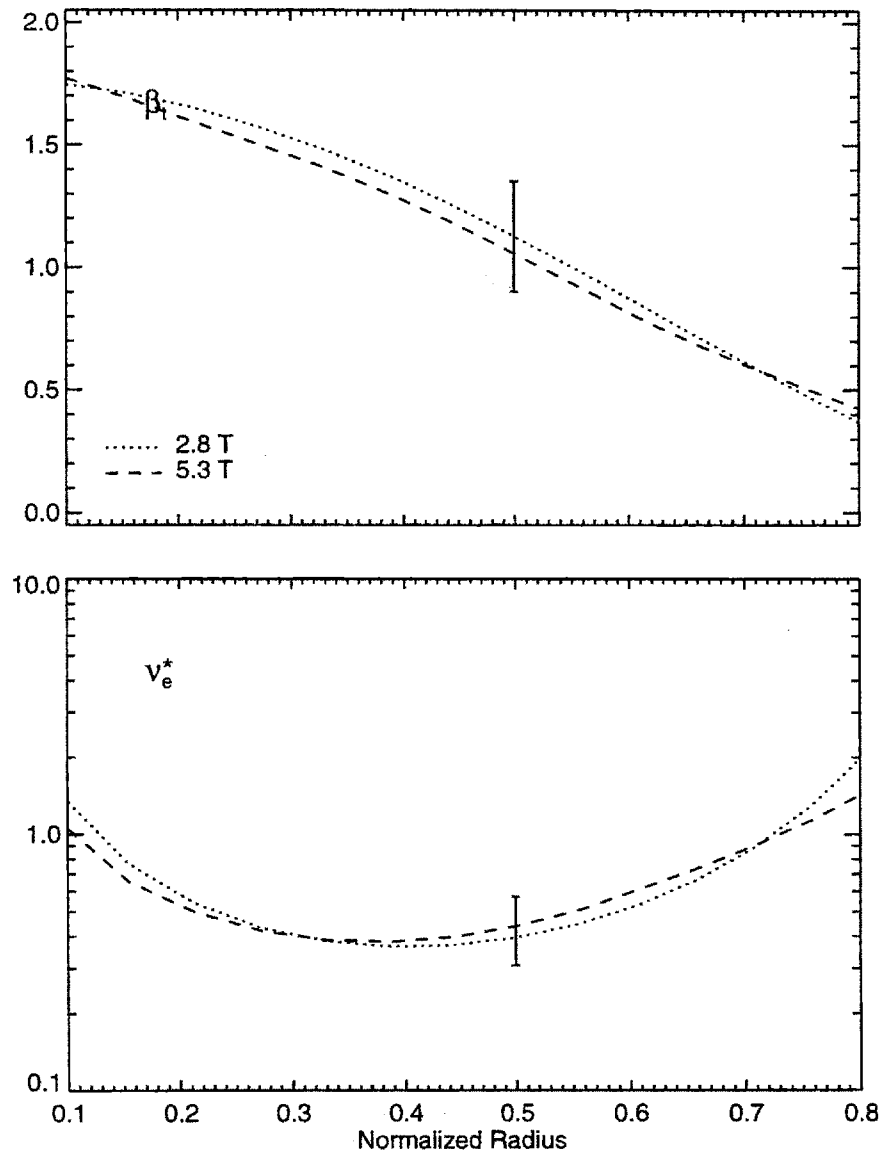


Figure 4.40: Dimensionless parameter profiles for H-Mode group b shots

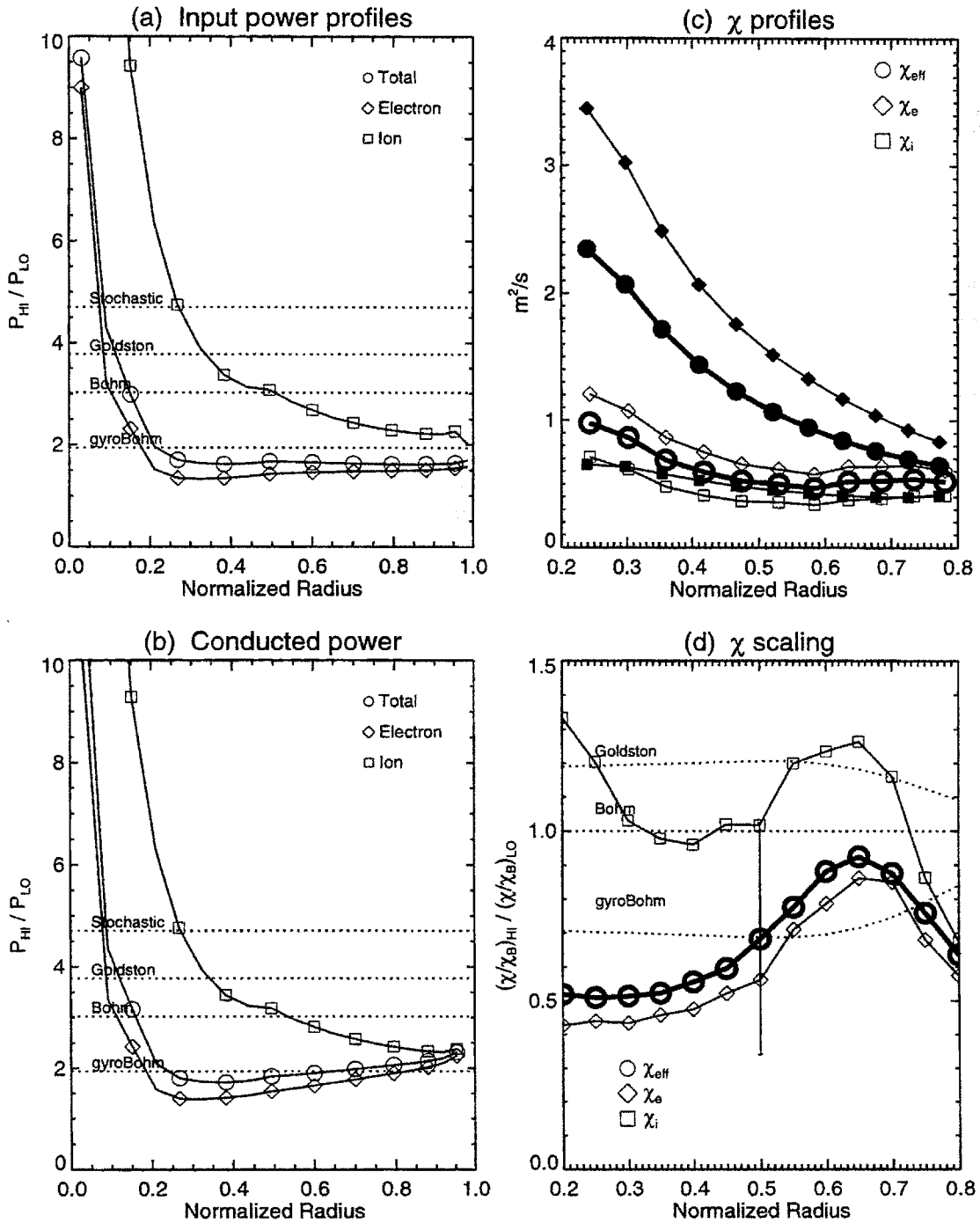


Figure 4.41: Local analysis plots for ρ^* scaling H-Mode group *b*. Shown are the ratios of P_{in} and P_{cond} (high field/low field), the χ profiles for both shots (filled = low, open = hi), and the ratio $\mathcal{R}(\chi/\chi_B)$ for each species.

4.6 ν^* Scaling

The ν^* range of shots in the core table, shown in figures 4.4 and 4.31 for L-Mode and H-Mode respectively, is almost two orders of magnitude. It should therefore be possible to get a reasonable accurate picture of the ν^* scaling of transport despite the large uncertainty in ν^* .

Analogous to ρ^* scaling, ν^* scaling is the scaling of $B\tau_E$ or χ/χ_B with ν^* , this time keeping ρ^* as well as the other dimensionless parameters fixed.⁸ Unlike ρ^* scaling, there are no particular named scalings identified because of physics considerations.

Once again, B is used as the independent variable. For ρ^* , β_p , and $q_{\psi 95}$ to remain fixed as ν^* varies requires

$$I_p \sim B^1, \quad (4.74)$$

$$n_e \sim B^0, \quad (4.75)$$

$$T_e \sim B^2, \quad (4.76)$$

in analogy to (4.15)-(4.17). This gives $\nu^* \sim B^{-4}$.

C-Mod's *LOGBOOK* RDB database made possible the identification of shots suitable for ν^* scaling. No dedicated runs were attempted. The following sections present results for L-Mode and H-Mode plasmas.

4.6.1 L-Mode

The data in the set \mathcal{L} are replotted in figure 4.42 as β_p and $q_{\psi 95}$ versus ρ^* to facilitate the identification of shots suitable for ν^* scaling. The strong ρ^* scalings measured in the previous section indicate that it is particularly important to obtain a good match in ρ^* . The best match that could be obtained for the shots in \mathcal{L} is 10%. Assuming Bohm scaling for L-Mode, $B\tau_E \sim \rho^{*-2}$, a mismatch of ρ^* this large gives a 20% difference in $B\tau_E$. In summary, the criteria for judging shots as similar was taken to be:

$$|\beta_p - \beta_{p0}| < 0.05 \quad (4.77)$$

$$|q_{\psi 95} - q_{\psi 950}|/q_{\psi 950} < 8\% \quad (4.78)$$

$$|\rho^* - \rho^*_0|/\rho^*_0 < 10\%, \quad (4.79)$$

where the subscript "0" refers to a shot with $B_t \neq 5.3$ T selected as the group center. Only one group of shots was identified, as shown by the filled symbols. Table 4.17 summarizes the parameters for this group.

Figure 4.43 plots a regression of $B\tau_E$ versus ν^* . The uncertainty in ν^* was added in quadrature to the uncertainty in $B\tau_E$, as done for the ρ^* regressions. Treating

⁸For ν^* scaling the term *dimensionally similar* will refer to shots with all dimensionless parameters matching save ρ^* .

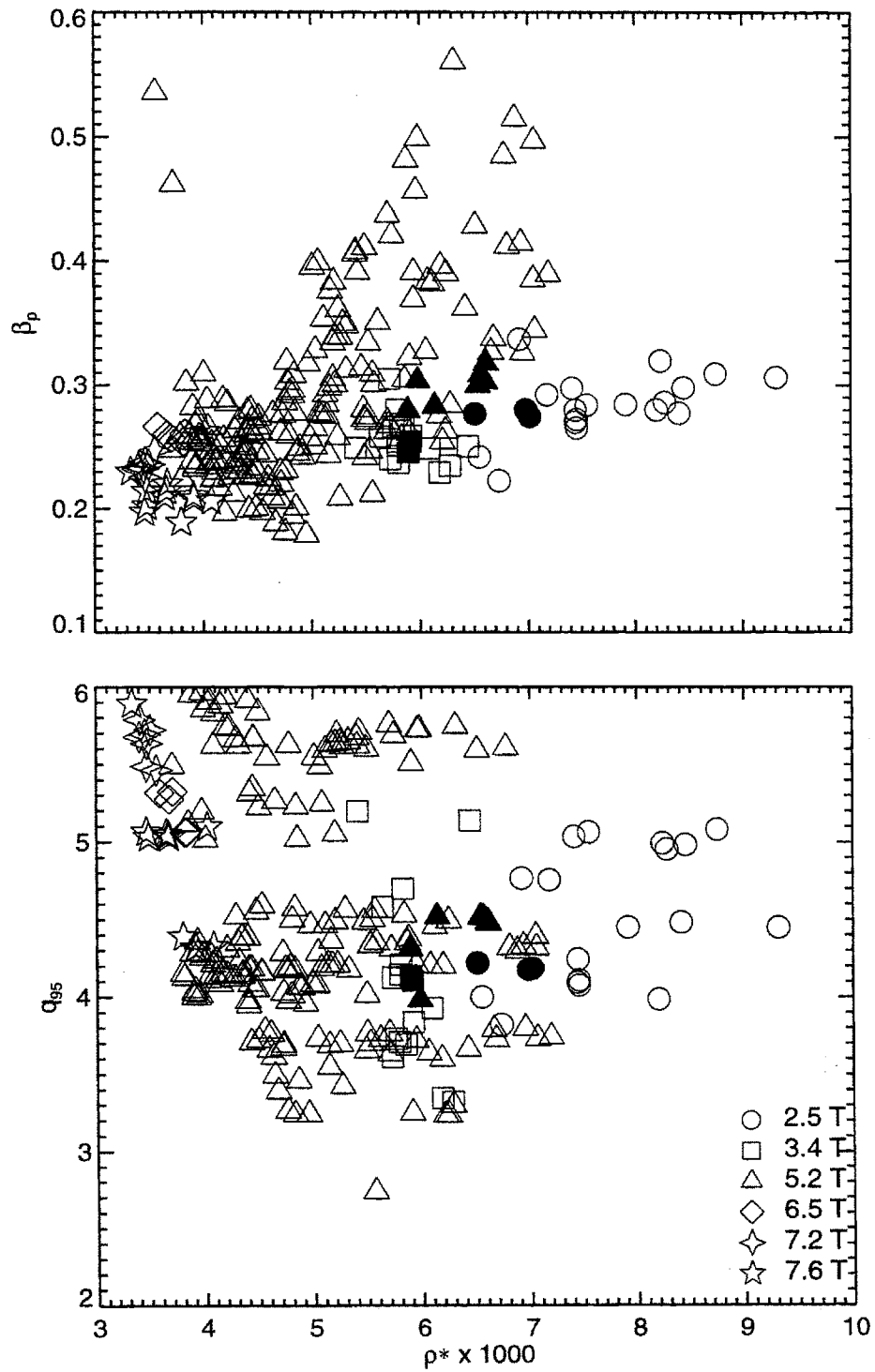


Figure 4.42: Distribution of L-Mode shots for ν^* scaling shot selection. One group is identified (filled symbols).

Group	$q_{\psi 95}$		β_p		$\rho^* \times 1000$	
	average	s.d.	average	s.d.	average	s.d.
a	4.28	0.06	0.28	0.01	6.33	0.11

Table 4.17: L-Mode shot groups for global ν^* scaling

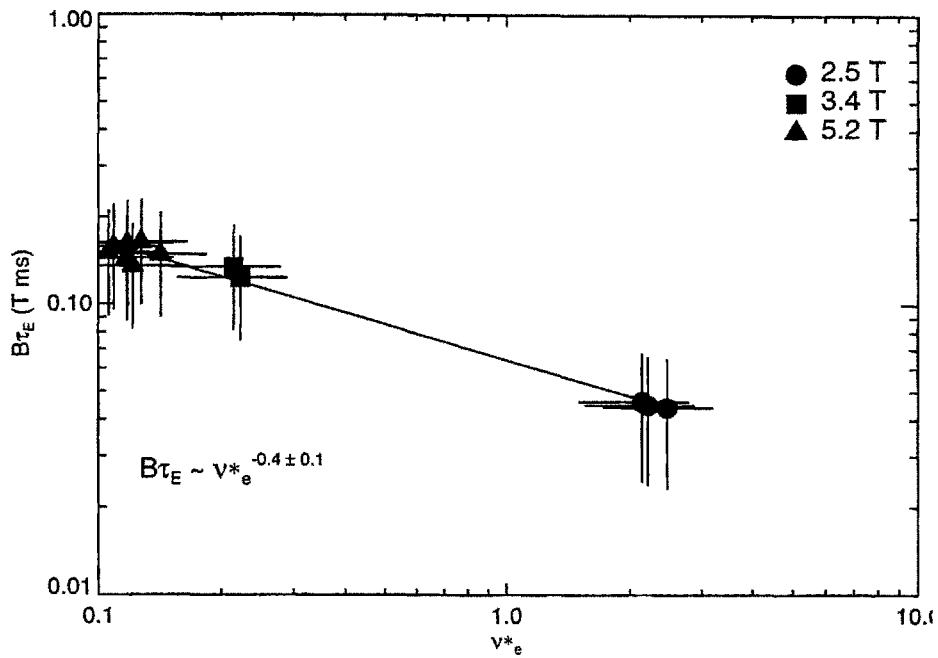


Figure 4.43: $B\tau_E$ versus ν^* for L-Mode.

Shot	960201027	950215018
t_0 (s)	0.71	0.81
B_t (T)	2.6	5.3
I_p (MA)	0.4	0.8
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	1.13	1.09
P_{RF} (MW)	0.00	1.02
U/B^2	1.69	1.73
β_t	0.43	0.41
β_p	0.35	0.35
$q_{\psi 95}$	4.26	4.49
$\rho^* \times 1000$	3.79	4.00
T_i/T_e	0.86	0.87
κ	1.56	1.70
Z_{eff}	1.49	2.32
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.24	0.24
ν^*_e	1.74	0.15
$B\tau_E$ (T s)	0.05	0.17
TRANSP run	TRANSP01	TRANSP01

Table 4.18: Parameters for L-Mode, group a

the points as independent gives $B\tau_E \sim \nu^{*-0.4 \pm 0.1}$. This is slightly stronger than the dependence seen in the residuals from the ρ^* regressions ($\sim \nu^{*-0.3}$) but consistent within the uncertainty.

Two shots were selected from this group for local analysis: 960201027 at 2.6 T and 950215018 at 5.3 T. Now, the profiles of β_p and ρ^* must match, which they do as shown in figure 4.44. Table 4.18 summarizes the parameters for the two shots. Note that the higher field shot is RF-heated.

Figure 4.45 plots the scalings of χ_{eff} , χ_e , and χ_i with ν^* . The scaling is plotted as γ defined by:

$$\gamma = \frac{\log(\mathcal{R}(\chi/\chi_B))}{\log(\mathcal{R}(\nu^*))}, \quad (4.80)$$

where \mathcal{R} is defined by (4.41). Assuming (2.44), the scaling of χ_{eff} is consistent with the global scaling within the uncertainty range over most of the profile. At $r/a = 0.5$, $\chi_{\text{eff}}/\chi_B \sim \nu^{*0.5 \pm 0.4}$. Once again, the errors in χ_e and χ_i are large. The center of their uncertainty ranges roughly show the same scaling with ν^* as χ_{eff} .

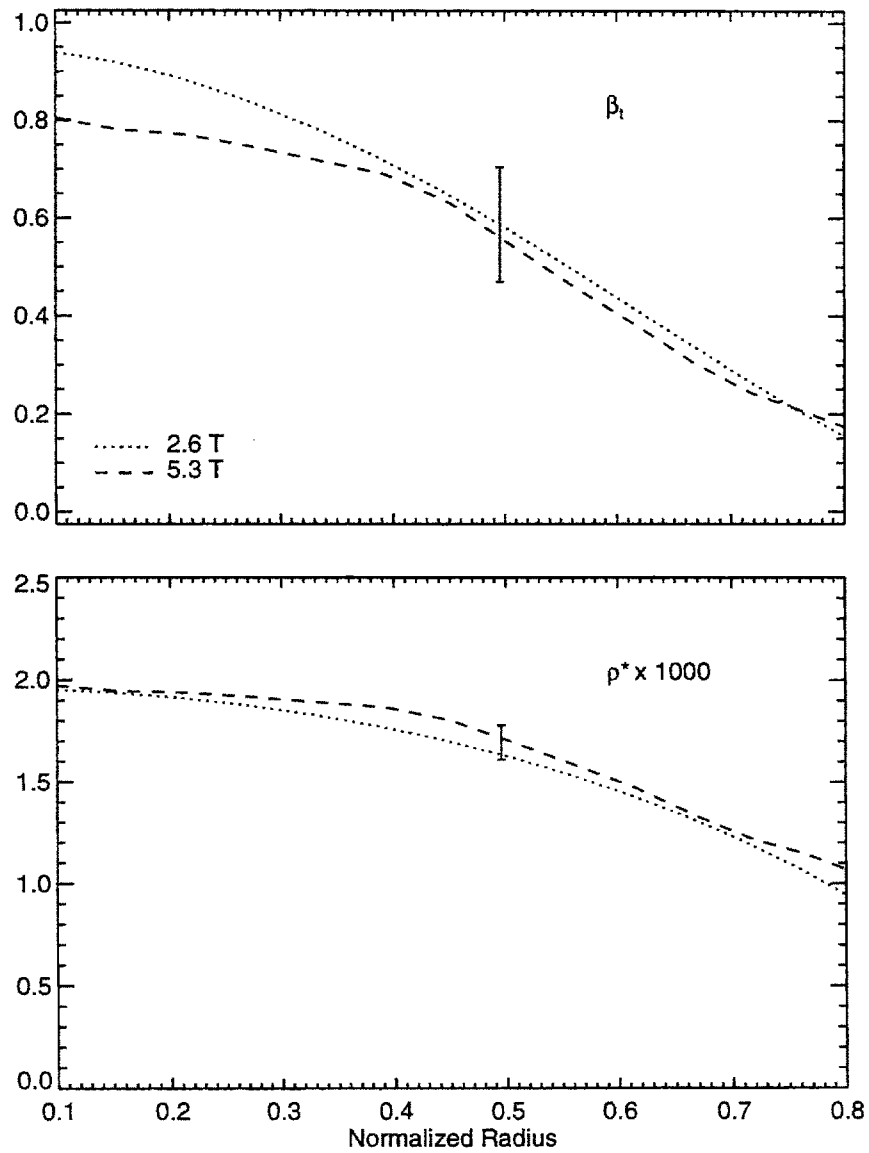


Figure 4.44: Dimensionless parameter profiles for L-Mode ν^* scaling

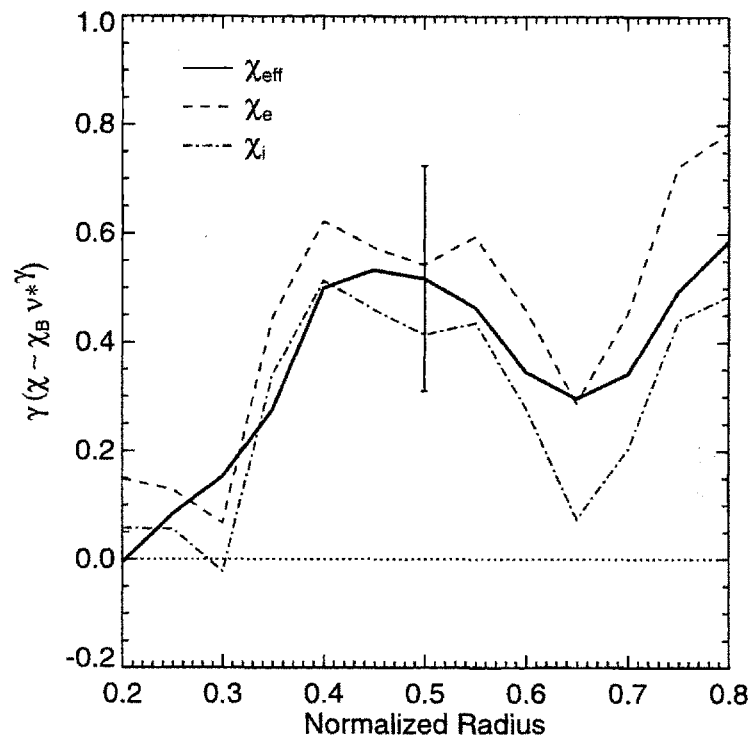


Figure 4.45: Scaling of χ with ν^* for L-Mode. $\gamma = \mathcal{R}(\chi/\chi_B)/\mathcal{R}(\nu^*)$.

Group	$q_{\psi 95}$		β_p		$\rho^* \times 1000$	
	average	s.d.	average	s.d.	average	s.d.
a	3.20	0.01	0.48	0.01	7.38	0.05

Table 4.19: H-Mode shot groups for global ν^* scaling

4.6.2 H-Mode

The procedure for ν^* scaling analysis in H-Mode is the same as for L-Mode. It was possible to find one group of similar shots in \mathcal{H} satisfying the following constraints:

$$|\beta_p - \beta_{p0}| < 0.05 \quad (4.81)$$

$$|q_{\psi 95} - q_{\psi 95_0}|/q_{\psi 95_0} < 5\% \quad (4.82)$$

$$|\rho^* - \rho^*_0|/\rho^*_0 < 5\%, \quad (4.83)$$

Figure 4.46 identifies the group among the shots in \mathcal{H} . Table 4.19 summarizes the parameters for this group.

Note that there are only four shots in the group, two at low field, and two at higher field. The regression of $B\tau_E$ versus ν^* , plotted in figure 4.47, is therefore subject to greater uncertainty. The result is

$$B\tau_E \sim \nu^{*-1.0 \pm 0.2} \quad (4.84)$$

which is a much stronger dependence on ν^* than shown by the whole dataset \mathcal{H} or by the residuals from the ρ^* regressions.

The two shots selected from this group for local analysis are 960118030 and 960130020. Their parameters are summarized in table 4.20, while figure 4.48 plots their dimensionless parameter profiles. The fractional RF power absorption for 960118030 could not be measured so was assumed to be 70% as per the discussion in § 3.5.3.

The scaling of χ_{eff} , which at $r/a = 0.5$ is $\chi_{\text{eff}}/\chi_B \sim \nu^{*1.0 \pm 0.4}$, is consistent with the global scaling as shown in figure 4.49. This assumes that $\tau_E \sim 1/\chi_{\text{eff}}$, as given by (2.44). Examining the center of the even larger ranges of uncertainty of χ_e and χ_i shows that the electrons have nearly the same scaling with ν^* as χ_{eff} , while the ions have a weaker scaling.

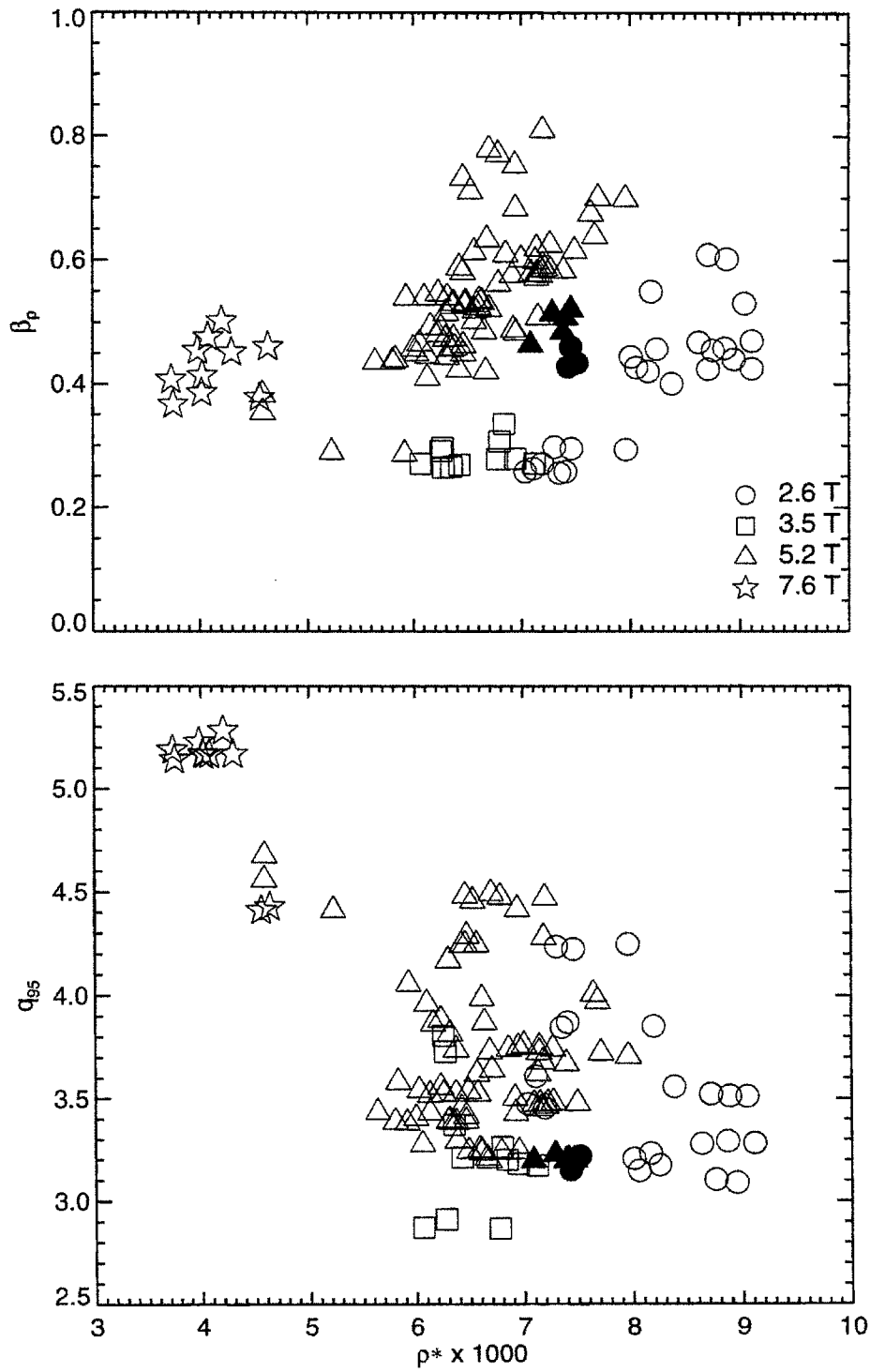
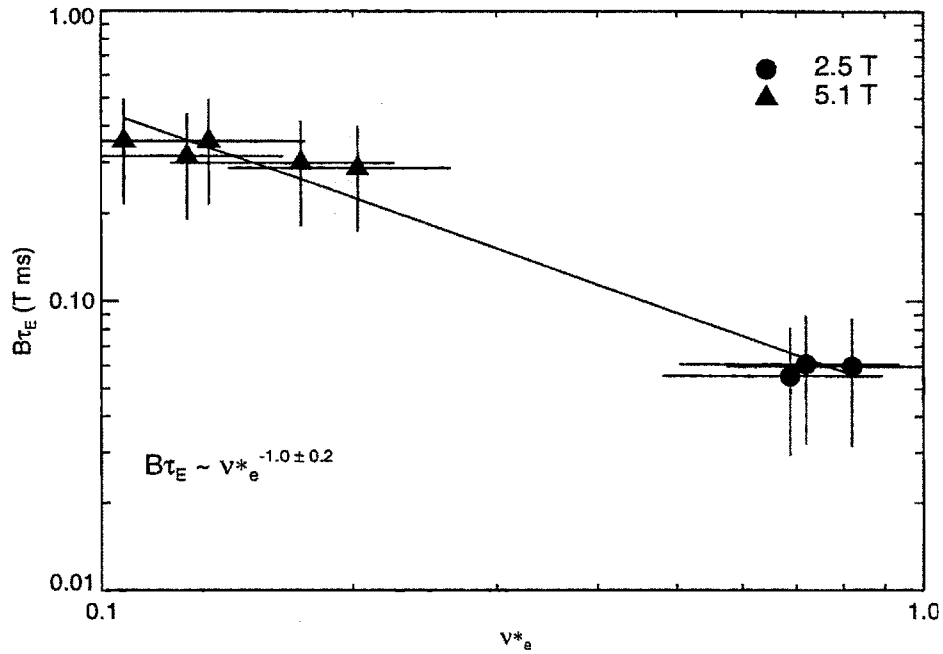


Figure 4.46: Distribution of H-Mode shots for ν^* scaling shot selection. One group is identified (filled symbols).

Figure 4.47: $B\tau_E$ versus ν^* for H-Mode

Shot	960118030	960130020
t_0 (s)	1.06	0.86
B_t (T)	2.6	5.3
I_p (MA)	0.6	1.1
$\langle n_e \rangle$ ($10^{20}/\text{m}^3$)	2.25	2.60
P_{RF} (MW)	0.93	1.99
U/B^2	5.26	4.76
β_t	1.28	1.14
β_p	0.53	0.49
$q_{\psi 95}$	3.24	3.19
$\rho^* \times 1000$	4.90	4.68
T_i/T_e	0.92	1.00
κ	1.62	1.60
Z_{eff}	1.05	2.39
$P_{\text{rad}}/P_{\text{in}}$ ($r/a = 0.5$)	0.11	0.08
ν^*_e	1.10	0.23
$B\tau_E$ (T s)	0.06	0.30
TRANSP run	TRANSP01	TRANSP02

Table 4.20: Parameters for H-Mode, group a

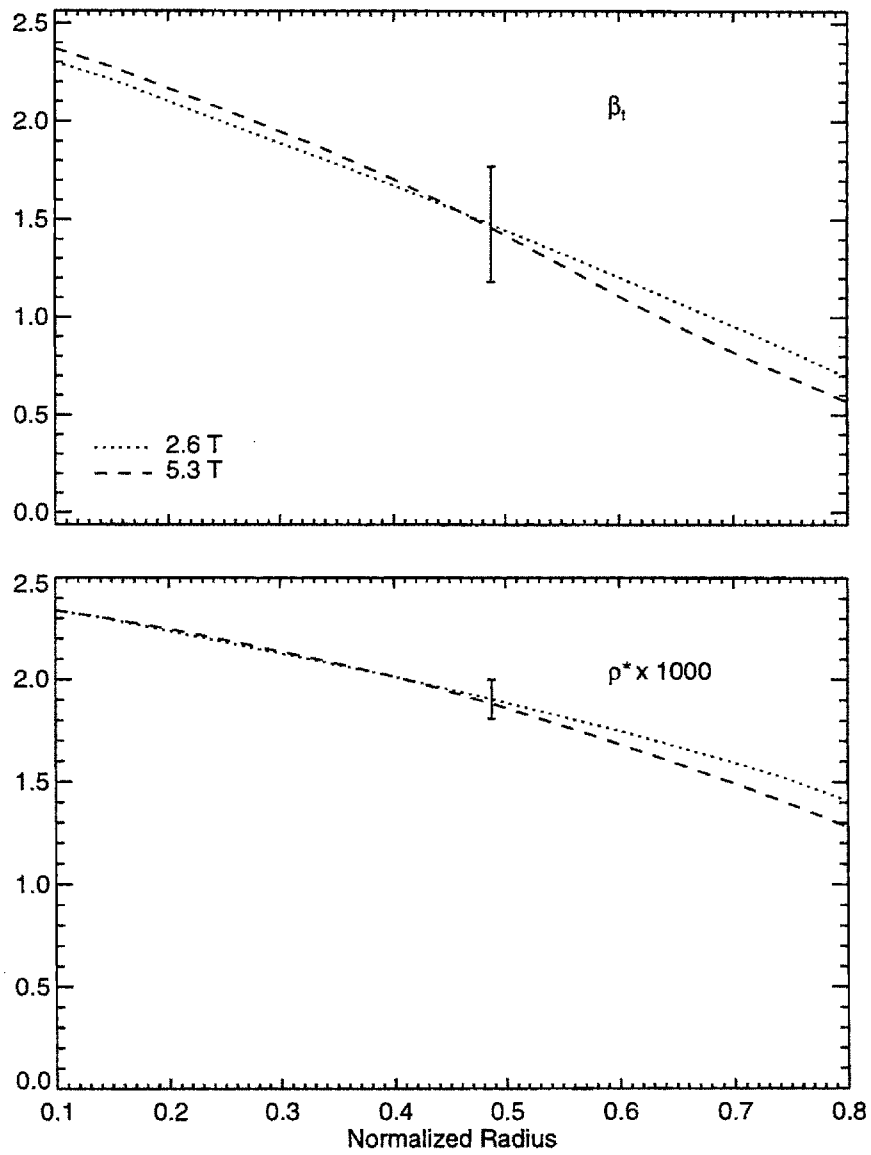


Figure 4.48: Dimensionless parameter profiles for H-Mode ν^* scaling

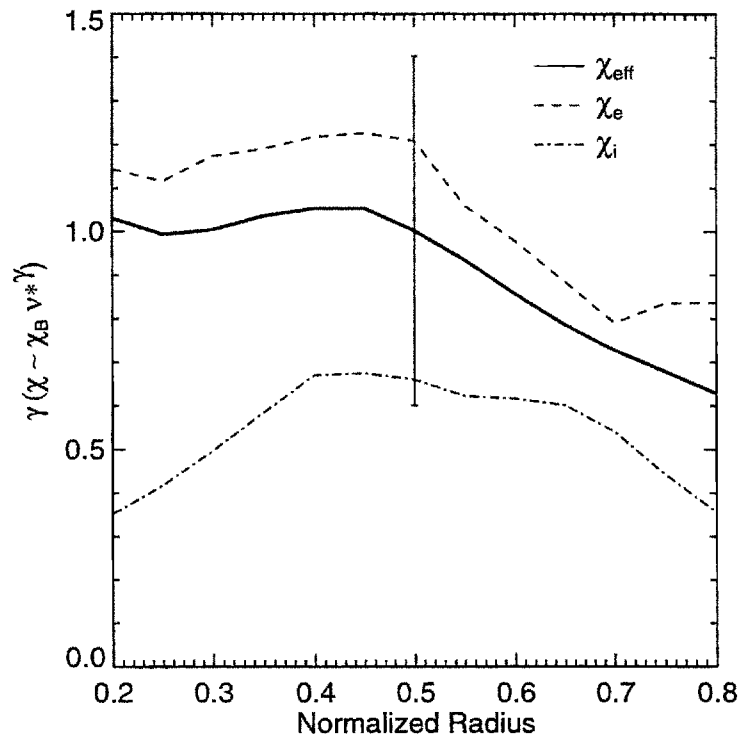


Figure 4.49: Scaling of χ with ν^* for H-Mode. $\gamma = \mathcal{R}(\chi/\chi_B)/\mathcal{R}(\nu^*)$.

4.7 Discussion

This section summarizes the results from the ρ^* scaling experiments on Alcator C-Mod and compares them to the results from experiments on other tokamaks. The implications for ITER from these results are also presented.

4.7.1 Summary of ρ^* Scaling Results

Three of the four groups of dimensionally similar L-Mode shots studied (*La*, *Lc*, *Le*) had Bohm-like global scaling. Averaging the results from the three groups gives $B\tau_E \propto \rho^{*-2.0 \pm 0.4}$, or $\alpha = 0.0 \pm 0.4$. Group *Lc* had shots with $B_t = 8$ T. The fourth group, *Lb*, was Goldston-like, with $\alpha = -0.8 \pm 0.6$. Other than being at slightly higher q_{95} and ν^* , no significant difference was found between *Lb* and the other three groups. The average of the scalings from all four groups is $B\tau_E \propto \rho^{*-1.7 \pm 0.3}$, or $\alpha = -0.3 \pm 0.3$, which is identical to the scaling obtained by regressing the entire L-Mode dataset against ρ^* .

Local ρ^* scaling of L-Mode plasmas was inconclusive. It was possible to identify a pair of shots with matching β_p and ν^* profiles from three of the four groups: *La*, *Lb*, and *Le*. The first showed a Goldston-like scaling for χ_{eff} , consistent within its uncertainty range with the global scaling of the group. The other two showed a χ_{eff} scaling much worse than Goldston, that is $\alpha \ll -0.5$, where $\chi_{\text{eff}} \propto \chi_B \rho^{*\alpha}$. However, for all four groups the radiated power fraction $P_{\text{rad}}/P_{\text{in}}$ was not well matched, being much higher at 2.6 T than at 5.3 T, and was instrumental in determining the scaling of χ_{eff} . Since ρ^* scaling is based in part on the assumption that atomic physics effects such as radiation are not important, the L-Mode ρ^* scaling results may not be valid.

The H-Mode plasmas studied were not affected by P_{rad} , and so the conclusions are more robust. Two groups of dimensionally similar shots were identified (*Ha* and *Hb*). Both showed gyroBohm scalings in $B\tau_E$ and χ_{eff} . Averaging the global results from the two groups gives $B\tau_E \propto \rho^{*-3.3 \pm 0.9}$, or $\alpha = 1.3 \pm 0.9$, which is consistent with the scaling observed for the entire H-Mode dataset. The averaged local scaling is $\chi_{\text{eff}}(r/a = 0.5) \propto \chi_B \rho^{*1.6 \pm 1.0}$.

In both L-Mode and H-Mode, it was not possible to unambiguously determine the scalings of the individual electron and ion channels. This was because a precise measurement of the ion temperature profile was not available, and because the uncertainties in χ_e and χ_i from the electron-ion energy exchange term Q_{ei} were very large. Examining the center of their uncertainty ranges, the scaling of χ_e with ρ^* was consistently stronger than for χ_i , with most groups showing a gyroBohm scaling. In L-Mode the two channels were strongly coupled, so that the scaling of χ_{eff} was intermediate between χ_e and χ_i . In H-Mode it was observed that at low field (2.6 T), the ions were weakly heated by the 2nd harmonic minority H ICRF. The scaling of χ_{eff} was therefore close to the scaling of χ_e .

ν^* scaling results were as follows. In L-Mode, the pair of dimensionally similar

shots studied had $B\tau_E \sim \nu_*^{-0.4 \pm 0.1}$. This is consistent with the scaling of the entire set of L-Mode shots ($B\tau_E \sim \nu_*^{-0.27 \pm 0.9}$) and with the scaling of the residuals from the global ρ_* analysis ($B\tau_E \sim \nu_*^{-0.3 \pm 0.03}$). These ν_* scalings are nearly identical with (4.6), the dimensionless form of ITER89-P (4.6). Local scaling had $\chi_{\text{eff}} \propto \chi_B \nu_*^{0.5 \pm 0.4}$, which is also consistent within the uncertainty.

H-Mode ν_* scaling was stronger than observed for L-Mode. The pair of dimensionally similar shots had $B\tau_E \sim \nu_*^{-1.0 \pm 0.2}$ and $\chi_{\text{eff}} \sim \chi_B \nu_*^{1.0 \pm 0.4}$. The set of all H-Mode shots as a whole showed a weaker dependence: $B\tau_E \sim \nu_*^{-0.5 \pm 0.1}$. These scalings are slightly stronger than the ν_* scaling of the dimensionless form of ITER93-H (4.8).

In closing, recall that the relative insensitivity of the L-Mode global scaling results to the radiated power, and the general observation on C-Mod that the H-factor in L-Mode is independent of the radiated power fraction [11], suggest that a marginal stability model may be more appropriate for χ . The IFS-PPPL model treats χ as marginally stable; the applicability of this model to C-Mod is examined in the next chapter. A brief comparison of the ρ_* scaling results on C-Mod to those from other tokamaks is first presented.

4.7.2 ρ_* Scaling on Other Tokamaks

DIII-D

The most comprehensive set of ρ_* scaling experiments performed on another tokamak were on DIII-D [108] by Petty and Luce [80–82, 89, 109, 110]. Their well-resolved T_i profiles [111], together with the weak coupling between electrons and ions in DIII-D's low density plasmas ($\langle n_e \rangle = 0.1 - 0.5 \times 10^{20}/\text{m}^3$), allowed unambiguous separation of the electron and ion channels. They were the first to show that two species have different ρ_* scalings [109].

In L-Mode, they observe that the electrons are consistently gyroBohm, while the ions vary from Goldston to Bohm-like as $q_{\psi 95}$ decreases. The scalings of χ_{eff} and $B\tau_E$ depend on the heating scheme and the strength of the coupling between species. For ECRF-heated plasmas at low density, the power stays in the electron channel, so the χ_{eff} and $B\tau_E$ scalings are determined by the electrons and are gyroBohm. At higher density the coupling was stronger, so that the ions affected the the overall confinement, making it Bohm-like, while the scalings of the electrons and ions were unchanged. The scalings were shown to be independent of the heating method by repeating the experiments using neutral beam heating, which heated both species equally. Overall, they published the scalings $\chi_e \propto \chi_B \rho_*^{1.1 \pm 0.3}$ and $\chi_i \propto \chi_B \rho_*^{-0.5 \pm 0.3}$.

They also studied ELMy H-Mode plasmas at ITER relevant parameters. Once again the electrons were consistently gyroBohm. For low $q_{\psi 95}$ plasmas, the ions were also observed to be gyroBohm, suggesting that the change in transport between L- and H-Mode was occurring in the ion channel. At higher $q_{\psi 95}$ they observed Bohm-like ions. They interpret these results as suggesting that the deviation from gyroBohm

scaling of the ion channel occurs for short scale lengths of the density profile (L-Mode) or current profile (high $q_{\psi 95}$) [110].

No mention was made of the relative magnitude of the radiated power P_{rad} to the total input power P_{in} in L-Mode, where on C-Mod it was observed to be significant and affect the assumptions underlying ρ^* scaling. It is therefore unclear how C-Mod's results for L-Mode compare to theirs. In H-Mode, they observed $P_{\text{rad}}/P_{\text{in}} = 0.15$, comparable to what was observed on C-Mod.

For ν^* scaling in H-Mode, they observed that $\tau_E \sim \nu^{*-0.35}$. Combined with their β scaling result, $B\tau_E \sim \beta^{0.1}$, and an assumption that $\tau_E \propto I_p^{1.0}$, they were able to suggest an engineering scaling for τ_E in H-Mode based on their dimensionless scaling experiments:

$$\tau_E \propto I_p^{1.0} B_t^{0.09} P_{\text{in}}^{-0.45} n_e^{0.27} a^{1.82}. \quad (4.85)$$

Other Tokamaks

ρ^* scaling experiments [83, 112–114]. were also conducted on JET [115]. In L-Mode, their one set of three dimensionally similar shots exhibited a Bohm-like global scaling [114]. Modelling the T_i profile using $\chi_i \propto \chi_{i,\text{nc}}$, the total conducted heat flux showed a Bohm-like scaling as well, from which they infer that χ_{eff} is also Bohm-like. No mention was made of the radiated power in L-Mode. For ELMy H-Modes, they observe a gyroBohm scaling globally. This is also true for the scaling χ_{eff} within $r/a = 0.6$ measured on two pairs of dimensionally similar shots.

On TFTR [108] two sets of four L-Mode shots were studied [116], one set at low density and the other at high density. In both cases their total conducted power was observed to have a Bohm-like scaling with ρ^* , while the global confinement time was observed to have a Goldston scaling. They explain the difference in scalings as the result of changes in the power deposition profile shape over the scans, a consequence of using neutral beam heating. They did not discuss the radiated power. Standard H-Mode plasmas are not observed on TFTR.

L-Mode on Tore Supra [117] showed a gyroBohm scaling for both $B\tau_E$, χ_{eff} , and χ_e [118]. This is consistent with the L-Mode results from DIII-D for ECRF heating at low density. ECRF heating was also used for the ρ^* scaling experiments on the W7-AS Stellarator [119], with similar results for L-Mode.

Comparison to C-Mod

Qualitatively, these results are the same as those obtained from C-Mod. In all L-Mode experiments with strongly coupled electrons and ions or directly heated ions, the scaling of $B\tau_E$ and χ_{eff} was Bohm-like. χ_{eff} was also Bohm in these cases, which is consistent with C-Mod's result, though the latter is not definitive. The global and χ_{eff} gyroBohm scaling in H-Mode measured by DIII-D and JET is also seen on C-Mod.

As the error in separating the channels on C-Mod was great, it is not possible to make a definitive comparison with the DIII-D results. On every pair of shots examined, the center of the error range of χ_e showed a stronger scaling with ρ^* than the center of the error range of χ_i . In most cases the former was gyroBohm. As the channels were always strongly coupled, χ_{eff} was determined by both, coming out in between the electron and ion scalings. It was not possible to resolve a change in χ_i scaling from Bohm to Goldston with increasing $q_{\psi 95}$. χ_i in H-Mode showed a Bohm-Goldston scaling on C-Mod. DIII-D's observations suggest that this could be the result of a short density or current scale length. The normalized scale length is defined as

$$\frac{L_X}{a} = \frac{X}{a \nabla X} \quad (4.86)$$

where X is either the density or current, and a is the minor radius. In H-Mode, C-Mod's density scale length is $L_n/a \simeq 3$, with a large uncertainty because of the problems with the Abel inversion of the interferometer chords [93]. This value is actually larger than the scale length observed in DIII-D H-Modes with gyroBohm ions. Comparing current scale lengths is not straightforward, as current profile measurements are not available on C-Mod. DIII-D observed Bohm-like ions in H-Mode for $q_{\psi 95} = 7$ and gyroBohm ions for $q_{\psi 95} = 3.8$. Assuming that $q_0 = 1$, ∇q can be approximated as $(q_{\psi 95} - q_0)/a$. Of course, this form does not result in a meaningful scale length L_q . The high and low $q_{\psi 95}$ DIII-D cases have $\nabla q \simeq 11/\text{cm}$ and $5/\text{cm}$ respectively. C-Mod's H-Mode ρ^* scaling shots had $q_{\psi 95} = 3.5$, corresponding to $\nabla q \simeq 11$. It is more likely that the large difference in the ion RF heating as calculated by SPRUCE and FPPRF was responsible for the ion scaling on C-Mod.

Chapter 5

Testing the IFS-PPPL Model of ITG Turbulence

This chapter presents the results of a comparison of the predictions from a particular theoretical model of tokamak transport — the IFS-PPPL model — to Alcator C-Mod plasmas. First, § 5.1 describes the model and how it is used to predict tokamak transport. The specific methods followed in the comparison study on C-Mod are presented in § 5.2, with the results of the study in § 5.3. A summary and discussion of the results is in § 5.4.

5.1 The IFS-PPPL Model

As discussed in chapter 2, theoretical descriptions of anomalous transport are difficult because of the complexity of the nonlinear, electromagnetic toroidal gyrokinetic equation. Analytical solutions of this equation are only possible in drastically simplified situations in which potentially important effects are neglected, and have thus far been unable to accurately describe the experimental observations on tokamaks. Historically, numerical solutions have focussed only on linear or quasilinear versions of the equations, omitting nonlinear, turbulence related effects.

The advent of new nonlinear computer simulation techniques [120,121] and comprehensive models of wave-particle interactions [122,123] have permitted increasingly accurate modelling of tokamak transport. At the forefront of these new numeric treatments is the IFS-PPPL model of Beer, Dorland, Hammett, and Kotschenreuther [31]. Their successful (post-facto) predictions [31, 124] of TFTR [108] L-Mode transport, and the model's implications for the performance of ITER have been the focus of much recent attention

At the heart of their model are two complementary components. The first is a comprehensive linear gyrokinetic code describing the linear dynamics of plasma instabilities on time scales longer than the gyromotion [125]. For TFTR plasmas the code indicated [31] that the dominant instability is the ion temperature gradient

(ITG) mode [29,30], driven unstable when the temperature gradient ∇T exceeds some critical value ∇T_{crit} calculated precisely by the code.

Having identified ITG modes as the primary instability, the second component of the IFS-PPPL model, a nonlinear gyrofluid simulation of ITG driven turbulence, calculates its transport consequences to provide a form for χ suitable for comparison to experiment. The nonlinear code includes the effects of kinetic resonances and toroidal geometry for both the deuterium ITG mode and the weaker, carbon impurity ITG mode, which on TFTR was important only in the supershot regime [126]. In the nonlinear model the electron response is treated as adiabatic. χ_e is therefore determined from the quasilinear χ_e/χ_i ratio calculated by the linear code [126].

5.1.1 ITG Modes

The name “ion temperature gradient (ITG) mode” suggests that the mode is a drift-type instability as discussed in § 2.2.3. In a slab geometry [127], this is indeed the case. For toroidal systems, gradients in the magnetic field, and field-line curvature, play important roles in the physics of the instability, so the mode has an interchange-like character as well. This section provides a simple picture of the physical mechanism behind the ITG instability based on the general discussion of electrostatic anomalous transport presented in § 2.2.3.

The description begins with a perturbation in temperature \tilde{T} with components in the radial and poloidal directions, located on the outside midplane, and labelled “hot” and “cold” in figure 5.1. The ∇B and curvature drifts can be combined in the low β limit, giving a drift velocity:

$$\mathbf{v}_d = \frac{v_{\parallel}^2 + \frac{1}{2}v_{\perp}^2}{\Omega_c B^2} \mathbf{B} \times \nabla B \sim \frac{T}{qB^3} \mathbf{B} \times \nabla B. \quad (5.1)$$

Note that \mathbf{v}_d depends on the the temperature T , has no mass dependence, and its direction depends on the sign of the charge. On average, the particles in the hotter regions of the perturbation ($\tilde{T} > 0$) drift faster than those in the colder regions of the perturbation ($\tilde{T} < 0$); this is represented by the length of the thin vertical arrows in figure 5.1. Consider the surface S in the (r, ϕ) plane. The downward flux of ions through S will be greater than the upward flux of electrons, as the ions are hotter. A positive charge perturbation therefore accumulates in the region around S . The same is true between other extrema of the temperature perturbation as shown. The electric fields from the charge perturbations result in the $\mathbf{E} \times \mathbf{B}$ eddies described in § 2.2.3. Because the charge perturbations arise between the extrema of the temperature perturbations, the relative spatial phase of \tilde{T} and the concomitant $\tilde{\phi}$ is such that the eddies are able to drive transport, in the directions shown by the v_E arrows.

The temperature gradient is the factor responsible for the instability. On the

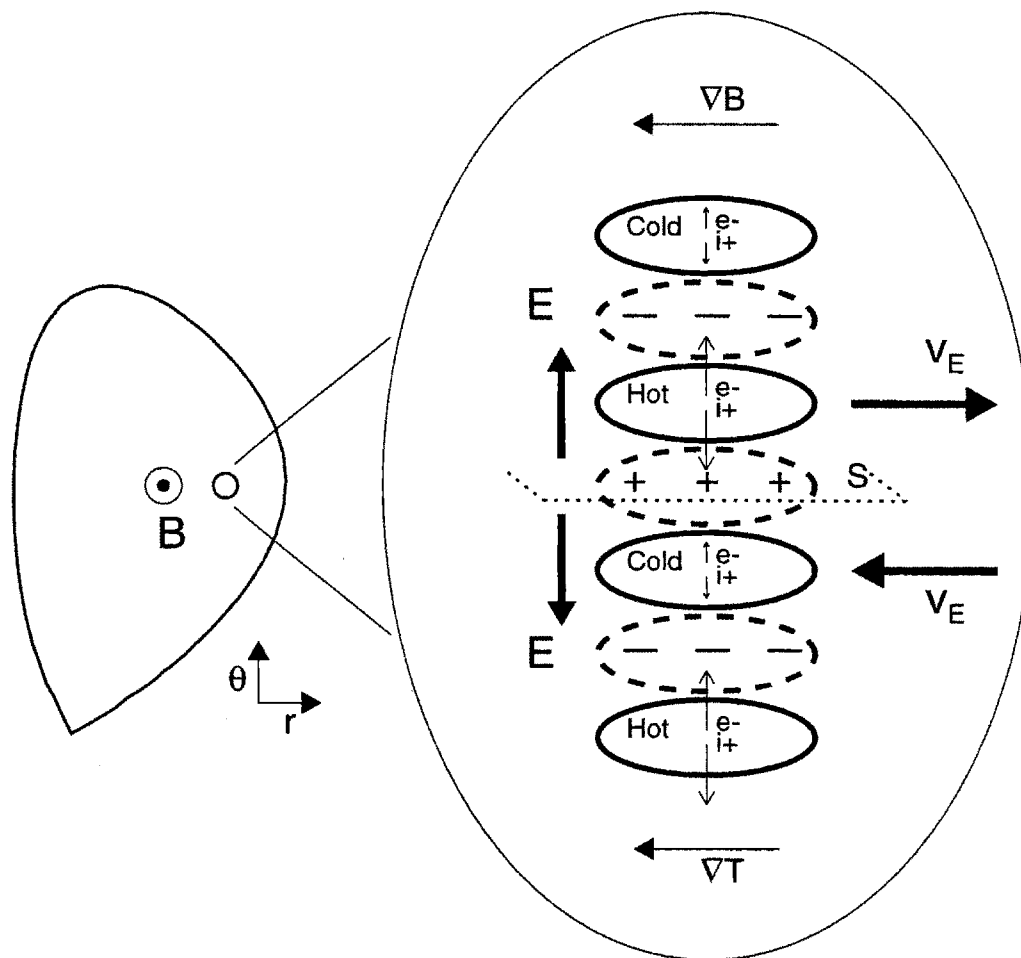


Figure 5.1: Simple picture of the ITG mode (after Beer [28]). A temperature perturbation on the outboard midplane (labelled “hot” and “cold”) gives rise to charge separation due to the temperature dependence of the ∇B drift (indicated by the length of the thin vertical arrows). $\mathbf{E} \times \mathbf{B}$ eddies are driven by the resulting electric field as described in § 2.2.3. The temperature gradient causes the eddies to reinforce the perturbation, as they convect hotter plasma into the regions where $\tilde{T} > 0$ and colder plasma into the regions where $\tilde{T} < 0$. On the inboard midplane, the perturbations are suppressed because ∇T is in the opposite direction.

outboard midplane, the $\mathbf{E} \times \mathbf{B}$ flux convects hotter plasma down the gradient into the hot regions of the perturbation, and colder plasma up the gradient into the cold regions of the perturbation. The magnitude of \tilde{T} is thus increased, resulting in the positive feedback of an instability. On the inboard midplane the direction of the $\mathbf{E} \times \mathbf{B}$ flux is the same, but ∇T is in the opposite direction, so the perturbation is quenched.

Many other phenomena, not represented in this simple picture, affect the ITG mode. Magnetic shear, particle trapping and Landau damping are all crucial factors in determining whether the mode is unstable. A detailed description of the ITG mode in toroidal geometry is beyond the scope of this thesis; more information can be found in [28]. One effect worth noting is the stabilizing influence of rotational velocity shear. The rotation velocity can be from externally driven torques, or internal $\mathbf{E} \times \mathbf{B}$ or diamagnetic effects. Basically, the shear helps to stabilize the transport by twisting and poloidally stretching the $\mathbf{E} \times \mathbf{B}$ eddies, thus decorrelating and shrinking them. Velocity shear is most stabilizing when the shearing rate is greater than or equal to the mode's nonlinear growth rate. Waltz [128] describes the physics behind rotation shear stabilization in more detail.

5.1.2 Details of the IFS-PPPL Model

The electron and ion diffusivities from the IFS-PPPL model are described as functions of local plasma parameters by an interpolation formula, fit to the results from runs of the linear and nonlinear codes conducted over a wide range of plasma parameter space [31]. Only first-principles theory is used in the preparation of these parameterizations; no free parameters that can be adjusted to fit experimental measurements are included, so they provide a rapid, unbiased means of testing the theory.

χ_i is parameterized as a function of the deuterium and carbon ITG mode χ 's as follows [126]:¹

$$\chi_i = 12 \max(\chi_i^{(1)}, \chi_i^{(2)}) \rho_i^2 v_{ti} / R, \quad (5.2)$$

$$\chi_i^{(1)} = \frac{(q/\tau_b)^{1.1}}{1 + \hat{s}^{0.84}} \left(1 + \frac{6.7\epsilon}{q\nu^{0.26}} \right) \mathcal{Z}(Z_{\text{eff}}^*) \mathcal{G}^{(1)}(R/L_T) \quad (5.3)$$

and

$$\chi_i^{(2)} = \frac{0.33\tau_b^{-0.8}}{1 + \hat{s}} \max[0.5, (Z_{\text{eff}}^* - 3)] \mathcal{G}^{(2)}(R/L_T). \quad (5.4)$$

Here, $\mathcal{G}^{(i)} \equiv G(R/L_T - R/L_{T_{\text{crit}}}^{(i)})$, $G(x) \equiv \min(x, x^{1/2})H(x)$, and $H(x)$ is a Heav-

¹The parameterizations shown were obtained from directly from Dorland [126].

inside function. The temperature ratio $\tau \equiv T_i/T_e$, $\tau_b \equiv \tau/(1 - \sigma_b)$, and $\mathcal{Z} = \min[1, (3/Z_{\text{eff}}^*)^{1.8}]$. The expression $Z_{\text{eff}}^* \equiv (n_D + 36n_C)/(n_D + 6n_C)$, where n_D is the thermal hydrogenic ion density, and only carbon impurities n_C have been considered. [High Z impurities tend to cause little dilution and can usually be ignored, while low- Z impurities such as helium are not well described by this parameterization.] The collisionality parameter is

$$\nu \equiv 2.1 R n_{e19}/(T_e^{1.5} T_i^{0.5}) \quad (5.5)$$

where R is given in meters and the temperatures are in keV. The magnetic shear $\hat{s} \equiv \frac{r}{q} \frac{dq}{dr}$. All other symbols are standard [116].

The critical gradient scale length of the deuterium mode is approximately

$$R/L_{T_{\text{crit}}}^{(1)} = f(\{p_j\}) g(\{p_j\}) h(\{p_j\}), \quad (5.6)$$

where

$$f = 1 - 0.2 Z_{\text{eff}}^{*0.5} \hat{s}^{-0.7} (14\epsilon^{1.3} \nu^{-0.2} - 1), \quad (5.7)$$

$$g = (0.7 + 0.6\hat{s} - 0.2R/L_n^*)^2 + 0.4 + 0.3R/L_n^* - 0.8\hat{s} + 0.2\hat{s}^2, \quad (5.8)$$

$$\text{and } h = 1.5 (1 + 2.8/q^2)^{0.26} Z_{\text{eff}}^{*0.7} \tau_b^{0.5}. \quad (5.9)$$

The parameter $R/L_n^* \equiv \min(6, R/L_n)$ and all density scale lengths are assumed to be equal. The critical gradient scale length of the carbon mode is approximately

$$R/L_{T_{\text{crit}}}^{(2)} = 0.75(1 + \tau_b)(1 + \hat{s}) \mathcal{D}(R/L_n^*) \mathcal{E}(Z_{\text{eff}}^*), \quad (5.10)$$

in which $\mathcal{D} \equiv \max(1, 3 - 0.67R/L_n^*)$ and $\mathcal{E} \equiv 1 + 6 \max(0, 2.9 - Z_{\text{eff}}^*)$.

The electron χ_e is obtained from the ratio of the quasilinear electron and ion heat fluxes found with the comprehensive linear code. For the deuterium mode,

$$\chi_e^{(1)}/\chi_i^{(1)} = 0.72\epsilon^* \nu^{0.14} (q/\hat{s})^{0.3} \tau^{0.4} \mathcal{J}_{1.5}(R/L_n), \quad (5.11)$$

where $\epsilon^* \equiv \max(0.17, \epsilon)$. For the carbon mode,

$$\chi_e^{(2)}/\chi_i^{(2)} = 0.26\tau\nu^{0.22} \mathcal{J}_2(R/L_n^*). \quad (5.12)$$

In these expressions, $\mathcal{J}_\xi \equiv \max[\xi, (1 + 0.3R/L_n^*)]$. Finally,

$$\chi_e = 12 \max(\chi_e^{(1)}, \chi_e^{(2)}) \rho_i^2 v_{ti}/R. \quad (5.13)$$

Certain effects are not represented in these models of χ_e and χ_i . The most important of these is the transport driven by trapped particle modes, which are sometimes observed to unstable in the edge regions of the plasma. Consequently, one boundary condition for the solution is the measured temperature at some radius $r_b < a$. (The

other boundary condition is $dT/dr = 0$ at $r = 0$.) The stabilizing effects of sheared $\mathbf{E} \times \mathbf{B}$ flows, both internally and externally driven, are also not included, nor are finite β effects or the electron version of the ITG mode, the ETG mode. Finally, sawteeth were not modeled, so that the predicted profiles may be too large compared to the measured time-averaged profiles.

The forms for χ_i and χ_e in (5.2) and (5.13) were obtained from fits to the results of many simulations conducted with the linear and nonlinear codes, and are considered valid only for the range of parameters over which the simulations were run. These are: $0.7 < q < 8$, $0.5 < \hat{s} < 2$, $0 < R/L_n < 6$, $0.5 < T_i/T_e < 4$, $1 < Z_{\text{eff}} < 4$, $0.1 < r/R < 0.3$, and $0.5 < \nu < 10$. This last parameter ν is defined as

$$\nu = 2.1 \frac{Rn}{T_e^{1.5} T_i^{0.5}}, \quad (5.14)$$

where n is measured in $\times 10^{19}/\text{m}^3$, R is in m, and the temperatures are in keV.

5.1.3 Marginality and C-Mod

The IFS-PPPL model finds that the ITG mode is unstable only when ∇T exceeds some critical value ∇T_{crit} . Below this value, the model assumes that the mode does not drive any transport, so the system is purely neoclassical. The transport resulting from this mode is therefore well described by the marginal stability picture described in § 2.3.2. As the following arguments will show, C-Mod is very close to the marginal stability threshold, making it a vital test bed for the model.

For $\nabla T > \nabla T_{\text{crit}}$, the IFS-PPPL χ_i has the form:

$$\chi_{\text{ITG}}(\nabla T) = \chi_B \rho^* \mathcal{F}(\nu^*, \dots) (L_T - L_{T,\text{crit}}). \quad (5.15)$$

The total power conducted by the ions across a given flux surface at $r = r_0$ by the ITG mode is

$$\begin{aligned} P_{\text{ITG}} &= \int dS n \chi_{\text{ITG}} \nabla T \sim 4\pi^2 a R n \chi_{\text{ITG}} \nabla T \\ &= P_{\text{in},i} \end{aligned} \quad (5.16)$$

where $P_{\text{in},i}$ is the net power input to the ions within r_0 . At the boundary point of the model, $P_{\text{in}0}$ and all the other parameters in (5.15)-(5.16) except ∇T are known,

so these equations can be rewritten to solve for $(L_T - L_{T,\text{crit}})$:

$$\begin{aligned}
 (L_T - L_{T,\text{crit}}) &= \frac{P_{\text{in}}}{4\pi^2 a R n \chi_B \rho^* \mathcal{F}(\dots) \nabla T} \\
 &= P_{\text{in},i} \frac{B^2}{n T^{5/2}} \mathcal{G}(\dots) \\
 &\equiv \frac{P_{\text{in},i}}{P_{\text{gB}}} \mathcal{G}(\dots)
 \end{aligned} \tag{5.17}$$

where $\chi_B \sim T/B$, $\rho^* \sim T^{1/2}/aB$, $\nabla T \sim T/a$, and \mathcal{G} is a function of dimensionless parameters. The factor $P_{\text{gB}} \equiv B^2/nT^{5/2}$ arises from any gyroBohm χ . The details of the specific model are encapsulated in the function \mathcal{G} . For C-Mod, the ratio $P_{\text{in},i}/P_{\text{gB}} \ll 1$, so $L_T \simeq L_{T,\text{crit}}$.

Figure 5.2 plots P_{ITG} normalized to $P_{\text{in},i}$ versus $L_{T,\text{crit}}/L_T$ for two representative C-Mod plasmas, plasmas from DIII-D and TFTR, and for a plasma expected on ITER. The specific values used in calculating P_{ITG} for each are listed in table 5.1; note that they are taken at $r = 0.8a$.

Parameter		C-Mod L-Mode	C-Mod H-Mode	DIII-D H-Mode	TFTR L-Mode	ITER
n	$\times 10^{20}/\text{m}^3$	1.2	2.2	0.2	0.2	1.3
T_e	keV	2.0	1.4	0.7	1.3	5.0
T_i	keV	2.0	1.4	0.8	1.3	5.0
B	T	5.3	5.3	0.9	4.8	5.7
Z_{eff}		1.7	2.5	1.7	2.6	1.5
q		2.7	2.3	2.8	2.5	2.0
\hat{s}		2.8	2.5	1.0	1.7	2.7
R	m	0.67	0.67	1.7	2.5	8.1
a	m	0.22	0.22	0.5	0.8	2.8
L_n/a		1.0	1.0	1.9	0.4	12
$P_{\text{in},i}$	MW	0.8	2.0	2.0	6.1	180
Shot		960301009	960116024	82788	45950	1001

Table 5.1: Parameters at $r/a = 0.8$ for comparison of $P_{\text{in}}/P_{\text{ITG}}$ for Alcator C-Mod, DIII-D, TFTR, and ITER. Data for tokamaks other than C-Mod taken from the ITER profile database; $P_{\text{in},i}$ for these tokamaks was estimated as half the total input power.

As evident in figure 5.2, the IFS-PPPL model has C-Mod very close to its critical gradient scale length $L_{T,\text{crit}}$ in both L- and H-Mode, and is thus marginally stable. Since the IFS-PPPL model is considered by Beer, Dorland, et al. to be most suscep-

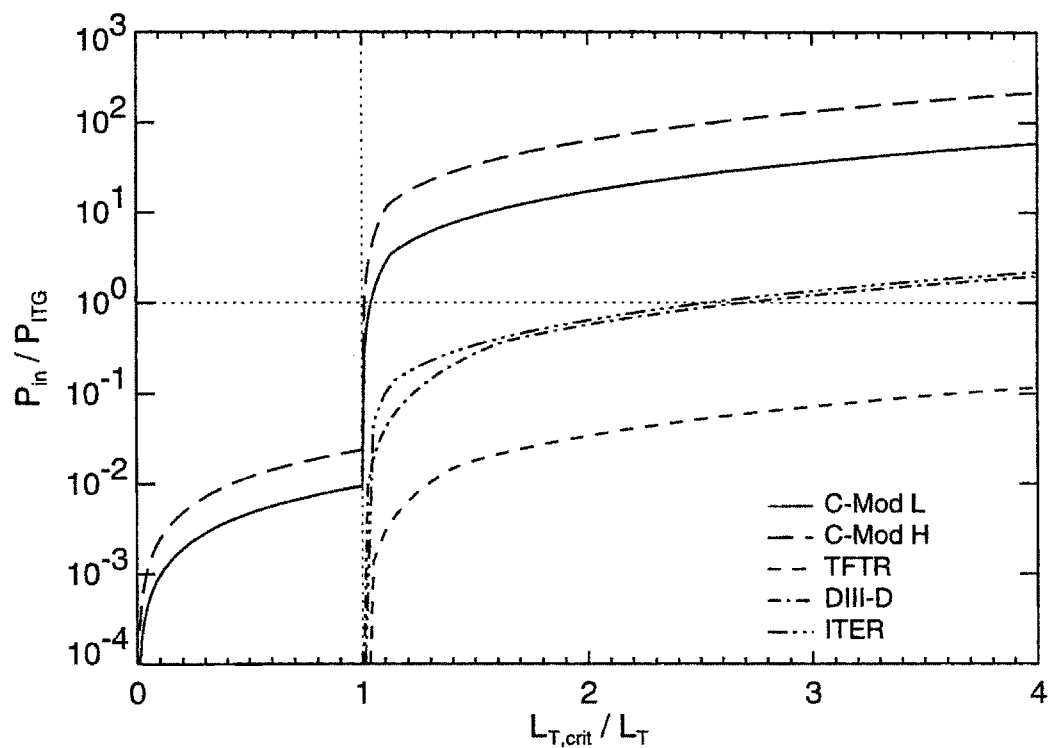


Figure 5.2: Proximity to marginal stability threshold for C-Mod, DIII-D, TFTR, and ITER. P_{ITG} is calculated using (5.17) and the values in table 5.1, and is normalized to $P_{in,i}$. The value of $L_{T,crit}/L_T$ expected for a particular tokamak is where $P_{ITG} = P_{in,i}$. C-Mod has $L_T \approx L_{T,crit}$, and therefore represents a better comparison for ITER than DIII-D or TFTR.

tible to error for plasmas near the marginal stability threshold, testing it on C-Mod is particularly important, as it is much closer to marginality than the other tokamaks so far studied.

5.2 Methods

This section describes the code `nt`, used to calculate the temperature profiles based on the IFS-PPPL model, and the methods used to compare these profile to experimental measurements.

5.2.1 Transport Simulation Using the IFS-PPPL Model

The IFS-PPPL model was compared to C-Mod data by using the parameterizations of χ_i and χ_e (5.2)-(5.13) to solve the electron and ion energy balance equations for the temperature profiles predicted by the model. These equations are modified forms of (2.4), with $\mathbf{q}_j = n_j \chi_j \nabla T_j$:

$$\nabla \cdot \mathbf{q}_e - Q_{ei} - Q_{\text{conv}e} - \frac{\partial W_e}{\partial t} + Q_{\text{RF}e} + Q_{\text{OH}} - Q_{\text{rad}} \quad (5.18)$$

$$\nabla \cdot \mathbf{q}_i = Q_{ei} - Q_{\text{conv}i} - \frac{\partial W_i}{\partial t} + Q_{\text{RF}i} \quad (5.19)$$

The code `nt` written by Dorland et al. was used to calculate the solutions for T_i and T_e , self-consistently with $\nabla \cdot \mathbf{q}_j$ and its constituent parameters $\tau = T_i/T_e$ and L_T , as well as Q_{ei} and $Q_{\text{conv}j}$. Plasma measurements input to `nt` were the density, Z_{eff} , R , q , ϵ , and Q_{rad} . TRANSP calculations of $Q_{\text{RF}j}$ and Q_{OH} were input as well, so these were not necessarily consistent with the resulting temperature profiles. The runs conducted for this thesis were at one fixed time, so the time derivatives of the stored energy densities at that time were input to `nt` as well. `nt` contains an ad hoc model of the stabilizing effects of toroidal rotation shear; this was tested for one H-Mode shot for which rotation measurements were possible.

A FORTRAN namelist is used to control the various models within `nt`, and to provide the numeric parameters for the solution of the energy balance equations. There are three parameters of interest that need to be determined: the boundary condition for the temperature profile solution r_b , the time step dt over which to evolve the solution when iterating, and the number of time steps n_s over which to iterate. The optimal values found for each of these parameters are presented as part of § 5.3.

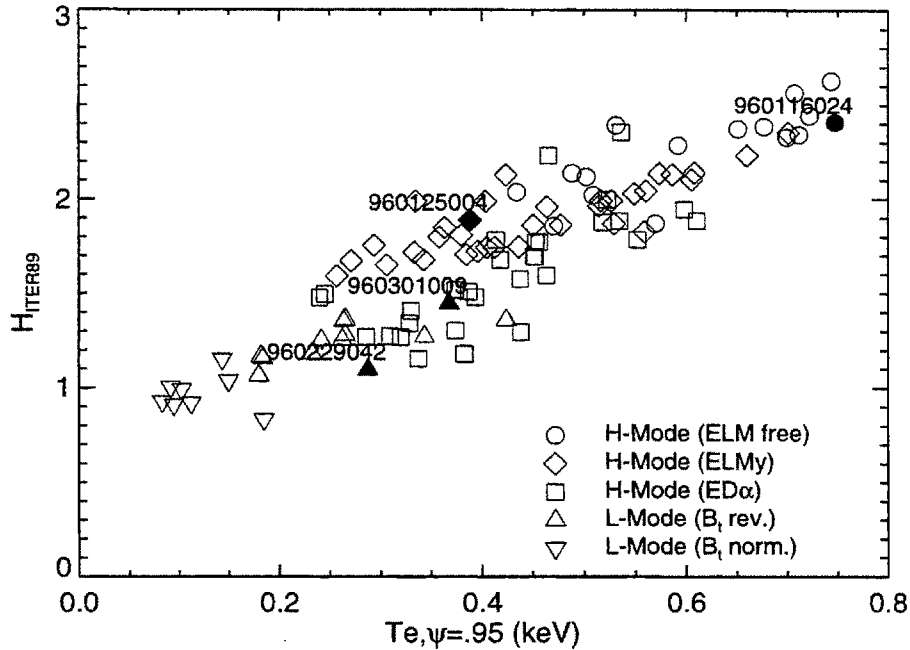


Figure 5.3: H-factor vs. edge T_e for shots used in comparison

5.2.2 Comparison to Experiment

Four C-Mod shots were chosen for analysis: two reversed-field L-Modes (960229042 and 960301009) and two H-Modes, one ELMy (960125004) and one ELM-free (960116024). All four shots were selected based on their being post-boronization, and having good edge T_e and some Hirex T_i measurements.

Figure 5.3 plots the H-factor ($H = \tau_E / \tau_{E,ITER89-P}$) versus the temperature at the 95% flux surface for the four shots selected and the 97 others in the core table of *LOGBOOK* that meet these criteria.² As evident from this plot, the H-factor shows a strong dependence on the edge temperature [11]. The selected shots span the range from $H=1.1$ to $H=2.4$. Table 5.2 summarizes their parameters.

TRANSP runs of each of these shots were conducted to prepare the inputs for nt. Table 4.4 in chapter 4 lists the diagnostics used for this analysis. All shots were at 5.3 T so there were no problems measuring T_e or f_{PRF} . Hirex measurements were available for at least two spatial locations for each shot. For the three shots with $T_{e0} \gtrsim 4$ keV, the Hirex measurements were corrected for the fact that the peak emission from He-like argon is not at the innermost approach of the chord to the magnetic axis, which is nominally chosen as the spatial location corresponding to the measurement. The correction algorithm was basically to determine the temperature at this innermost approach position that would be necessary to duplicate the spectrum

²The complete SQL query for selection was “(gradte_gpc is not null) and (shot between 960000000 and 960399999)”.

Shot	Mode	H-factor	$T_{e\psi=95}$ (keV)	T_{e0} (keV)	I_p (MA)	P_{RF} (MW)	$\langle n_e \rangle$ $\times 10^{20}/\text{m}^3$
960116024	H	2.4	0.7	3.6	1.0	2.3	2.4
960125004	H	1.9	0.4	3.1	1.0	2.6	2.9
960229042	L	1.1	0.3	2.9	1.1	2.2	1.8
960301009	L	1.4	0.4	3.4	0.8	2.8	1.2

Table 5.2: Shots for IFS-PPPL model comparison

observed along the chord [72]. In all cases, quantitative comparisons to the IFS-PPPL predicted profiles were conducted using the T_i profile estimated from the T_e profile and the T_{i0} measurement from the neutrons by the method described in § 3.6.3. These “estimated” profiles agreed well with the Hirex measurements for all shots.

Because sawteeth were not modelled, the two profiles were compared for $r \geq 5$ cm. Since the predicted profile is set equal to the measured profile at $r = r_b$, only radii out to $r = r_b - 2$ cm were considered. Over the range $r = 5$ cm to $r = r_b - 2 \simeq 15$ cm, a quantitative comparison was made by calculating the RMS difference between the two profiles, expressed as a percentage:

$$\Delta_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{r=5\text{cm}}^{r_b-2\text{cm}} \frac{(T_{r,\text{predict}} - T_{r,\text{measure}})^2}{T_{r,\text{measure}}^2}} \quad (5.20)$$

where N is the number of radial points in the predicted T_i grid between $r = 5$ cm and $r = r_b - 2$ cm.

The sensitivity of the predictions to the density, Z_{eff} , edge temperature, and the ICRF power deposition profiles was determined as these all have large uncertainties. Since the IFS-PPPL parameterizations for χ_e , χ_i , and ∇T_{crit} (5.2)-(5.13) depend on q and the shear, both of which are not measured on C-Mod, the sensitivity of the predictions to these parameters was also investigated.

It is important to note that C-Mod plasmas do not strictly lie within the range of validity of the parameterizations (5.2)-(5.13) as C-Mod’s value of ν , defined by (5.5), is larger than the maximum of $\nu_{\text{max}} = 10$. Table 5.3 lists the values of ν for the four shots measured at $r/a = 0.8$. Despite the fact that the parameterizations are not strictly valid for C-Mod, they are used anyway in the subsequent sections to test the model, as their dependence on ν is expected to be weak [129]. Beer, Dorland, et al. are currently working to extend the range of validity to higher values of ν .

Shot	ν
960116024	16
960125004	63
960229042	60
960301009	23

Table 5.3: Values of $\nu \equiv 2.1RnT_e^{-1.5}T_i^{0.5}$ for C-Mod plasmas. The IFS-PPPL model parameterizations in (5.2)-(5.13) are valid only for $0.5 < \nu < 10$.

5.3 Results

This section presents the results of the comparison of the IFS-PPPL model to C-Mod measurements. A brief presentation on the selection of the optimal numeric control parameters for *nt* is given first in § 5.3.1. The results from each of the four shots are then presented in § 5.3.2. For two of these shots, the T_e prediction was particularly poor; an attempt to improve the prediction using an ad hoc sawtooth model is examined in § 5.3.4. The Z_{eff} profile input to *nt* for the runs shown in § 5.3.2 was flat; the effects of a hollow or peaked Z_{eff} profile on the predictions is investigated in § 5.3.5. A test of *nt*'s ad hoc rotational velocity shear stabilization is presented in § 5.3.6.

5.3.1 Optimal Input Parameters

As mentioned in § 5.2, optimal values for three numeric control parameters needed to be determined: the boundary condition r_b , the time step dt , and the number of steps over which to evolve the solution n_s . Examination of the optimal r_b is deferred until after the results from the individual shots are examined. For these initial runs, the value used was $r_b = 0.8a \simeq 17$ cm. The other two parameters are discussed here.

The solution of the energy balance equations for the temperature profiles was obtained at a single time slice. An iterative method was used to solve these equations, so some evolution in time was necessary to allow the solutions to reach steady state. The parameter dt controls the step size of one iteration in time. If dt is too large, the resulting self-consistent χ profiles will be unphysically jagged from numerical effects. The more marginal a device, the smaller dt must be. Figure 5.4(a) plots the predicted χ_i profiles for two values of dt : 0.0005 s and 0.00005 s. Clearly the latter value is more optimal for C-Mod. Note that this is about three orders of magnitude smaller than the parameter used in TFTR simulations, which is reasonable since the arguments in § 5.1.3 indicate that C-Mod should be this much closer to marginality.

The parameter n_s should be set to the number of steps necessary to reach steady state. This was evaluated by plotting the predicted T_{i0} as a function of the number

of steps³ for each shot as shown in figure 5.4(b). Shot 960116024 required $n_s = 5000$ to reach steady state, while the other shots were run with $n_s = 1000$.

5.3.2 Examination of Individual Shots

Shot 960301009 - L-Mode

Figure 5.5 shows four plots used to examine the prediction of the IFS-PPPL model. Plot (a) shows the measured and predicted temperature profiles, and the estimated T_i profile used in the TRANSP run. The most obvious point to observe is that the predicted T_e is much higher than the measured T_e for $r < 10$ cm. The RMS difference between the profiles, using (5.20), is $\Delta_{\text{RMS}} = 31\%$. Referring to figure 5.5(b), the χ_e profile is much lower than the χ_i profile over the entire range, which is contrary to the observations from TRANSP that suggest $\chi_e \simeq \chi_i$. The IFS-PPPL model is likely underpredicting the electron transport, so that its T_e prediction is too high.

The T_i profile is also not well predicted on this shot. The estimated T_i profile appears to be a reasonable approximation for T_i . Based on the difference between the Hirex and neutron measurements at the center, the uncertainty in the estimated profile is $\sim \pm 20\%$. The RMS difference between the estimated and predicted T_i profiles is 30%, greater than this uncertainty. Note that with T_e poorly predicted, the parameter $\tau = T_i/T_e$ used in (5.2)-(5.13) to evaluate χ_i and χ_e is erroneous. However, for this shot, the T_i prediction did not noticeably change when the simulation was rerun with T_e set to be the measured profile.

Figure 5.5(c) plots the ratio of the critical gradient scale length $L_{T,\text{crit}}$ to the profile scale length from the predicted and the estimated T_i profiles. According to the IFS-PPPL model, the measured ∇T is up to $2.5 \times \nabla T_{\text{crit}}$, which is not expected based on the arguments in § 5.1.3. Averaging the profiles over the range $r \in (5, 15)$ cm, the estimated profile has $\langle L_{T,\text{crit}}/L_T \rangle = 2.1$ while the predicted profile has $\langle L_{T,\text{crit}}/L_T \rangle = 1.1$.

A firm conclusion on the success of the model must be based on the variations in the prediction from uncertainties in the input data, and in the temperature profile to which it is being compared. These are addressed in figure 5.6. Plot (a) shows the measurements for T_e and T_i , and the estimated T_i profile. The uncertainty in the estimated T_i profile, $\pm 20\%$, is also shown with the horizontal shading. The dashed line shows the prediction for T_i from the IFS-PPPL model using the nominal inputs. The vertical shaded area shows the range in this prediction from a $\pm 30\%$ uncertainty in Z_{eff} . The model is unambiguously wrong within this range for $r < 11$ cm.

A similar plot for a $\pm 50\%$ variation in the ICRF input power is shown in figure 5.6(b). The actual uncertainty in the RF power deposition is at most $\pm 20\%$. A variation of $\pm 50\%$ was chosen to emphasize the fact that since the profiles are at

³There was no special reason why T_{i0} was chosen as against any of the other predicted scalar parameters.

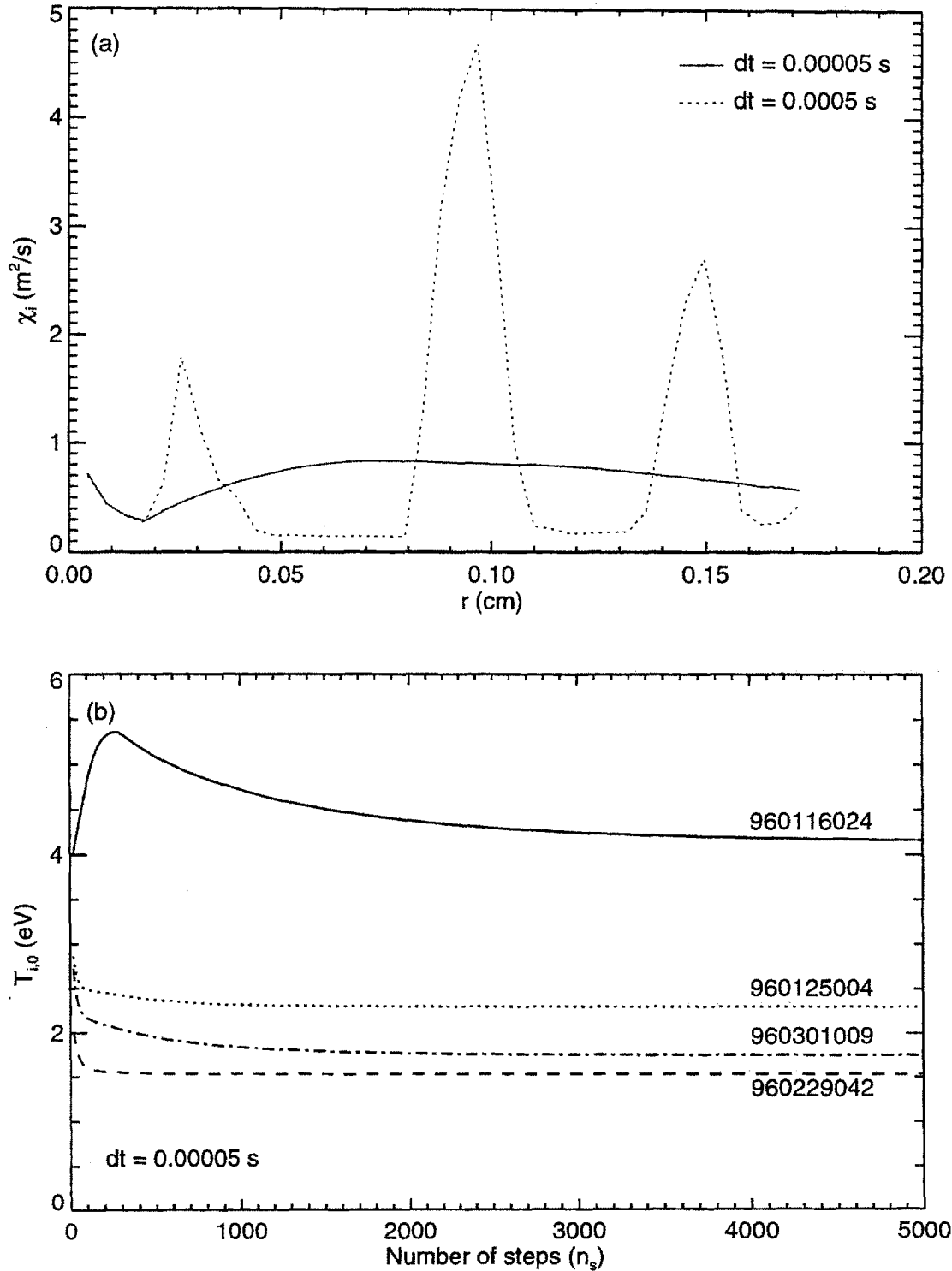


Figure 5.4: Determining optimal dt and n_s for simulations

marginal stability, as indicated by figure 5.5(c), they are insensitive to the input power over a wide range. To see a noticeable difference in the T_i prediction, the input power would need to be $\pm 90\%$.

Figure 5.6(c) plots the predictions for T_i (thick dashed and dotted lines) using the two q profiles shown (thin lines). The q profile from TRANSP, steeper than the one from EFIT, has a greater shear $s = \frac{r}{q} \frac{dq}{dr}$. Increasing shear helps to stabilize the transport, so the T_i prediction using the TRANSP q profile is slightly higher. Neither q profile is necessarily more accurate; the real profile probably lies somewhere between the two, so the predicted T_i using the real profile would be slightly lower.

Finally, the sensitivity to the edge temperature used as the boundary condition for the prediction is shown in figure 5.6. Since the gradient scale length L_T of the predicted profile is fixed at $L_{T,\text{crit}}$, adjusting the edge temperature does adjust the slope ∇T_{crit} , since for two edge temperatures T_1 and T_2

$$L_T = L_{T,\text{crit}} \Rightarrow \frac{T_1}{\nabla T_1} = \frac{T_2}{\nabla T_2} \Rightarrow \nabla T_2 = \frac{T_2}{T_1} \nabla T_1. \quad (5.21)$$

The estimated uncertainty in the edge temperature, $\pm 20\%$, is not enough to change ∇T_{crit} at the boundary to the value observed in the estimated profile. Since the latter is $L_T \simeq 2.5L_{T,\text{crit}}$ at the boundary location ($r_b = 17$ cm), the edge temperature would need to be increased by a factor of 2.5 for the edge gradient of the predicted profile to match the estimated profile.

Shot 960229042 - L-Mode

Figure 5.7(a) shows that the predicted T_i profile is again systematically lower than the estimated profile, with an RMS deviation $\Delta_{\text{RMS}} = 25\%$. The uncertainty in the estimated profile is $\pm 10\%$, lower than normally used because of the close agreement with the many Hirex measurements. The estimated profile has $\langle L_{T,\text{crit}}/L_T \rangle = 2.1$ over the range $r \in (5, 15)$ cm, while the predicted profile has $\langle L_{T,\text{crit}}/L_T \rangle = 1.3$. The sensitivity of the prediction to the measurement uncertainties is presented in figure 5.8. As seen for shot 960301009, the predicted T_i is systematically too low even with a 30% uncertainty in Z_{eff} . The uncertainty in P_{RF} does not affect this conclusion, which is important for this shot as the SPRUCE singularity problem was significant for this shot (see § 3.5.2). The q profile from EFIT is shallower, so as for shot 960301009, the T_i profile predicted using the actual q profile would be slightly lower than that obtained from the TRANSP q profile. For this shot, a 20% uncertainty in the edge T_i barely overlaps the prediction with the estimated profile for $r > 5$ cm, although the slope of the predicted profile is still too shallow.

For this shot, The T_e prediction agrees well with the measured profile for $r \in (5, 17)$ cm, with $\Delta_{\text{RMS}} = 7.5\%$. The shape of the profile suggests that the temperature on axis is diverging. Turning to figure 5.7(c), this is because the χ_e value on axis $\rightarrow 0$.

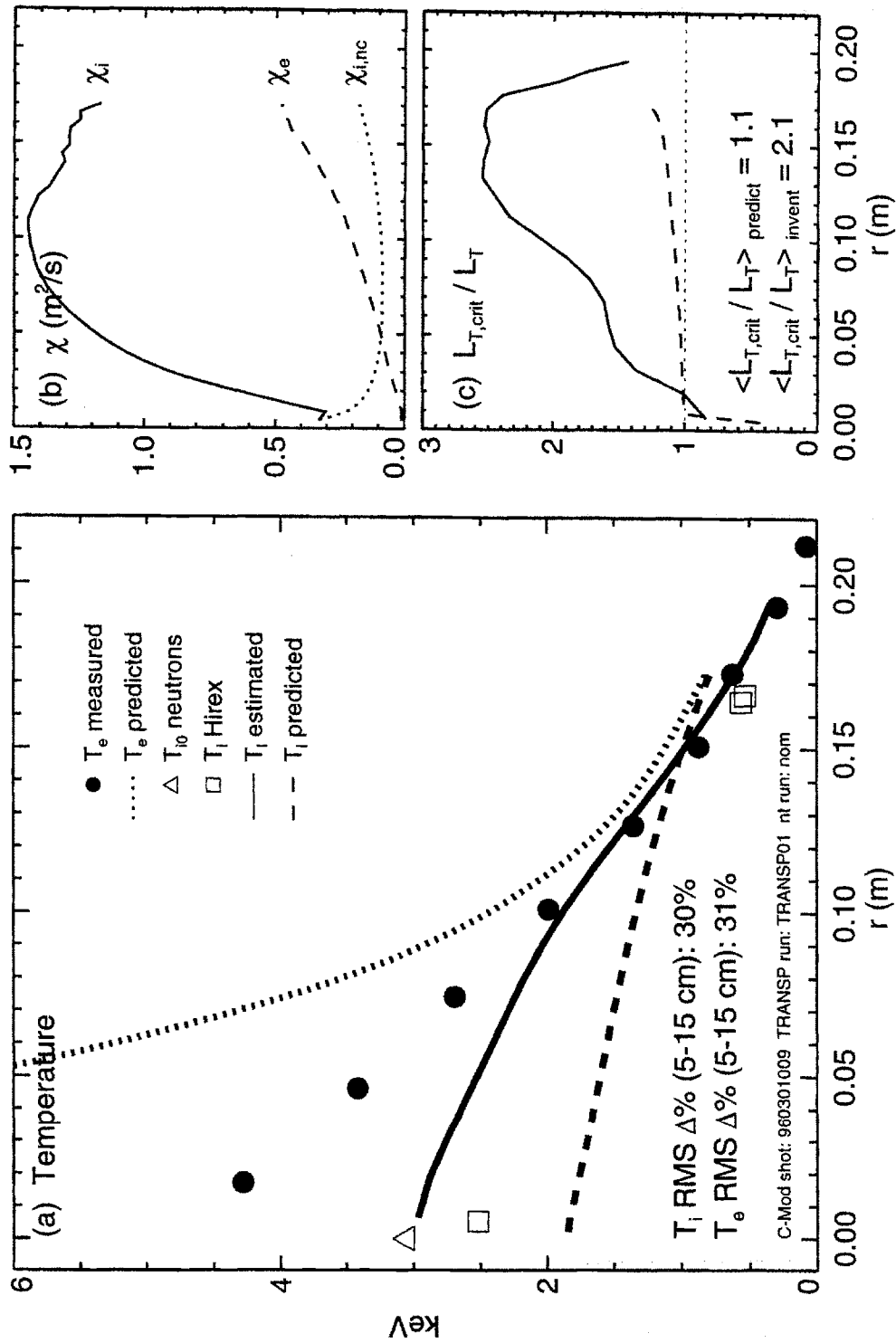


Figure 5.5: Comparison of IFS-PPPL model for shot 960301009

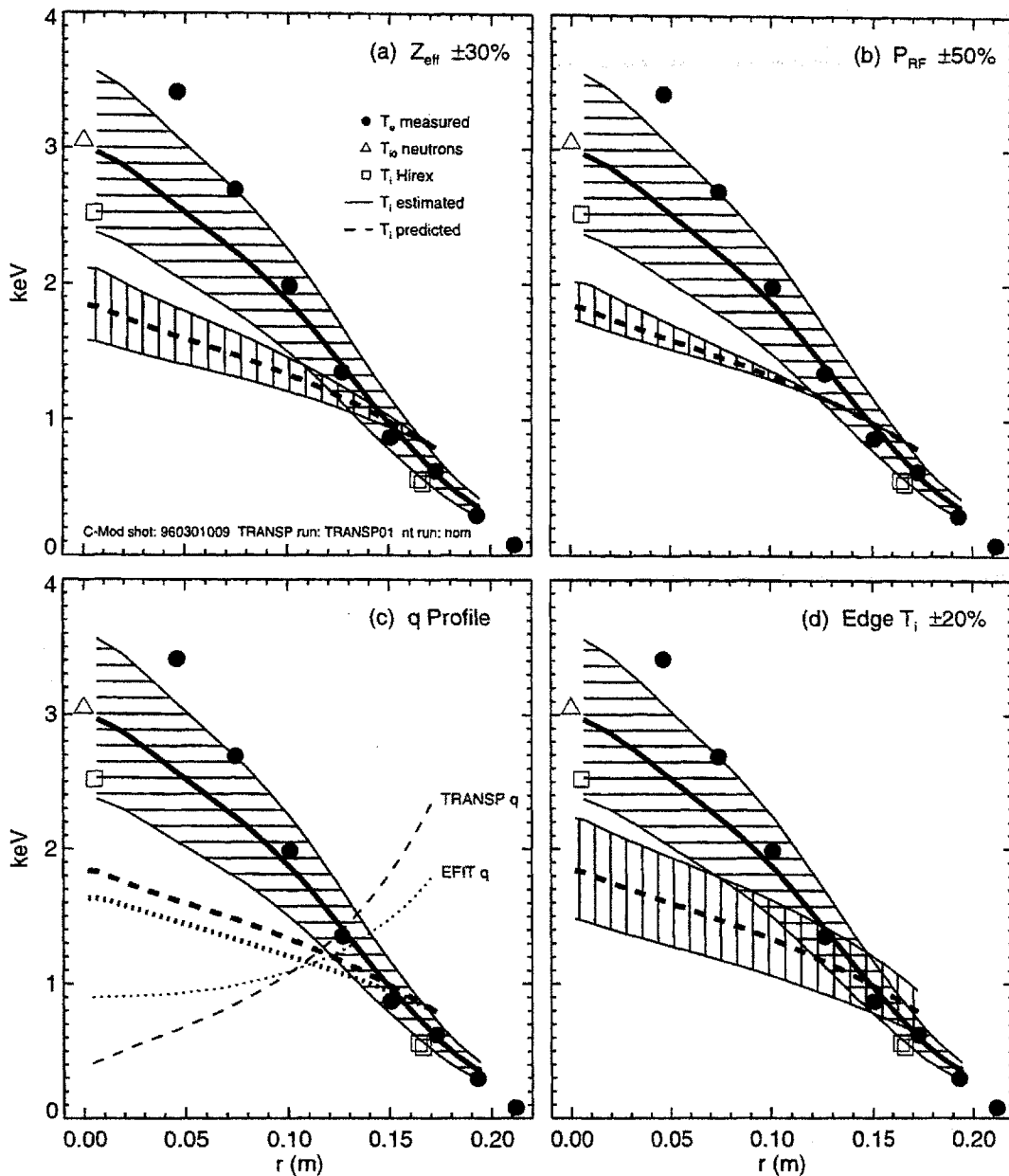


Figure 5.6: Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960301009. The vertical shading around the predicted profile shows the variation in the prediction resulting from a change in one of the parameters on which it depends. The parameters examined are Z_{eff} ($\pm 30\%$), the RF input power ($\pm 50\%$ — larger than the actual uncertainty to illustrate the marginality of the profile), the q profile (TRANSP vs. EFIT), and the edge T_i boundary condition.

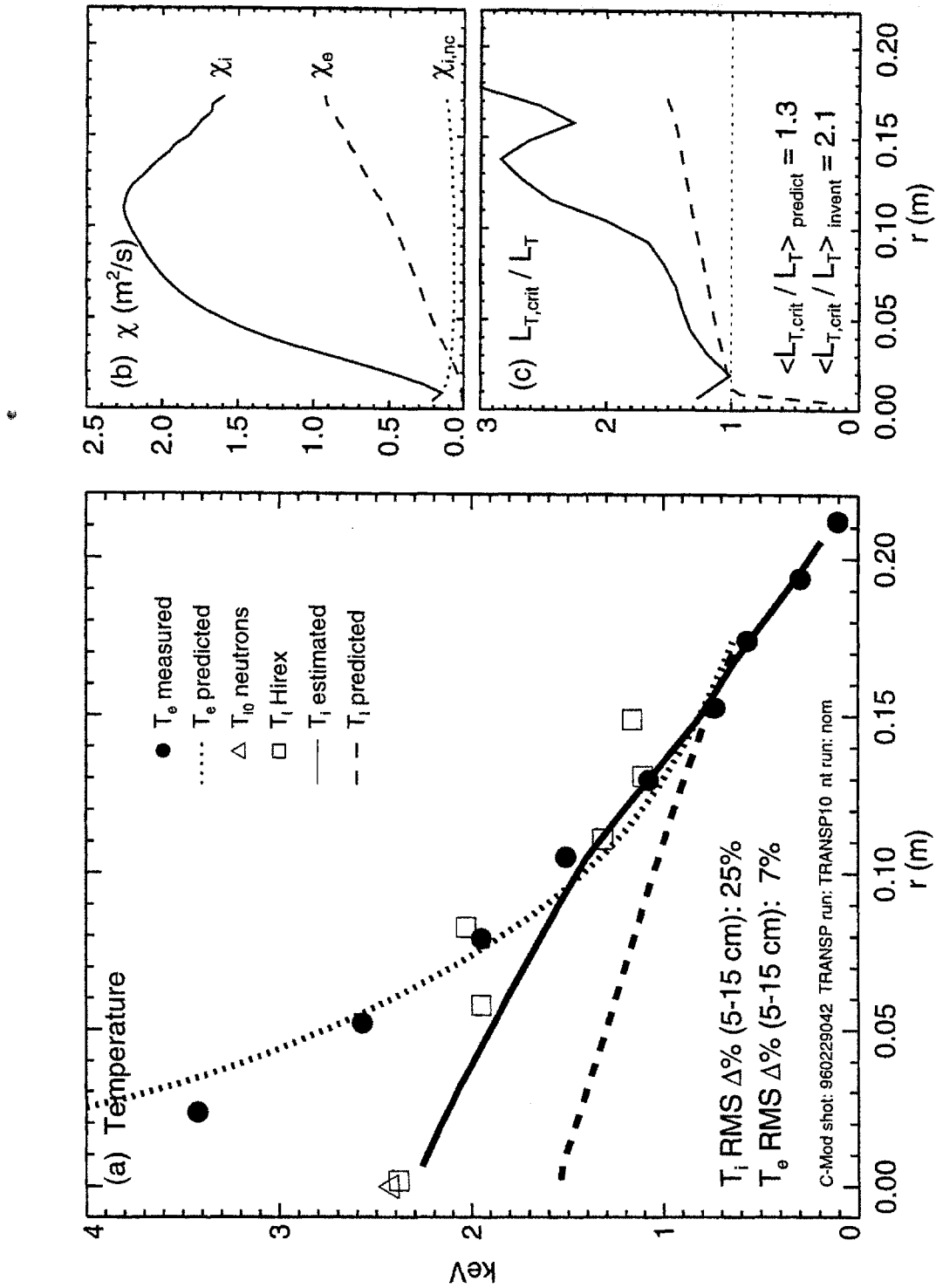


Figure 5.7: Comparison of IFS-PPPL Model for Shot 960229042

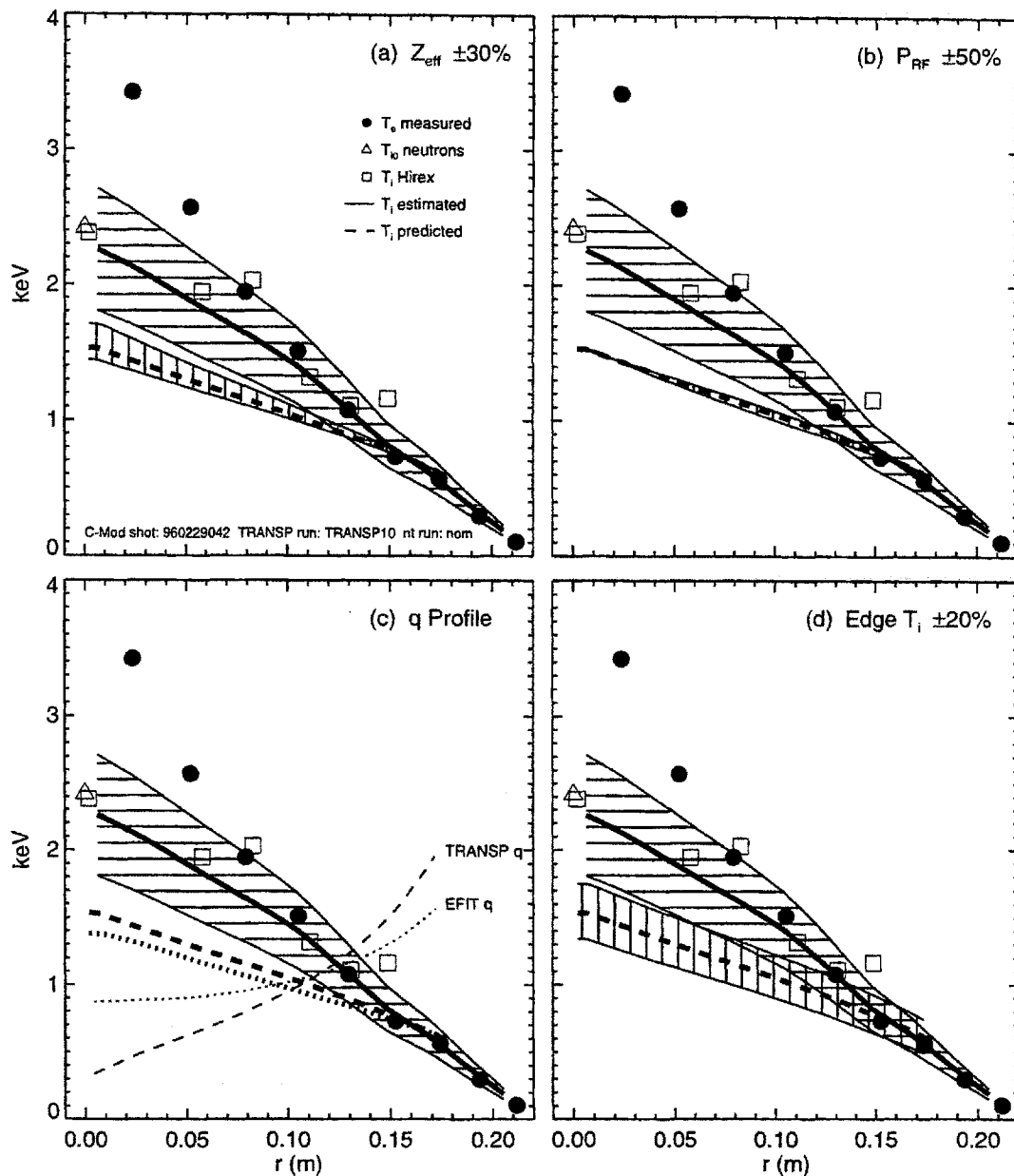


Figure 5.8: Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960229042. The vertical shading around the predicted profile shows the variation in the prediction resulting from a change in one of the parameters on which it depends. The parameters examined are Z_{eff} ($\pm 30\%$), the RF input power ($\pm 50\%$ — larger than the actual uncertainty to illustrate the marginality of the profile), the q profile (TRANSP vs. EFIT), and the edge T_i boundary condition.

Shot 960125004 - ELMy H-Mode

The difference between the predicted and estimated T_i profiles is slightly lower for this shot, with $\Delta_{\text{RMS}} = 17\%$ over the range $r \in (5, 15)$ cm, which is just within the uncertainty of the estimated profile ($\pm 20\%$). The estimated T_i profile scale length relative to $L_{T,\text{crit}}$ is shorter than the ratio $L_{T,\text{crit}}/L_T$ of the predicted profile for $r > 5$ cm, although the averages of the two over $r \in (5, 15)$ cm are closer for this shot than for the L-Mode shots already shown: $\langle L_{T,\text{crit}}/L_T \rangle = 1.5$ for the estimated profile and $\langle L_{T,\text{crit}}/L_T \rangle = 1.1$ for the predicted profile.

When factoring in the uncertainty in the predicted profile, shown in figure 5.9, the IFS-PPPL model does better. The predicted and estimated profile uncertainty ranges overlap considerably for a 30% error in Z_{eff} , and a 20% error in the edge T_i . Once again the error in P_{RF} is not important to the prediction because T_i is marginally stable, so the problems with the RF power deposition profile are not important here. The anticipated downward shift from using the actual q profile, which lies somewhere between the TRANSP and EFIT profiles, is not enough to weaken the success of the model.

The T_e prediction works well for this shot as well, with $\Delta_{\text{RMS}} = 8\%$ for $r \in (5, 17)$ cm.

Shot 960116024 - ELM free H-Mode

The ion temperature prediction from the IFS-PPPL model is most successful for shot 960116024, as shown in figure 5.11. This time, the predicted T_i is slightly higher than the estimated profile. The RMS difference between them over the range $r \in (5, 15)$ is only 2%, well within the uncertainties of both, as shown in figure 5.12. Though the details of the profile shape are different for the two shots, as shown by the plots of $L_{T,\text{crit}}/L_T$ in figure 5.11(c), the average values of $L_{T,\text{crit}}/L_T$ over the aforementioned range are identical at 1.1. Because of the close agreement of the two profiles, not much is learned by studying the sensitivity of the prediction to uncertainties in the measured parameters in figure 5.12. Plot (c) that the model does well even using the much shallower q profile from EFIT. Shot 960116024 also encountered a significant problem with the singularity in SPRUCE (see § 3.5.2), though this should not be significant since the profile is marginal. Note that $\chi_i \rightarrow \chi_{i,\text{nc}}$ as $r \rightarrow 0$. This is because ∇T drops below ∇T_{crit} thus stabilizing the ITG mode, so that only neoclassical transport is present.

However, the T_e prediction for this shot is not good, with $\Delta_{\text{RMS}} = 75\%$ over $r \in (5, 15)$ cm. Since T_e was poorly predicted, the value of T_i/T_e used in the parameterizations was erroneous. Unlike for shot 960301009, when the simulation was rerun with T_e set to be the measured profile, the T_i profile prediction noticeably changed, as shown in figure 5.13. Using the measured T_e , the RMS deviation of the T_i prediction was $\Delta_{\text{RMS}} = 10\%$, still within the uncertainty of the estimated profile.

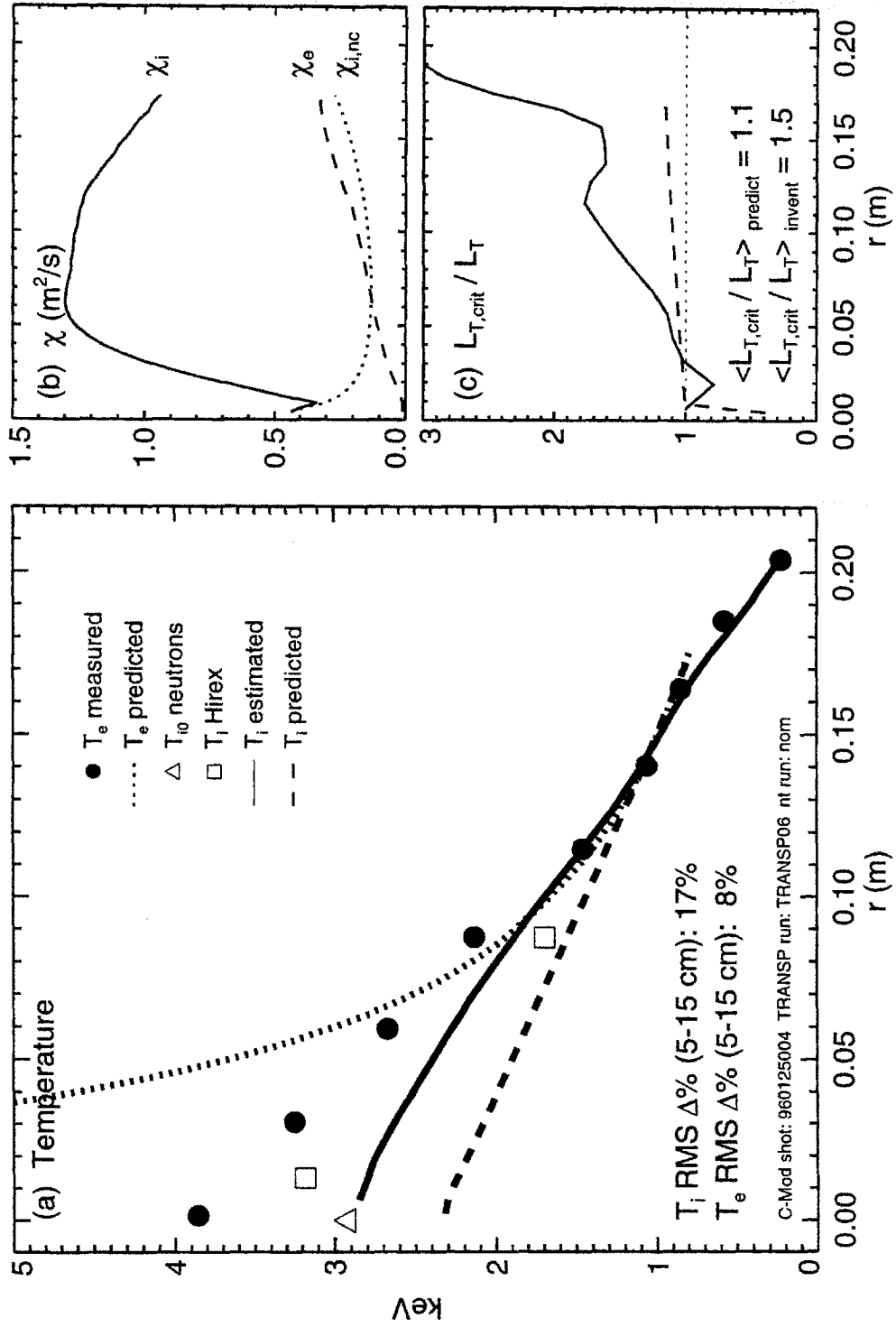


Figure 5.9: Comparison of IFS-PPPL Model for Shot 960125004

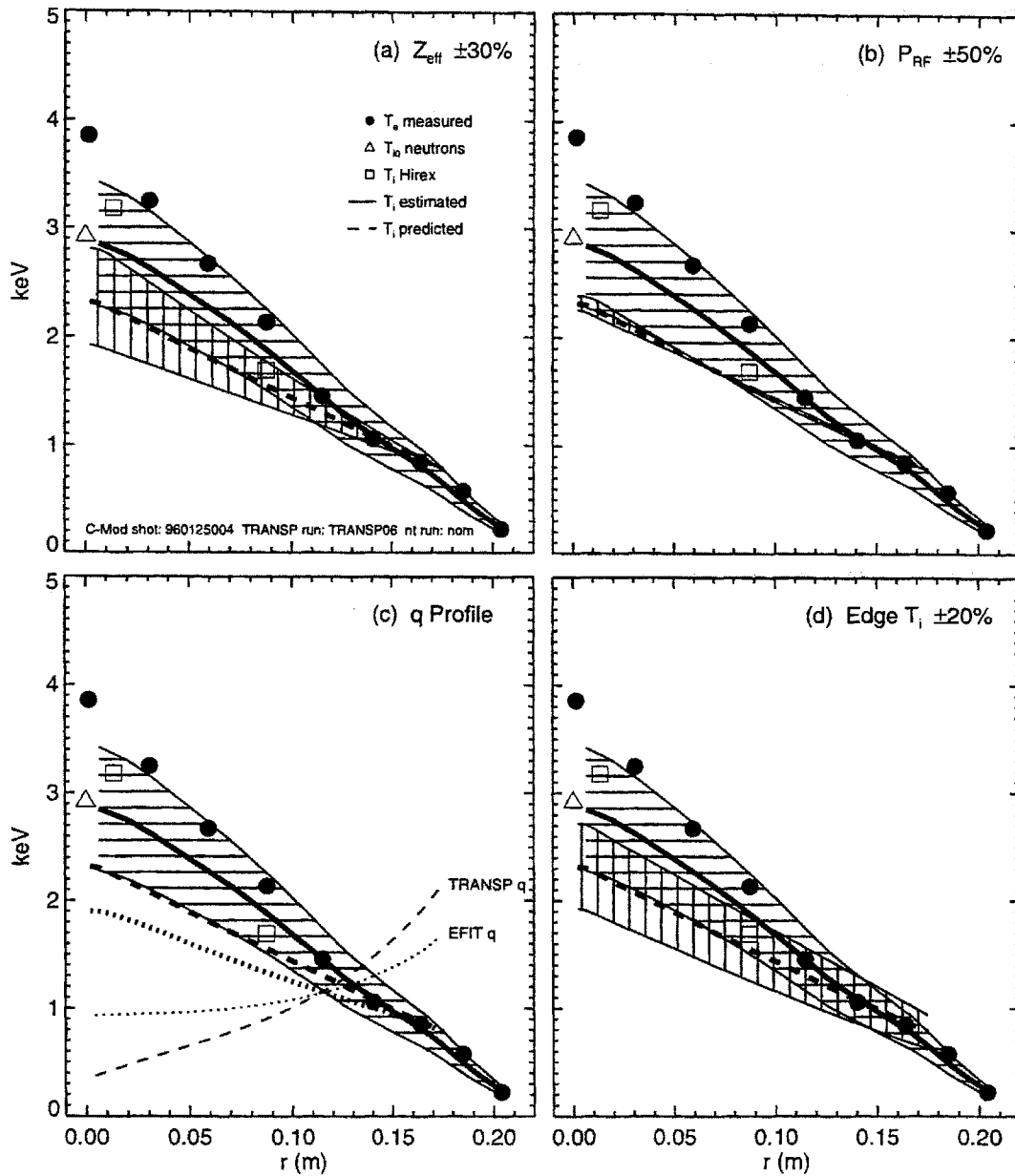


Figure 5.10: Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960125004. The vertical shading around the predicted profile shows the variation in the prediction resulting from a change in one of the parameters on which it depends. The parameters examined are Z_{eff} ($\pm 30\%$), the RF input power ($\pm 50\%$ — larger than the actual uncertainty to illustrate the marginality of the profile), the q profile (TRANSP vs. EFIT), and the edge T_i boundary condition.

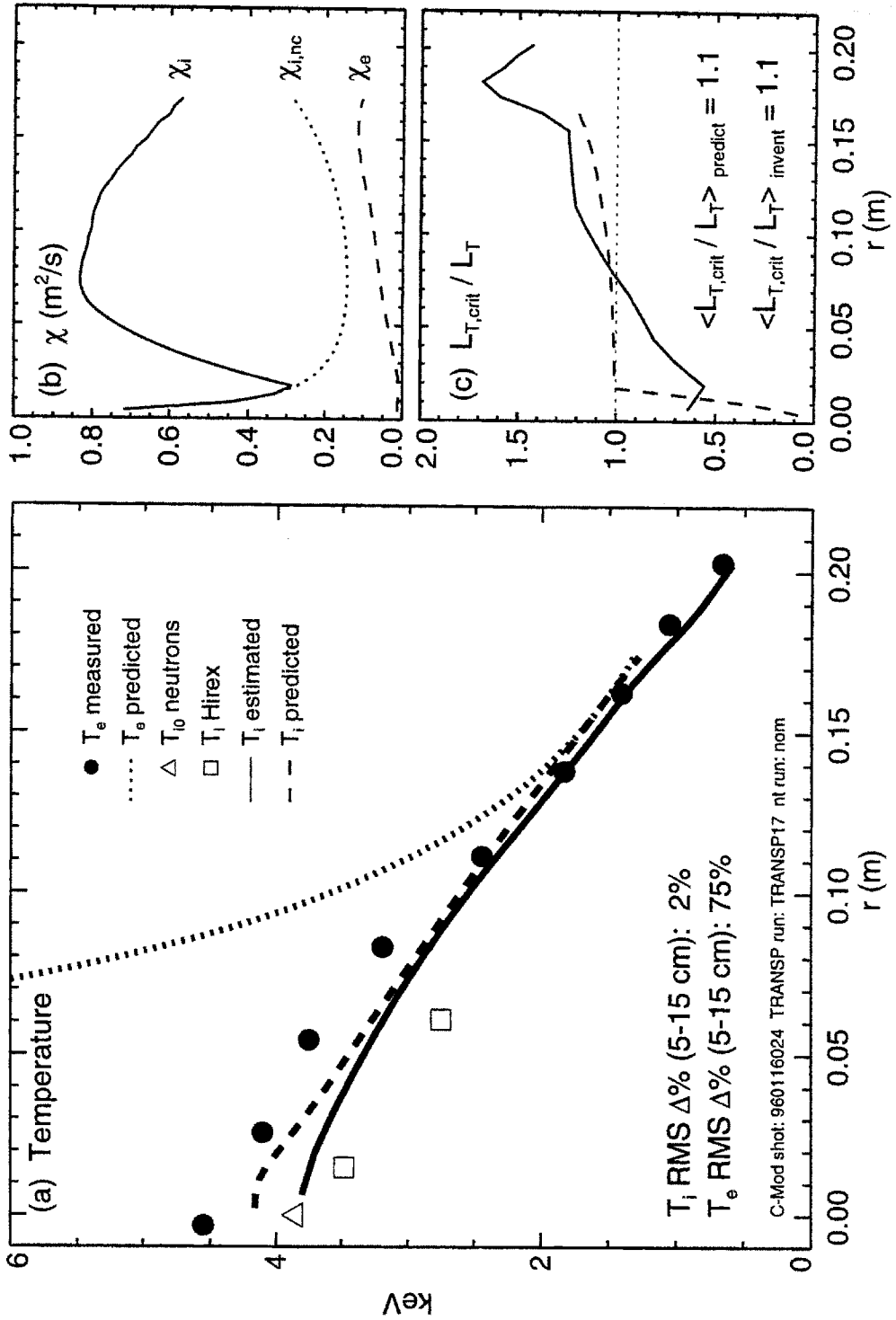


Figure 5.11: Comparison of IFS-PPPL Model for Shot 960116024

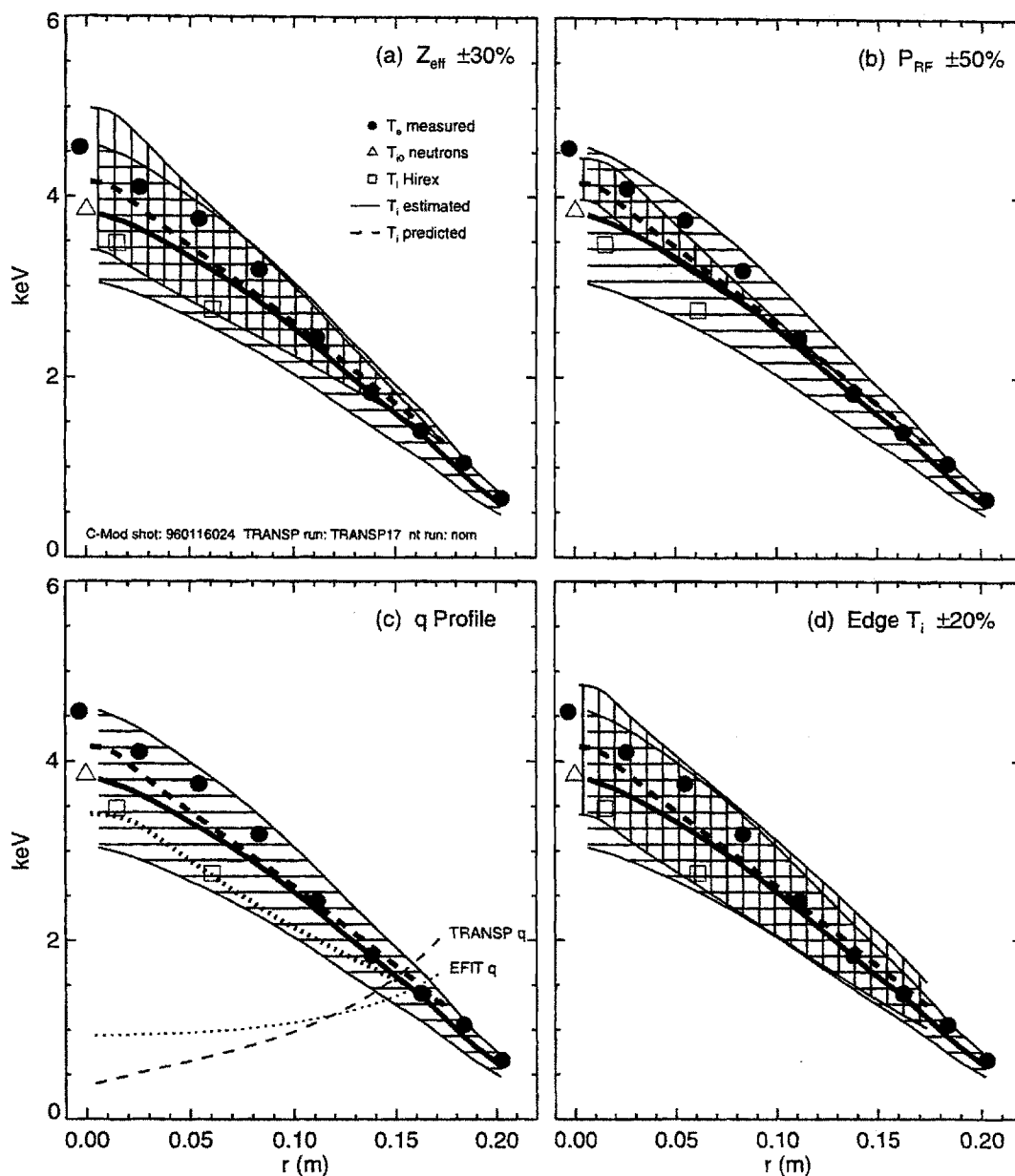


Figure 5.12: Sensitivity of predicted T_i profile to uncertainties in various parameters for shot 960116024. The vertical shading around the predicted profile shows the variation in the prediction resulting from a change in one of the parameters on which it depends. The parameters examined are Z_{eff} ($\pm 30\%$), the RF input power ($\pm 50\%$ — larger than the actual uncertainty to illustrate the marginality of the profile), the q profile (TRANSP vs. EFIT), and the edge T_i boundary condition.

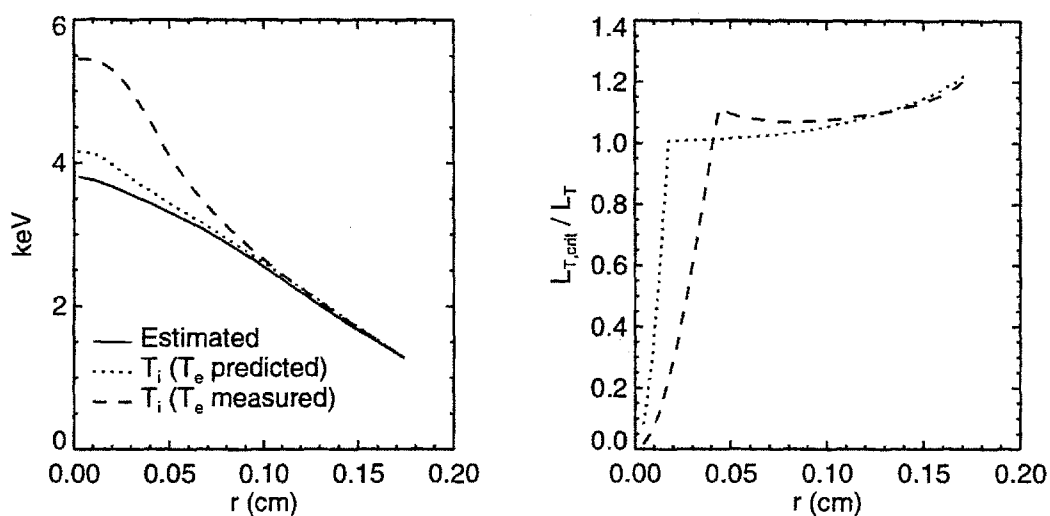


Figure 5.13: Checking sensitivity of prediction to T_i/T_e . The dotted line shows the T_i prediction obtained while also predicting T_e . The dashed line shows the T_i prediction obtained using the measured T_e profile. Even though T_e is poorly predicted for this shot, the T_i prediction is not significantly affected.

5.3.3 Boundary Condition and Edge Temperature

The effects of varying the location of the r_b boundary condition were examined for each of the four shots, with the results shown in figure 5.14.

For the two L-Mode shots (960301009 and 960229042), changing r_b basically shifts the predicted profile up or down, though there appears to be some steepening of T_i near the edge for $r_b = a$. The predicted profiles are systematically lower than the estimated profile for every value of r_b . While decreasing r_b/a improves the agreement between the two, it does not add any credibility to the IFS-PPPL model prediction, so the value $r_b/a = 0.8$ used in the runs presented in § 5.3.2 is adequate.

The H-Mode shot 960125004 shows a more significant improvement in agreement when r_b/a is lowered from 1.0 to 0.8. The critical gradient also increases to be closer to the measured profile, suggesting that for this shot the edge transport effects are important. Further improvement is obtained for $r_b/a = 0.6$, though this value does not provide a better test of the model. On shot 960116024, the IFS-PPPL model works very well for all values of r_b , suggesting that edge transport effects are not important here, and that the profile is critical all the way to the top of the edge pedestal [35]. Of all four shots, this one has the highest edge temperature, though the implications of this to the IFS-PPPL model are not clear.

5.3.4 T_e Predictions and Sawteeth

One possible explanation for the extremely high T_e predicted in the center is the neglect of sawtooth mixing. The code *nt* provides an ad hoc treatment of sawteeth, which sets χ_e equal to a specified constant value χ_{saw} inside the mixing radius r_m . The effects of this ad hoc model for shot 960116024 are shown in figure 5.15 for $\chi_{\text{saw}} = 0.5 \text{ m}^2/\text{s}$. The mixing radius is at roughly $r = 10 \text{ cm}$, based on the q profile from TRANSP shown in figure 5.12 on page 202. For the T_e prediction, plotted with the thin lines, the sawtooth model does improve the prediction of T_e in the center, changing the value from $T_{e0} \simeq 30 \text{ keV}$ to $T_{e0} \simeq 4.8 \text{ keV}$, much closer to the measured value of 4.5 keV. However, the model ∇T_e is already too large at $r = r_m$, so that the predicted profile is significantly higher than the measured profile for $r \in (5, 10) \text{ cm}$. This is reflected in the stored energy calculated using the predicted profile, which is 35% higher than the measured value. Overall, the ad hoc model is not useful, as the fact that the profile cannot be predicted for $r < r_m$ limits the range in radius over which the IFS-PPPL model can be tested. The effects of the sawtooth model on T_i are also shown; these are only from the difference in $\tau = T_i/T_e$ and are therefore small.

5.3.5 Z_{eff} Profile Shape

The analysis of the sensitivity of the IFS-PPPL model prediction to uncertainty in Z_{eff} used flat Z_{eff} profiles. Since the parameterizations in (5.13)-(5.2) depend on the

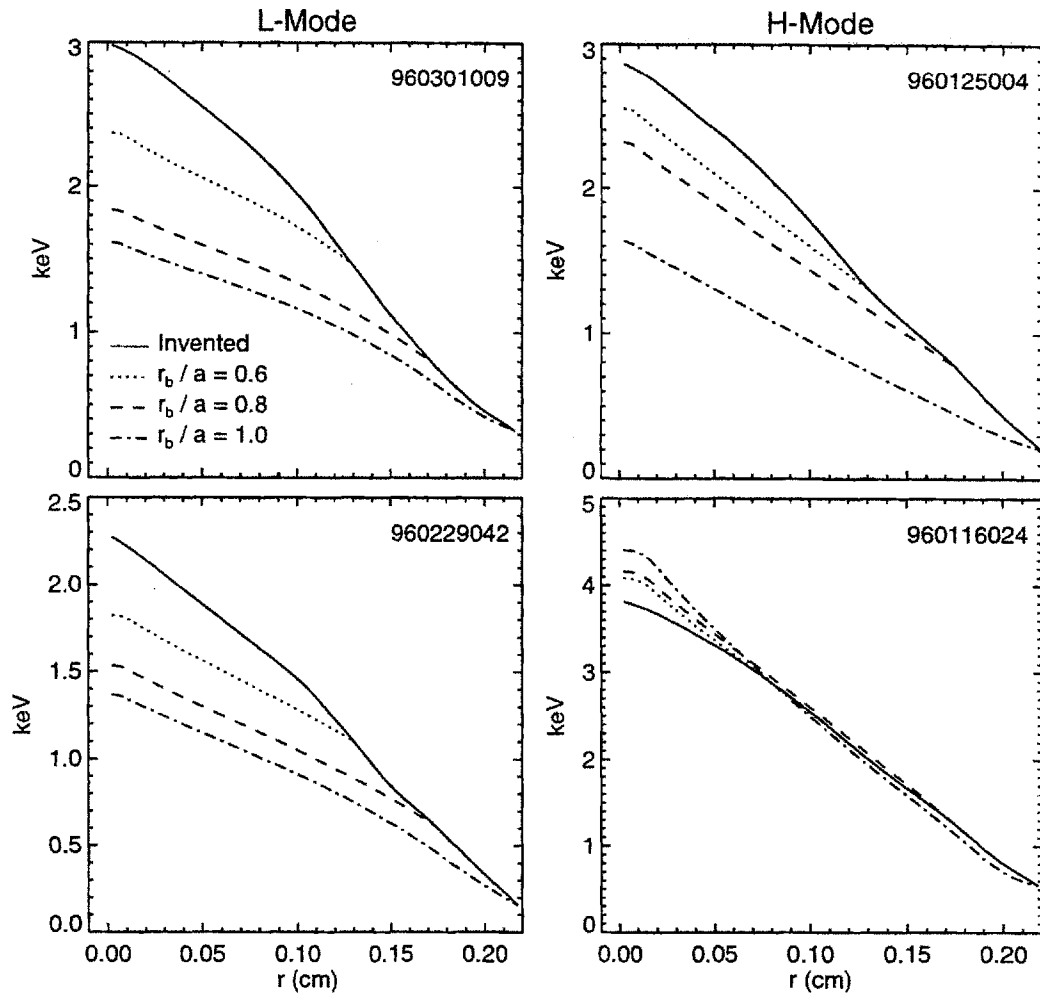


Figure 5.14: Comparison of estimated T_i profile to predicted profile using different r_b

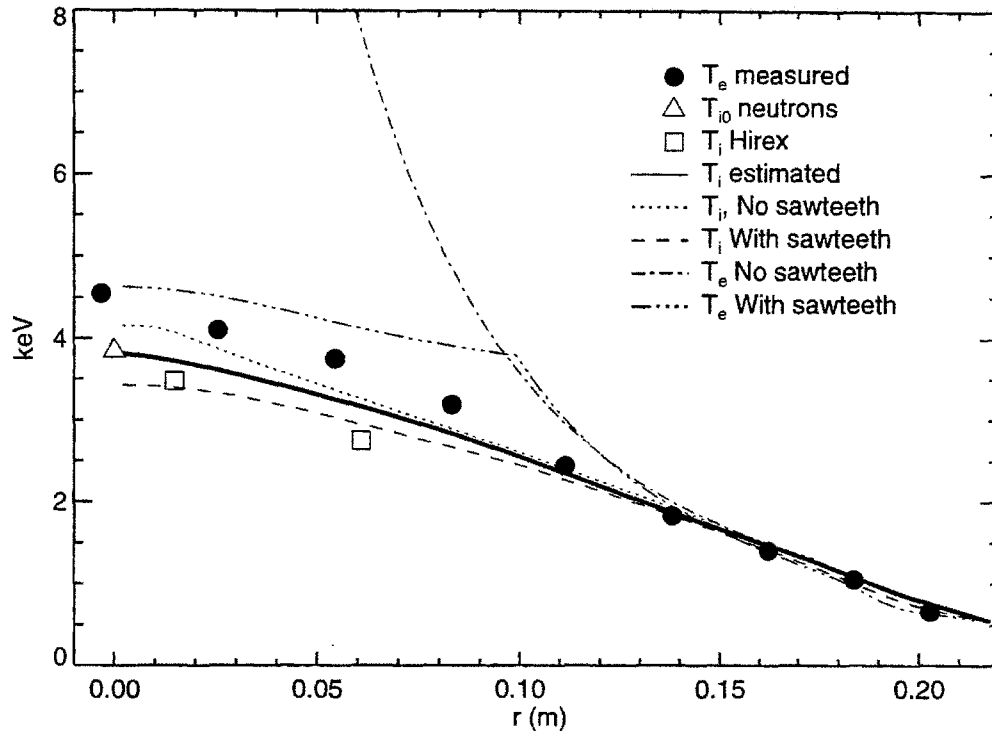


Figure 5.15: Effects on prediction of ad hoc sawtooth model, shot 960116024. Predictions obtained with and without the sawtooth model are shown. Sawtooth model sets $\chi_e = 0.5 \text{ m}^2/\text{s}$ inside the mixing radius.

local Z_{eff} , it is possible that the predictions can be more sensitive to non-flat profile shapes. Figure 5.16(a) plots the flat Z_{eff} profile used in the nominal prediction of shot 960229042, as well as hollow and flat profiles created artificially to study these effects. The extremes of the created profiles were set to be $\pm 30\%$ of the nominal value.

As shown in figure 5.16(b), the T_i prediction is insensitive to the Z_{eff} profile shape. The T_e profile does show more of a dependence, but only in for $r < 5 \text{ cm}$ where the prediction fails anyway. The fact that the predictions are not more sensitive to profile shape differences is comforting, as the Z_{eff} profile measurements on are subject to large uncertainties.

5.3.6 Toroidal Rotation

Measurements of the toroidal rotation velocity v_ϕ on Alcator C-Mod are possible with the Hirex X-ray spectrometer system [71,130]. A tangential-viewing spectrometer can provide central rotation measurements, while the main set of 5 perpendicular-viewing spectrometers can provide profiles. However, profile measurements have been possible to date on only one set of shots: 960130030-032. The Hirex system measures the

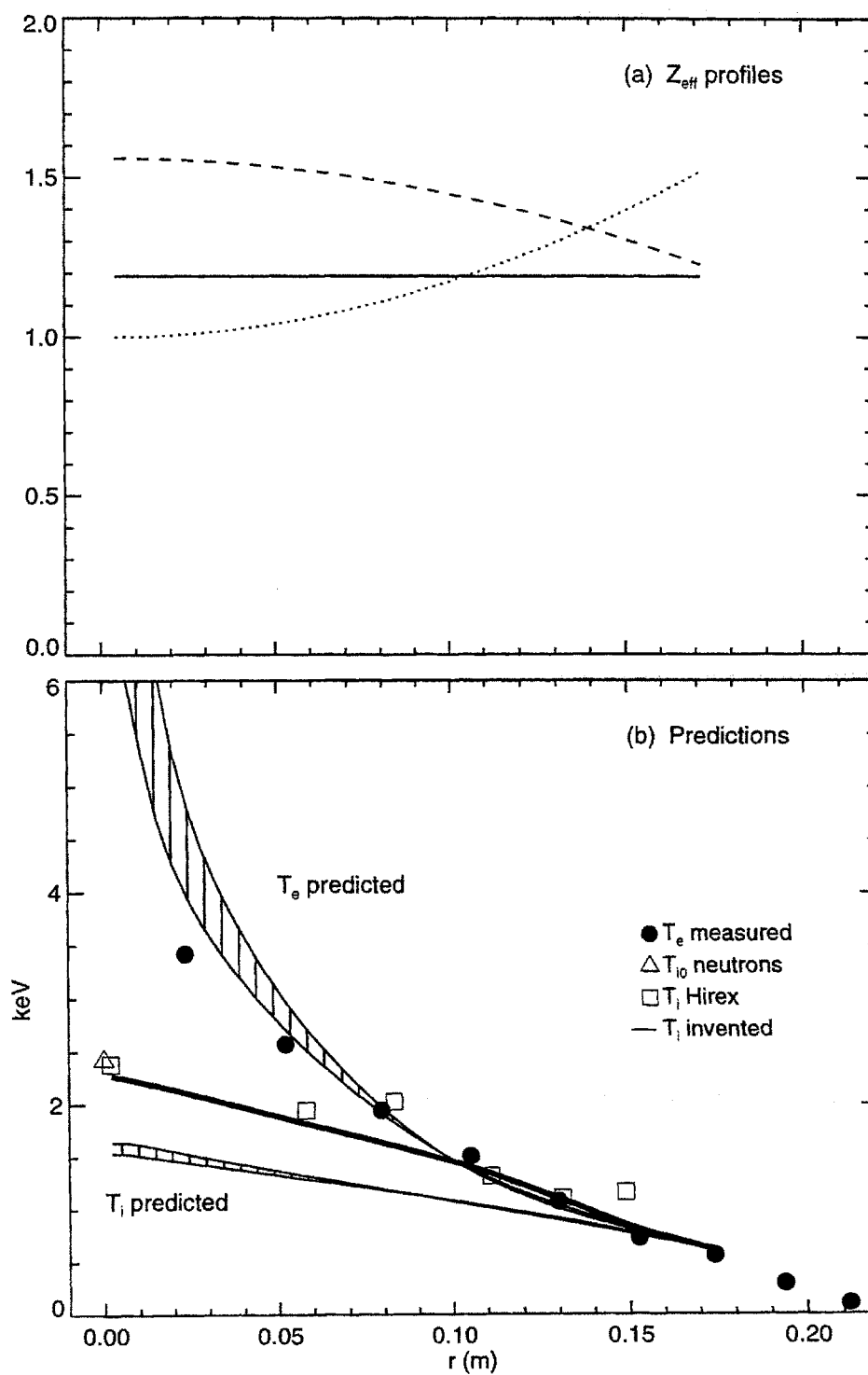


Figure 5.16: Effect of Z_{eff} profile shape on IFS-PPPL model prediction, shot 960116024

rotation of the argon impurity, as it is deduced from shifts in the spectra of Argon radiation. The toroidal rotation velocity of the deuterium ions is derived from the momentum balance equation using the Hirex measurement together with measurements of the deuterium and argon pressure gradients and neglecting poloidal rotations. For C-Mod, the resulting toroidal rotation velocity is about $1.5\times$ greater than the argon velocity, so the Hirex measurements used in this section were multiplied by a correction factor of 1.5 to obtain the rotation velocity of the deuterium majority species. Note that since C-Mod does not have neutral beams there is no external source of torque to drive the rotation.

Rotation measurements were not available for any of the four shots studied in § 5.3.2. A central measurement was available for shot 960116025, which had values B_t , I_p , $\langle n_e \rangle$, T_{e0} , T_{i0} , P_{in} , and U_{kin} similar to shot 960116024. This central measurement was combined with the profile shape from shots 960130030-032 to give a crude estimate of the profile for shot 960116024. The resulting profile is shown in figure 5.17(a) with the dotted line. The ratio of v_ϕ to the ion sound speed c_s is defined as the “Mach-number”, and is plotted with the solid line.

The T_i profile prediction from nt employing this v_ϕ profile is shown in figure 5.17(b) with the dashed line. The estimated T_i profile and the profile obtained from nt using only its diamagnetic rotation calculation are shown for comparison with the solid and dotted lines respectively. Since T_e was poorly predicted for this shot, the measured T_e profile was used for the T_i predictions. The rotation has no discernable effect on T_i , with Δ_{RMS} changing from 10% to 8%. The plot of $L_{T,crit}/L_T$ in figure 5.17(c) shows that the L_T obtained with rotation is only slightly lower than that obtained without over the range $r = 5$ to $r = 10$ cm. Overall it appears that the shearing rate of the rotation on C-Mod is not strong enough to effect the transport from ITG modes.

5.4 Summary and Discussion

The results from the tests of the IFS-PPPL model on the four shots described in § 5.3.2 are summarized in table 5.4. Within the estimated uncertainty ranges, the model is not successful at predicting T_i in L-Mode. The ratio of the measured temperature profile scale length L_T to the critical scale length of the model is $L_{T,crit}/L_T \simeq 2$, while the predicted profiles were closer to marginality with $L_{T,crit}/L_T \simeq 1$. On the other hand, the model’s predictions for T_i in H-Mode are more accurate. The effects of toroidal rotation on the predictions were negligible for the one shot examined.

Assuming that ITG turbulence is in fact the mechanism responsible for tokamak transport, so that the general physics in the IFS-PPPL model is correct, it is instructive to examine where the model might be wrong. A qualitative, schematic picture of the predicted transport is shown in figure 5.18 as the solid curve.

At C-Mod’s input power level P_{in} , their model predicts that in L-Mode $\nabla T \simeq \nabla T_{crit}$, when in fact it is observed that $\nabla T \simeq 2\nabla T_{crit}$. If the marginal stability model

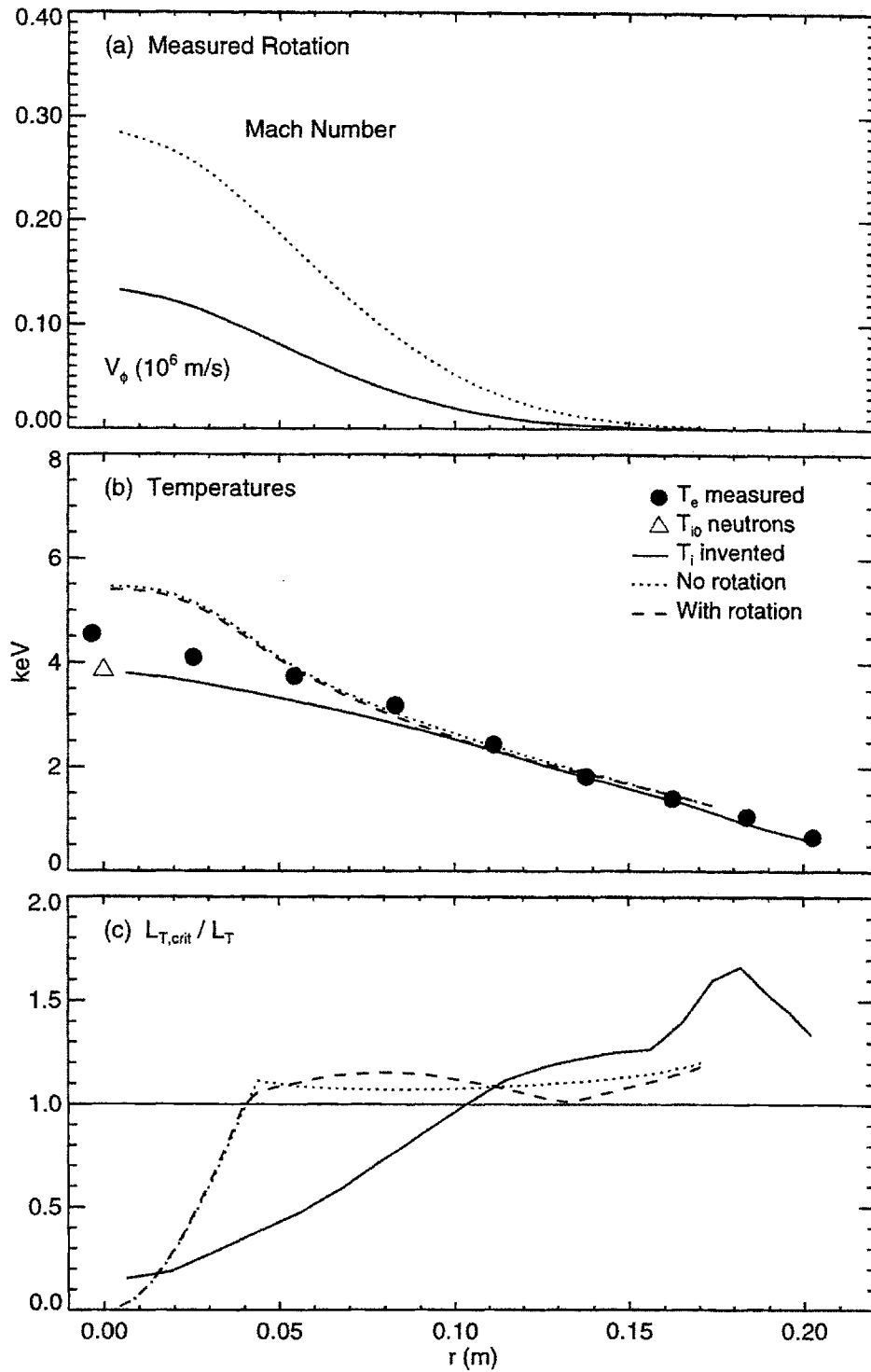


Figure 5.17: Effect of toroidal rotation on IFS-PPPL model prediction

Shot	Mode	$T(r_b)$ keV	ν/ν_{\max}	T_i Δ_{RMS} %	T_e Δ_{RMS} %	$L_{T,\text{crit}}/L_T$ Predicted	$L_{T,\text{crit}}/L_T$ Measured
960116024	H	0.00	1.60	0.0	0.0	0.0	0.0
960125004	H	0.00	6.30	0.0	0.0	0.0	0.0
960229042	L	0.00	6.00	0.0	0.0	0.0	0.0
960301009	L	0.00	2.30	0.0	0.0	0.0	0.0

Table 5.4: Summary of IFS-PPPL model comparison. Listed for each shot is the RMS difference between the predicted and measured profile for T_e and T_i , and the ratio $L_{T,\text{crit}}/L_T$ from the predicted and estimated T_i profiles. Also listed are the temperature at the boundary point (assuming $T_e=T_i$) and the ratio of the ν value for the shot calculated from (5.14) to the maximum ν for which the IFS-PPPL model is valid.

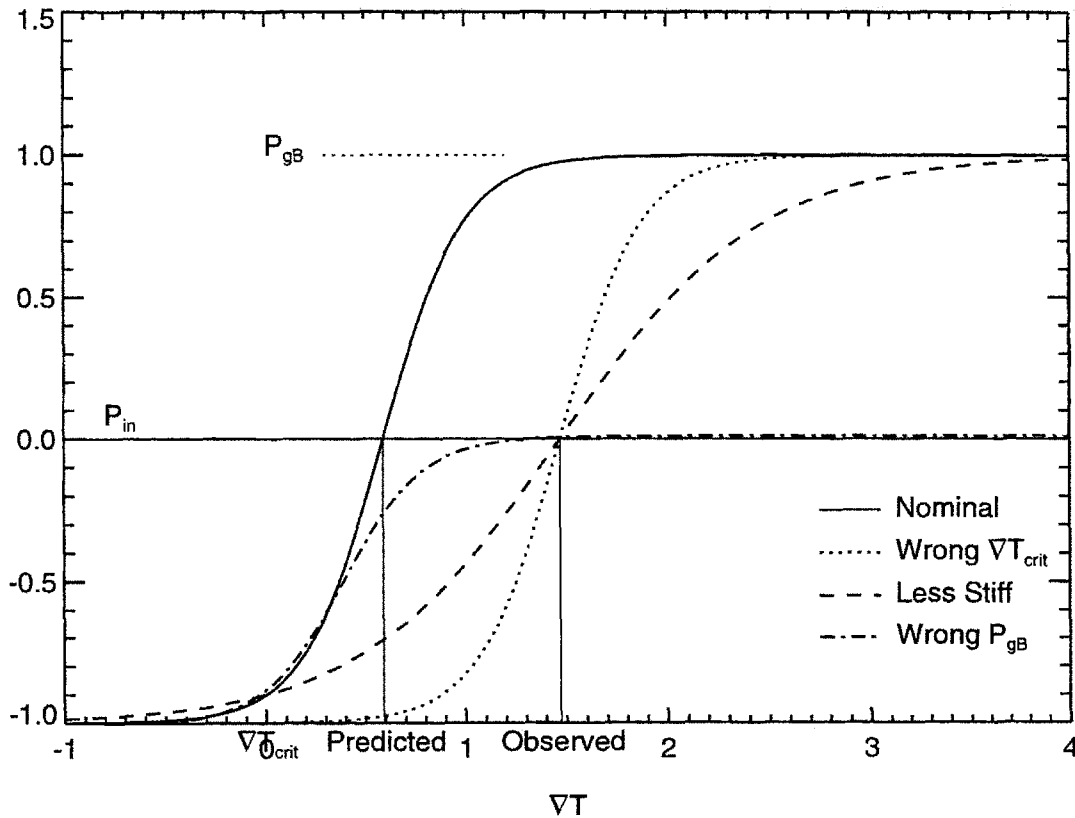


Figure 5.18: Possible ways in which the IFS-PPPL model can be wrong. Dotted line: ∇T_{crit} incorrect. Dashed line: Real transport less stiff. Dot-dashed line: Saturation is lower.

is correct, then there are three places in which the model could be incorrect. The first is in its prediction of ∇T_{crit} . If the actual behavior of the transport is represented by the dotted curve, the observed profiles would in fact be marginally stable (*i.e.* $L_T \simeq L_{T,\text{crit}}$). Assuming that the IFS-PPPL model's ∇T_{crit} is correct, a second place where it can be wrong is in the slope of the conducted power P_{cond} versus ∇T . If the actual transport is less stiff, so that the slope $dP_{\text{cond}}/d\nabla T$ is less steep as shown by the dashed curve, it would be possible for C-Mod to be at the marginal stability threshold with the observed profiles. A final possibility is that the IFS-PPPL calculation of the saturation level of the turbulence is wrong, and that the actual P_{gB} is given by the dot-dashed curve. Referring back to figure 5.2 on page 186 for a quantitative estimate, the plot of $P_{\text{in}}/P_{\text{gB}}$ for a typical C-Mod L-Mode indicates that the IFS-PPPL model χ would need to be reduced by almost an order of magnitude to obtain $\nabla T = 2\nabla T_{\text{crit}}$ assuming its calculation of ∇T_{crit} is correct. Of course, it is always possible that the model is just plain wrong as well.

The second of these points, the stiffness of the model, warrants additional attention. The IFS-PPPL model underestimates [126] the stabilizing effects of self-generated sheared $\mathbf{E} \times \mathbf{B}$ flows [131] that act to reduce the stiffness of the model. If C-Mod is in fact very close to the instability threshold, the growth rate of the instability, which scales as $\gamma \sim (L_{T,\text{crit}}/L_T - 1)$ can be very small, so that even if the shearing rate from the self-generated flows is small the two can be comparable. If this is the case, the actual transport would be less than what the model predicts, so that the predicted T_i is too low. However, the T_i predictions are more greatly underestimated in L-Mode than in H-Mode, while the rotation velocity is observed to be lower is expected to be lower in L-Mode than in H-Mode [130], so it is not clear that the effects of $\mathbf{E} \times \mathbf{B}$ shear are important here.

The success of the model at predicting T_e is also mixed. For two shots, one L-Mode and one H-Mode (960229042 and 960125004), its prediction was quite good ($\Delta_{\text{RMS}} < 10\%$) over the range $r \in (5, 15)$ cm, while for the other two shots, also one L-Mode and one H-Mode (960301009 and 960116024), the predicted T_e was far too high. While the ad hoc sawtooth mode in the nt code does result in an overall lower temperature, it does not significantly improve the agreement. Since the predictions depend on $\tau = T_i/T_e$, the fact that T_i was well predicted for shot 960116024 cannot be counted as a success because of the poor T_e prediction, though the prediction does not seem to be sensitive to τ . Since the IFS-PPPL model only includes the ITG mode's effect on the electrons, it is possible that there is another, electron based instability present for the C-Mod plasmas. This is currently being investigated by Dorland et al. It is also possible that the quasilinear method used to obtain χ_e is not sufficient. One of the recent improvements in their model is a full treatment of non-adiabatic electrons in the nonlinear code. As soon as revised parameterizations of χ_e based on this improvement become available, the results from C-Mod should be reexamined.

Figure 5.19 plots the Δ_{RMS} values for the T_e and T_i profiles from each shot versus the temperature at the boundary $r = r_b$ and versus the ratio ν/ν_{max} , where ν_{max} is the

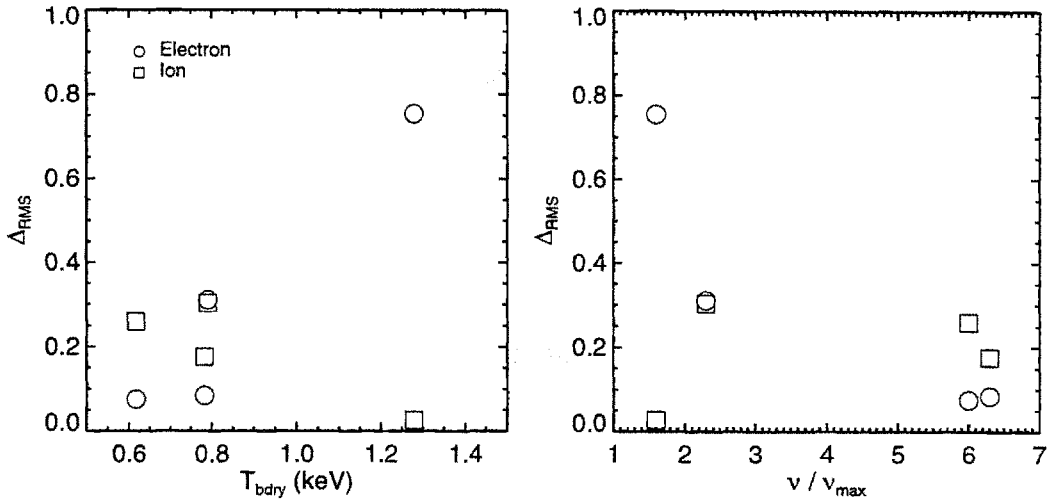


Figure 5.19: Plotting RMS difference between predicted and measured profiles versus the temperature at the boundary and the ratio ν/ν_{max} , where ν is defined by (5.14) and $\nu_{\text{max}} = 10$ is the maximum valid ν value for the IFS-PPPL model parameterizations.

maximum value of ν for which the IFS-PPPL model parameterizations are valid. The shot with the best T_i prediction (960116024) has the highest boundary temperature and the lowest ratio ν/ν_{max} . This shot also has the worst T_e prediction. There does not appear to be a systematic trend with T_{bdry} for either profile prediction if this shot is omitted. The T_i prediction seems to be independent of ν/ν_{max} , while the T_e prediction improves as ν increases beyond ν_{max} .

Given the fact that the parameterizations (5.2)-(5.13) are not strictly valid for any of the shots studied because of C-Mod's high ν values (see table 5.3 on page 190), these results cannot be considered final. Work is currently underway to extend the range of validity of the IFS-PPPL model to C-Mod's values of ν .

Chapter 6

Conclusions and Future Work

This chapter summarizes the results from both the ρ^* scaling experiments and the tests of the IFS-PPPL model, and discusses their implications. Future experiments that could either answer questions raised these results or improve upon their accuracy are suggested.

6.1 Summary

ρ^* Scaling

Empirical ρ^* scaling experiments are important because of their relevance to predicting the performance of ignition-scale tokamaks and because they can provide insight into the underlying plasma physics mechanisms responsible for anomalous transport [33]. The experiments on C-Mod have extended the range of plasma parameter space over which the scalings have been tested in both magnetic field (to 8 T) and density (to $\langle n_e \rangle = 3.8 \times 10^{20}/\text{m}^3$). In L-Mode, scaling of the global confinement time τ_E showed an overall Bohm-like dependence on ρ^* ($B\tau_E \propto \rho^{*-2}$) as did the scaling of the local one-fluid effective diffusivity ($\chi_{\text{eff}} \propto \chi_B \rho^{*0}$). However, the local scaling results are not definitive, as discrepancies in the radiated power fraction $f_{\text{rad}} \equiv P_{\text{rad}}/P_{\text{in}}$ call into question the underlying assumptions of the ρ^* scaling approach. Specifically, ρ^* scaling requires all dimensionless parameters to remain fixed as ρ^* is varied, and also assumes that atomic physics effects not described by the Boltzmann equation are negligible. For the L-Mode measurements, f_{rad} was not well matched for any of the shot pairs examined, and in fact was a crucial factor in determining the scalings.

H-Mode plasmas showed a gyroBohm scaling both globally ($B\tau_E \propto \rho^{*-3}$) and locally ($\chi_{\text{eff}} \propto \chi_B \rho^{*1}$). For these plasmas f_{rad} was well matched and low ($\lesssim 10\%$), so the assumptions behind ρ^* scaling should be valid. In both L- and H-Mode, determining the scaling of the individual electron and ion channels was not possible due to the large uncertainty in the electron-ion coupling associated with the high

density.

Tests of the IFS-PPPL Model

The IFS-PPPL Model is a comprehensive first-principles description of ion temperature gradient (ITG) driven turbulence and its consequent transport, developed by Beer, Dorland, Hammett, and Kotschenreuther [31]. The model employs a full numerical treatment of the linear gyrokinetic equation indicating that ITG modes are the predominant instability in the plasma core. It also includes a detailed nonlinear numerical treatment of the gyrofluid behavior of ITG modes providing parametric models for χ_i and χ_e useful for experimental comparison. Transport from the ITG turbulence is described by a marginal stability model; the temperature gradient scale length $L_T \equiv |T/\nabla T|$ is the critical parameter that determines the stability of the system. For $L_T > L_{T,\text{crit}}$, the mode is unstable and can drive transport. Alcator C-Mod plasmas are predicted to be very close to the marginal stability threshold in both L- and H-Mode, with $\langle L_{T,\text{crit}}/L_T \rangle \simeq 1.1 - 1.3$, where the average is taken over the range $r \in (5, 15)$ cm. The most critical test of the IFS-PPPL model is of its predictions very near to the threshold. C-Mod plasmas are much closer to this threshold than those from any other tokamak studied to date.

Comparisons of the model's predictions for T_i and T_e were conducted on four C-Mod plasmas. For the two L-Modes, the T_i predictions were systematically low, with an RMS difference of $\Delta_{\text{RMS}} = 25 - 30\%$ measured over the range $(\frac{a}{4}, \frac{3a}{4})$. The measured profiles had $\langle L_{T,\text{crit}}/L_T \rangle \simeq 2$ and were therefore much steeper than predicted. If the IFS-PPPL model critical scale length $L_{T,\text{crit}}$ is correct, the model is overestimating χ_i in L-Mode by nearly an order of magnitude. Predictions of T_i for the two H-Mode plasmas were more accurate, giving an RMS difference of $\Delta_{\text{RMS}} = 5 - 20\%$ over the same range, which is well within the estimated uncertainty. Predictions for T_e did not show a systematic difference between L- and H-Mode. In each case, the T_e prediction was accurate for one shot ($\Delta_{\text{RMS}} < 10\%$) and very inaccurate for the other ($\Delta_{\text{RMS}} > 50\%$).

Implications

The gyroBohm scaling of transport in H-Mode is consistent with the picture of gyroBohm scale turbulence as the underlying mechanism responsible for transport. Possible mechanisms for the increase in turbulence scale length in L-Mode are described in [87, 88]. gyroBohm scaling in H-Mode is favorable for ignition scale machines, because as large machines they naturally have small ρ^* values.

However, the problems encountered with the radiated power in the L-Mode ρ^* scaling experiments, and the general observation that the H-factor appears to be independent of the radiated power fraction in L-Mode, suggest that a local diffusive assumption for χ is not entirely appropriate.

The success of the IFS-PPPL model on C-Mod was mixed, however, it is encouraging that the model was close in some cases. Further testing is required before a firm picture of its ability to describe C-Mod plasmas can be formed. If the model is correct in describing the plasma as a marginally stable system, more appropriate parameters for quantifying transport are the critical temperature gradient scale length $L_{T,\text{crit}} \equiv |T/\nabla T|$ and the stiffness $d\mathbf{q}/d(\nabla T)$ where \mathbf{q} is the conducted heat flux. Given this, the most critical factor in the design of future ignition scale tokamaks from a transport point of view assuming they are marginally stable is not ρ^* , but the edge temperature.

6.2 Suggested Future Experiments

Many avenues are open for further exploration of the topics covered in this thesis. Several relate to increasing the precision of the measurements used in the ρ^* scaling and IFS-PPPL model analyses given the planned improvements in Alcator C-Mod's diagnostic capability. Others are suggested based on their results.

Both experiments suffered from a lack of well-resolved ion temperature measurements, requiring that a T_i profile be "invented" based on the central measurement available from the neutron rate, the T_e profile, and some assumptions on the coupling of the species. One of the capabilities of the diagnostic neutral beam (DNB) from the University of Texas [132] is finely resolved T_i profile measurements using charge exchange recombination spectroscopy [133]. These profiles will allow more thorough comparisons to the predictions from the IFS-PPPL model, and will help in determining the ρ^* scaling of the individual electron and ion channels.

Improvements in the density profile measurements from the planned tangential two-color interferometer will help to reduce the uncertainties in the τ_E and χ measurements, thus allowing a more unambiguous determination of the ρ^* scalings. These will also hopefully allow more accurate Z_{eff} profile measurements, which in addition to improving the precision of the TRANSP calculations would help in identifying the resistivity model that best describes C-Mod plasmas. Though knowledge of the resistivity model is not a crucial factor for either experiment, it is important to the overall understanding of transport.

Plans to include the full wave code TORIC as an option for the ICRF modelling capabilities in TRANSP will be of great assistance in the analysis of C-Mod plasmas, as TORIC is not plagued by the singularity problem like SPRUCE. It is noteworthy that the problem with SPRUCE was not fully recognized as a severe limitation until the code was run on C-Mod data.

Once the 40 MHz ICRF system is fully operational, it will be possible to strongly heat plasmas at fields lower than 5.3 T. One experiment suggested by this capability is to conduct the "ITER demonstration" discharges on C-Mod at the ν^* and β values anticipated for ITER, which are easier to obtain at lower field. This would allow a

more direct prediction of ITER's performance from extrapolation of C-Mod observations. It is recommended that future L-Mode ρ^* scaling experiments be conducted with boronized walls at reversed field. The former should reduce the radiated power fraction [10,11] so that the assumptions of ρ^* scaling are met, while the latter provides more flexible operation in L-Mode since the H-Mode power threshold is higher [134]. H-Mode scaling at 8 T will be possible once C-Mod is run at higher current. Finally, additional ν^* scaling experiments are desirable, as are β scaling studies, which could not be conducted in this thesis.

The tests of the IFS-PPPL model conducted in this thesis are preliminary. It is necessary to examine many more plasmas over a wider variety of conditions before firm conclusions on the model's veracity are possible. Suggested experiments include studying the success of the model as a function of edge temperature and input power. An important complementary test of the IFS-PPPL model will be to compare its predicted fluctuation spectra [135] to measurements obtained from beam emission spectroscopy [136] using the diagnostic neutral beam once it is installed. Detailed empirical studies of the scaling of transport with ∇T would also complement tests of the IFS-PPPL model.

Appendix A

TRANSP and MDSplus

Central to any local transport analysis is a computer code that calculates the transport coefficients and quantities. Given measured plasma quantities such as electron temperature and density, magnetic geometry, plasma current, surface voltage, Z effective, and radiated power, the code inverts a set of transport equations to obtain transport coefficients and fluxes as functions of space and time in the plasma.

The work in this thesis relied heavily on the the PPPL code analysis code TRANSP [47]. The physics models used in TRANSP are described in chapter 3. This appendix describes the work done to make TRANSP usable at Alcator C-Mod. In particular, it describes the modifications made to TRANSP integrating it with the MDSplus data system, and the tools constructed to facilitate its use. Also provided is some information on how to run TRANSP, and how to understand its outputs.

I write this section to describe the effort I have placed into getting TRANSP to work over the past four years¹, and also as a guide for the lucky soul who will take over TRANSP management after I graduate and move on. In cases where details would bog down the discussion, I refer users to separate documents that either I, or someone at PPPL, have written. The documents that I have written are in appendix B.

A.1 Motivation

TRANSP as provided by PPPL is self-contained, that is, after the initial effort of installing the code on a given computer system, no further programming is necessary to run it. Why then was so much time and effort expended in integrating TRANSP with MDSplus? The answer to this question lies in the cumbersome manner in which TRANSP is traditionally run, and the numerous advantages and simplifications offered by MDSplus.

There are three steps at which the user interacts with TRANSP, as outlined in figure A.1. The first two are at the input stage to the code. The measured plasma

¹Much gratitude is owed to Tom Fredian and Doug McCune for their copious help.

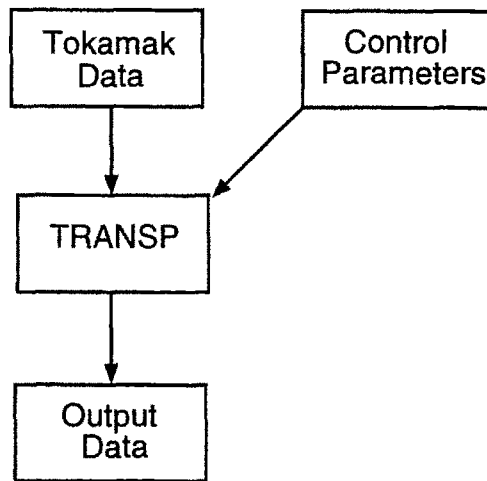


Figure A.1: Dataflow through TRANSP

quantities (henceforth referred to as “inputs”) must be supplied in a format that the code can read, and processed to a state where the analysis can be successfully applied. For example, since some of the computation involves calculating spatial and temporal derivatives, the inputs must be smoothed to prevent singularities in their calculation.

Inherent in local transport analysis is a certain amount of modelling. For example, particle balance relies on quantities that aren’t easily measured in a tokamak, and so must be modelled given some assumptions. In addition to supplying measured plasma data to TRANSP, the user must also select and control the models TRANSP uses in its calculations.

The third stage at which the user interacts with the code is at the output stage. The numerous outputs provided by TRANSP must be examined, and can be used for further computation.

TRANSP as supplied by PPPL runs “standalone”, that is, it is not designed to interface with the data system of a given tokamak experiment. This makes it usable by many experiments; indeed, there are eight sites around the world² that use TRANSP for their local transport analysis. However, the lack of an interface to the data system increases the work a user must do in running TRANSP, adding many steps at each stage of interaction to an already complex task.

TRANSP’s inputs are individual files provided by the user in UFILE format. Preparation of the inputs is a manual task. That is, each user must collect data from the tokamak data archive, and use the myriad of data processing tools included as part of the TRANSP package. Each processing step generates another UFILE, adding to the number of files the user must organize. Each tool runs as a separate, keyboard

²PPPL, MIT, GA, ORNL, JET, TEXTOR, ASDEX-U, and IPR in India.

menu-driven program, making manual preparation tedious.³ Control of the calculations is provided by a FORTRAN namelist, also contained in a separate file supplied by the user.

At the output stage, TRANSP supplies data in a format designed for its dedicated analysis tool, RPLOT. RPLOT is a powerful tool, allowing the user to plot data in a variety of ways, to integrate and differentiate the data, and to perform mathematical operations on the data. However, it too is keyboard menu-driven, sometimes requiring many keystrokes to perform simple operations. Another problem is that the output data is not readily exported to other analysis tools. Output data must be written as UFILES and then read back in. If a user wishes to view the inputs and outputs of another user's TRANSP run, he must be able to locate the proper files and have the proper system access to them.

Users must therefore keep track of many files for each run, and learn many different tools to prepare runs and examine their outputs. They must also devise a system for maintaining the context of a run; since the run is identified only by a number, and is otherwise a set of files, a separate mapping of run number to context must be developed. Finally, the user must be able to archive all these files in a systematic fashion. These problems complicate the use of standalone TRANSP.

Enter MDSplus. As Alcator C-Mod's data system, it provides a revolutionary means of acquiring, storing, viewing, and analyzing data.⁴ The central philosophy of MDSplus is integration. All data from a given shot are stored in one structure, in a common format. This includes raw data from the CAMAC digitizers, calibrated data and the calibrations themselves, CAMAC setup, comments, and results from further analysis. All this information can be examined using the same X-windows based graphical tools: scopes, the traverser, and IDL. Users need only know the names in the data tree, or "node" of the data signal in which they are interested to gain access to that data. Maintaining the context of data is simple, as comments and text can be stored with the data in the same structure. Archiving is maintained centrally and seamlessly, so users do not need to know where their data is physically located in order to access it, or to back it up themselves.

Only one data structure outside of MDSplus is used at Alcator C-Mod: a relational database⁵ storing scalar data extracted from the shots. MDSplus is not designed to handle multiple shots efficiently in the way that, for example, would be needed to answer the question "return the shot numbers of all shots with maximum plasma current greater than 1.0 MA and that were diverted". The relational database, for historical reasons known as *LOGBOOK*, is used in these cases where scalar data from multiple shots is desired.

Given the philosophy of MDSplus, it was natural to integrate TRANSP as well. For any given run, all inputs, outputs, and the namelist could be stored together

³The tools can be run in automatic mode, saving the user some effort.

⁴For more information on MDSplus, see [137,138].

⁵DEC's RdB

with comments in a subtree of the shot being examined, thus retaining its context. Inputs could be culled directly from the other subtrees, eliminating the need for UFILES. Every processing step could be recorded for each input signal, so that the run could be easily reproduced and understood. Outputs could be examined with the pre-existing MDSplus tools, so even novice TRANSP users, and non-users interested in TRANSP data, would be able to view them without learning new tools. New tools for viewing the output could be constructed using the same X-windows based IDL graphical tools. Experienced users would find these tools easier to use interactively, while new users would more easily learn to run TRANSP being able to concentrate on the physics in the code rather than its interfaces. The many TRANSP runs could be organized with a table in *LOGBOOK*, so that users could keep track of their runs.

The remaining sections in this appendix describe first the interface between TRANSP and MDSplus, and then the user tools, that allow TRANSP to be integrated with MDSplus.

A.2 TRANSP Specifics

The name “TRANSP” is used throughout in three different meanings. One refers simply to the binary executable *TRANSP.EXE* that is the actual analysis program. Another meaning for “TRANSP” is as a reference to the entire package of programs and utilities supplied by PPPL, including *TRANSP.EXE*. Some of these programs are run as pre- or post-processors as a matter of course. Other programs are data preparation tools or output viewing tools. The final meaning for “TRANSP” used in this thesis is as a reference to a given analysis on a given set of data, a so-called “TRANSP run”. Hopefully the context in which “TRANSP” is used will allow the reader to determine the proper meaning.

The entire TRANSP package is maintained by PPPL in a Unix cvs library (henceforth referred to as the “PPPL repository” or simply “repository”). Sites remote to PPPL obtain it over the internet via a local Unix node running cvs software. The Alcatraz C-Mod computing environment, in which all analysis and development is conducted, is VMS-based. PPPL developed a system for porting code from Unix to VMS and back at the local site, for receiving updates from the repository, and for sharing local modifications and improvements with the repository and therefore the world. There are many documents within the TRANSP distribution describing this system, how to set it up, and how to maintain it. See for example `TRANSP$:[CMSREF.VAXPORT]INSTALL.INF`, `TRANSP$:[CMSREF.VAXDOC]PSHARE_UPDATE`, and `TRANSP$:[CMSREF.DOC]UNIX_COMMIT.DOC`. The documents in § B.3 and § B.16 are useful references for the local implementation. They describe how to set up the system requirements for TRANSP, and provide an outline of the procedure to update the code. Alcatraz C-Mod was the first remote site to setup the PPPL system; consequently the work done here helped PPPL debug and perfect it. The steps followed

are documented in `TRANSP$: [CMSREF.VAXDOC]NFS_VMS.DOC`.

Code development in VMS must be conducted within the “TRANSP environment”, involving the definition of a large set of logical names and symbols. PPPL intended that a separate computer account be used for this environment; this type of account is not allowed by PFC computer policies, so TRANSP development must be done from within a user’s account. To avoid conflicts between the user’s environment and the TRANSP environment, a DCL command procedure (`TRANSP$: [USER]TRANSP_LOGIN.COM`) is used to first clear the former before setting up the latter. The entire code package is contained in the directories under `TRANSP$: [000000]`. There are other file areas important to TRANSP itself, and several are used by the C-Mod specific tools. See § B.13 for a list of these areas.

To see how MDSplus was hooked into the TRANSP system, first examine figure A.2, which shows the traditional sequence of steps executed for any TRANSP run.⁶ The first step is collecting and processing inputs from various diagnostics, using their processing utility programs in the distribution. The next step is preparing the namelist, which is where most of the thought and effort is expended. The namelist is complicated, and until recently was simply a text file with sparse comments. PPPL just completed an X-Windows based namelist tool, XTranspin, greatly simplifying this step. For non-circular plasmas, TRANSP’s MHD fixed-boundary equilibrium solver uses the fourier-decomposed moments of the plasma boundary. These moments are calculated by the program SCRUNCHER, whose outputs are fed to the next stage. With the input data including the boundary, and namelist prepared, the program TRDAT is run to map all the inputs to a common spatial coordinate and timebase, write all the inputs to the file read in by TRANSP, and check the namelist for inconsistencies and errors. One more step is executed before running TRANSP itself: PLABEL is run to generate labelling and indexing information for the outputs. After TRANSP.EXE runs, POPLT translates its temporary ASCII output to direct access binary files, which together with the files from PLABEL are used by the output-viewing program RPLOT.

Figure A.3 shows the MDSplus hooks into the TRANSP system. The application PRETRANSP⁷ collects and processes the inputs, read from the various diagnostic trees and written to the TRANSP tree⁸. PRETRANSP also provides access to the namelist, whether through a text editor or through XTranspin. TRDAT was modified so that, if the namelist variable `KMDSPLUS = 1`, data is read from the TRANSP tree rather than from a set of UFILES. The `PRExxx` and `EXTxxx` variables in the TRDAT namelist then instruct TRDAT to read node `xxx`.⁹ SCRUNCHER was also modified to allow input and output from an MDSplus tree. The PLABEL stage remains unchanged,

⁶“Traditional” means the standalone, non-MDSplus version.

⁷See § A.4 for information on PRETRANSP.

⁸The TRANSP tree is described in the next section.

⁹There is in general more to reading inputs than setting these switches. See `TRANSP HELP` for more information.

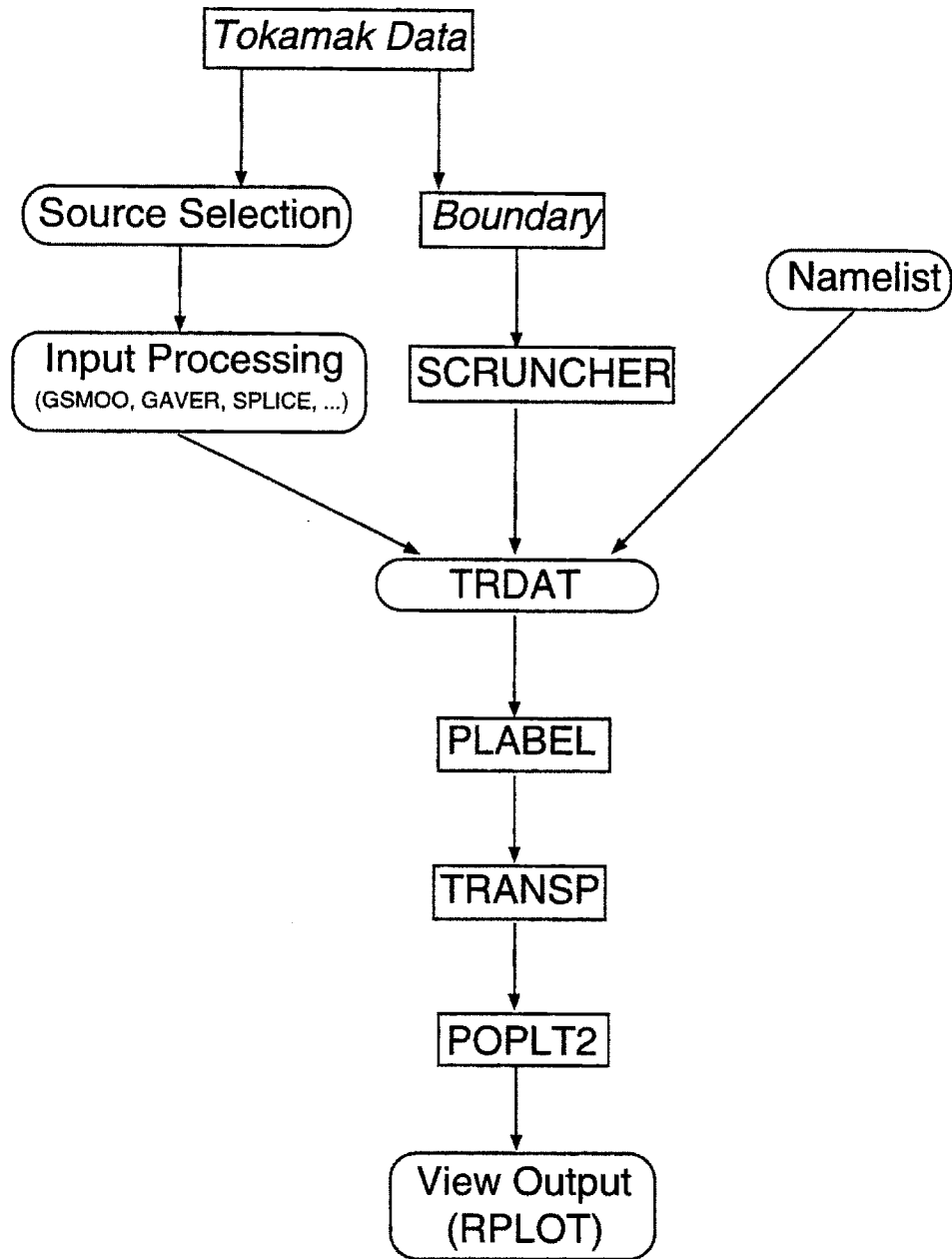


Figure A.2: Steps followed in traditional TRANSP run. Oval boxes are user interface steps. Programs are capitalized.

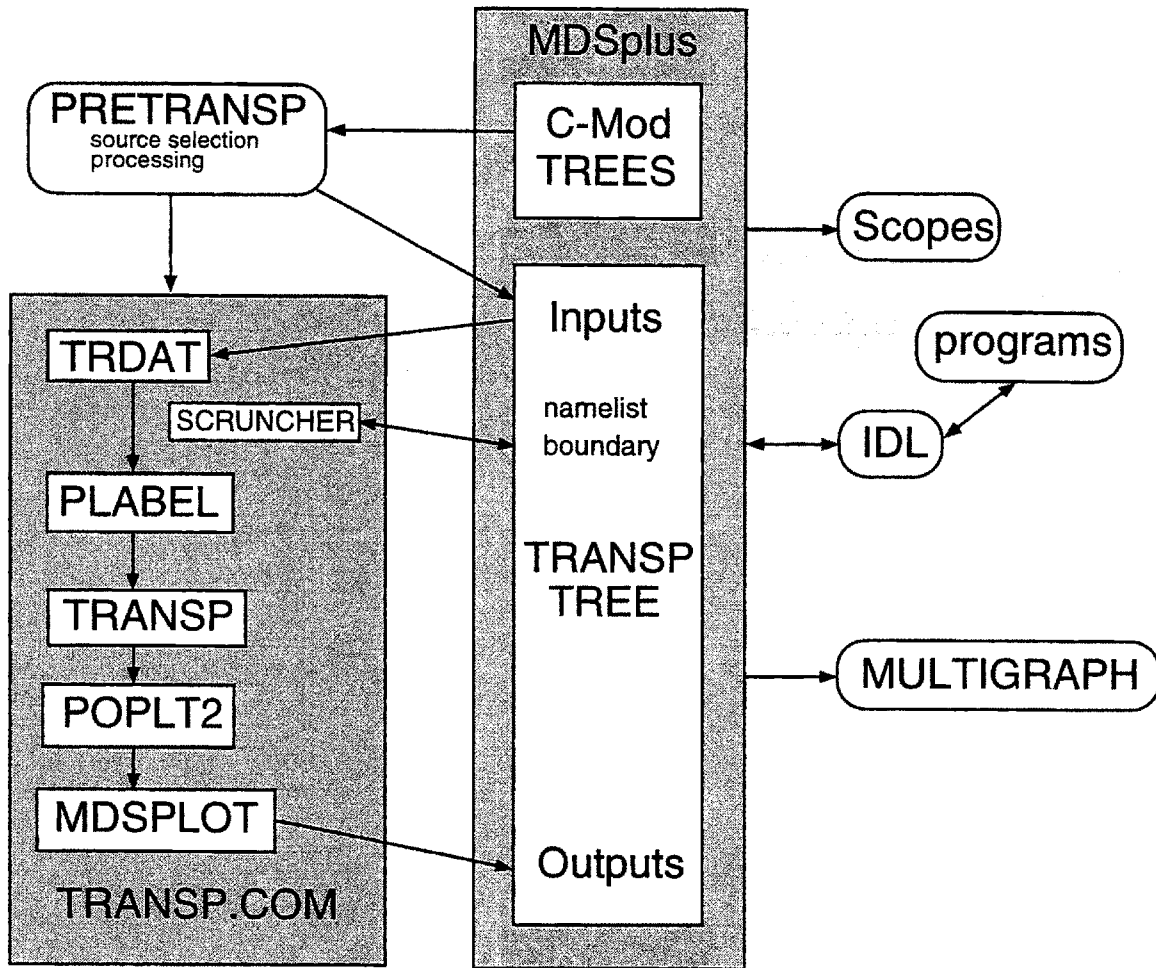


Figure A.3: Steps followed in MDSplus TRANSP run. Oval boxes are user interfaces. Lines indicate dataflow to/from tree.

as does the POPLT stage. Rather than running RPLOT interactively, MDSPLUS runs automatically to write the output data back into the TRANSP tree. The node names of the outputs are identical to their labels in RPLOT. Details of the specific modules in TRANSP modified for MDSplus integration are provided in the document `TRANSP$: [CMSREF.DOC]MDSPLUS.DOC` and in § B.4.

These modifications are now a permanent part of the TRANSP repository at PPPL. In the future, other tokamak experiments using MDSplus would be able to take advantage of them to use TRANSP.

A.3 MDSplus Specifics

This section describes the MDSplus side of the integration. The structure of the TRANSP tree is shown in figure A.4, mirroring the flow out data into and out of TRANSP, as depicted in figure A.3. Each major section of the tree/data flow will be dealt with in turn.

For future reference, the main structure nodes `\TOP.INPUTS`, `\TOP.MAPPING`, `\TOP.OUTPUTS`, and `\TOP.TRANSP_OUT`, will be referred to by their tags: `\INPUTS`, `\MAP`, `\OUTPUTS`, and `\TRANSP_OUT` respectively.

A.3.1 Inputs

Each potential input, named by a three letter “trigraph”, has a corresponding signal node in the TRANSP tree under the structure node `\INPUTS` from which TRDAT reads its data; this signal node is referred to as the *input* node. The input node has several child nodes underneath it providing context for the input.¹⁰ The node `SOURCE` stores the unprocessed C-Mod data. The processing steps that map the source data to the input node are recorded in the node `PROCESSING`. The nodes `COMMENT` and `LABEL` are self-explanatory. The data flow is from C-Mod tree, to source node, through processing steps, to the input node, and then to TRDAT.

The source node is loaded by an IDL procedure designed to get data from the C-Mod tree and return an IDL array. A copy of this procedure is stored in the node `SOURCE_PRO`. For most signals, the source procedure is simple: get data from a node in the C-Mod tree and return it as an array with its dimensions. Some signals must first be processed to some extent. For example, all the profiles must be mapped to a fixed, normalized radial grid, as TRANSP cannot handle radial positions that change with time. Storing the procedure in the tree with the input signal keeps a permanent record of the steps used to generate the source.

Given that the source procedure can alter the data from its state in the C-Mod tree, and that it is desirable to keep a record of the processing steps, it should be clear why there are two distinct nodes — input and source — for every TRANSP input. The

¹⁰The input node is therefore a signal node with structure underneath it.

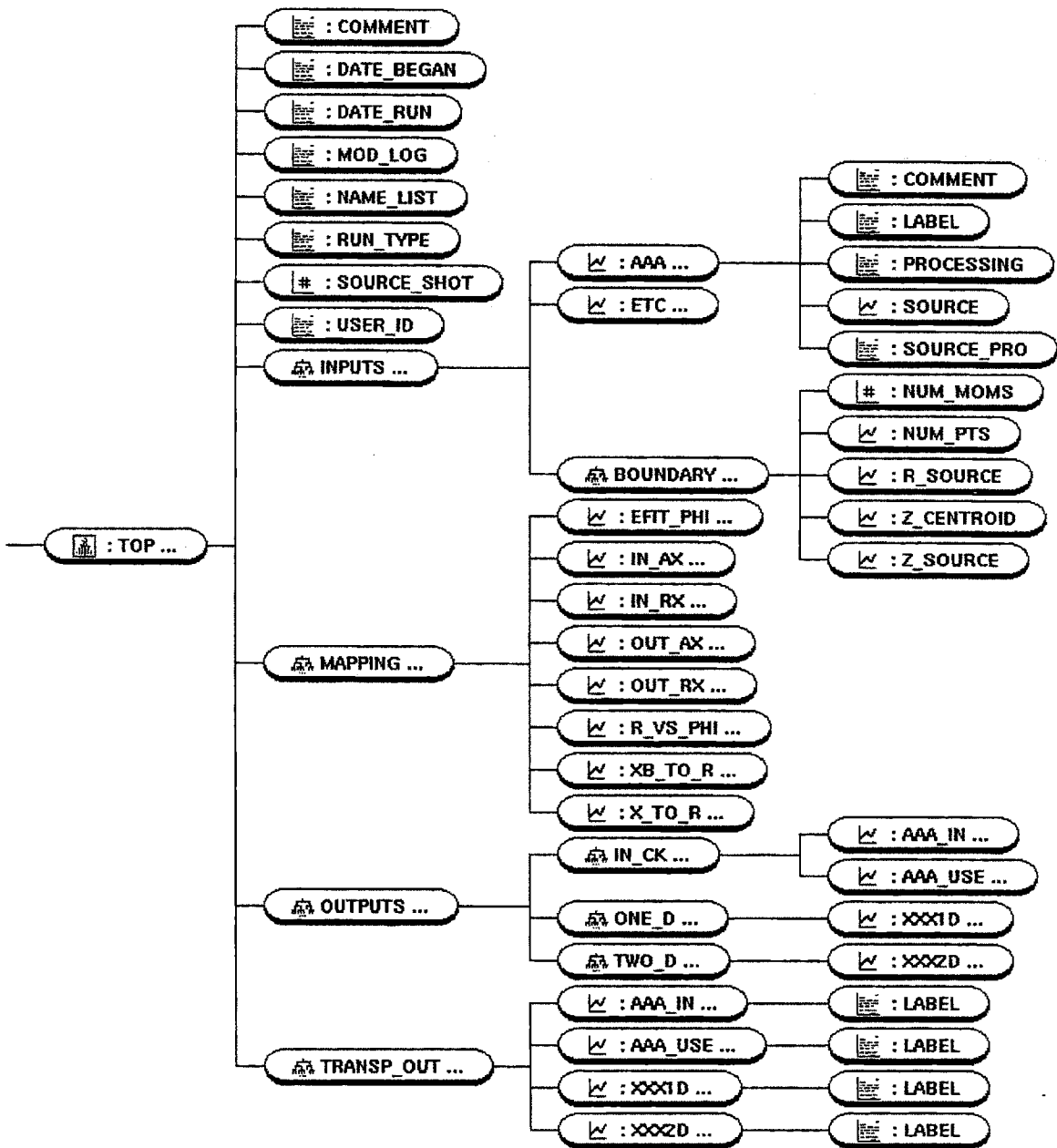


Figure A.4: TRANSP tree structure

source node stores the signal output by the source procedure, while the input node stores the processed version of the signal. Thus it is possible for example to determine if the data in the C-Mod tree has been changed, which happens occasionally, or to start processing with a fresh version of the source, without additional effort.

In practice only a small subset of the possible inputs are used. The list of all potential outputs, kept in the TRANSP distribution as `TRANSP$: [CMSREF.MISC]TRDATGEN.SPEC`, is used to generate the structure of the tree. See the document in § B.16 for more information. Some inputs that would never be used at C-Mod, in particular those that refer to the neutral beams, are not included in the TRANSP tree.

Also under the `\INPUTS` structure is the child `BOUNDARY`. This contains nodes storing the EFIT-supplied boundary location as functions of time that are inputs to the SCRUNCHER code. SCRUNCHER writes its outputs to some of the input nodes: `RMO`, `RMM`, and `YMM` for up-down symmetric plasmas, `MRV` for up-down asymmetric plasmas.

A.3.2 Outputs

On the output side, data from TRANSP flows through MDSplot to the output nodes. Each potential output has a corresponding node under the structure node `\TRANSP_OUT` where MDSplot actually writes the output data. Users look for outputs under `\OUTPUTS.ONE_D` or `\OUTPUTS.TWO_D`. In general, each signal under `TRANSP_OUT` has a corresponding signal node under `\OUTPUTS`. However, outputs are not guaranteed to retain their name forever, and outputs have in the past changed names. To avoid breaking user tools that rely on these nodes, the old names for the signals are retained under the `\OUTPUTS` structure along with the new ones. Both map to the same node under `\TRANSP_OUT`. The list of all potential outputs is kept in the file `TRANSP$: [CMSREF.OUTCOR]XPLOT7.FOR`. As with the inputs, nodes not relevant to C-Mod are not included. Details on keeping the output structure up-to-date are provided in § B.16.

A.3.3 Namelist

Ideally, a structure similar to `\INPUTS` and `\OUTPUTS` would have been used for the namelist: a `NAMELIST` structure with children nodes for each of the parameters. In addition, TRANSP and TRDAT could have been modified to read the namelist variables directly out of the tree, avoiding the need for a separate namelist file. These modifications were not implemented, as the large number of parameters, and their volatility, made the task too large for someone wishing to graduate before old age.¹¹ Instead, the namelist is stored as a text array in the `NAME_LIST` node, and is written to the file read by TRDAT and TRANSP immediately prior to submitting the run.

¹¹Some day they may yet be included, especially now that XTranspin works and would make organization of the parameters more straightforward.

This file is never intended to be directly accessed by the user. Only a sufficiently masochistic user would edit or delete it after it is created.

A.3.4 Other Nodes

The last major structure in the TRANSP tree is `MAPPING`. This was added to aid users in understanding the various coordinate systems used and in mapping between them. See § A.3.7 for more details.

In addition to the inputs, outputs, and namelist, other information is also stored in the TRANSP tree: the userid of the person executing the run, the date the run was created, the date the run was executed, comments on the run, and a short string used to classify the run (`RUN_TYPE`). A log of the modifications to the tree structure is kept in `MOD_LOG`.

A.3.5 Trees

Often, when running TRANSP, a given shot is analyzed many times. Sometimes, this is because it is not clear which namelist settings are optimal, so several successive runs are required to tweak the namelist settings. Other times this is because the results obtained from different models are compared. Many of the runs on a given shot are temporary, not requiring permanent archiving. To allow the distinction of temporary runs from permanent ones, two separate tree structures are used.

When first creating a run, it is contained in a shot of the TRANSP tree, completely independent of the C-Mod tree. Shots in the TRANSP tree are identified by a four-digit shot number, and each is logically associated with a C-Mod shot (the C-Mod shot number is stored in the node `\TOP:SOURCE_SHOT`). After running, if the user decides the run merits permanent archiving, the run is moved, or “pasted” into a subtree of the C-Mod tree’s `TRANSPORT` structure, as represented by figure A.5. The subtrees are named `TRANSPnn`, where `nn` is a two-digit number determined incrementally ranging from 01 to 20.¹² Pasted shots lose their old TRANSP tree shot number, and cannot be deleted.

A.3.6 Run Management

Database

A RDB table `TRANSP_RUNS` is used to keep track of the TRANSP runs.¹³ The primary key of the table is the TRANSP shot number, the C-Mod shot number, and a field called `RUN_ID`. For pasted runs, `RUN_ID` is the name of the subtree of the run: `TRANSPnn`, while the TRANSP shot is zero. Unpasted runs have `RUN_ID` equal to “X”.

¹²If it is necessary to store more than 20 runs for a given C-Mod shot, contact TWF or JAS for additional information.

¹³At time of writing the table is in its own database. Eventually it will be a table in `LOGBOOK`.

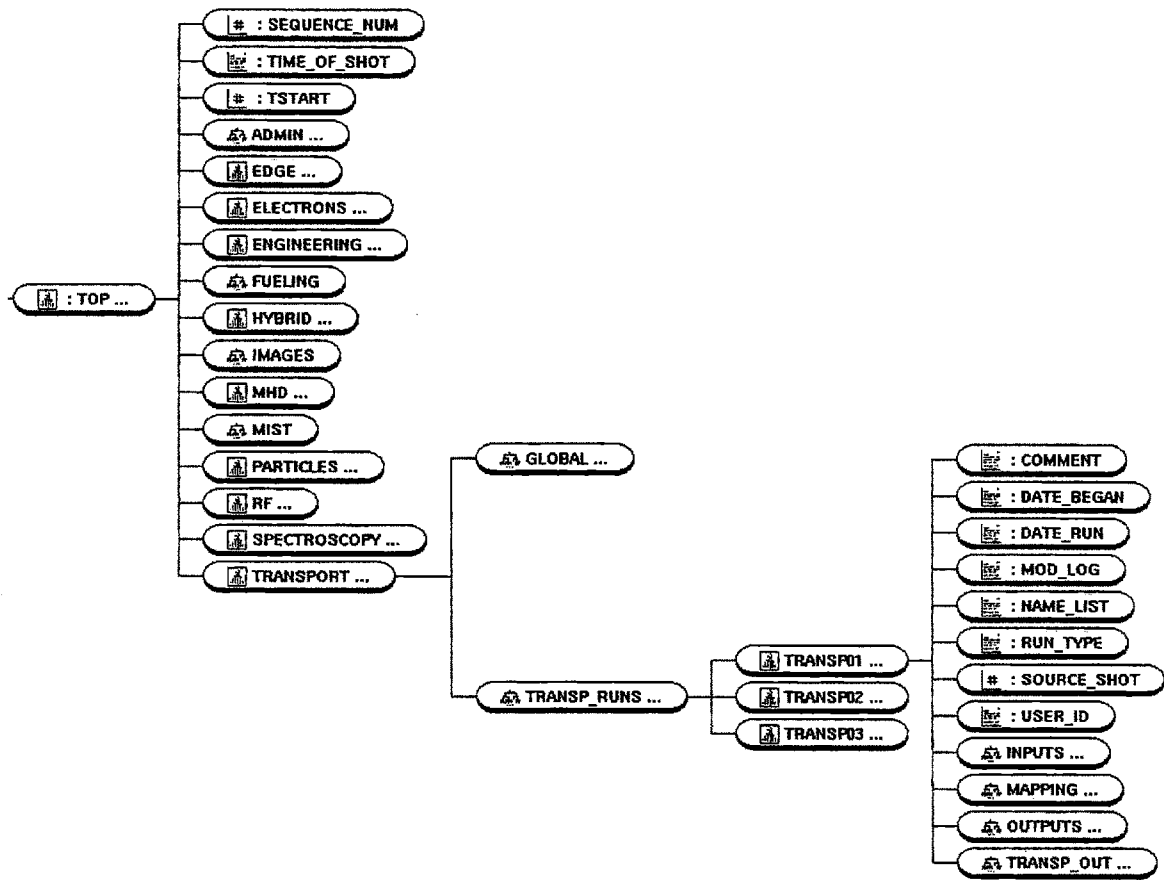


Figure A.5: Pasted TRANSP Run

Value	Status
-1	Locked
0	Not Run
1	Running
2	Run

Table A.1: Run Status Values

The table also stores other information about the run, some of which is also stored in the tree: the userid of the person creating the run, the dates it was created and run, the comments and `RUN_TYPE`, and other useful information. Some namelist switches are also included to allow easy location of runs with particular switch settings.

A field called `RUN_FLAG` holds the status of the run. Table A.1 shows the possible values. A run is locked when it is being edited in `PRETRANSP`, preventing collisions. Runs are marked as “run” only if they complete successfully, without any errors.

The tool `SHOT_SELECT` is used everywhere it is necessary to view or select a run. Depending on the circumstances, a subset of the runs is displayed for selection. For example, users may only select a run to process in `PRETRANSP` from the set that they themselves created. Users may also restrict the display of runs with an SQL statement. `SHOT_SELECT` is always called by another IDL application (see the section on `POSTTRANSP`).

POSTTRANSP

`POSTTRANSP` integrates many if not all of the run management functions necessary for `TRANSP`. Its functions are divided into two categories: those that are specific to a selected run, and those that are general to all runs.

Once a run is selected in `POSTTRANSP`, a user may examine the run’s log file and `TRANSP_RUNS` database information, and the comments for the run. If the run belongs to the user, the comments may be edited. `POSTTRANSP` is also the only means to view the namelist and signal source and processing information from a run once it has been successfully completed. An incorrectly set run status flag (see § A.3.6) can be reset using `POSTTRANSP`. This would be necessary if for example, `PRETRANSP` crashes during processing of a run, or a submitted run crashes without updating its status. Sometimes submitted runs that crash have completed `TRANSP.EXE` successfully but were unable to write outputs to the tree, either because the disk is full, or because the computer went down. `POSTTRANSP` allows the user to recover the data from these runs without having to rerun the `TRANSP` itself. Finally, `POSTTRANSP` is used to paste good runs into the C-Mod tree, and to delete unnecessary runs.

`POSTTRANSP` is also the front end for a generic `SHOT_SELECT` tool that allows users to browse through the runs in the database, and for the `NODE SEARCH` utility. This tool allows users to translate the codes used for the input and output signal

names into english. A final function not specific to a selected run is management of currently enqueued runs.

See the document in § B.6 for more information on POSTRANSF.

A.3.7 Coordinate Systems

This section explains the different coordinate systems encountered in running TRANSP.

TRANSP uses the normalized square root toroidal flux as its internal position coordinate. This is defined to be:

$$\hat{\phi}(x) = \sqrt{\Phi(x)/\Phi_{bdry}}, \quad (\text{A.1})$$

where the flux at the magnetic axis is defined to be $\Phi(0) = 0$. Calculations are performed on a fixed $\hat{\phi}$ grid, stored as \XIRSYM, from -1.05 to 1.05 in 0.025 increments. Negative $\hat{\phi}$ here refers to the *inboard* flux surfaces. The flux range 0 to 1.00 is divided into 20 “zones” of width 0.05 for use in the calculations. In addition there is an edge zone from 1.00 to 1.05. Since the flux surfaces can change shape or position as a function of time, the mapping $\hat{\phi} \Rightarrow R$ is in general a function of time. Because of this, and because most data at C-Mod is not plotted versus $\hat{\phi}$, some discussion on the coordinate systems used for the inputs and the outputs is necessary to avoid confusion. The document in § B.12 contains most of this information.

Inputs

Several possible coordinate systems are allowed for the data input to TRANSP; these are listed in table A.2 and defined with reference to figure A.6. Regardless of which is used, the radial coordinates of any given profile cannot be time varying. It is allowable to have different input profiles on different coordinate systems. It is also possible to input the data on *normalized* coordinates. The normalization is defined by

$$x = \frac{R - R_{\text{start}}}{R_{\text{finish}} - R_{\text{start}}}, \quad (\text{A.2})$$

where R_{start} and R_{finish} are defined in table A.2 for each allowed coordinate system.

PRETRANSP prepares all input data on a normalized minor radial grid, which corresponds to the “NRI value” of 4 in table A.2.¹⁴ That is, the data is input on a fixed normalized grid calculated as

$$x = \frac{R - R_0}{a}, \quad (\text{A.3})$$

¹⁴The NRI value refers to the setting of the namelist switch `NRlxxx`. For a normalized minor radial grid, `NRlxxx=-4` is used. See § A.4.4.

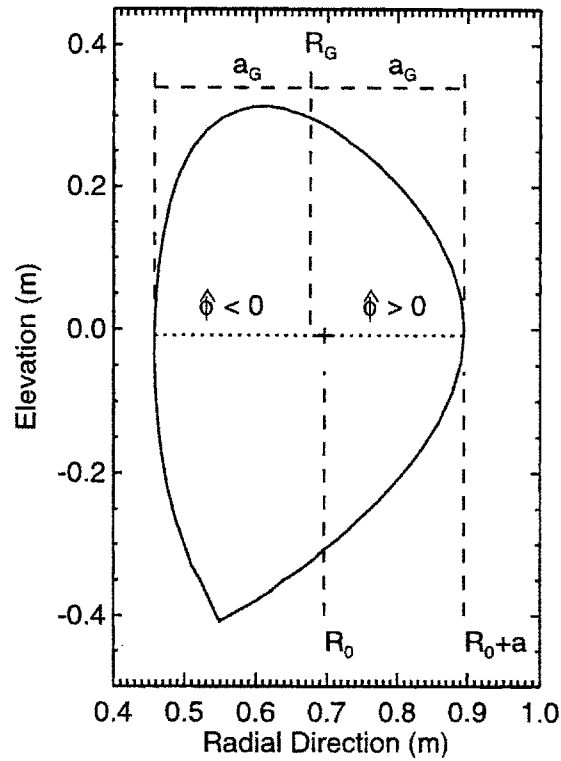


Figure A.6: Coordinate system shown on typical C-Mod plasma. R_0 is the location of the magnetic axis, which is shifted outboard from the geometric center of the plasma at R_G . The distance along the outboard midplane from the magnetic axis to the boundary is a ; this is in contrast to the geometric minor radius a_G .

where R is the major radial coordinate of a point on the profile, R_0 is the major radial location of the magnetic axis, and a is the minor radius defined as the distance from R_0 to the boundary along the outboard midplane (see figure A.6).

Although TRDAT can in general handle input profiles dimensioned as $f(r, t)$ or $f(t, r)$, PRETRANSP and the other TRANSP utilities at C-Mod are currently configured to expect the input data as $f(t, r)$, that is the first dimension is time, and the second dimension is position.

The option to input profile data versus normalized poloidal flux $\hat{\psi}$ is new. Since this is the natural coordinate from EFIT, and many of the C-Mod profiles are mapped to the $\hat{\psi}$ grid as stored in the MDSplus trees, it would be preferable to change over to this coordinate system for the inputs, though significant programming alterations in PRETRANSP and in the TDI functions would be required.

NRI value	Coordinate system	Start	Finish
1	Outboard Major Radial	R_G	$R_G + a_G$
2	Inboard Major Radial	$R_G - a_g$	R_G
3	Entire Major Radial	$R_G - a_G$	$R_G + a_G$
4	Minor Radial	R_0	$R_0 + a$
5	$\hat{\phi}$	0	1
6	$\hat{\psi}$	0	1

Table A.2: Input Coordinate Systems. R_0 , R_G , a , and a_G are defined in figure A.6. $\hat{\phi}$ is the normalized square root toroidal flux defined in (A.1). $\hat{\psi}$ is the normalized poloidal flux (eg. from EFIT). “NRI value” refers to the namelist switch NRIxxx — see § A.4.4.

Outputs

All of the profile outputs have one of four possible signals listed in table A.3 as their position dimensions.¹⁵ Actually, there are a handful of signals that have a position dimension not listed in table A.3. See § B.12 for more information. One of these four signals will be the result of a `DIM_OF(node,0)` mds\$value call. `DIM_OF(node,1)` returns the time dimension of the signal, which is also stored in `\TIME2D`.¹⁶

The nodes under the `MAPPING` structure, listed in table A.4, store conversions from the $\hat{\phi}$ output vectors to major radius R , the location of the magnetic axis, and the minor radius.

Several TDI functions are provided to assist users as well. See the document in § B.10 for more information.

A.4 Running TRANSP

This section describes the interface program `PRETRANSP`, designed to prepare `TRANSP` runs, and provides some information on how to run `TRANSP`.

In summary, `PRETRANSP` is used to create new runs as `TRANSP` tree shots,¹⁷ to select sources for each of the inputs, to process the sources into inputs, to prepare the namelist, and to submit the run, all using an X-windows based interface. The main screen of the interface, shown in figure A.7, explicitly shows the tasks that `PRETRANSP` performs.

¹⁵A

¹⁶The ordering of the dimensions for inputs and outputs is therefore different. Inputs are given as $\mathcal{I}(t,r)$ while outputs are $\mathcal{O}(r,t)$.

¹⁷Henceforth “runs” and “shots” will be used interchangeably.

Signal	Description
X	$\hat{\phi}$ values of the flux zone centers: 0.025 to 0.975 in 0.05 increments. Time independent.
XB	$\hat{\phi}$ values of the flux zone boundaries: 0.05 to 1.00 in 0.05 increments. Time independent.
RMJSYM	The major radial locations (in cm) of the $\hat{\phi}$ surfaces, ie. the mapping from $\hat{\phi}$ to R . Time dependent. DIM_OF(\RMJSYM ,0) = XIRSYM.
RMAJM	A subset of RMJSYM - the major radial locations of the flux zone boundaries $\hat{\phi}$ from -1.00 to 1.00. Time dependent.

Table A.3: Output Coordinate Systems. These four output signals are the list of possible output coordinate systems. Every output profile has one of these signals as its position dimension.

Node	Contents
X_TO_R	$R(X)$, ie. $R(\hat{\phi} = 0.025, 0.075, \dots, 0.975)$
XB_TO_R	$R(XB)$, ie. $R(\hat{\phi} = 0.05, 0.10, \dots, 1.00)$
R_VS_PHI	same as RMJSYM, ie. $R(\hat{\phi} = -1.05, -1.025, \dots, 1.05)$
OUT_RX	TRANSP's magnetic axis R_0 vs. time (cm)
OUT_AX	TRANSP's $a = R_{bdry} - R_0$ vs time (cm)
IN_RX	EFIT's R_0 vs. time (same as \IN:POS)
IN_AX	EFIT's $a = R_{bdry} - R_0$ vs. time (same as \IN:RMN)
EFIT_PHI	EFIT's calculation of R vs. $\hat{\phi}$ at the TRANSP output times.

Table A.4: Mapping Nodes

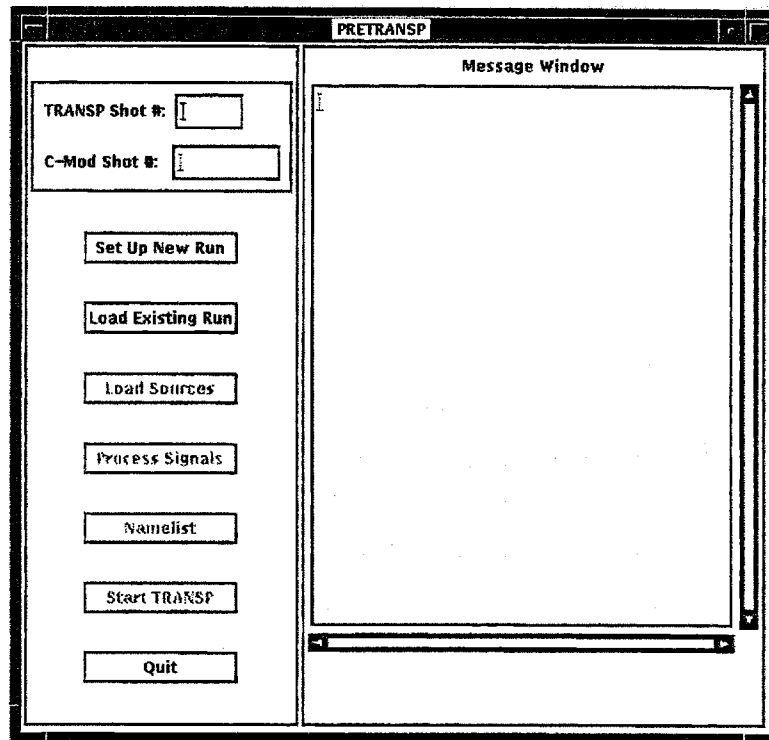


Figure A.7: PRETRANSP main screen

A.4.1 Choosing the TRANSP Run

The first operation is to select a run with which to work. A new run can be created using “Set Up New Run”, or a previously created run can be reloaded. Every run, identified by a four digit TRANSP tree shot number, is associated with a C-Mod shot; this association is assigned when the run is first created and can never be changed. Users also have the option of defining the 6 character field `RUN_TYPE` as a brief classification of the run, and to input comments on the run. When creating a new run, users have the option of duplicating, or “cloning” a previously created run. This function saves time when it is desired to rerun a case with only minor changes to either the inputs or the namelist. A valid C-Mod shot must be input before a run can be duplicated. A user may only edit the runs that he created, though other users’ runs may be duplicated.

After a run is selected, PRETRANSP allows three possible actions: “Load Sources”, “Process Signals”, and “Namelist”. The usual course for newly created, non-duplicated, runs is to do “Load Sources”, so this operation will be discussed next.

A.4.2 Loading Sources

EFIT Data

PRETRANSP first loads the source signals which come from EFIT and are never expected to change. These are: the major and minor radii, POS and RMN respectively¹⁸; the toroidal field, input as $RBZ = R \times B_\phi$; the surface voltage VSF; and the poloidal elevation of the midplane ZPL. The boundary information, and the MAPPING node EFIT_PHI, are also loaded.

Boundary

After the EFIT signals are loaded, the user is prompted for the number of moments to use in SCRUNCHER's fourier decomposition. Historically, since TRANSP was only able to handle up-down symmetric representations of the boundary shape, four moments is the default number. An up-down symmetric fourier decomposition of a C-Mod boundary shape including an x-point would often result in unphysical moments, producing loops as shown in figure A.8. The more moments in the expansion, the worse the problem became. Now that TRANSP can handle up-down asymmetric plasmas, the loop problem is no longer an issue, and using more moments generally does a better match to the boundary shape, as shown upon close inspection of figure A.9. While 12 moments do a better job of matching the shape overall, the difference between four and 12 is small, so using four moments even for up-down asymmetric decompositions should be sufficient.

Other Nodes

With the EFIT signals and the boundary moments loaded, the user proceeds to load sources for the remaining inputs of interest. The vast majority of runs only use a handful of the possible inputs, listed in table A.5. Seven of these inputs, central to the calculations TRANSP performs, *must* be supplied.¹⁹ Beyond the mandatory inputs, several optional ones are used as well. The radiated power BOL is supplied for electron power balance, either as a 1D or 2D signal. Some means of determining the ion temperature is also necessary, either a profile (TI2), or feedback data (NTX or TIT).²⁰ EFIT's calculation of $\ell_i/2 + \beta$, given as L2B, can be used as an initial condition for the poloidal field diffusion equation. For RF runs, the ICRF input power RFP is needed, and the sawtooth model, with sawtooth times in SAW, is usually turned on to broaden the RF power deposition profile.

¹⁸These are input for comparison only; TRANSP uses the boundary surface and its own reconstruction to get the major and minor radius.

¹⁹Some of these quantities can be input as time-independent via the namelist, but this is highly unusual.

²⁰See § 3.6 for details on Ti.

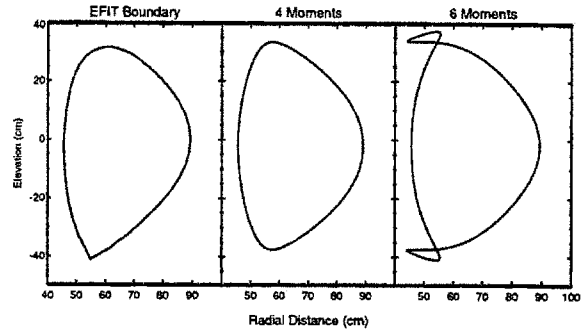


Figure A.8: Symmetric Fourier decomposition

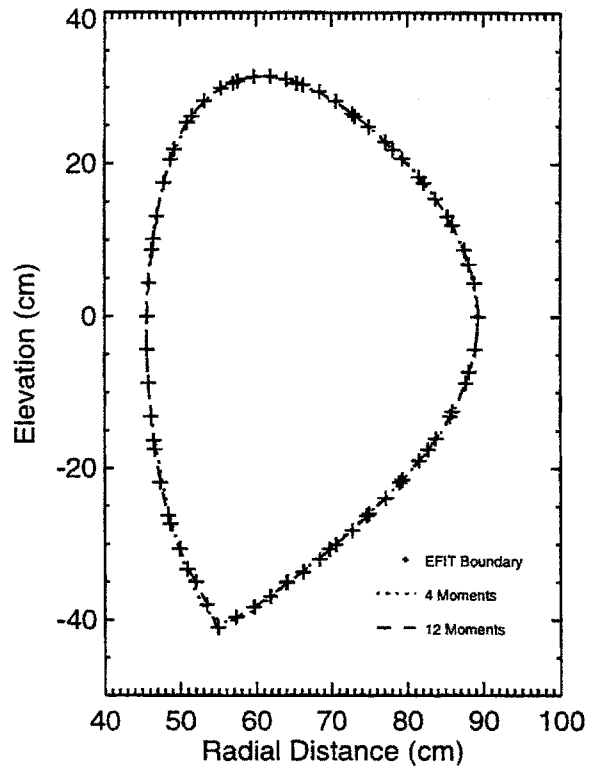


Figure A.9: Asymmetric Fourier decomposition

Input type	Node	Plasma parameter	Usual C-Mod source
Mandatory	CUR	plasma current	\MAGNETICS::IP
	MRY †	boundary moments	EFIT RBBBS, ZBBBS, NBBBS
	NER	electron density	\ELECTRONS::NE_EFIT
	RBZ *	$R \times$ toroidal field	EFIT BTAXV×RMAGX
	TER	electron temperature	Michelson, GPC, or YAG
	VSF *	surface voltage	$2\pi \frac{d}{dt} (\backslashANALYSIS::EFIT_SSIBRY)$
	ZEF	Z effective	\SPECTROSCOPY::Z_67
Optional	BOL	radiated power	Bolometer MAIN_POWER
	RFP	ICRF power	\RF::RF_POWER_NET
	SAW	sawtooth event times	from GPC data
	NTX	neutron rate	Neutrons NEUT_RATE
	TI2	ion temperature profile	from Hirex or Te profile
	TIT	central ion temperature	from Hirex or Neutrons ION_TEMP
	L2B	$\ell_i/2 + \beta$	EFIT ALI/2 + BETAP
Comparison	POS *	Major Radius	\EFIT_AEQDSK:RMAGX
	RMN *	Minor Radius	EFIT ROUT+AOUT-RMAGX
	ZPL *	Poloidal Elevation	\EFIT_AEQDSK:ZMAGX

†from SCRUNCHER. *automatic from EFIT.

Table A.5: Usual TRANSP Inputs

Default source procedures, suitable for most runs, are available for the inputs listed in table A.5. If other source procedures are desired, the document in § B.9 describes how to construct them for optimal use with PRETRANSP. Recall from § A.3.7 that all inputs must be specified on a fixed, normalized minor radial grid. Users may save their own default source procedures as well; see § B.17.

As mentioned in § A.3.1, inputs not relevant to C-Mod are not in the tree, and therefore are not presented as options to the user.

Supplying a source signal for a particular input is not sufficient to tell TRANSP to use it. The TRDAT section of the namelist must include the PRExxx and EXTxxx switches for the input, and in some cases other switches must be set as well to cause the data to be read. For example, NLNTX = .TRUE. must be set to read the neutron data NTX. Automatic setting of these switches based on the source selection is a future improvement for PRETRANSP.

A.4.3 Processing Inputs

Source signals are loaded into their corresponding input nodes during the next step in run preparation: input processing. PRETRANSP can perform five processing operations: smoothing, truncation, resampling, dehollowing, and mathematical functions. Processing is only possible on signals whose sources have been loaded first; see § A.4.2 for more information. Before discussing the processing operations in detail, the overall structure of the processing widget will be presented.

Processing Widget

The processing widget consists of a plotting area, pulldown menus for display and processing options, and “quit” and “done” buttons, as shown in figure A.10. The plotting area is divided into left and right regions. Pressing the “done” button writes the signal displayed in the left panel to the input node. For processing operations, data is displayed in a “before and after” mode, with the initial signal on the left, and the result of processing that signal on the right. After each operation is performed, the user is prompted to select the signal with which to continue before subsequent operations are allowed.

Several options are provided for displaying two-dimensional signals. Either the whole signal can be viewed at once in a surface or contour plot, or the signal can be “sliced” to give a time history at a given radius, or a profile at a given time. It is also possible to display the signal using PSLICE.²¹ One dimensional signals, or slices of two dimensional signals, can be plotted in “side-by-side” mode as described above, or in overlay mode. In the latter mode, it is possible to use the mouse to zoom in on a portion of the plot; ranges must be set explicitly in the former mode.

²¹PSLICE is an IDL procedure written by the author for 2D signal display. See IDL HELP for more information.

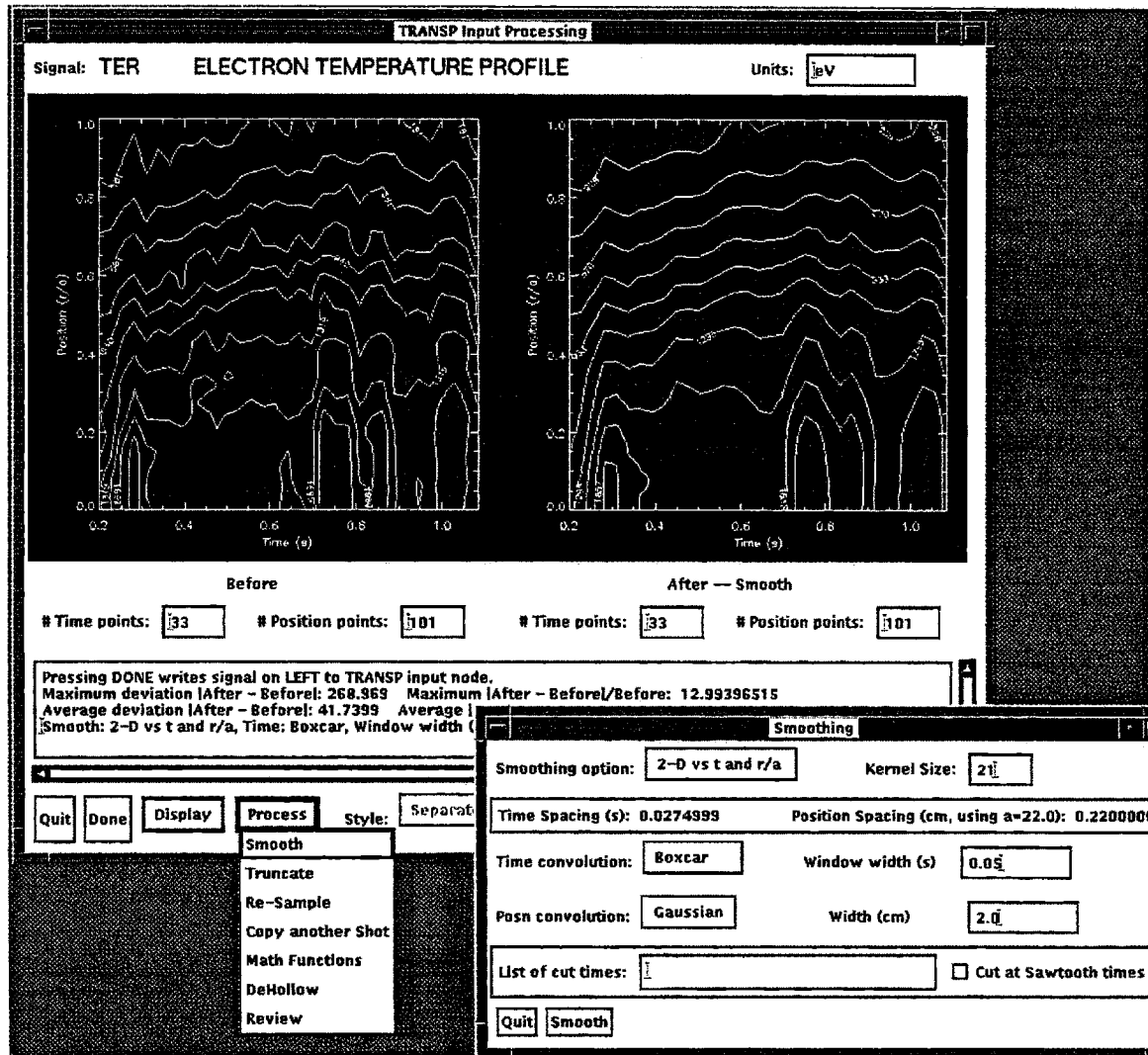


Figure A.10: Processing Widget

If a signal that has already been processed and saved in its input node is reexamined in the processing widget, the user is first presented with the option of continuing with the current input signal, or restarting processing on the original source signal.

Smoothing

Smoothing is the most important operation performed on input data, and the least quantifiable. Determining how much smoothing to apply is more of an “artistic judgement” on behalf of the user rather than a deductive method. Noise must be eliminated while preserving interesting and important features in the data. In PRETRANSP, smoothing is performed using a convolution method. The input data is convolved with a kernel of a particular shape to obtain a smoothed signal (using the IDL function CONVOL).

The kernel can be any shape in general, though it must have an equal number of grid points in both dimensions. Three shapes are currently included in PRETRANSP. Smoothing in the position dimension can be with either a traditional “boxcar” kernel, which uses equal weighting for the points on either side of a point in question, or a gaussian kernel, which weights the points with a gaussian. Temporal smoothing can be with either a boxcar kernel, or with an “RC filter”. The latter only uses the portion of the signal earlier in time to the point in question, and is therefore in some sense causal. The results of using these two kernels on a test “square-wave” signal are shown in figure A.11. Notice that the boxcar kernel has the smoothed signal

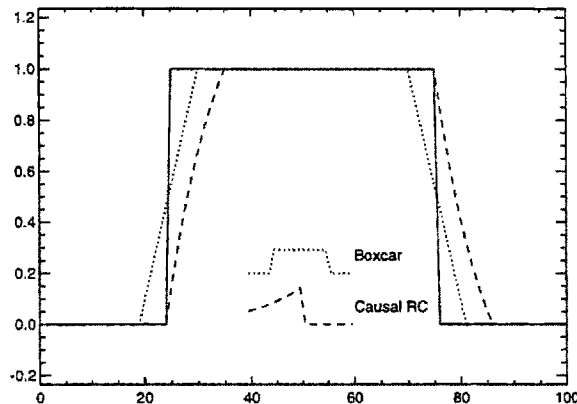


Figure A.11: Comparison of smoothing kernels

rising earlier than when the step occurs, while the RC kernel does not. Which of the two kernels to use depends on the specific application. In circumstances where only random noise is being smoothed, the acausality of the boxcar kernel should not be important. When rapid transitions larger than the background noise occur, an RC filter may be more appropriate.²²

²²PRETRANSP provides another way of dealing with rapid transitions. See page 241.

Since the convolution is a matrix operation, it is implicitly assumed that the input signal is on a regular grid, which is not always the case for C-Mod data. Digitization rates are sometimes adjusted during a shot to “zoom-in” on a portion of the shot, or certain times may be lost due to errors in the data. Input data is therefore interpolated onto a fixed grid, with a spacing equal to the minimum spacing in the original signal, before any smoothing occurs, and the result is interpolated back onto the grid provided with the input. The position dimension of 2D signals is by default a fixed, regular grid, so the interpolation need only be done in time.

PRETRANSP asks for the smoothing widths in physical units: seconds and/or cm. These are converted to a matrix size for the kernel by dividing by the spacing of the fixed grid of the input signal. For a symmetric kernel (boxcar or Gaussian), the half-width is set to the desired smoothing width. This means for example that smoothing with a time window of Δt averages all points within Δt of a given point at t_0 , that is points with $|t - t_0| \leq \Delta t$. Note that input profiles are on an r/a grid. Rather than inputting the smoothing range in r/a , the range in cm is converted to r/a by dividing by 22.0 cm.²³ While not exact, most C-Mod plasmas have a minor radius close to this value.

Both the smoothing widths specified by the user and the kernel size determine the amount of smoothing applied. For example, assume a 1D signal with time spacing $dt = 0.001$ s, a smoothing width $\ell_w = 0.05$ s, and a kernel size $k = 21$ points. The kernel covers a range $\Delta t = k \times \ell_w = 0.021$ s, less than the requested smoothing width. To obtain $\Delta t = 0.05$ s in this case, the user would have to increase k to at least 51 points. The maximum size of the kernel is limited by the size of the input signal. For example, in smoothing a 2D signal with dimensions (m, n) , the kernel size must satisfy $k < \text{Min}(m, n)$. Of course, in cases where $m, n \gg 1$, smoothing with a large kernel could require lots of CPU time. PRETRANSP will determine a default kernel size if none is specified explicitly.

Cut Times In cases where discontinuous events in the data are to be preserved during smoothing, PRETRANSP allows “cut times” to be specified. For each cut time, the signal will be smoothed in two segments: up to but not including the cut time, and past the cut time. Figure A.12 shows the difference between smoothing through a discontinuity and using a cut time. This feature is most often used when the ICRF power turns on or off, when a pellet is fired, or when it is desired to preserve the sawteeth in the data. Manually inputting cut times for sawteeth can be tedious, and is unnecessary since the sawtooth times are already supplied in `\IN:SAW`. As long as the sawtooth times are input, the user can specify cut times at each sawtooth event with one button push.

²³This value is hard-coded into the procedure `w.sig.pro`.

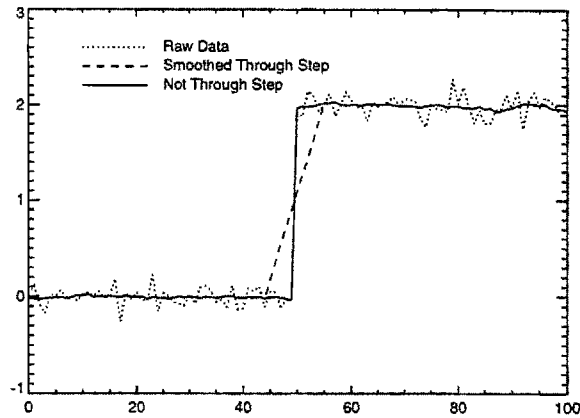


Figure A.12: Smoothing and cut times

Resampling

Resampling is most often performed when it is desired to reduce the number of time samples in a signal. Some signals, for example T_e from the GPC, can be larger than necessary for TRANSP analysis. To resample a signal, the user specifies the desired time spacing for the signal. The signal is resampled by the following method. For a given point at time t_0 and a resampling time of Δt , points with $t - t_0 < \Delta t$ are dropped. The first point with $t - t_0 > \Delta t$ is kept, and the process repeated. Note that this does not guarantee a fixed time spacing, or that the minimum time spacing will be Δt .

It is also possible to resample over a finer time grid. In this case, the starting and ending times remain the same, and the signal is interpolated onto a regular grid with the specified spacing.

Truncation

Some signals cover too wide a range in time. For example, the plasma current signal `\MAGNETICS::IP` covers a time range from -2 seconds to nearly +8 seconds, whereas the plasma only exists from 0 to typically $1 \rightarrow 1.5$ seconds. To eliminate the extraneous data, the signal is truncated. Other signals that are commonly truncated include the ICRF power `\IN:RFP`, and the density and temperature profiles (which are sometimes bogus during rampup or rampdown). It is also possible to truncate the position range of a 2D signal, though this shouldn't be necessary if the default position grid is used.

Math Functions

A math operation is performed by composing a valid IDL command string, which is then executed using the `execute()` function. The signal data is defined as an IDL

variable s , its time dimension as variable t , and its position dimension (for 2D signals) as variable rx . The command is set to start with “ $s =$ ”, and the user specifies the rest. For example, to add 10% to the signal, the user would enter $s * 1.1$, so that the command executed would be $s = s \times 1.1$. *Any* IDL command can be invoked this way, even multiple commands can be run using the $\&$ separator. This makes the math function operation very powerful, but also very dangerous. A masochistic user could enter $s \& \text{exit}$ or some other destructive command.

Dehollowing

This operation makes a profile monotonic, using the IDL operator $>$. It was included to avoid problems with negative χ 's, though it doesn't always provide a solution.

Recording Steps

All processing steps are recorded, in the order performed, in the PROCESSING node underneath each input node. The format for describing each step is set so that the steps can be read in and re-executed if necessary. This also allows users to save default processing steps for each input in their copy of the TRANSP model (see § B.17), and to copy the steps used in other shots.

The option “review” allows the user to examine the steps performed and to set defaults. A future improvement for PRETRANSP would be to allow individual steps to be “undone”. Currently, if a step performed earlier and accepted is subsequently determined to be unacceptable, the only recourse is to start over with the source signal.

A.4.4 Namelist

This section covers how to edit the TRANSP namelist, and the commonly used settings for C-Mod runs. More information on the physics models controlled by the namelist is in chapter 3, while details on how the namelist is integrated with MDSplus are given in § A.3.3. Comprehensive information on the namelist can be found in the online TRANSP documentation accessed with the DCL command

```
HELP TRANSP OPERATIONS NAMELIST.
```

This command is subsequently referred to in the text as “TRANSP HELP”.²⁴

Editing as a File

Before XTranspin was released, the only way to edit the namelist was as a text file in a text widget or editor. This option is still available for users unfamiliar with XTranspin,

²⁴Namelist info is also at the URL http://w3.pppl.gov/tftr/helpdocs/transp/transp_hlp.html.

and to maintain the comments in the file added to assist the user. Eventually, with XTranspin working and properly tested, text editing will be phased out.

In the meantime, some information on how to use the namelist editor with the namelist file is prudent. The namelist file is actually two separate namelists: the TRANSP namelist and the TRDAT namelist. The latter is at the end of the file, and is delimited by the \$TRDATA line. The namelists are broken up into sections dealing with each of the major namelist areas. Reference to the appropriate TRANSP HELP command is given at the head of each section.

When the "namelist" button on the PRETRANSP main menu is pressed, the namelist is read and displayed in a text widget, allowing rudimentary editing capability as well as access to the utilities designed to aid in preparation. Individual switches can be edited manually. Comments are marked with a ! character. If a switch is commented out, it will assume its default value set by the TRANSP modules PRESET or DPRSET.

As for the options available from the namelist widget, the user can load a namelist from another shot or from a file on the disk (the latter is not usually done or recommended), save the namelist as their default or as a file, and reset to the last saved version. There is also an option of editing the namelist using their favorite text editor.²⁵ A primitive search facility is built into the widget as well, allowing simple text searches, and the namelist can be compared to other namelists from other shots or in files on the disk. The comparison is via the VMS command DIFFERENCES; either whole namelist files, or namelists parsed to collect only the switch names and values, can be compared. The latter method is preferable if using XTranspin, since it strips all comment lines and reorders the namelist in alphabetical order.

Editing using XTranspin

XTranspin is an X-windows widgets based tool for namelist preparation. From PRETRANSP it is accessed through the namelist tool. There are many advantages to using XTranspin for namelist preparation. All namelist switches are grouped according to function, so features requiring more than one switch to be adjusted are easy to activate (for example: using T_i profiles versus modelling them, including ICRF modules). Furthermore, it is not necessary to remember the names of the individual switches and their corresponding numeric values. For example, the type of MHD equilibrium to use is controlled by the switch LEVGEO. With XTranspin, it is not necessary to remember that a symmetric equilibrium corresponds to LEVGEO=5 while an asymmetric equilibrium corresponds to LEVGEO=6, as these options are presented as text in a pulldown menu, XTranspin also allows the namelist to be tested for consistency before submitting the TRANSP run, reducing the chance that the run will crash.

XTranspin reads the namelist from a temporary file prepared by PRETRANSP from the namelist stored in the text node \TOP:NAME_LIST. When the user is finished

²⁵Right now this is hard-coded to be EVE; changing this should be straightforward.

editing the namelist using XTranspin, it should be saved using the “Save” option on the “File” menu. The user should *not* change the filename from XTranspin. After the temporary file is saved, PRETRANSP reads it back in and presents it in the namelist tool. It is not written back into the tree until the user selects “Exit, Save to Node”. To use the consistency check feature in XTranspin, save the namelist first, then select “Check Data” from the “File” menu before exiting.

Once XTranspin is used to edit a namelist, it must be used for all subsequent access to that namelist in order to understand and translate all the namelist settings, as it strips all the comments providing this information in the text-based version of the namelist.

As XTranspin continues to mature, it will be more thoroughly integrated with TRANSP. Users are encouraged to explore its features now, as it will eventually become the standard namelist tool. As a precaution it is suggested that the user keep in their own file area a copy of a text-based namelist which contains all the comment information, in case it is ever necessary to edit a namelist stripped by XTranspin without access to it.

Common Settings

This section provides a summary of the switches usually adjusted for TRANSP runs on Alcator C-Mod. It should be used in conjunction with the online help: \$HELP TRANSP OPERATIONS NAMELIST. They are presented in the order that they appear in the text-based version of the namelist. It should be straightforward to apply this information to the namelist as organized by XTranspin.

Input Data The signals that TRANSP actually reads as inputs are set in the TR-DAT namelist at the end of the file. MDSplus users should have KMDSPLUS=1. For the input signal represented by the trigraph “XYZ” to be read by TRANSP, it is necessary (but sometimes not sufficient) to set extXYZ='XYZ' and preXYZ='x', where x is one letter A-Z, but can be any letter. These must be set manually for now; a planned future modification for PRETRANSP is to set them automatically based on the source selection stage. Typically, the only time the user would need to adjust these is if T_i is input, or if the RF model is desired.

Time Limits Set TINIT and FTIME to be the start and stop time for the run, based on the shot's timebase. Ensure that the input data covers this range; otherwise TRANSP will extrapolate the data.

Plasma Composition The plasma majority species is set by APLASM and BACKZ — for deuterium these are 2 and 1 respectively. If there is more than one background species, these can be set to an array. AIMP and XZIMP refer to the primary impurity

species responsible for Z_{eff} . For high density shots, this is typically between carbon and oxygen. For low density shots, it is roughly Molybdenum³²⁺.

Initial Conditions These switches are for the particle balance calculation, and are only used when there is more than one background plasma species.

Z Effective See § 3.3 for information on the physics models here. Typically the central Z_{eff} value is used for the plasma composition Z_{eff} (NLZFIN=.true. and NLZFI2=.false.). The switch NLZFIM should always be false for the magnetic Z_{eff} .

Magnetics See § 3.3. Neoclassical resistivity is selected with NLSPIZ=.false, Spitzer with NLSPIZ=.true. If using neoclassical resistivity, the bootstrap current model is usually turned on as well with NLBOOT=.true. To fix V_{surf} and have the magnetic Z_{eff} calculated, set NLVSUR=.true. If NLVSUR=.false, Z_{eff} is set to be the plasma composition Z_{eff} and V_{surf} is calculated. Initial conditions for the magnetic field diffusion equation are typically given by NEFLD=3. If NEFLD=4 is used, the $\ell_i/2 + \beta_p$ data must be read in as well: NLI2PB=.true and set preL2B,extL2B in the TRDAT namelist.

Sawteeth To turn the sawtooth model on, set NLSAW=.true. It is typically applied only to the ICRF fast ions (NLSAWIC=.true). DTSAWD must be set smaller than the sawtooth period. XSWID1 and XSWID2 are typically both 0. The TRDAT switches preSAW,extSAW must also be set.

Particle Balance The particle confinement time that determines the source strength is set by TAUPH. See § 3.2 for more information. The remaining options are only used if more than one species is present. These are too complicated to explain here; see TRANSP HELP for more information.

Neutrals Model FRANTIC is typically used (NSOMOD=1), and the recombination neutrals are used in the calculations (NLRECO=.true). See TRANSP HELP for more information.

Convection Both ALPHOE and ALPHOI are set to 0.6, giving $\frac{3}{2}nTu$ as the convection term in energy balance (see § 3.1).

Ion Temperature and Conductivity Here the choice is made to use input T_i data, or model T_i using the methods described in § 3.6.2. There are several switches of importance here; table A.6 summarizes their settings. In addition, the TRDAT switches must be set to read the T_i profile (preTI2, extTI2), T_{i0} (preTIT, extTIT) and the neutron rate (preNTX, extNTX) as appropriate.

Switch	To use measured T_i	To model T_i
NLTIPRO	true	false
FIEFAC	must be 1.0	does not matter
NLTI2	true	false
NLFXKF	false	true
NLXKIE	false	false= $\chi_i \propto \chi_{i,nc}$ true= $\chi_i \propto \chi_e$
NLTIFO	false	true = feedback on T_{i0}
NLTNTX	false	true = feedback on R_{DD}
NLNTX	always true if there is neutron data	
NRITI2	-4	comment out
NSYTI2	0	comment out

Table A.6: Namelist settings for ion temperature control.

Minimum Values These are typically not adjusted

RPLOT Output and Restart Records These are also typically not adjusted. For simplicity, the timebases of the 1D outputs (STEDIT) and the 2D outputs (SEDT) should be the same. Do not change MRSTRT!

Input Interpolation These are also not usually adjusted. TGRID1=TGRID2=0.005.

ICRF See § 3.5. To turn the ICRF heating models (SPRUCE/FPPRF) on, set NLICRF=.true and set the TRDAT switches to read the power (preRFP, extrRFP). Also set the minority species with AMINI and XZMINI — this will either be (1,1) for H minority, or (3,2) for ^3He . FRMINI is the minority fraction. TCRFON and TCRFOF control the times at which the RF model is turned on and off. TCRFON must be greater than TINIT. Optimal settings for the antenna and vacuum vessel are: SEPICHA=25.6, VVRMOM=64.5, 25.0, 3.25 and VVZMOM=0.0, 57.3, -3.25. The frequency is FRQICHA=80e6 for 80 MHz. For second harmonic minority heating, set NHARMICH=2. If the ICRF models are not desired, set NLICRF=.false and *comment out* FRMINI and the TRDAT switches preRFP, extrRFP.

MHD Geometry See § A.4.2. LEVGEO=5 is for symmetric decompositions (required for old SPRUCE), LEVGEO=6 is for asymmetric decompositions. The former needs the TRDAT switches preRMO, extrRMO, preRMM, extrRMM, and preYMM, extYMM set; the latter requires only preMRY, extMRY.

Pellets Please see TRANSP HELP. Examples can be found in C-Mod shot 950609013, TRANSP runs TRANSP06 or TRANSP07.

Profile Symmetrization See § A.3.7. All profile inputs XYZ should have `nriXYZ=-4` and `nsyXYZ=0`.

Write UFILES By setting `SCLREQ(1)='$UFCOPY'`, TRDAT will write out the input data read from the tree as a set of UFILES. This is useful for sending data to PPPL or elsewhere. The shot number of the UFILES (see § B.15) is controlled by NSHOT.

A.4.5 Submitting the Run

With all the inputs and the namelist prepared, the user proceeds to submit the run to a batch queue, as `TRANSP$: [USER]TRANSP.COM`, for analysis. `TRANSP.COM` runs the programs outlined in § A.2. It requires as input the TRANSP shot number of the run, which is passed as a parameter in the submit command. Several temporary files are generated to be used as `sys$input` for the TRANSP programs. After the last stage in the TRANSP analysis, MDSPLOT, the command procedure cleans up the TRANSP temporary files, archives the RPLOT files, cleans the tree, and sets the run status flag `RUN_FLAG`.

Submission is only allowed when the namelist has been saved, and when all the nodes with specified sources have been processed. If some nodes do not require processing, they may be loaded automatically with the “load remaining” button. Nodes may also be loaded with default processing; however, without examining the inputs unpredictable results may be obtained.

Users are encouraged to submit their jobs after regular hours. 7 pm. is the default time. Also, users are encouraged to submit their runs on the computers designated for their use, though after hours it is acceptable to run on any computer.²⁶

Currently there is no easy method of determining the status of the run while it is in progress. The log file can be searched for the string `GFRAME` to list the time steps that have been completed, but the user must remember or look up the finish time for the run `FTIME` in order to know what fraction of the run has been completed.

Runs sometimes crash, so `TRANSP.COM` is designed to update the status of the run to “not run” if there is an error. Users must check their runs manually - no notice is supplied of whether the run completed successfully or whether it crashed. A large fraction of the crashes occur at the TRDAT stage, either because the namelist has inconsistent or incorrect settings, or because data required for analysis has not been supplied. If the run crashes for these reasons, the user must check the log file for the error, correct it, and resubmit the run. Using the “check data” feature in XTranspin should prevent many namelist-related crashes.

²⁶See Martin Greenwald about which computers are acceptable.

A.5 Examining Outputs

This section describes how to view TRANSP outputs, in particular with the dedicated tools constructed.

As explained in § A.3.2, the outputs are all written to the nodes under `\TRANSP_OUT`, but users access the data through the nodes under `\OUTPUTS`. The latter nodes are divided between 1D outputs under `\ONE_D` and 2D outputs under `\TWO_D`. Every output has a unique name, so to save typing these tags, an output `xxxx` can be referenced directly as `\xxxx`, though the dimension of the signal is then not obvious from the node name. Remembering the translations of the RPLOT output names used for the node names can be difficult, so the `NODE SEARCH` tool in `POSTRANSF` is available to assist users. It searches the node names, and the contents of the nodes' `LABEL` and `COMMENT` subnodes for a specified string. See the document in § B.6 for more information.

The coordinate systems for the outputs are described in § A.3.7. All 2D output signals have position as their 0th dimension and time as their 1st dimension. In general, the timebases of the 1D and 2D signals are identical, though it is possible to set the namelist so that they are different. The 1D timebase is in the node `\TIME1D` and the 2D timebase is in `\TIME2D`.

The usual methods for accessing MDSplus data — the language interfaces for IDL, FORTRAN, and C, and `Scopes` — can be used to view TRANSP data. Scope pads for all the inputs and outputs are available in `CMOD$MODELS:[TRANSPORT.SCOPE]` to assist with scope use. These are updated as part of the tree update procedure described in the document in § B.16. To be viewed as traces in a scope, 2D signals must be “sliced”, using the TDI functions `TSLICE` and `RSLICE`. Several other TDI functions are available for TRANSP outputs. See the document in § B.10 for more information.

A.5.1 Multigraph

Often it is desired to view several TRANSP outputs on the same graph. `RPLOT` provides so-called “multigraphs” for such a purpose. At C-Mod, the tool `Multigraph` (or `MG`) was written in IDL using modern X-windows technology for the same purpose. `MG` is designed to plot multiple MDSplus signals with complete user control over the appearance of the plot, and can perform mathematical operations on the data. While it can plot *any* MDSplus signal, full realization of its potential can only be had with TRANSP data. Furthermore, `MG` is customizable, while `RPLOT`'s multigraph's are hardcoded (in `XPLOT7.FOR`). Figure A.13 shows its main screen with one of its control menus activated.

`MG` is designed to make 1D plots. 2D signal slicing is controlled in the menu “Data Functions”. Up to 20 signals may be plotted simultaneously, using any combination of six line types, eight symbols, and eight colors. Signals can be averaged, stacked,

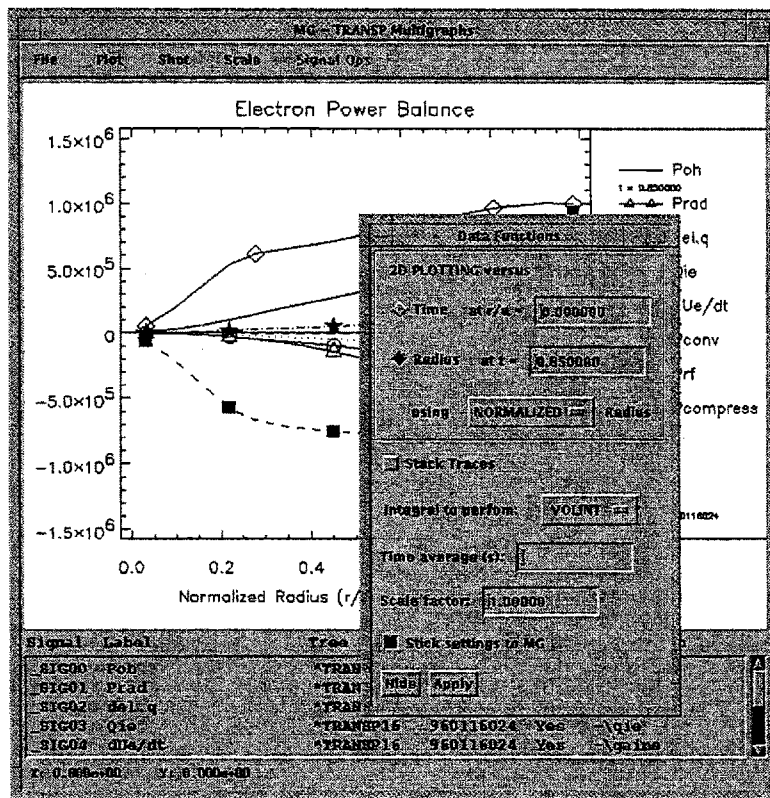


Figure A.13: Multigraph program

and scaled. On 2D output data, MG can perform volume or surface integrals, volume averages, and derivatives as well. The 2D data viewing utility PSLICE can be invoked with any of the 2D signals in a multigraph. “Variable signals” can be defined, which for example make comparing data between shots easier. Two signals pointing to the same variable expression but with different shot numbers can be entered, and then only one change is necessary to switch the signal being compared.

The “Plot Style” menu controls the plot title, axes labels and tickmarks, and the character and symbol sizes. A legend can also be placed on the plot, with optional automatic display of the TRANSP run number and slice value. Several of the scope features are replicated: mouse zoom, pointer, and the autoscale functions.

Default multigraphs are stored in `CMOD$MODELS:[TRANSPORT.SCOPE]*.MG`. Users may store their own multigraphs in their own file areas.

MG is versatile enough to make everything from quick plots for routine analysis to presentation quality plots for viewgraphs, or even theses. For a concise description of its functions, see the document in § B.5.

A.5.2 Other Tools

Several tools not specific to TRANSP or MDSplus have been constructed to assist in viewing data. Two are described here.

The procedure PSLICE is used to display 2D data in an intuitive manner. The user is presented with contour and surface plots, and two 1D “slice” plots of the signal versus each of its dimensions, as shown in figure A.14. The slice plots displayed can be changed by moving crosshairs over the contour plot, or over the slice plots themselves. If the procedure is called with no arguments, it will prompt the user for an MDSplus expression which should return a 2D signal. The procedure can also be called with either one or three arguments. The first is always the 2D data, while the remaining two arguments are the dimensions for the data. If no dimensions are supplied, the data is plotted versus index number. PSLICE is part of the MDSplus IDL function library, stored in `MDS$ROOT:[IDL]` (though the reference copy is kept in `CMOD$MODELS:[TRANSPORT.IDL.CMSREF]`). Help on it should be available through `MAN_PROC`. Additional information is also in § B.8

The program GP was written to provide a means of organizing the plethora of small IDL plotting procedures that invariably accumulates during analysis. It allows the small procedures to be grouped into two files: one to store the means of loading the data to be plotted, and one to store the plot procedures. This separation allows the user to edit the appearance of the plot without having to reload the data every time. Keeping all the separate plotting procedures together in one file makes it easier to keep track of useful plots. For more information on GP, see the document in § B.2.

A.5.3 Scalar Database

Local analysis scalar parameters can be stored in the `TRANSP_SCALARS` table of `LOGBOOK`, in analogy to the `CORE` table for global analysis scalars. When runs are pasted, the user has the option of flagging them for later extraction into the table. A tool called `TS` has been written to facilitate the extraction. For a given run, it extracts `TRANSP` outputs into the table, with an option to first smooth the data. Since the database tables are not optimally designed to store an entire profile, the values of the profile data at three radial locations are stored: $r/a = 0.25, 0.50,$ and 0.75 . A complete list of the fields in the table can be obtained by executing the SQL command `show table (columns) TRANSP_SCALARS`. More information on `TS` can be found in § B.14.

Appendix B

TRANSP Reference

This appendix contains the online documentation written by the author. It is included so that this thesis can provide a more complete reference. However, readers must be cautioned that the online documents are subject to change. One should always check the date of the online version to see if the document within this section has been superseded.

There are also many documents within the TRANSP distribution that, although referenced by this thesis, are not included here. Pointers to them are provided when necessary.

The table below lists the titles of the documents in this appendix, and their locations online. All documents can be found in `CMOD$MODELS:[TRANSPORT.DOC]` unless otherwise specified.

Section	Filename	Contents
B.1	DOCLIST.DOC	List of documents
B.2	GP.DOC	GP plotting tool
B.3	MDSPLUS_SETUP.DOC	Details on setup and maintenance
B.4	MDSPLUS_TRANSP.DOC	Overview of MDSplus-TRANSP integration
B.5	MULTIGRAPH.DOC	Multigraph
B.6	POSTRANSP.DOC	POSTRANSP run management
B.7	PRETRANSP.DOC	PRETRANSP
B.8	PSLICE.DOC	PSLICE
B.9	SOURCES.DOC	User-specified sources for PRETRANSP
B.10	TDI.DOC	Useful TDI functions
B.11	TDI_OBSOLETE.DOC	Obsolete TDI functions
B.12	TRANSP_COORDINATES.DOC	Explanation of TRANSP coordinate system
B.13	TRANSP_FILE_AREAS.DOC	List of the file areas in TRANSP system
B.14	TRANSP_SCALARS.DOC	Tool to load TRANSP_SCALARS table
B.15	UFILES_AND_IDL.DOC	UFMDSIO routines to read/write UFILES
B.16	UPDATING_TRANSP.DOC	Keeping TRANSP distribution up-to-date
B.17	USER_DEFAULTS.DOC	Maintaining user defaults for PRETRANSP

Table B.1: Online Documentation

B.1 DOCLIST.DOC

DOCLIST.DOC

97.05.24

=====

Documents relevant to the MDSplus TRANSP manager and users at C-Mod:

TRANSP\$: [CMSREF.DOC]MDSPLUS.DOC

a VERY brief word about the MDSplus integration, which mostly explains how non-MDSplus users are affected

In CMOD\$MODELS:[TRANSPORT.DOCS] :

GP.DOC

Instructions for GP, the general purpose plotting tool

MDSPLUS_SETUP.DOC

Overview on infrastructure necessary for MDSplus-TRANSP use

MDSPLUS_TRANSP.DOC

most of the details of the MDSplus-TRANSP integration

MULTIGRAPH.DOC

information on the MULTIGRAPH data plotting program

POSTRANSF.DOC

information on the POSTRANSF tool, which has a suite of functions used for run management

PRETRANSP.DOC

information on the PRETRANSP data preparation program

PSLICE.DOC

help for the IDL function PSLICE - this should be included in

IDL help

SOURCES.DOC

(referenced by PRETRANSP.DOC) How to specify data sources for
TRANSP inputs

TDI.DOC

TDI functions useful for examining TRANSP inputs and outputs

TDI_OBSOLETE.DOC

Obsolete TDI functions - these either don't work or are no
longer supported. If you use them, switch now!

TRANSP_COORDINATES.DOC

An explanation of the coordinate systems used in TRANSP and how
they relate to MDSplus and PRETRANSP

TRANSP_FILE_AREAS.DOC

File area summary

TRANSP_SCALARS.DOC

how to run the program TS - used to load TRANSP output data into
the TRANSP_SCALARS table of LOGBOOK

UFILES_AND_IDL.DOC

description of UFMDSIO sharable image used to read and write
UFILE data to and from MDSplus and IDL

UPDATING_TRANSP.DOC

how to keep the C-Mod TRANSP distribution up-to-date,
and how to modify it

USER_DEFAULTS.DOC

How to manage your own defaults for processing steps and the namelist. NOTE!!! Given the current state of PRETRANSP it is NOT possible to store these defaults. The system requires that each user keep their own copy of the TRANSP tree model, and has not been sufficiently tested to allow general use.

B.2 GP.DOC

GP.DOC
=====

96.12.07

GP is a tool to help in making quick plots of IDL data. Without GP, users typically have a separate IDL procedure in its own file for each plot, or they have one procedure with many plots and some kind of clumsy menu system that allows them to scroll through the list of plots. Code allowing the plot to be printed as a postscript file must be added to each procedure. Also, the editing and replot cycle can be tedious, requiring lots of `.run` commands.

GP solves these problems. Instead, users have *one* file with many procedures, each generating a different plot. GP provides a widget-based menu allowing selection of the plot to display and print, and provides a standardized print interface. Users can then have one file for each group of plots they need, helping to maintain the context of the plots and making it easier to organize the files. Editing and replotting is easy, as the user can edit the file in a separate window, and compile it on the fly at the touch of a button.

To run GP:

```
$ @CMOD$MODELS:[TRANSPORT.IDL]GP
```

The file containing the plot routines **MUST** be a `.PRO` file, and must be structured in a particular manner. Assume it is called "PLOT_PROC.PRO", then it must look like the following:

```
PRO plot_one
  plot,...
END

PRO plot_two
  plot,...
END

...

PRO plot_n
  plot,...
```

```
END

PRO plot_proc,plot_list
  plot_list=['plot_one', 'plot_two', ... , 'plot_n']
END
```

The last procedure in the file PLOT_PROC.PRO must be called PLOT_PROC, and must have one and only one parameter: plot_list - a string array of the names of the plot procedures. This array is used in the widget menu only - *all* the procedures are compiled.

To start, type the name of the procedure containing the plots, and press enter or hit the "Recompile" button. Then select the plot you wish to display using the droplist. To print the plot, press the print button and use the popup widget to set the parameters.

To edit and recompile, the user should have an editor running in another window editing the file PLOT_PROC.PRO. After writing the file in the editor, the user need only hit "Recompile" to get the latest version of the plot. (The IDL command RESOLVE_ROUTINE is used to do this.) If there are errors in the file that prevent its recompilation, the program will not usually complain (nor will it crash). Check the decterm from which you ran GP for errors if something unexpected happens after recompiling.

Another time-saving feature is included in GP. Sometimes, obtaining the data to be plotted in a procedure can take many CPU seconds. If a user is making cosmetic changes to a plot, it would be time consuming to reobtain the data every time a replot is desired. To solve this, GP allows a separate procedure for the data generation. This means that two files must be maintained instead of one (one for data, one for plots), but this should be a minimal inconvenience compared to all the savings.

Once again, the file has to have a particular structure in order to compile properly. If the file is DATA_PROC.PRO, the *last* procedure in it *must* be called data_proc. All subroutines and functions called must appear *before* DATA_PROC (just as in PLOT_PROC.PRO). Procedures in other files can be called, but the user must ensure that the proper resolve_routine commands are in DATA_PROC.PRO. GP

recompiles this file and calls the procedure DATA_PROC every time the button "Load Data" is pressed.

To share the data between DATA_PROC.PRO and PLOT_PROC.PRO, the user can use handles or common blocks. If handles are used, there must be one common block to share the ID of the main handle. For example, in DATA_PROC.PRO:

```

PRO get_data,data
  mds$open,...
  data=mds$value(...)
  mds$close
END
PRO data_proc
  common data,data
  get_data,data
END

```

and in PLOT_PROC.PRO:

```

PRO plot_one
  common data,data
  plot,data
END
PRO plot_proc,plot_list
  plot_list=['plot_one']
END

```

Users are encouraged to use handles rather than common blocks, as they are more flexible. They would be used as follows.

DATA_PROC.PRO:

```

PRO get_data1,data1
  mds$open,...
  data1=mds$value(...)
  mds$close
END
PRO get_data2,data2
  mds$open,...
  data2=mds$value(...)
  mds$close

```

```

END
PRO data_proc
  common data_handle,data_handle
  data_handle=handle_create()
  get_data1,data1
  get_data2,data2
  handle_value,data_handle,{data1:data1, data2:data2}, /set
END

```

PLOT_PROC.PRO:

```

PRO plot_one
  common data_handle,data_handle
  handle_value,data_handle,data
  plot,data.data1
  oplot,data.data2
END
PRO plot_proc,plot_list
  plot_list=['plot_one']
END

```

There are other elegant solutions to the problem of sharing data between the procedures. Once a user comes up with a favorite, they should use it in every procedure pair for GP.

An alternative to using the "Load Data" to save reloading data every time a replot is done is as follows. This method needs only the file PLOT_PROC.PRO.

In PLOT_PROC.PRO have a procedure get_data. Then call it from the plot_proc procedure only when it is necessary to reload the data. For example:

```

PRO plot_one
  common data,data
  plot,data
END
PRO get_data
  common data,data
  data=mds$value(...)

```



```
END
PRO plot_proc,plot_list
  plot_list=['plot_one']
  get_data ; comment this line out if a
           ; reload is not required.
END
```

B.3 MDSPLUS_SETUP.DOC

MDSPLUS_SETUP.DOC

97.06.27

=====

This document contains an outline of the infrastructure necessary to run TRANSP in the C-Mod MDSplus environment.

File areas and logical names

The document TRANSP_FILE_AREAS.DOC contains a list of the logical names and file areas that need to be allocated. The TRANSP code and binaries, if all the VAX, AXP, and Unix implementations are kept together, need a total of 2GB of disk space. Currently on C-Mod they are all kept on a dedicated disk CMOD2\$DKA100:, which has the logical name DISK\$TRANSP.

The command procedure CMOD\$MODELS:[TRANSPORT]TFREE.COM gives a list of the disk usage and quota if applicable for each TRANSP file area.

Queues

The VMS TRANSP code management software needs queues on one VAX and one AXP node that have access to NAG libraries. Currently these are:

Queue name	Node
-----	-----
TRANSP_VAX	CMOD2
TRANSP_AXP	CMODA

These queues should have their login command procedures defined to be TRANSP\$: [USER]LOGIN_TRANSP.COM.

Tree

The TRANSP tree, TRANSP\$DATA, should be constructed as specified in

appendix A of Jeff's thesis, and in MDSPLUS_TRANSP.DOC.

MDSplus Linking

As explained in MDSPLUS_TRANSP, the MDSplus calls are linked at run time using LIB\$FIND_IMAGE_SYMBOL. This feature is available only in VMS. The MDSplus library referenced by the TRANSP modules is MDSLIB.

Software requirements

*** This information is not complete - contact Doug McCune ***
*** (dmccune@pppl.gov) at PPPL for the latest requirements ***

VMS:

CMS: ???

MMS: ???

NAG libraries: one VAX and one AXP node must have NAG libraries available (see queues above)

FORTRAN: current FORTRAN version is ???

C: ???

TEDI: ???

Unix:

Unix TRANSP on C-Mod is only used as an intermediate step in the update process from PPPL. TRANSP itself is not run on Unix nodes. This lessens the software requirements in Unix to:

cvms: ???

make: ???

diff: ???

TRANSP Development Environment

Any time code work on TRANSP itself is necessary, or when one of the TRANSP binaries is run, the command procedure LOGIN_TRANSP.COM in TRANSP\$:[USER] should first be run to setup the necessary TRANSP logical names. This procedure first clears all user-defined symbols and logical names (to avoid collisions with the TRANSP ones), defines logical names and symbols useful for the local implementation (if new ones are added, they must first be tested against the TRANSP symbols and logicals for conflicts), and then calls the TRANSP login procedure TRANSP\$:[CMSREF.ALLCOM]MASTER_LOGIN.COM. Logicals and symbols needed by XTranspin are also set (TRANSP\$:[CMSREF.TRANSPIN]XTRANSPIN.COM is not used).

TRANSP runs are submitted by PRETRANSP and use the command procedure TRANSP\$:[USER]TRANSP.COM, which first runs LOGIN_TRANSP.

Jeff's IDL Tools Development Environment

The IDL tools I've built are in CMOD\$MODELS:[TRANSPORT.IDL...]. They are kept in the cms library [.CMSLIB] with reference copies in [.CMSREF]. PLEASE USE THE CMS LIBRARY SYSTEM TO MAINTAIN THEM!!! That is, reserve and replace them when making edits, and use the comments to document the changes made.

The tools are run from IDL as restored savesets. The savesets need to be rebuilt every time a change is made to the IDL code. They can be rebuilt using my MAKE.COM command procedure. For example:

```
$ @MAKE foo
```

The MAKE procedure requires three files for each program (for example, FOO). 2 are kept in the CMS library:

foo_MODULES.PRO contains .run command for each external IDL procedure used by the program

foo_MAKE.PRO calls the foo_MODULES procedure and then issues the command:

```
save,/routines,filename='cmod$models:[transport.idl]foo.SAV'
```

The third file is a command procedure in CMOD\$MODELS;[TRANSPORT.IDL]:

foo.COM calls TRANSP_SETUP to check for a lock on the TRANSP code, TRANSP_LOGICALS to setup the logical names used by the tools, and then IDL. It also contains the IDL commands to restore the saveset foo.SAV and issues the command to start the program.

The procedures can be compiled in "debug" mode using the command procedure DBG.COM. For example:

```
$ @DBG foo
```

will run IDL, load the modules used by FOO (specified in the foo_MODULES file), and then give an IDL prompt. The code can then be run interactively.

Code Maintenance

Once TRANSP has been installed, the following batch jobs need to run on the VMS and Unix nodes to keep the code up-to-date. See the document TRANSP\$:[CMSREF.VAXPORT]INSTALL.INF and related documents in the TRANSP distribution for information on how to install the code, and the document UPDATING_TRANSP.DOC for details on how to keep the code up-to-date.

On the Unix node, the following cron jobs should run:

```
xshare:
```

```
00 20 * * * /transp/xshare/transp/daemon/update.daemon
    >/transp/xshare/transp/log/update.cron.log
```

pshare:

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * *
    /transp/pshare/transp/daemon/master.daemon >
    /transp/pshare/transp/log/master.cron.log
```

These can be submitted using the files `cron_xshare` and `cron_pshare` respectively. The commands are:

```
crontab cron_xshare
crontab cron_pshare
```

The batch job `TRANSP$:[CMSREF.CODESYS]CODE_DAEMON` should be submitted to `TRANSP_VAX`. It runs nightly, submitting a slave job to `TRANSP_AXP`. Ignore the errors from the jobs `TRMONITOR` and `TRSTART`. These programs handle `TRANSP` run enqueueing and management, and are not used at C-Mod.

Runs RDB Table

The logical name `TRANSP_RUNDB` must point to the database containing the `TRANSP_RUNS` table. An RMU dump of the currently existing database, `CMOD$MODELS:[TRANSPORT.RUN_DB]TRANSP_RUNS.RDB`, is kept in the same directory as the database.

Backing Up

Code backups should be done before every code update, after every code update, after many runs have accumulated, or if development work on the code itself, or the IDL tools, is in progress. The command procedure `CMOD$MODELS:[TRANSPORT]BACKUP.COM` can be used for this. Note that a special set of RMU commands are included in this procedure to backup the `TRANSP_RUNS` database.

*** THIS PROCEDURE MUST BE CHANGED IF THERE ARE CHANGES TO THE DEFINITIONS OF THE LOGICAL NAMES USED FOR THE TRANSP FILE AREAS.***

B.4 MDSPLUS_TRANSP.DOC

MDSPLUS_TRANSP.DOC

96.11.29

=====

TRANSP is now MDSplus compatible. That is, data can be read into TRANSP through TRDAT from an MDSplus tree as opposed to a set of UFILES. SCRUNCHER can also read from and write to an MDSplus tree. TRANSP output can be written to an MDSplus tree using MDSPLOT (which is a cut-up version of RPLOT; only the data read/write routines were kept).

TRANSP's MDSplus capabilities cannot be exploited without some setup external to the TRANSP code system. See the document MDS_SETUP.DOC for an overview. Non-MDSplus users need not worry about these documents - all TRANSP defaults are constructed so that TRANSP will run as before the MDSplus modifications.

TREE

In MDSplus mode, TRDAT and SCRUNCHER will expect their inputs to reside in a TRANSP tree shot. To coordinate the TRANSP files with the TRANSP tree, a four digit shot number is used for the TRANSP tree which is also the run number for the TRANSP files (nnnnXY.ABC). By convention, we use run numbers in the range 1000-9000. MDSPLOT also requires the TRANSP tree.

The TRANSP tree must have a particular structure. For complete details on the TRANSP tree, see Jeff's thesis. Here is a basic outline:

```

\TRANSP::TOP
  :NAME_LIST (\NL)
  INPUTS (\INPUTS)
      :xx1
      :xx2
  BOUNDARY
      :NUM_MOMS (\N_MB)
      :R_SOURCE (\RMB)
      :Z_SOURCE (\ZMB)
  OUTPUTS

```



```

ONE_D (\ONE_D)
      :www1
      :www2
TWO_D (\TWO_D)
      :yyy1
      :yyy2
IN_CHK (\IN_CHK)
      :xx1_IN
      :xx1_USE
      :xx2_IN
TRANSP_OUT
      :www1
      :www2
      :yyy1
      :yyy2
      :xx1_IN
      :xx1_USE
      :xx2_IN

```

xx1...xxN are the trigraphs for all TRANSP inputs (see TRDATGEN.SPEC for a complete list) and are known as the "input nodes". An existing TRANSP model's input nodes can be updated when TRDATGEN.SPEC changes - see UPDATING_TRANSP.DOC. Two dimensional inputs (vs. time and position) MUST be stored with time as dimension 0, and position as dimension 1. (Setting LFIXUP=2 in the namelist will allow either ordering for TRDAT itself - provided the units are correct - but the IDL-based utilities are not guaranteed to work if the order is switched.) Right now, the position dimension is taken as $x=(R-R_x)/A_x$, where R_x is the major radius of the magnetic axis, and A_x is the distance from the magnetic axis to the boundary surface on the midplane. (See TRANSP_COORDINATES.DOC.)

A complete list of the possible TRANSP outputs is contained in XPLOT7.FOR. www1...wwwN are the one dimensional TRANSP outputs, with time as their only dimension. yyy1...yyyN are the 2D outputs, stored with position as their 0th dimension and time as their first. The position dimension is usually \TWO_D:X or \TWO_D:XB (see TRANSP_COORDINATES.DOC for complete details). The nodes under IN_CHK store the input profiles which TRANSP knows as xxx_IN and xxx_USE.

In order to write to a node, it must already be present in the tree. No facility exists to add a node to the tree if there is an output

available with no corresponding node. The outputs that MDSPLOT could not write to the tree are stored in the file `nnnn_NOWRITE.DAT`, where `nnnn` is the 4 digit shot number.

SCRUNCHER input is stored under `\INPUTS.BOUNDARY` (nodes are known as "boundary source nodes").

USAGE

To use the MDSplus capability within TRANSP itself, the only requirement is that a TRANSP tree exists with a shot number corresponding to the TRANSP run number `nnnn`. All the steps in `MDS_SETUP.DOC` must be performed once before TRANSP can be used.

TRDAT:

The TRDAT namelist switch `KMDSPLUS` controls where TRDAT looks for its input. If `KMDSPLUS=0`, TRDAT runs in its usual manner, looking for a set of UFILES as specified in the list of `PRExxx` and `EXTxxx` namelist switches. If `KMDSPLUS=1`, TRDAT will look in the TRANSP tree. The node it will search for a given trigraph `xxx` (specified in the namelist by `EXTxxx`; `PRExxx` is ignored) is the node `\INPUTS:xxx`. The input nodes are loaded by the IDL-widgets based preprocessor `PRETRANSP` (see `PRETRANSP.DOC`).

MDSPLOT:

MDSPLOT would only be used by an MDSplus user. No switches need be set for it to run properly. It uses the `nnnnMF.PLN`, `nnnnNF.PLN`, and `nnnnTF.PLN` files generated by `POPLT2`, which is run after TRANSP itself completes.

SCRUNCHER:

To use SCRUNCHER with MDSplus, use `+` as the option for the data source. The boundary source nodes are loaded by `PRETRANSP`, which for Alcator C-Mod gets its boundary data from EFIT output stored in the C-Mod tree.

MDSplus-TRANSP is much easier to use if the IDL-based utilities for data preparation are also used. See `TRANSP_FILE_AREAS.DOC` on where to

find them, and the document named after the utilities on how to use them.

If using the utility PRETRANSP to prepare TRANSP runs, do not use the queue management software supplied in the TRANSP package. Instead, use the command procedure ??? [ed note: the command procedure is local to MIT for now. It will be committed to one of the .COM file CMS libraries soon, and this file will be updated. Eventually, PRETRANSP and the other IDL utilities will be integrated with the TRANSP queue management software. - JMS 95.06.15]

TRANSP CODE MODIFICATIONS

The MDSplus modifications to the TRANSP code are all delimited by

```
C*** MDSPlus Modification
C*** end MDSPlus
```

Most are self-explanatory. The following modules were modified:

The input stage of TRDAT was modified to read from a TRANSP tree if KMSPLUS=1. See the files UF1GET.FOR, UF2GET.FOR, UF3GET.FOR and the new UFMSPLUS.FOR in SOURCEROOT:[TRDATSUB]. DDATA.FOR was modified to read the namelist variable KMSPLUS, which is also in the include file TRDATA.BLK. DPRSET.FOR was modified to have the default value KMSPLUS=0.

SCRUNCHER.FOR in SOURCEROOT:[SCRUNCHER] was also modified.

The new program module SOURCEROOT:[MDSLOT]MDSLOT.FOR was written to handle output.

All MDSplus references included in TRANSP itself have !^ after them, so they are VMS only. This is because LIB\$FIND_IMAGE_SYMBOL is used to gain access to the MDSplus shareable image library (MDSSHR.EXE). Using a shareable image for the MDSplus routines saves the necessity of including the library (or at least a dummy library) with the PPPL TRANSP distribution in order for non-MDSplus users to link the code.

The MDSplus functions used are in their own (NEW) subroutine library
SOURCEROOT:[MDSTRANSF]. The functions in this library are;

```
FIND_IMAGE_SYMBOL
MDS_CLOSE
MDS_OPEN
MDS_PUT      <- this used by MDSPLOT
MDS_PUT1D   ]
MDS_PUT2D   ] these 3 used by SCRUNCHER
MDS_PUT3D   ]
MDS_VALUE
```

B.5 MULTIGRAPH.DOC

MULTIGRAPH.DOC

96.12.17

=====

This program is used primarily to display TRANSP input and output data. It can however display any valid MDSplus data.

SIGNALS

To plot a signal, click on "Add Signal". Input a "valid expression" in the signal input field which pops up. "Valid expression" generally means a TDI expression that returns a signal (ie. it must have a dimension associated with it). The signal returned can be 1D or 2D. (There is more to this story - we'll see it later.) Any node reference must be absolute, as no default node can be set.

A tree and shot must also be specified. If several signals from a given shot are to be plotted, use the "Set Default Shot" option under the "Shot" pulldown menu. If a given multigraph is to always look at a specific shot, then choose the "fix shot to MG" option under the "Shot" pulldown menu. This will ensure that when the multigraph is loaded in the future, it will come up with the specified shot, and not the current default.

If it is desired to mix data between TRANSP trees and C-Mod trees, the following feature makes it easier to identify shot numbers. Set the default shot to be a TRANSP run, either pasted or unpasted. TRANSP tree signals are plotted in the usual way. To plot a signal from a C-Mod tree corresponding to the shot number from the TRANSP run, enter the name of the C-Mod tree, and 0 for the shot number. Multigraph will use the contents of the node \TOP:SOURCE_SHOT from the default run to determine the C-Mod shot to use.

Ultimately, when the data is plotted, only "y vs. x" is allowed - ie. 1D data. 2D signals are therefore "sliced" to give data vs. time at a given position or data vs. position at a given time. In most cases, the slicing options on the "Functions" popup under the "Plot" pulldown menu will suffice. Select whether to plot vs. time or position

(radius), and input the radius or time at which to slice the data. MG uses the TDI functions RSLICE and TSLICE. See the document CMOD\$MODELS:[TRANSPORT.DOC]TDI.DOC for more information.

To edit a signal that has already been defined, click on the signal appearing in the signal list.

A signal can be made "not valid" by defining the slice value to be outside an allowable range, or by performing an illegal operation on the data (see the "Data Functions" section below). To determine the reason a signal is invalid, edit the signal, and click the "Apply" button.

There are several "special" signal expressions and features:

Callable IDL

NEW FEATURE November 1996!!!

To plot the result of an IDL procedure, you no longer need to enclose the name of the procedure and its parameters in parentheses. The IDL command, including all parameters, can simply be entered as the signal expression as is. For example, to call the procedure TEST_PLOT with 3 arguments, enter the following as the expression:

```
TEST_PLOT,arg1,arg2,arg3
```

The procedure TEST_PLOT must be in the IDL path, which the user must set in their startup procedure or using the logical name IDL_PATH.

The procedure MUST return a 1D signal, stored in the "Ken" Variable "_SIG". For example, to plot a parabola, the procedure would look like:

```
PRO TEST_PLOT
  x=findgen(100)
  y=x^2
  dummy=mds$value('_SIG=BUILD_SIGNAL($,,$)',y,x)
END
```

If the procedure can be successfully compiled, but it does not return

a valid signal in _SIG, Multigraph will display an error message stating this.

See the section "Expression Evaluation" below for more information.

Math Operations on Signals

NEW FEATURE November 1996!!!

Mathematical operations can now be performed on signals, without the need for an explicit "build_signal" command in the expression. For example, the ion to electron temperature ratio can be plotted simply by entering:

\TI / \TE

Multigraph detects that this is a mathematical operation, and does the "build_signal" itself. (The build_signal is necessary because math operations on signals in TDI result in arrays, not signals.) CAUTION!!! Multigraph is assuming that the operation is being performed ONLY on 2D TRANSP outputs having radial dimensions \X (see TRANSP_COORDINATES.DOC). If outputs and inputs are mixed, or if outputs with different radial dimensions are mixed, the results could be bogus. THE USER MUST CHECK THAT ALL SIGNALS HAVE THE PROPER RADIAL DIMENSION!!! Multigraph will not detect this itself.

Variable expressions

10 "variable expressions" can be defined from the "Variable Expressions" popup under the "Plot" pulldown. These can then be used as signals by putting the following into the signal input box:

&N

where N is the index (0-9) of the variable expression. This feature is useful for example if you have several shots in which you want to look at the same data. It is also useful if you want to look at several slices of the same signal. If you input:

`&N,SLICE`

where SLICE is a floating point number, you will get the signal defined by variable expression N sliced at the value SLICE. The slicing method is determined on the "Functions" popup. If you are plotting vs. position, then the slices will be time slices, and vice versa.

Error Plots

Errors in TRANSP are usually determined by rerunning a shot with scaled inputs. To display these errors, first enter the signal and its "nominal" shot. Then enter the same signal and the two shots which should determine the maximum and minimum ranges of the signal. Finally, go back to the nominal signal, and press the "Error range" button. The number of points plotted can be altered by changing the number of points slider.

For example, say you have shot 1000 which is the nominal case. It was rerun as shots 1001 and 1002 with TER -> 1.1 TER and 0.9 TER respectively. To plot the errors in chi_e, do the following:

- define the variable expression 0 as \CONDE
- enter the signal &0 with shot 1000
- enter the signal &0 with shot 1001
- enter the signal &0 with shot 1002
- go back to the first signal and press "Error Range"

Note that the signal and its two error shots MUST BE CONSECUTIVE in the signal list.

"Ken Variables"

A signal expression can include a TDI variable `_Pn`, where n is an integer between 0 and 9. The values of these "parameters" are set on the "TDI Parameters" popup menu. These are useful if you want to plot several different slices from the same shot without having to enter

File

Load Multigraph	Load a new multigraph
Save Multigraph	Save the current mg (if no filename is defined, ask for one)
Save As...	Allows renaming of a mg
Print...	print the mg (pops up a dialog box with printing parameters first)
New Multigraph	Clears/Resets the current settings
Quit	Quit MG altogether

If any changes are outstanding on a mg, you will be asked if you want to save them first before loading, clearing, or exiting.

Plot

Functions	operations to perform on the data (see separate section)
Style	change appearance of plot (labels,ticks,charsize,etc.)
TDI Parameters	edit "Ken Variables" referenced by signals (see above)
Variable Expressions	define variable expressions (see above)

Shot

Set Default Shot	define the default TRANSP run to use
(Un)Fix Default to MG	Use when a specific shot should always be used

for a given mg

Scale

Autoscale	sets xstyle and ystyle=2 and sets the x and y ranges to the maximum covered by the data
Autoscale Y	sets the yrange to be the maximum range within the currently defined X range
Set Ranges	pops up a dialog box asking for the range values

Signal Ops (only active when a signal is selected - see "Signal Box")

Deselect	deselects signal
Scale to	sets scale to range defined by signal
Delete	delete signal (will ask to confirm)
Copy	Create another copy of the selected signal
PSLICE	Call PSLICE to display 2D signal, if it is 2D
MB3 - POINT	if a signal is selected when MB3 is used in the draw window, the pointer will trace over the selected signal (ala Scopes)

DRAW WINDOW

MB1 (left) ZOOM	Press to start zoom box, drag over desired area, and release (ala Scopes)
MB2 (middle) AUTOSCALE	click to autoscale

MB3 (right) POINT Place crosshairs on the window and display the location in the lower right corner of the widget (see "signal box" for special case)

SIGNAL BOX

* ADD SIGNAL * Add another signal to plot (up to 20 maximum)

Single click Select signal for further operations (See "Signal Ops" above)

Double click edit signal

DATA FUNCTIONS

The following functions are available for use on Multigraph signals. The settings are usually carried over from one Multigraph to the next. For example, assume the electron power balance multigraph is loaded, and the VOLINT operation is selected. If the ion power balance multigraph is then loaded, VOLINT will remain selected.

Since many different functions are performed on a Multigraph in the normal course of observation, the settings are not saved in the Multigraph definition file by default. If you wish to save them, for example, to always plot the volume integrals when the Multigraph is first loaded, then press the "Stick settings to MG" button. If a newly-loaded Multigraph is not using the functions you set from the previous one, check the setting of this button.

2D plotting:

Determine the slice method to use (time history at a given position, or profile at a given time). The TDI slicing functions TSLICE and RSLICE are used. Please read the document TDI.DOC for more information on them. If you specify a value for the time or position

at which to plot the data that is outside the range contained in the signal, the signal will be invalid (see above).

Stack Traces:

 Useful for "accounting" type plots. This plots each successive signal "on top" of the previous one. That is, for signals s_0, s_1, \dots, s_N , it plots: $s_0, s_0+s_1, s_0+s_1+s_2, \dots, s_0+s_1+s_2+\dots+s_N$. Signals have the same dimensions for this to work (ie. be on the same time and position grid).

Integrals:

 On TRANSP OUTPUTS ONLY, the following operations can be performed:

VOLINT: Volume integrates the signal by multiplying it by the volume elements in `\TWO_D:DVOL` and then summing. The result is a 2D signal versus r and t . Data vs. r means "the volume integral of the signal from 0 to r ". The volume elements are the volumes contained between successive flux surfaces. Since these are specific to the TRANSP outputs, this operation is only allowed on TRANSP outputs.

VOL: Multiplies the signal by `\TWO_D:DVOL` but does not sum.

VOLAVG: Calculates the total volume integral of the signal and divides by the plasma volume `\ONE_D:PVOL`.

SURFINT: Surface integrates the signal by multiplying it by the surface elements in `\TWO_D:DAREA` and then summing. The result is a 2D signal versus r and t . Data vs. r means "the surface integral of the signal from 0 to r ". The surface elements are the surface areas between successive flux surfaces.

TGRAD: Calculates the gradient of the signal, returning $\text{grad}(S)$ versus r and t . This operation can be performed on TRANSP inputs as well.

Time Average:

 Average the signal over the time specified.

Scale Factor:

Multiply the data by the input factor. The factor is displayed on the Y-axis label if it is automatically determined.

B.6 POSTRANSP.DOC

POSTRANSP.DOC

96.11.29

=====

POSTRANSP is the front end for a variety of run management options.

It allows the user to paste and delete runs; examine the namelist, logfile, database information, and signal processing information for a run; and to recover a run from various crash states.

POSTRANSP's functions are broken down into two categories: those that are performed on a previously selected run, and those that are not specific to a run.

Actions specific to selected run

After a run is selected using the "Select Run" button, the following options are available. Note that the actions marked with * can only be performed on a user's own runs.

Show LOG file: Display the log file from the TRANSP run in a read-only TPU window.

Database Info: Display all the values stored in the TRANSP_RUNS database for the run.

Show NAMELIST: View the NAMELIST for a run using the NAMELIST tool (also used in PRETRANSP). This is how to examine the NAMELIST for a successfully completed run (as PRETRANSP will not allow the run to be loaded).

Show/Edit Comments: Display the comments and the RUN_TYPE for a run. A user may edit the comments from runs belonging to himself.

Signal Info: Display the source procedure and the processing steps used for the inputs.

* Recheck status: Useful if PRETRANSP crashes, leaving a run locked, or if a submitted run crashes without updating its status to 'Not run'.

- * Recover data if run aborted: Useful if a submitted run aborts without loading its output data into the tree. Check the log file first to make sure that the run aborted. ONLY USE THIS FOR ABORTED RUNS! IF IN DOUBT, ASK FIRST.

- * Paste run into C-Mod tree: Paste the run into the TRANSP tree as a subtree under \TOP.TRANSPORT.TRANSP_RUNS if it is desired to permanently archive the run. A "run id" will be assigned, and the run will lose its 4 digit TRANSP shot number. The pasted run will automatically be selected for further operations after being pasted.

- * Delete run from TRANSP tree: Removes all trace of run. Can only delete unpasted runs (ie. runs in TRANSP tree with 4 digit shot number).

POSTRANSP should prevent the user from performing an unauthorized action (such as deleting another user's run, or pasting an already pasted run). If you think that POSTRANSP is letting you do something it shouldn't, please make note of it and inform Jeff immediately (and of course, don't do it!).

General actions

These actions are not specific to a given run.

Show all runs: Brings up "shot_select" widget for browsing runs in database. An SQL query can be specified to limit the runs displayed.

- * Show/stop enqueued runs: Lists a user's currently submitted runs and their queues. Clicking on a run will stop it, so click 'Quit' if no changes are desired.

Show/search node names: Brings up the "NODE SEARCH" tool which allows browsing of the input and output nodes in the TRANSP tree. Useful for translating the obscure codes into meaningful information. A search string can be entered to limit the nodes displayed - any node whose name or label contains the string will be listed.

B.7 PRETRANSP.DOC

PRETRANSP.DOC

97.05.24

=====

Running TRANSP - with PRETRANSP

To set up and submit a TRANSP run, run the TRANSP preprocessor PRETRANSP that I wrote. It is in the form of an IDL save set, and can be run by doing

```
$@CMOD$MODELS:[TRANSPORT.IDL]PRETRANSP.COM
```

PRETRANSP is widgets based, so it must be run in an X-Windows environment. TRANSP can be run on both VAXen and Alpha's. All data associated with a TRANSP run are stored in a TRANSP tree shot.

USER DEFAULTS

Note that this feature is not currently enabled. See USER_DEFAULTS.DOC for more information. There are places in this document which refer to the option of storing defaults. These options are not currently provided.

SET UP NEW SHOT

When setting up a new source, the TRANSP shot number is automatically generated. A valid C-Mod shot must be associated with the TRANSP shot of interest.

You have the opportunity to enter a 6 character "RUN TYPE" and a comment. The run type is intended to be a quick way of identifying the purpose of the run.

Every run is stamped with the username of the person creating it and the date it was created.

LOAD EXISTING SHOT

It is possible to quit PRETRANSP before a run has been completely

prepared. This option allows you to come back to the run and continue preparing it. Also, if a run does not complete successfully, it can be reloaded and corrected.

LOAD SOURCES

TRANSP inputs associated with EFIT signals are loaded automatically every time this button is pressed. You are prompted for the number of moments to use in the fourier decomposition of the boundary.

Users have control over which signals TRANSP can use as inputs and can determine via IDL procedures the source TRANSP uses for each input. See SOURCES.DOC. The source for a signal must be loaded before it can be processed.

PROCESS SIGNALS

All the signals except a few associated with EFIT must be copied from \IN:xxx:SOURCE to \IN:xxx. This is the stage where the user can preprocess the data.

There are several processing options available:

SMOOTHING: 1D and 2D signals can be smoothed. 2D signals can be smoothed in one dimension at a time or both dimensions at once. Smoothing is done by convolution with a smoothing kernel. Time kernels can be boxcar or RC causal. Position kernels can be boxcar or Gaussian

TRUNCATION: A signal can be chopped within a specified time and/or position range.

RESAMPLING: A signal can be resampled over a different time base. If the timebase is coarser, PRETRANSP finds the time points in the signal closest to the specified spacing. If the timebase is finer, PRETRANSP will interpolate the data.

MATH: any valid IDL mathematical operation which can be specified in one line after "SIGNAL = " can be used on a signal. I caution against this, because units can get screwed up, and you might end up giving

TRANSP bogus data.

COPY ANOTHER SHOT: The processing steps from another TRANSP run with the same C-Mod shot can be copied.

All processing steps are recorded during the processing stage, and can be reviewed by selecting "Review" from the "Process" pull-down menu. When a signal is written to the tree (ie. when "Done" is pressed), the processing steps are recorded in the tree (in the node `\IN:xxx:PROCESSING`). If the user defaults system were enabled, users could set their own default processing by clicking on the "Default" button next to the step of interest after opening the "Review" widget. See `USER_DEFAULTS.DOC`.

The signals are displayed as "Before & After" the given processing step. Pressing "After" will accept the current step and move on to the next; "Before" will undo the current step. The signal displayed on the left-hand-side of the plot window is the signal that will be written to `\IN:xxx` when "Done" is pressed.

If you process a signal `xxx`, loading `\IN:xxx`, and come back to process it again, you will first be prompted to select the signal to process. Selecting "Input" will continue from the signal loaded to `\IN:xxx` earlier. "Source" will start fresh with the source signal `\IN:xxx:SOURCE`.

Display options:

If the plots on the screen get garbled by other windows, select "Redisplay" to replot them.

2D signals can be displayed as surface or contour plots, or they can be "sliced" to give a 1D signal vs. time or r/a . Xsurface can also be used.

1D signals can be displayed in two modes: "Separate" and "Overlay". The former displays the "before" signal (which would be written when "Done" is hit) on the LHS, and the "after" signal on the RHS. "Overlay" plots the "before" and "after" signals on the same graph.

You can adjust the range of the signal by selecting "Ranges" under the

"Display" pulldown menu. When a 1D signal is plotted in "Overlay" mode, you can use Mouse Button 1 (MB1, the LH-most button) to zoom, as on a scope.

NAMELIST

IT IS IMPERATIVE THAT YOU HAVE A GOOD UNDERSTANDING OF THE TRANSP NAMELIST AND WHICH SWITCHES ARE IMPORTANT FOR THE TASK IN HAND!!! See \$HELP TRANSP OPERATIONS NAMELIST for more information.

You have total freedom at present to edit the namelist. This can be dangerous, especially when it comes to telling TRANSP which inputs to use and how to use them, so be careful to change only those parameters you are sure you need to. Eventually, the sensitive parts of the namelist will be protected. A default namelist can be saved. See USER_DEFAULTS.DOC.

SUBMITTING

Before you can submit TRANSP, the sources must be loaded, the input nodes must all be loaded through the "Process Signals" button, and the namelist file must be written (using the "Namelist" button). If you exit PRETRANSP after having completed all 3 but before starting TRANSP, and come back to PRETRANSP later, you cannot start TRANSP immediately. If no further changes are required, and you do want to submit the job right away, the quickest way is:

- (a) press "Set Up Shot"
- (b) press "Process Signals"
- (c) press "Quit" on the pop up widget with the list of nodes
- (d) press "Namelist"
- (e) press "Exit" on Namelist widget to rewrite namelist file
- (f) press "Start TRANSP"

TRANSP can be submitted to any node. The default node is the one that PRETRANSP is run on. SEVERE PENALTIES will be imposed for bogging down someone else's machine. You are allowed to choose the time at which the job starts executing. The default time is 7pm the same evening. Please try not to run TRANSP jobs on the C-Mod cluster during the day, as they are very CPU intensive. The test shots I have run so far covered a time range of 0.3 s with a time step of 0.001 s, did not use the RF code, and took over 1/2 an hour to run on a VAX-4000/90. The asymmetric runs can take up to 5 times longer than the symmetric runs. Running on an AXP cuts the time down by a factor of ~ 3 .

The DCL procedure `TRANSP$:[USER]TRANSP.COM` is used to run TRANSP itself. Only thorough knowledge of TRANSP, or total insanity (the causal connections between the two are clear), would allow successful understanding and changes of `TRANSP.COM`.

OUTPUT AND UTILITIES

There are several utilities available to make your life running TRANSP easier. Run management functions are performed using `POSTTRANSP`. See the document `POSTTRANSP.DOC` for more information. There are also scopes and scope pads available in `CMOD$MODELS:[TRANSPORT.SCOPE]`. The scopes have extensions `.DAT` while the pads have extensions `.PAD`. I find it useful to have the "NODE SEARCH" tool from `POSTTRANSP` running whenever I'm looking at data, as the signal names in the pads are the obscure codes used by the UFILE people. There are TDI functions available to facilitate access to the data. See `TDI.DOC`.

`RPLOT` is available for use as well. It needs the `nnnnMF.PLN`, `nnnnNF.PLN`, and `nnnnTF.PLN` files, which are in a ZIP file in `TRANSP$RUNS:[OUTPUT]` (see `PASTING` section). To run it, set up an XTERM window (NOT a Decterm) and do the following:

```
$SET DEF your_directory
$COPY TRANSP$RUNS:[OUTPUT]RPLOT_nnnn.ZIP your_directory
$UNZIP RPLOT_nnnn
$RPLOT
```

I am not very familiar with it.

POSTRANSF (and PASTING)

TRANSP runs are initially stored in the TRANSP tree; a run's four digit shot number is the same as its TRANSP run number. It is intended that TRANSP runs which are deemed interesting or useful should be moved, or "pasted", into the C-Mod tree (under the TRANSPORT subtree). Use POSTRANSF to paste a shot into the C-Mod tree. Shots that are not pasted in will eventually be deleted.

Once a shot is pasted, it is deleted from the TRANSP tree. Ordinarily, all files which TRANSP itself reads and writes are deleted as well (including the .LOG file, and the RPLOT files). It is possible to save them to the scratch area, from which they can be copied to elsewhere.

See POSTRANSF.DOC for more information.

Help and documentation

There is little documentation on the programs I wrote currently available. See me for help if you need it.

Help on TRANSP itself is available at the DCL level by typing \$HELP TRANSP. NAMELIST help is particularly useful. Look in

\$HELP TRANSP OPERATIONS NAMELIST

The code is stored in the directory tree TRANSP\$:[...]. The ACLs should be set so that you cannot modify any of it, but please don't try anyway! You can look at it for clarification on what TRANSP is doing. I cannot promise any answers yet if you come to me for help on the code itself, but I will try.

TRANSP documentation is stored in TRANSP\$:[DOC] and TRANSP\$:[CMSREF.DOC]. These documents may reference others in different directories.

The log file of the run you submit is stored in TRANSP\$RUNS:[LOG] as

TRANSP_nnnn.LOG, where nnnn is the TRANSP shot number. POSTTRANSP can be used to view the log file (see POSTTRANSP.DOC).

CAUTION

While you have write permission in the TRANSP\$RUNS directories, please DO NOT WRITE IN THEM DIRECTLY, and DO NOT DELETE ANYTHING in them. Should the need arise (I can't see how it would, but you never know), please contact me immediately.

Also, though it should be obvious, DO NOT EDIT OR directly ALTER the contents of the TRANSP TREES in anyway!!! You shouldn't be able to, but it doesn't hurt to be warned.

The general rule of caution is 'see me before doing anything unusual, including writing or deleting any files in the directories TRANSP uses'.

Good luck running TRANSP!

B.8 PSLICE.DOC

PSLICE.DOC

96.11.29

=====

The General Purpose Profile Slicer

Purpose:

Displays two dimensional data in a user friendly manner. Data is usually a function of (r,t), but it can be a function of anything.

Invocation:

MDSplus mode:

```
IDL> pslice
```

The button "SIGNAL" brings up a window for you to enter a TDI expression for ANY two-dimensional signal. If the signal is built with units, the units will be displayed.

IDL mode:

```
IDL> pslice,Z,[X,Y]
```

Z is a two dimensional array. X and Y are optional - each is a one dimensional array with dimensions corresponding to Z. That is:

```
Z=ARRAY(m,n)
```

```
X=ARRAY(m)
```

```
Y=ARRAY(n)
```

Either BOTH of X and Y must be specified, or neither. In the latter case, Z is plotted against index number. That is

```
X=INDGEN(m)
```

```
Y=INDGEN(n)
```

Usage:

Four plots are displayed in a 2x2 arrangement. (IT=top, B=bottom, R=right, L=left). A contour plot of the data is in the TR panel. Slices of the data along the X and Y directions are plotted in the TL and BR panels respectively. The coordinates at which the slices are taken are plotted with crosshairs (red on a color screen). To change the slices plotted, use MB1 - click and drag to scan, or just click and release to jump. Scanning in TR allows movement in both the X and Y directions. Scanning in TL or BR allows scanning in only the Y or X directions respectively.

The BL panel shows a surface plot of the data. The two slices plotted in TL and BR are shown on the surface plot (in red on a color screen). The surface plot can be rotated with the sliders on the bottom line of the widget.

The "TRANSPOSE" button transposes the data, so that the slice in TL is displayed in BR and vice versa.

B.9 SOURCES.DOC

SOURCES.DOC

96.11.29

=====

USER-DEFINED SOURCE PROCEDURES for TRANSP INPUTS

Users can use any of the possible TRANSP inputs by providing an IDL procedure to calculate the source signal. The calculation can be something as simple as doing an MDS\$VALUE on a C-Mod subtree, or it can be as complicated as you like. Examples are the default source procedures stored in CMOD\$MODELS:[TRANSPORT.IDL.SOURCES].

The name and path of the source procedure for the input signal XXX are specified after hitting "Load Sources" and are stored in XXX:SOURCE_PRO. When the procedure is run to actually generate the source, the source signal is stored in XXX:SOURCE, and a copy of the procedure used is stored in XXX:SOURCE_PRO.

Defaults are stored for common sources in CMOD\$MODELS:[TRANSPORT.IDL.SOURCES], and users can store their own defaults (see USER_DEFAULTS.DOC). If you want to use the default sources in CMOD\$MODELS, you do not need to read the rest of this document.

The user's procedure is called from IDL using CALL_PROCEDURE. This places restrictions on how the procedure can be written and called. Here are the important points to remember:

- 1) The procedure MUST be called with ONLY these parameters IN THE ORDER SPECIFIED:

PRO name,CMOD_SHOT,SIGNAL,RANK,DIM_0,DIM_1,STATUS,UNITS=UNITS

If the signal is only one dimensional (ie. vs. time only), DIM_1 is still needed as a parameter. Make sure it is explicitly defined in your procedure to some dummy value. See CMOD\$MODELS:[TRANSPORT.IDL.SOURCES]CUR.PRO for how I handle this in the default procedures.

- 2) RANK is an integer giving the dimension of the signal. If the signal is versus time only, RANK=1. If the signal is versus time and

position, RANK=2

3) UNITS is a string array of dimension RANK+1. UNITS(0) is the units of the signal itself. UNITS(1...) are the units of each of the dimensions. See TRDATGEN.SPEC for the list of allowable units.

4) Only the TRANSP tree of the shot of interest is open when the procedure is called, and only this tree should be open when the procedure is done. In other words, user's procedures must open their own C-Mod subtrees AND CLOSE ONLY THOSE trees when they are done. DO NOT USE MDS\$CLOSE by itself. Please specify the C-Mod subtree name and the shot number as well. The variable CMOD_SHOT will contain the C-Mod shot number associated with the TRANSP shot. Example:

GOOD	BAD
----	---
mds\$open,'electrons',cmod_shot	mds\$open,'electrons',cmod_shot
te = mds\$value('\ECE_TE')	...
...etc...	...
mds\$close,'electrons',cmod_shot	mds\$close

5) STATUS can be used to tell the calling procedure whether the user's procedure succeeded in its attempt at generating the source.

6) Starting with IDL 4.0, procedures are recompiled every time they are loaded. This should allow the user to debug their procedures within the PRETRANSP context. The procedures are recompiled every time using the IDL function RESOLVE_NAME.

7) If the user procedure needs subroutines, the structure of the file should be as follows. (Assume the name of the procedure being called is TEST).

In the file TEST.PRO:

```
PRO subroutine1,...
```

```
RETURN
```

```
END
```

```
;---
```

```
PRO function1...
```

```
RETURN
END
```

```
;---
```

```
PRO subroutine2...
```

```
RETURN
END
```

```
;---
```

```
PRO subroutineN...
```

```
RETURN
END
```

```
;---
```

```
PRO TEST,CMOD_SHOT,SIGNAL,RANK,TIME,POSITION,STATUS,UNITS=UNITS
```

```
subroutine1,...
dummy=function1(...)
```

```
RETURN
END
```

8) Subroutines in other files can be called. However, there can only be one subroutine per file, and the filename must be the same as the procedure name (unless you get fancy and use the RESOLVE_NAME function yourself). Also, files other than the one named as the source procedure will not be stored in \INPUTS:xxx:SOURCE_PRO, so no permanent record of them will be available.

9) EFIT signals are available to the user's procedure, so there is no need to get them independently. To include the EFIT signals, put the following line near the top of the procedure (this will give you access to the EFIT common block):

```
@CMOD$MODELS:[TRANSPORT.IDL.CMSREF]W_EFIT.PRO
```

Examine the contents of this file and the file W_SOURCE.PRO (in the same directory as W_EFIT.PRO) to see what EFIT signals are available and how they are defined. If you would like additional EFIT information, it is always possible to open the tree and get it yourself, but I suggest letting me know so it can be added to the common block.

10) The position dimension of a 2D signal MUST be r/a and MUST be independent of time. (see TRANSP_COORDINATES.DOC). There is an IDL function available to convert signals defined versus R or r (which could itself be a function of time) to a fixed r/a grid. Call

```
NORM_RADIUS, IN_SIG, IN_TIME, IN_RADIUS, OUT_SIG, OUT_RADIUS, NR
```

where

```
IN_SIG = signal plotted versus R or r
IN_TIME = time base of input signal
IN_RADIUS = position dimension of signal (R or r)
            [can be R(t) or r(t)]
OUT_SIG = signal interpolated to fixed r/a grid
OUT_RADIUS = fixed r/a grid (this should be DIM_1)
NR = # of points in fixed grid:
      OUT_RADIUS = FINDGEN(NR)/(NR-1)
      (default value for NR is 101; can specify another value)
```

11) To display a message in the message window, use the subroutine SOURCE_MSG. Call it as follows:

```
SOURCE_MSG, string
```

12) DO NOT UNDER ANY CIRCUMSTANCES MESS WITH THE TRANSP TREE DIRECTLY!!! You do not need to do an MDS\$PUT into the TRANSP tree - my program does that for you provided the procedure returns a valid signal. I AM NOT RESPONSIBLE FOR THE CONSEQUENCES if you do mess with the TRANSP tree.

Users can test their source procedures before calling them from PRETRANSP. Run procedure @CMOD\$MODELS:[TRANSPORT.IDL]SOURCE_TEST.COM, and specify a valid C-Mod shot, the path, and the name of the

procedure. If an error occurs, it will hopefully be reported gracefully, but even if the program crashes, it should be possible to restart using retail. SOURCE_TEST was developed to avoid the problem of having PRETRANSP crash because of a faulty source procedure (though it would take a very bad error to get PRETRANSP to crash).

B.10 TDI.DOC

TDI.DOC

96.11.29

=====

SCOPES and TDI FUNCTIONS

=====

The scopes in CMOD\$MODELS:[TRANSPORT.SCOPE] have been protected so that you cannot edit them. If you want to customize a scope in there, copy it into your own scope directory.

To view data in a TRANSP shot (ie. not yet pasted), set the default experiment to TRANSP and use the 4 digit shot number.

To view data in a shot pasted into the C-Mod tree, set the default experiment to TRANSPnn, where nn is the number of the run_id, and use the 9 digit C-Mod shot number.

The scopes will tell you which tree and shot number you are looking at.

The tags \IN, \ONE_D, and \TWO_D are used to point out what type of signal can be passed to the function. You may leave off the \ONE_D and \TWO_D for the outputs.

Two very important TDI functions are TSLICE and RSLICE, which return a profile at a given time slice, and a time history at a given position slice, respectively.

TSLICE - returns a signal which is a profile for a given time

call as follows: TSLICE(node, time, [postype], [kenvar])

node: The 2D signal you wish to slice (either an output \ABCDEF or an input \IN:XXX)

time: the time at which you want the profile. The function will find the *nearest* time - it will not interpolate.

postype: an optional input. This specifies which position coordinate

you want the data returned with:

```

0 : NORMALIZED RADIUS (R-Raxis)/(Rbdry-Raxis)
1 : MAJOR RADIUS
2 : MINOR RADIUS
3 : "internal coordinate"
    inputs: r/a (same as 0)
    outputs: one of the four signals mentioned in (2)

```

kenvar: an optional output - a "Ken variable". This returns the actual time of the profile.

For example, in IDL:

```
signal=mds$value('TSLICE(\CONDE,0.5,1,_A)')
```

this returns a profile at the time nearest to 0.5 plotted versus major radius

```
act_time=mds$value('_A')
```

this returns the actual time for the profile

RSLICE - returns a time history at a given position. I urge caution
----- here, as the meaning of "position" is fuzzy.

call as follows: RSLICE(node, posn, [kenvar])

node and kenvar as the same as for TSLICE

posn: the position at which you want the time history. The function will return the history at the nearest value to posn (with the actual value used returned in kenvar).

- For an input, this would be the r/a value (note that the major radial coordinate corresponding to a given r/a value can and will change as a function of time).

- For an output, this would be the internal coordinate on which the signal is stored (one of the four signals mentioned in (2)). Then, you

must specify a value which is appropriate to the coordinate.

X, XB: posn would be the phi value
 RMJSYM, RMAJM: posn would be the major radial value

Note that for all four of these cases the major radial coordinate will change as a function of time, so you won't necessarily be looking at the same major radius over the entire time history.

For example, in IDL:

```
signal = mds$value('RSLICE(\CONDE,0.5,_A')
```

returns the time history of chi_e at the phi position nearest to 0.5

```
act_pos = mds$value('_A')
```

this returns the actual phi value for the time history

There are several other TDI functions which are used to look at TRANSP signals. All TDI functions can be used in any scope panel, or any MDS\$VALUE call.

I20(\IN:sig)

interpolates a 2d input signal onto the transp output timebase
 (replaces INTERP2D())

INTERP(sig,time)

interpolates the 1d signal sig to the timebase time

OUTRDIM(\TWO_D:sig)

returns the position dimension of the signal in major radial
 coordinates

RMAJ_IN(time)

returns EFIT's magnetic axis position (\MAP:IN_RX) interpolated onto the timebase time

RMIN_IN(time)

returns EFIT's minor radius (\MAP:IN_AX) interpolated onto the timebase time

SURFINT(\TWO_D:sig)

returns a 2d signal which is the surface integral of sig vs r and t. You will want to use TSLICE or RSLICE with this.

TIMDIM(signal)

returns which of the two dimensions of a 2D signal is the time dimension (0 for inputs, 1 for outputs)

TIME_AV(signal, window)

returns signal averaged in time over a width window. Handles input and output signals, 1D and 2D.

UNSCRUNCH(time)

for symmetric scrunches, returns the boundary shape at time

UNSCRUNCH_ASYM(time)

as UNSCRUNCH, for asymmetric scrunches

VOL(\TWO_D:sig)

multiplies signal by the volume elements DVOL to convert a density to density*d3v. Summing the elements in this signal from 0 to r gives the volume integral of the signal from 0 to r (see VOLINT)

VOLAVG(\TWO_D:sig)

returns the volume average of a 2D signal

VOLINT(\TWO_D:sig)

returns a 2d signal which is the volume integral of sig vs r and t. You will want to use TSLICE or RSLICE with this.

See MDS\$ROOT:[TDI.TRANSP] for the source code of these functions. DO NOT EDIT THIS SOURCE CODE! The functions are kept in the CMS library CMOD\$MODELS:[TRANSPORT.TDI.CMSLIB].

B.11 TDI_OBSOLETE.DOC

TDI_OBSOLETE.DOC

96.11.29

=====

The following TDI functions in MDS\$ROOT:[TDI.TRANSF] are obsolete:

INTERP2D - replaced by I2D

IN_R_SLICE, OUT_R_SLICE - replaced by RSLICE

IN_T_SLICE, OUT_T_SLICE - replaced by TSLICE

NORMALIZED_RADIUS - replaced by TSLICE and RSLICE

RAXIS - superceded by node \MAP:OUT_RX

RBDRY - superceded by nodes \MAP:OUT_RX and \MAP:OUT_AX

RMAJ - superceded by node \MAP:X_TO_R

RMIN - superceded by node \MAP:R_VS_PSI

RMINOR - superceded by node \MAP:OUT_AX

RVSPSI - replaced by node \MAP:R_VS_PSI

TRESCALE - Maxim's mapping function - replaced by nodes

B.12 TRANSP_COORDINATES.DOC

TRANSP_COORDINATES.DOC

96.11.29

=====

COORDINATES and RECONSTRUCTION

TRANSP performs its own reconstruction of the flux surfaces. Before the latest version it was constrained to an up-down symmetric reconstruction; hence, it's own coordinate system will be different than the "real" coordinate system of C-Mod (as defined by EFIT etc.).

The construction TRANSP performs is based on the real C-Mod boundary shape, as given by EFIT. The C-Mod boundary shape is fourier decomposed, usually to four moments, and the fourier moments are used by TRANSP. The (old) symmetric fourier decomposition looks like:

$$\begin{aligned} R &= R_0 + \text{SUM } R_i * \text{COS}(i * \text{theta}) \\ Y &= \text{SUM } Y_i * \text{SIN}(i * \text{theta}) \end{aligned}$$

where the sum is over i from 1 to n (usually n=4).

The latest version of TRANSP allows up-down asymmetric plasmas for OHMIC RUNS ONLY (a symmetric decomposition is still necessary for the ICRF code). The asymmetric decomposition looks like:

$$\begin{aligned} R &= R_0 + \text{SUM } [R_{i,c} * \text{COS}(i * \text{theta}) + R_{i,s} * \text{SIN}(i * \text{theta})] \\ Y &= \text{SUM } [Y_{i,c} * \text{COS}(i * \text{theta}) + Y_{i,s} * \text{SIN}(i, \text{theta})] \end{aligned}$$

where $R_{i,c}$ $Y_{i,c}$ are the cosine moments and $R_{i,s}$ $Y_{i,s}$ are the sine moments (for $i=1$ to n).

INPUTS

Profile data input to TRANSP is mapped to a normalized r/a coordinate system before TRANSP sees it. (This means that all relevant switches in the NAMELIST must be set accordingly, including $NR_{ixxx} = -4$ and $NSY_{xxx} = 0$). The diagnostics own radial coordinate is mapped to r/a using EFIT's data. Note that r/a is defined to be $(R - R_{magx})/A_{magx}$

where

R = major radial coordinate of diagnostic
 Rmagx = major radius of magnetic axis (\EFIT_AEQDSK::RMAGX)
 Amagx = distance from magnetic axis to intercept of outermost
 flux surface with midplane (calculated as follows:

$$\text{Amagx} = \text{\EFIT_AEQDSK::ROUT} + \text{\EFIT_AEQDSK::AOUT} - \text{\EFIT::AEQDSK::RMAGX}.$$

The input nodes \INPUTS:POS and \INPUTS:RMN are loaded with EFIT's values for Rmagx and Amagx respectively.

OUTPUTS

TRANSP stores all output profile data in SQUARE ROOT NORMALIZED TOROIDAL FLUX coordinates, ie. NOT IN NORMALIZED RADIUS r/a. Note that a fixed normalized flux grid is used, so the major radial mapping of the flux grid is a function of time. Negative flux values refer to the inboard side of the magnetic axis.

ALL of the 2D TRANSP outputs have one of the following five possible signals as their position dimensions.

X : these are the normalized square root toroidal flux values (ϕ) of the flux zone centers. Values: 0.025 to 0.975 in 0.05 increments. Time independent.

XB : these are the ϕ values of the flux zone boundaries (used for signals which are "flux through the surface" related - like \CONDE or \PCNDE). Values: 0.05 to 1.00 in 0.05 increments. Time independent.

RMJSYM : these are the major radial locations (in cm) of the flux zone centers and boundaries. Essentially, the mapping from ϕ to major radius. Values correspond to the ϕ values -1.05 to 1.05 in 0.025 increments. Time dependent.

RMAJM : A subset of RMJSYM - the major radial locations of the flux zone boundaries from -1.0 to 1.0 (ie, covering only the

plasma). Time dependent.

RFMAJ : major radius in *metres* of the zones used in the
FPPRF/SPRUCE calculations. Used ONLY for dispersion relation
outputs \REKP2 and \IMKP2

You will get one of these five signals when you do a
DIM_OF(\ABCDEF,0). (Incidentally, the time dimensions of the 1D
signals are all \TIME1D and the 2D signals \TIME2D.) To get the major
radial mapping of the position dimension of any TRANSP output, use the
TDI function OUTRDIM:

```
position = mds$value('OUTRDIM(node)')
```

The TRANSP output XIRSYM contains the complete flux grid from -1.05 to
1.05 in 0.025 increments. DIM_OF(\XIRSYM,0) returns RMJSYM. THESE
TWO SIGNALS SHOULD BE USED TO CONVERT FLUX COORDINATE TO MAJOR RADIAL
COORDINATE!

Under the structure \TRANSP.MAPPING (tag \MAP) are several signals
which summarize the coordinate system that TRANSP uses. This should
make it easier to convert between the internal coordinates of the
TRANSP outputs, and major or minor radial coordinates.

The signals added are:

```
\TOP.MAPPING:      (or \MAP:)
  X_TO_R           - mapping from X to major radius (cm)
  XB_TO_R          - same for XB (cm)
  R_VS_PHI         - the same as RMJSYM, with DIM_OF(\MAP:R_VS_PHI,0)
                    equal to the normalized flux values corresponding
                    to each position (cm)
  OUT_RX           - TRANSP's magnetic axis vs time (cm)
  OUT_AX           - TRANSP's Rboundary-Raxis vs time (cm)
  IN_RX            - EFIT's magnetic axis vs time (same as \IN:POS)
  IN_AX            - EFIT's (Rbdry-Raxis) vs time (same as \IN:RMN)
  EFIT_PHI         - EFIT's calculation of square root normalized
                    toroidal flux vs. position as the TRANSP output
                    times.
```

For example, the TRANSP output for the electron thermal diffusivity is
\CONDE. Its position dimension is XB (ie. flux values 0.05 to 1.0 in

0.05 increments). To plot \CONDE versus major radius do the following in IDL:

```
chi=mds$value('\CONDE')

r=mds$value('OUTRDIM(\CONDE)') (or, since we know XB is the internal
coordinate, r=mds$value('\MAP:XB_TO_R') will also work)

i=[index of time slice you wish to plot]

plot,r(*,i),chi(*,i)
```

To plot chi versus minor radius, do

```
r=mds$value('OUTRDIM(\CONDE)') - mds$value('\MAP:OUT_RX')
```

There are two other radial output coordinates that could be of use: \TWO_D:RMNMP and \TWO_D:RMJMP. DO NOT PLOT DIRECTLY VERSUS EITHER!!! RMJMP is the major radial location of the center of each flux surface zone used in TRANSP's calculations. RMNMP is the distance from the center of the flux surface zone to the boundary of the zone. Therefore, by adding the two, you obtain the major radial location of the boundaries of the flux surface zones, which is just what you'd get by converting XB to a major radial grid using XIRSYM and RMJSYM.

I encourage you to examine these signals on your own, and contact me with questions. Please don't try to "reinvent the wheel" by writing your own coordinate mapping routines. I have placed a lot of thought into what TRANSP is doing here.

TDI SLICING FUNCTIONS

A long time ago I wrote the TDI functions IN_R_SLICE, IN_T_SLICE, OUT_R_SLICE, and OUT_T_SLICE, which would get either a time history for a given position or a profile for a given time.

I have condensed all these down to two functions: TSLICE and RSLICE. This should make it easier to type and remember. (The old functions will still work, but they are very inefficient and are not guaranteed to be maintained forever). See TDI.DOC for more information.

COMPARING TO INPUTS

TRANSP passes as outputs the interpolated input profiles that it used (on its own coordinate system). The profiles that it used can then be compared to the profiles that it was fed as input. For example, to compare the temperature profiles, plot \INPUTS:TER and \TWO_D:TE on the same graph. CAUTION!!! The input and output signals will in general be on different time bases. To interpolate a 2D \INPUTS signal to the TRANSP timebase, use the TDI function I20:

```
result = I20(\INPUTS:node)
```

Using I20 and the slice functions makes profile comparison easier. To compare the profiles vs. r/a at (say) $T=0.5$ s, plot the following:

```
TSLICE(\TWO_D:TE, 0.5, 0)  
TSLICE(I20(\INPUTS:TER), 0.5, 0)
```

There are still slight differences between the profiles that TRANSP uses and the profiles that it was given. The origin is under investigation.

For up-down symmetric restricted runs, the values for Rmagx and Amagx that TRANSP calculates will be (slightly) different than the EFIT values. The intention in the data mapping is to have the input and output profiles identical when mapped to an r/a normalized radial coordinate system. They will then appear slightly different when plotted vs. major or minor radius.

B.13 TRANSP_FILE_AREAS.DOC

TRANSP_FILE_AREAS.DOC

96.12.19

=====

TRANSP Logical Names

=====

Set by system:

DISK\$TRANSP	CMOD2\$DKA100:	(for rooted logical names, substitute this definition for DISK\$TRANSP)
TRANSP\$	DISK\$TRANSP:	[TRANSP.]
TRANSP\$AXP	DISK\$TRANSP:	[TRANSP.AXP.]
TRANSP\$DATA	CMOD\$ROOT:	[CMOD.TRANSP_RUNS]
TRANSP\$RUNS	DAQA\$DRA6:	[CMOD.SHOTS.TRANSP_RUNS.]
TRANSPnn\$DATA	CMOD\$NROOT:	[TRANSPORT]

Set by TRANSP login or TRANSP_SETUP.COM (for IDL tools):

TRANSP_IDL	CMOD\$MODELS:	[TRANSPORT.IDL.CMSREF]
TRANSP_XSHARE	DISK\$TRANSP:	[UNIX.TRANSP.PSHARE.TRANSP.]
TRANSP_PSHARE	DISK\$TRANSP:	[UNIX.TRANSP.XSHARE.TRANSP.CODESYS.VAXPORT.SOURCE.]

Note that the settings for unix<->VMS transfers in TRANSP's DO_CONFIG and UPDATE_SETUP use the translations of TRANSP_XSHARE and TRANSP_PSHARE rather than "TRANSP_XSHARE" and "TRANSP_PSHARE", as they need to be rooted logical names.

TRANSP Disk Areas

=====

VMS

TRANSP\$	TRANSP source and VAX binaries
TRANSP\$AXP	AXP binaries
TRANSP\$DATA	TRANSP trees
TRANSP\$RUNS	"PPPL Files" (nnnnXX.YYY) and log files

```

CMOD$MODELS:[TRANSPORT...] All MDSplus and IDL code, runs database
TRANSPORT$DATA             TRANSPORT tree
TRANSPnn$DATA              Pasted TRANSP runs (nn = 01 to 20 now)
TRANSP_XSHARE              LOST xshare unix->VMS file transfer area
TRANSP_PSHARE              LOST pshare transp area
SCRATCH_DISK:[TRANSP...]   TRANSP WORK area (volatile)
MDS$ROOT:[TRANSP.TDI]     TDI functions
                           (CMSLIB is CMOD$MODELS:[TRANSP.TDI])

```

Unix

mount to these:

```

daqa:disk$transp:[unix.transp]    /transp
cmod3:scratch_disk:[transp.log]    /transp/cmod

```

It is no longer necessary to export any Unix file areas to VMS. Since the disk is actually an nfs mounted VMS disk already, direct disk access is possible for unix<->VMS transfers. The following areas are accessed by the VMS implementation of TRANSP. The logical names TRANSP_XSHARE and TRANSP_PSHARE are defined in LOGIN_TRANSP and are for convenience only - they are not used by any of the TRANSP management scripts.

```

/transp/xshare/transp/codesys/vaxport/source  TRANSP_XSHARE:[000000]
/transp/pshare/transp                        TRANSP_PSHARE:[000000]

```

Also obsolete are the nfs mounts of the PPPL disks:

```

sr01.pppl.gov:/users/xshare/repos  /transp/pppl/repos  $CVSROOT
sr01.pppl.gov:/users/xshare/codata /transp/pppl/codata $PPPLDATA

```

These have been replaced by rsh access to the PPPL repository as part of cvs versions 1.5 and higher.

B.14 TRANSP_SCALARS.DOC

TRANSP_SCALARS.DOC

97.05.24

=====

=====

The program "TS" (for "TRANSP Scalars") is used to interactively load the TRANSP_SCALARS table of the LOGBOOK database. It is accessed by executing the command procedure TS.COM in CMOD\$MODELS:[TRANSPORT.IDL].

The table is used to store TRANSP output data, both 1D and 2D. The data is stored with the key (SHOT, RUN_ID, TIME_SLICE, R_POSITION). RUN_ID refers to the RUN_ID of the particular TRANSP run (see POSTRANSF.DOC for information on RUN_ID's and pasting). There can therefore be multiple entries in the table for a given shot, each corresponding to a different TRANSP run.

R_POSITION will be either 1,2, or 3, corresponding to $r/a=0.25$, 0.5 , and 0.75 . Thus profile data is stored at these three locations. Scalar data is independent of position, and so would be the same for each R_POSITION. For example, the data stored for a particular shot might look like:

SHOT	RUN_ID	TIME_SLICE	R_POSITION	TAUE	TEMP_E	HDIF_E
9601300020	TRANSP02	0.94	1	0.043	2.315e3	1.062e4
9601300020	TRANSP02	0.94	2	0.043	1.467e3	5.657e3
9601300020	TRANSP02	0.94	3	0.043	7.498e2	6.300e3

To load the table, a run must first be selected. Only pasted runs may be loaded into the table. After selecting the run, the time slice must be selected. There is an option to select one of the time slices from the core table for that shot, so that joined queries are possible.

After the time slice is selected, press the button "Process data and load database". This brings up the "signal widget" - a menu of the parameters stored as columns in the table. The buttons are labelled by the column name; the widget provides a description of each. Each parameter corresponds to a particular TRANSP output signal. For example, POH corresponds to \TWO_D:POH, and TAUE corresponds to \ONE_D:TAUA1.

Typically, it is desired to load all the columns at once. This is

accomplished by pressing the "Load All" button. You are prompted for smoothing parameters over which to smooth the TRANSP data, and then can press "Proceed with auto load" to load them all. Note that with this method, the same smoothing is applied to each signal. To determine the optimal smoothing parameters, it is useful to examine the signals individually. This is done by pressing a signal button. It is also possible to apply different smoothing to each signal, or to load only specific columns. This is done by visiting each signal in turn, smoothing it, and then pressing the "Load Database" button. This is explained in the following paragraphs.

Pressing a signal button pops up a "processing widget". This allows you to smooth the signal in time (and in space if its a 2D signal). The signal value that will be loaded into the table is highlighted on the plot with a * (this will be referred to as the "load value"). The signal is smoothed using a boxcar kernel. To smooth, enter a width for the boxcar, and press smooth. You will then be asked whether you wish to use the smoothed signal. If you press "Yes", the * will move to the smoothed signal at the time of interest, indicating that the smoothed signal will be used for the load value. If you press "No", the * will remain on the unsmoothed signal. Pressing clear resets the load value to the unsmoothed signal. If another smoothing window is entered, the smoothing will be applied to the original TRANSP output - it is not cumulative.

Press "Accept" to signify that the signal is ready to be loaded into the table. This will change the corresponding button on the signal widget to upper case. If "Hide" is pressed without "Accept", the signal will not be loaded into the table, but the smoothing will be remembered. When "Load Database" is pressed, only those signals whose corresponding buttons are shown in upper case will be loaded. Note that if it is desired to load *all* the columns with the same smoothing, it is not necessary to press "Accept" when examining the smoothing of a given signal. Remember - "Load Database" is used only when it is desired to smooth each signal differently, or to only load a subset of the signals,

If "Quit" is pressed on the signal widget after changes have been made but before they have been saved to the TRANSP_SCALARS table, the user will be prompted to make sure it is desired to quit, as after quitting all information on the smoothing is lost, and the load values are cleared. The user would then have to start over on preparing the

outputs for loading into the table.

The source code for TS is stored in the same CMS library as all the other TRANSP tools: `CMOD$MODELS:[TRANSPORT.IDL.CMSREF]`. If more columns are added to the `TRANSP_SCALARS` table (something TWF or JAS would do on request), TS would need to be modified to include them. TS would also need to be modified if it is desired to change the TRANSP output signal corresponding to a given column, or to delete a column. The file `TS_SIGNALS.PRO` in the aforementioned CMS library contains the information for each column TS can load into the table.

One column is special - `T_INTRO_FLAT`. This stores the difference in time between the time of interest and the beginning of the current flattop. The time of flattop must be determined manually - a special widget is popped up to do this. The widget plots `Bt`, `Ip`, and `Vsurf` vs. time. The time of interest that will be loaded into the core table is shown with a dashed line. The selected flattop start time is shown with a solid line. The solid line can be moved by pressing and holding any of the mouse buttons. Once the time is identified, press "Accept" to accept it. Note that `T_INTRO_FLAT` is NOT loaded automatically when "Load All" is pressed!

B.15 UFILES_AND_IDL.DOC

UFILES_AND_IDL.DOC

97.05.12

=====

Several FORTRAN subroutines are available to read and write UFILES from IDL and MDSplus trees. These are all stored in the shareable images UFMDSIO_VAX.EXE and UFMDSIO_AXP.EXE stored in the directory TRANSP\$:[USER.UFMDSIO].

Setup

At time of writing, UFMDSIO is not contained as part of the official TRANSP distribution. It therefore needs to be built manually. Hopefully it will be integrated with TRANSP to avoid this necessity.

In the meantime, the procedure REBUILD.COM in TRANSP\$:[USER.UFMDSIO] can be run to rebuild the sharable images manually. It will rebuild ONLY the image for the machine type on which it is run; to rebuild both versions, it must be run on both an AXP and a VAX.

The option files UFMDSIO_VAX.OPT and UFMDSIO_AXP.OPT list the TRANSP subroutine libraries on which UFMDSIO depends. The TRANSP manager must monitor the repository for changes to these libraries in order to keep UFMDSIO up-to-date.

Use

To use the sharable image library, a DCL logical name must be defined to be the full path spec of the image file. I use the logical name UFMDSIO. So for example:

```
$ define UFMDSIO TRANSP$:[USER.UFMDSIO]UFMDSIO_xxx.EXE
```

where 'xxx' is the machine type (either AXP or VAX).

The subroutines in UFMDSIO can then be called from IDL using a CALL_EXTERNAL command:

```
IDL> result=CALL_EXTERNAL('UFMDSIO',subroutine_name,p0,p1,...)
```


UFILE naming

UFILE names are composed as follows: Pnnnnn.XXX where

P is a one character "prefix"
 nnnnn is a 5 digit "shot number"
 XXX is a three letter "extension" (for input UFILES, this
 would be the trigraph)

In the subroutine parameter lists below, these are denoted by variable names prefix, exten, and ufile_shot respectively. ufile_shot MUST be a long integer.

Subroutines

UFtoMDS (tree,node,mds_shot,prefix,exten,ufile_shot)

reads UFILE and writes contents to node NODE, for shot MDS_SHOT of MDSplus tree TREE

MDStoUF (tree,mds_shot,node,label,prefix,exten,ufile_shot,status)

reads node NODE from shot MDS_SHOT of tree TREE and writes to UFILE. Can write LABEL for UFILE

UFtoIDLread (prefix,exten,ufile_shot,ndim0,ndim1)

call this first to read from a UFILE to an IDL variable. This returns the size of the variable, so that it can be properly initialized before read with the data (see below)

UFtoIDLget (variable,dim0,dim1)

after UFtoIDLread is called to determine the size of the variable, call this to actually return the data. For example, in IDL do:

```
x=call_external('UFMDSIO','UFtoIDLread','X','XXX',100011,n0,n1)
```

```
dim0=fltarr(n0)
dim1=fltarr(n1)
data=fltarr(n0,n1)
x=call_external('UFMDSIO','UFtoIDLget', data, dim0, dim1)
```

```
IDLtoUF (var,dim0,dim1,ndim0,ndim1,label,label0,label1,cmod_shot,
>      prefix,exten,ufile_shot)
```

writes the IDL data in variable VAR with dimensions DIM0 and DIM1 to UFILE. Uses label LABEL for data and LABEL0, LABEL1 for dimensions. CMOD_SHOT is used as the shot date in the UFILE for reference.

B.16 UPDATING_TRANSP.DOC

UPDATING_TRANSP.DOC
 =====

96.12.19

To receive code updates from the PPPL repository, follow the steps in TRANSP\$: [CMSREF.VAXDOC]PSHARE_UPDATE.DOC. This document provides a brief outline of this document, but is by no means a substitute for it. Also provided is information on how to update the TRANSP tree structure after a code update.

Outline of PSHARE_UPDATE.DOC:

- A. Make sure xshare is up-to-date with respect to ("wrt") pppl and pshare is up-to-date wrt VMS. This should be done automatically when running xshare's update.daemon and VMS's CODE_DAEMON
- B. LOCK TRANSP (PSHARE_UPDATE step 1)
 - edit CMOD\$MODELS: [TRANSPORT.IDL]TRANSP_SETUP: comment out goto NOFORCE
 - BUILD.LOCK doesn't matter since we aren't using TRMANAGER
- C. STOP CODE_DAEMON on TRANSP_VAX, and remove cron jobs from xshare and pshare (crontab -r)
- D. cvs -n update pshare
 - watch for conflicts or modified signals (C or M). If didn't make any mods locally, then conflicts are not caused by us, so can delete file and re-update
 - if make local mods, COMMIT THEM after E.
- E. run update.daemon on pshare
 - conflicts should be taken care of, so should just merge new code
 - will rebuild and take a while, but should be able to continue to step F after cvs update unless problems in code
 - if made local changes, COMMIT THEM! SEE PSHARE_UPDATE.DOC step 2 and UNIX_COMMIT.DOC
- F. run update.daemon on xshare

- shouldn't need to do anything as it should already be up-to-date wrt pppl
 - if commit changes at step D, will pick these up now
- G. `cv`s -n update xshare and pshare again
- both should be in same state wrt repository now
- H. run GETNEW on VMS
- I. run GETNEW_INSTALL on VMS
- J. restart pshare's cron job (`crontab ~pshare/cron_pshare`)
- needed for VMS's CODE_DAEMON
- K. run CODE_DAEMON on VMS
- L. if `TRANSP$:[CMSREF.MISC]TRDATGEN.SPEC` or `TRANSP$:[CMSREF.OUTCOR]XPLOT7.FOR` changed, then need to update tree structures. See the next section of this document.
- M. restart xshare's update.daemon (`crontab ~xshare/cron_xshare`)
- N. remove lock in `TRANSP_SETUP.COM`

Should be back in business.

Updating XTranspin

XTranspin is not built as part of the CODE_DAEMON job on VMS. Building must be done manually. Follow the instructions in `TRANSP$:[CMSREF.TRANSPI]INSTALL.VMS`. Basically, all that is needed is `MMS/DESC=XTRANSPI ALL`.

Updating the TRANSP tree

The TRANSP tree INPUTS and OUTPUTS structures are constructed from the TRDATGEN.SPEC and XPLOT7.FOR files respectively. If new inputs or outputs are added to TRANSP through changes in these files, it is necessary to update the TRANSP tree structure as well.

All programs and procedures necessary to update the tree structure are stored in CMOD\$MODELS:[TRANSPORT.TREE] and its subdirectories. The command procedure UPGRADE.COM takes care of the input nodes, the output nodes, and the default scope pads stored in CMOD\$MODELS:[SCOPES].

You must specify the shot number of the tree to update. Usually this would be the model, shot=-1. You would need to update another shot if for example that shot was duplicated from an old shot with an out-of-date structure. The update scripts keep a log of the nodes added, deleted, and changed, and write this log to the node \TOP:MOD_LOG.

Inputs

UPGRADE.COM calls UPGRADE_INPUTS.COM that in turn runs the IDL procedure [.CMSREF]UPGRADE_INPUTS.PRO. This procedure loads the input definitions from the file TRDATGEN.SPEC; they are the trigraphs found on lines beginning with the * character. Not all inputs are relevant to C-Mod. The irrelevant ones are filtered out by the function REMOVE_DATA. PLEASE READ THE COMMENTS IN THIS FUNCTION!!! Once the useful inputs are identified, the procedure checks to see if they are already included in the .INPUTS structure, and if not, adds them and their subsidiary nodes (COMMENT, LABEL, PROCESSING, SOURCE, SOURCE_PRO). It also deletes nodes in the .INPUTS structure that are no longer TRANSP inputs.

Outputs

UPGRADE.COM calls UPGRADE_OUTPUTS.COM that in turn runs the IDL procedure [.CMSREF]UPGRADE_OUTPUTS.PRO. This procedure loads the output definitions from the file XPLOT7.FOR; they are included in the lines with the characters C% at the start. Once again, the irrelevant outputs are filtered using the procedure READ_SWITCH (READ THE COMMENTS!!!). This time, however, instead of filtering individual

lines from the file, it filters blocks of lines, as the outputs are grouped into categories delimited by the lines beginning with C\$. The filter program acts on the C\$ lines, not on the C% lines.

Once the relevant outputs are identified, the procedure first checks to see if any nodes in the \TRANSP_OUT structure are not in the list of relevant nodes. If it finds any it deletes them. The procedure then checks to make sure that the nodes in \ONE_D, \TWO_D, and \IN_CHK all point to nodes in \TRANSP_OUT that exist. If not, they are flagged for "redirection". For example, if the TRANSP output ETA becomes obsolete after an update, the node \TWO_D:ETA, which points to \TRANSP_OUT:ETA, will return an error when checked, which flags it for redirection. The user is then supposed to decide whether to delete the node \TWO_D:ETA, or to have it point to a different \TRANSP_OUT node (in this case, probably \TRANSP_OUT:ETA_USE). This feature prevents scope pads and procedures which depend on the node names from breaking immediately after a code update. If redirected nodes are identified, the TRANSP manager should make every effort to change the tools and procedures using that node to the current version of the node name, as redirection is not guaranteed to be 100% foolproof.

The potential exists for the following scenario, for example: The 1D TRANSP output FOO exists and is included in the \TRANSP_OUT structure. Later on, the output becomes superceded by GUB, so that \ONE_D:FOO is redirected to \TRANSP_OUT:GUB from \TRANSP_OUT:FOO. Then later still, it is decided to have another TRANSP output named FOO, which translates to a completely different parameter. At this point the only logical course is to remove the redirection of FOO so that it points to the correct output, and to have users correct their tools and procedures that wish to use the old version of FOO.

After checking for redirected nodes, the procedure adds new nodes. Actually, the procedure scans for nodes needing to be redirected before adding new nodes, and actually carries out the redirection after adding new nodes, so that old outputs can be redirected to the new nodes.

Scopes

The scope pads CMOD\$MODELS:[TRANSPORT.SCOPE}*].PAD are updated with

respect to the TRANSP structures as well. They are broken down into a 1D pad, and 2D pads vs. time and vs. position, for each of the inputs and outputs. The pads are generated from the nodes under \INPUTS, \ONE_D, and \TWO_D. The IDL procedure [.CMSREF]UPGRADE_SCOPES.PRO is used for this

B.17 USER_DEFAULTS.DOC

USER_DEFAULTS.DOC

97.05.24

=====

=====

*** THE USER DEFAULTS SYSTEM IS NOT CURRENTLY AVAILABLE ***

PRETRANSP provides the option for each user to store their own default processing steps, source signals, and namelist. The current method for implementing this, requiring each user to store their own copy of the TRANSP tree model in their own file area, has not been tested, so this option has been disabled.

If user defaults were enabled, a new shot would be created from a copy of the model stored in the directory specified by the logical name TRANSP\$your_username. For example, my model would be stored in the directory TRANSP\$SCHACHTER. If the logical name TRANSP\$your_username is not defined, the model in TRANSP\$DATA will be used.

This system has never been sufficiently tested to allow general use. Furthermore, the requirement that each user store a copy of the model in their own area complicates the process of keeping the tree structure up-to-date (see UPDATING_TRANSP.DOC).

Therefore the options that allow individual defaults to be stored have been disabled. Either intensive testing, or a new system of storing the defaults, would be required before these features could be implemented.

In the meantime, do NOT define the logical name TRANSP\$your_username (for me this would be TRANSP\$SCHACHTER). Though the tools are set to ignore this logical name and therefore function properly, it is best not to have it defined to avoid any confusion.

Bibliography

- [1] R. D. Hazeltine and J. D. Meiss. *Plasma Confinement*. Addison Wesley, 1992.
- [2] L. A. Artsimovitch. Tokamak devices. *Nucl. Fusion*, **12**:215, 1972.
- [3] D. T. Garnier. *Lithium Pellet Injection Experiments on the Alcator C-Mod Tokamak*. PhD thesis, Massachusetts Institute of Technology, 1996.
- [4] J. P. Freidberg. *Ideal Magnetohydrodynamics*. Plenum Press, 1987.
- [5] J. Wesson. *Tokamaks*. Oxford University Press, 1987.
- [6] I. H. Hutchinson. Alcator C-Mod program. *J. Fusion Energy*, **10**(4):269, 1991.
- [7] M. Greenwald et al. Energy confinement of high-density pellet-fueled plasmas in the Alcator C tokamak. *Phys. Rev. Lett.*, **53**(4):352, 1984.
- [8] M. Greenwald et al. Confinement of ohmic and ICRF heated plasmas in Alcator C-Mod. In *Plasma Physics and Controlled Nuclear Fusion Research*, volume 2, page 4. International Atomic Energy Agency, November 1994.
- [9] M. Greenwald et al. Transport experiments in Alcator C-Mod. *Phys. Plasmas*, **2**(6):2308, 1995.
- [10] Y. Takase et al. Radiofrequency-heated enhanced confinement modes in the Alcator C-Mod tokamak. *Phys. Plasmas*, **4**(5):1647, 1997.
- [11] M. Greenwald et al. H-Mode confinement in Alcator C-Mod. *Nucl. Fusion*, **37**:793, 1997.
- [12] Y. Takase, P. T. Bonoli, S. N. Golovato, and M. Porkolab. ICRF heating scenarios in Alcator C-Mod. In *Proceedings of the Tenth Topical Conference on Radio-Frequency Power in Plasmas*, page 56. AIP, 1995.
- [13] Y. Takase et al. Survey of ICRF heating experiments and enhanced performance modes in Alcator C-Mod. *Plasma Phys. Control. Fusion*, **38**:2215, 1996.

- [14] S. N. Golovato et al. ICRF heating in the Alcator C-Mod tokamak. In *Proceedings of the Eleventh Topical Conference on Radio-Frequency Power in Plasmas*, page 23. AIP, 1996.
- [15] U. Frisch and S. A. Orszag. Turbulence: challenges for theory and experiment. *Phys. Today*, page 24, January 1990.
- [16] S. Ichimaru. *Basic principles of plasma physics: a statistical approach*. Addison-Wesley Book Publishing Company, Inc, 1973.
- [17] S. I. Braginskii. Transport processes in a plasma. In M. A. Leontovich, editor, *Reviews of Plasma Physics*, page 205. Consultants Bureau, New York, 1965.
- [18] F. L. Hinton and R. D. Hazeltine. Theory of plasma transport in toroidal confinement systems. *Rev. Mod. Phys.*, **48**(2):239, 1976.
- [19] B. Blackwell et al. Energy and impurity transport in the Alcator C tokamak. In *Plasma Physics and Controlled Nuclear Fusion Research*, volume 2, page 27. International Atomic Energy Agency, 1983.
- [20] P. C. Liewer. Measurements of microturbulence in tokamaks and comparisons with theories of turbulence and anomalous transport. *Nucl. Fusion*, **25**(5):543, 1985.
- [21] J. D. Callen. Anomalous transport in tokamaks: transport task force reviews. *Phys. Fluids B*, **2**(12):2869, 1990.
- [22] J. W. Connor and H. R. Wilson. Survey of theories of anomalous transport. *Plasma Phys. Control. Fusion*, **36**:719, 1994.
- [23] F. Wagner and U. Stroth. Transport in toroidal devices — the experimentalist's view. *Plasma Phys. Control. Fusion*, **35**:1321, 1993.
- [24] J. D. Callen. Transport processes in magnetically confined plasmas. *Phys. Fluids B*, **4**(7):2142, 1992.
- [25] B. A. Carreras. Transport mechanisms acting in toroidal devices: a theoretician's view. *Plasma Phys. Control. Fusion*, **34**(13):1825, 1992.
- [26] R. J. Bickerton. Magnetic turbulence and the transport of energy and particles in tokamaks. *Plasma Phys. Control. Fusion*, **39**:339, 1997.
- [27] F. F. Chen. *Introduction to Plasma Physics and Controlled Fusion*, volume 1. Plenum Press, 2nd edition, 1984.
- [28] M. A. Beer. *Gyrofluid Models of Turbulent Transport in Tokamaks*. PhD thesis, Princeton University, 1995.

- [29] B. Coppi, M. N. Rosenbluth, and R. Z. Sagdeev. Instabilities due to temperature gradients in complex magnetic field configurations. *Phys. Fluids*, **10**:582, 1967.
- [30] G. S. Lee and P. H. Diamond. Theory of ion temperature gradient driven turbulence in tokamaks. *Phys. Fluids*, **29**:3291, 1986.
- [31] M. Kotschenreuther, W. Dorland, M. A. Beer, and G. W. Hammett. Quantitative predictions of tokamak energy confinement from first principles simulations with kinetic effects. *Phys. Plasmas*, **2**(6):2381, 1995.
- [32] B. B. Kadomtsev. *Plasma Turbulence*. Academic Press, London, 1965.
- [33] R. E. Waltz, J. C. DeBoo, and M. N. Rosenbluth. Magnetic-field scaling of dimensionally similar tokamak discharges. *Phys. Rev. Lett.*, **65**(19):2390, 1990.
- [34] R. Goldston. Energy confinement scaling in tokamaks: some implications of recent experiments with ohmic and strong auxiliary heating. *Plasma Phys. Control. Fusion*, **26**(1A):87, 1984.
- [35] F. Wagner et al. Regime of improved confinement and high beta in neutral-beam-heated divertor discharges of the ASDEX tokamak. *Phys. Rev. Lett.*, **49**:1408, 1982.
- [36] P. N. Yushmanov et al. Scalings for tokamak energy confinement. *Nucl. Fusion*, **30**(10):1999, 1990.
- [37] D. P. Schissel and the H-Mode Database Working Group. Analysis of the ITER H-Mode confinement database. In *Proceedings of the 20th European Conference on Controlled Fusion and Plasma Physics*. European Physical Society, 1993.
- [38] K. Thomsen et al. ITER H-Mode confinement database update. *Nucl. Fusion*, **34**(1):131, 1994.
- [39] K. H. Burrell et al. Status of local transport measurements and analysis in toroidal devices. *Phys. Fluids B*, **2**(12):2904, 1990.
- [40] S. R. de Groot and P. Mazur. *Non-Equilibrium Thermodynamics*. Dover, 1984.
- [41] R. J. Bickerton, J. W. Connor, and J. B. Taylor. Diffusion driven plasma currents and bootstrap tokamak. *Nature (Physical Science)*, **229**:110, 1971.
- [42] A. A. Ware. Pinch effect for trapped particles in a tokamak. *Phys. Rev. Lett.*, **25**:15, 1970.
- [43] L. Onsager. Reciprocal relations in irreversible processes.I. *Phys. Rev.*, **37**:405, 1931.

- [44] L. Onsager. Reciprocal relations in irreversible processes.II. *Phys. Rev.*, **38**:2265, 1931.
- [45] J. P. Christiansen et al. Evidence for a local diffusive model of transport in a tokamak. *Plasma Phys. Control. Fusion*, **34**(13):1881, 1992.
- [46] T. C. Luce, C. C. Petty, et al. Experimental constraints on transport. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, November 1994.
- [47] R. J. Hawryluk. An empirical approach to tokamak transport. In B. Coppi et al., editors, *Physics of Plasmas Close to Thermonuclear Conditions*, volume 1, page 19. CEC, Brussels, 1980.
- [48] B. Coppi. Nonclassical transport and the principle of profile consistency. *Comments Plasma Phys. Control. Fusion*, **5**(6):261, 1979.
- [49] W. Manheimer and J. P. Boris. Marginal stability analysis — a simpler approach to anomalous transport in plasmas. *Comments Plasma Phys. Control. Fusion*, **3**(1):15, 1977.
- [50] D. McCune, R. Goldston, R. McCann, and R. Wieland. Generalization of PPPL transport analysis code TRANSP for time dependent magnetic field geometry. *Bull. Am. Phys. Soc.*, **29**(8):1304, 1984.
- [51] S. Tamor. Antic: A code for calculation of neutral transport in cylindrical plasmas. *J. Comput. Phys.*, **40**:104, 1981.
- [52] M. C. Zarnstorff et al. Parallel electric resistivity in the TFTR tokamak. *Phys. Fluids B*, **2**(8):1852, 1990.
- [53] S. P. Hirshman. Neoclassical current in a toroidally-confined multispecies plasma. *Phys. Fluids*, **21**(8):1295, 1978.
- [54] S. P. Hirshman, R. J. Harwryluk, and B. Birge. Neoclassical conductivity of a tokamak plasma. *Nucl. Fusion*, **17**(3):611, 1977.
- [55] L. L. Lao, H. S. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer. Reconstruction of current profile parameters and plasma shapes in tokamaks. *Nucl. Fusion*, **25**(11):1611, 1985.
- [56] L. L. Lao et al. Separation of β_p and ℓ_i in tokamaks of noncircular cross section. *Nucl. Fusion*, **25**:1421, 1985.
- [57] L. L. Lao et al. Equilibrium analysis of current profiles in tokamaks. *Nucl. Fusion*, **30**:1035, 1990.

- [58] L. L. Lao and T. H. Jensen. Magnetohydrodynamic equilibria of attached plasmas after loss of vertical stability in elongated tokamaks. *Nucl. Fusion*, **31**:1909, 1991.
- [59] S. P. Hirshman, U. Schwenn, and J. Nuhrenberg. Improved radial differencing for three-dimensional magnetohydrodynamic equilibrium calculations. *J. Comput. Phys.*, **87**(2):396, 1990.
- [60] P. J. O'Shea. *Measurements of ICRF power deposition and thermal transport with an ECE grating polychromator on the Alcator C-Mod tokamak*. PhD thesis, Massachusetts Institute of Technology, 1997.
- [61] D. N. Smithe et al. An algorithm for the calculation of three-dimensional ICRF fields in tokamak geometry. *Nucl. Fusion*, **27**:1319, 1987.
- [62] D. N. Smithe et al. SNARF analysis of ICRF heating on TFTR. In *Proceedings of the Eighth Topical Conference on Radio-Frequency Power in Plasmas*, page 190. AIP, 1993.
- [63] M. Evrard, J. Ongena, and D. van Eester. Improved dielectric tensor in the ICRF module of TRANSP. In *Proceedings of the Tenth Topical Conference on Radio-Frequency Power in Plasmas*. AIP, 1995.
- [64] G. W. Hammett. *Fast Ion Studies of Ion Cyclotron Heating in the PLT Tokamak*. PhD thesis, Princeton University, 1986.
- [65] D. McCune. Private communication, 1997.
- [66] M. Brambilla. Theory of Bernstein wave coupling with loop antennas. *Nucl. Fusion*, **28**(4):549, 1988.
- [67] M. Brambilla and T. Krucken. Numerical simulation of ion cyclotron heating of hot tokamak plasmas. *Nucl. Fusion*, **28**(10):1813, 1988.
- [68] P. Bonoli. Private communication, 1997.
- [69] C. Kurz and C. L. Fiore. Neutral particle diagnostics for Alcator C-Mod. *Rev. Sci. Instrum.*, **61**(10):3119, 1990.
- [70] T. C. Hsu. *The Submillimeter Wave Electron Cyclotron Emission Diagnostic for the Alcator C-Mod Tokamak*. PhD thesis, Massachusetts Institute of Technology, 1993.
- [71] J. E. Rice, F. Bombarda, M. A. Graf, E. S. Marmor, and Y. Wang. Observations of Alcator C-Mod plasmas from a five chord high energy resolution x-ray spectrometer array. *Rev. Sci. Instrum.*, **66**(1):752, 1995.

- [72] J. E. Rice. Private communication, 1997.
- [73] S. M. Kaye and the ITER Confinement and Database Modeling Group. Projection of ITER performance using the multimachine L and H mode databases. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, November 1994.
- [74] K. S. Riedel. Tokamak to tokamak variation and collinearity in scaling laws. *Nucl. Fusion*, **30**(4):755, 1990.
- [75] K. S. Riedel and S. M. Kaye. Uncertainties associated with extrapolating L-Mode energy confinement to ITER and CIT. *Nucl. Fusion*, **30**(4):731, 1990.
- [76] S. M. Kaye et al. Status of global energy confinement studies. *Phys. Fluids B*, **2**(12), 1990.
- [77] D. A. Belsley, E. Kuh, and R. E. Welsch. *Regression diagnostics: identifying influential data and sources of collinearity*. John Wiley and Sons, 1980.
- [78] G. W. Bluman and K. Sukeyuki. *Symmetries and Differential Equations*. Springer Verlag, New York, 1989.
- [79] J. W. Connor and J. B. Taylor. Scaling laws for plasma confinement. *Nucl. Fusion*, **17**(5):1047, 1977.
- [80] T. C. Luce and C. C. Petty. Dimensionless parameter scaling for transport physics and performance extrapolation. *Bull. Am. Phys. Soc.*, **41**(7):1572, 1996.
- [81] T. C. Luce, C. C. Petty, B. Balet, and J. G. Cordey. Implications from dimensionless parameter scaling experiments. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, October 1996.
- [82] C. C. Petty, T. C. Luce, B. Balet, J. P. Christiansen, and J. G. Cordey. Nondimensional transport experiments on DIII-D and projections to an ignition tokamak. In *Proceedings of the 23rd European Conference on Controlled Fusion and Plasma Physics*, volume 23. European Physical Society, 1996.
- [83] J. G. Cordey et al. A review of the dimensionless parameter scaling studies. *Plasma Phys. Control. Fusion*, **38**:A67, 1996.
- [84] B. B. Kadomstev. Tokamaks and dimensional analysis. *Sov. J. Plasma Phys.*, **1**(4):295, 1975.
- [85] P. M. Gerhart and R. J. Gross. *Fundamentals of fluid mechanics*. Addison-Wesley Publishing Company, 1985.

- [86] J. W. Connor. Invariance principles and plasma confinement. *Plasma Phys. Control. Fusion*, **30**(6):619, 1988.
- [87] X. Garbet and R. E. Waltz. Action at distance and Bohm scaling of turbulence in tokamaks. *Phys. Plasmas*, **3**(5):1898, 1996.
- [88] D. E. Newman, B. A. Carreras, P. H. Diamond, and T. S. Hahm. The dynamics of marginality and self-organized criticality as a paradigm for turbulent transport. *Phys. Plasmas*, **3**(5):1858, 1996.
- [89] C. C. Petty, T. C. Luce, et al. Nondimensional transport scaling in DIII-D: Bohm versus gyroBohm resolved. *Phys. Plasmas*, **2**(6):2342, 1995.
- [90] International Atomic Energy Agency, Vienna, Austria. *ITER Interim Design Report*, July 1995.
- [91] I. H. Hutchinson. *Principles of plasma diagnostics*. Cambridge University Press, 1987.
- [92] E. S. Marmor. The Alcator C-Mod diagnostic complement. *Rev. Sci. Instrum.*, **63**(10):4695, 1992.
- [93] T. C. Luke. *Measurement of particle transport coefficients on Alcator C-Mod*. PhD thesis, Massachusetts Institute of Technology, 1994.
- [94] J. A. Casey, R. Watterson, F. Tambini, E. Rollins, and B. Chin. Construction of a scanning two-dimensional Thomson scattering system for Alcator C-Mod. *Rev. Sci. Instrum.*, **63**(10):4950, 1992.
- [95] R. Watterson and K. Chen. Status of the Alcator C-Mod scanning two-dimensional Thomson scattering diagnostic. *Rev. Sci. Instrum.*, **61**(10):2867, 1990.
- [96] C. L. Fiore, R. Boivin, R. S. Granetz, T. Fuller, and C. Kurz. Status of the neutron diagnostic experiment for Alcator C-Mod. *Rev. Sci. Instrum.*, **63**(10):4530, 1992.
- [97] J. A. Goetz et al. Power balance and scaling of the radiated power in the divertor and main plasma of Alcator C-Mod. *J. Nucl. Mater.*, **220-222**:971, 1995.
- [98] M. E. Foord, E. S. Marmor, and J. L. Terry. Multichannel light detector system for visible continuum measurements on Alcator C. *Rev. Sci. Instrum.*, **53**(9):1407, 1982.
- [99] R. S. Granetz, I. H. Hutchinson, J. Gerolamo, W. Pina, and C. Tsui. Magnetic diagnostics in Alcator C-Mod. *Rev. Sci. Instrum.*, **61**(10):2967, 1990.

- [100] A. Hubbard. Private communication, 1996.
- [101] S. Wolfe. Private communication, 1996.
- [102] P. R. Bevington. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill Book Company, 1969.
- [103] M. Greenwald. Private communication, 1997.
- [104] Y. Takase. Private communication, 1996.
- [105] G. K. Bhattacharyya and R. A. Johnson. *Statistical concepts and methods*. John Wiley and Sons, 1977.
- [106] N. R. Draper and H. Smith. *Applied regression analysis*. John Wiley and Sons, 1966.
- [107] R. D. Cook and S. Weisberg. *Residuals and influence in regression*. Chapman and Hall, 1982.
- [108] J. L. Luxon and L. G. Davis. Big Dee — a flexible facility operating near breakeven conditions. *Fusion Technol.*, **8**:441, 1985.
- [109] C. C. Petty et al. Gyroradius scaling of electron and ion transport. *Phys. Rev. Lett.*, **74**(10):1763, 1995.
- [110] T. C. Luce and C. C. Petty. Experimental determination of the dimensionless parameter scaling of energy transport in tokamaks. In *Proceedings of the 22nd European Conference on Controlled Fusion and Plasma Physics*. European Physical Society, 1995.
- [111] P. Gohil, K. H. Burrell, R. J. Groebner, and R. P. Seraydarian. High spatial and temporal resolution visible spectroscopy of the plasma edge in DIII-D. *Rev. Sci. Instrum.*, **61**(10):2949, 1990.
- [112] J. G. Cordey and the JET Team. Energy confinement and H-Mode power threshold scaling in JET with ITER dimensionless parameters. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, October 1996.
- [113] J. P. Christiansen et al. The ρ^* scaling experiments for ELMy H-Modes in JET. *Bull. Am. Phys. Soc.*, **40**(11):1765, 1995.
- [114] J. P. Christiansen et al. The scaling of transport with normalized Larmor radius in JET. *Nucl. Fusion*, **33**(6):863, 1993.

- [115] R. J. Bickerton. Results and plans for the Joint European Torus. *J. Fusion Energy*, **7**:115, 1988.
- [116] F. W. Perkins et al. Nondimensional transport scaling in the Tokamak Fusion Test Reactor: is tokamak transport Bohm or gyroBohm? *Phys. Fluids B*, **5**(2):477, 1993.
- [117] The Tore Supra Group. First experiments in Tore Supra. In *Plasma Physics and Controlled Nuclear Fusion Research*, volume 1, page 3. International Atomic Energy Agency, 1990.
- [118] C. Laviron et al. Local analysis of transport and turbulence in Tore Supra. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, October 1996.
- [119] U. Stroth, G. Kuhner, H. Maassberg, H. Ringler, and the W7-AS Team. Dimensionally similar discharges in the W7-AS stellarator. *Phys. Rev. Lett.*, **70**(7):936, 1993.
- [120] M. Kotschenreuther et al. First principles calculations of tokamak energy transport. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, November 1990.
- [121] A. M. Dimits and W. W. Lee. Partially linearized algorithms in gyrokinetic particle simulation. *J. Comput. Phys.*, **107**(2):309, 1993.
- [122] G. W. Hammett, W. Dorland, and F. W. Perkins. Fluid models of phase mixing, Landau damping, and nonlinear gyrokinetic dynamics. *Phys. Fluids B*, **4**(7):2052, 1992.
- [123] G. W. Hammett and F. W. Perkins. Fluid moment models for Landau damping with application to the ion-temperature-gradient instability. *Phys. Rev. Lett.*, **64**(25):3019, 1990.
- [124] W. Dorland et al. Comparisons of nonlinear toroidal turbulence simulations with experiment. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, November 1994.
- [125] M. Kotschenreuther et al. Simulations for confinement in near-fusion experiments. In *Plasma Physics and Controlled Nuclear Fusion Research*, volume 2, page 11. International Atomic Energy Agency, 1992.
- [126] W. Dorland, M. Kotschenreuther, Q. P. Liu, M. A. Beer, and G. W. Hammett. 3D simulations of tokamak plasma turbulence: Experimental tests and predictions. In *Proceedings of the Joint Varenna-Lausanne International Workshop on Theory of Fusion Plasmas*, page 185, Varenna, Italy, 1996. Editrice Compositori.

- [127] S. C. Cowley, R. M. Kulsrud, and R. Sudan. Considerations of ion-temperature-gradient-driven turbulence. *Phys. Fluids*, **3**(10):2767, 1991.
- [128] R. E. Waltz, G. D. Kerbel, J. Milovitch, and G. W. Hammett. Advances in the simulation of toroidal gyro-Landau fluid model turbulence. *Phys. Plasmas*, **2**(6):2408, 1995.
- [129] W. Dorland. Private communication, 1997.
- [130] J. E. Rice, E. S. Marmor, F. Bombarda, and L. Qu. X-ray observations of central toroidal rotation in ohmic Alcator C-Mod plasmas. *Nucl. Fusion*, **37**(3):421, 1997.
- [131] T. S. Hahm and K. H. Burrell. Exb flow shear effects on radial correlation length of turbulence and gyroradius scaling of confinement. *Phys. Plasmas*, **3**(1):427, 1996.
- [132] W. L. Rowan et al. Neutral beam diagnostic for Alcator C-Mod. *Rev. Sci. Instrum.*, **68**(1):935, 1997.
- [133] R. C. Isler. An overview of charge exchange spectroscopy as a plasma diagnostic. *Plasma Phys. Control. Fusion*, **36**:171, 1994.
- [134] A. E. Hubbard et al. Local plasma parameters and H-Mode threshold in Alcator C-Mod. In *Plasma Physics and Controlled Nuclear Fusion Research*. International Atomic Energy Agency, October 1996.
- [135] A. J. Wootton et al. Fluctuations and anomalous transport in tokamaks. *Phys. Fluids B*, **2**(12):2879, 1990.
- [136] R. J. Fonck, P. A. Duperrex, and S. F. Paul. Plasma fluctuation measurements in tokamaks using beam-plasma interactions. *Rev. Sci. Instrum.*, **61**:3487, November 1990.
- [137] T. W. Fredian, J. A. Stillerman, and M. Greenwald. Data acquisition for Alcator C-Mod. *Rev. Sci. Instrum.*, **68**(1):935, 1997.
- [138] J. A. Stillerman, T. W. Fredian, K. A. Klare, and G. Manduchi. MDSplus data acquisition system. *Rev. Sci. Instrum.*, **68**(1):939, 1997.

Acknowledgments

After spending many months composing an impersonal treatise on the work I have done over the last n years, I am glad that I can now take the time to offer a word of thanks to the many people who have contributed both to my research and to my entire graduate student experience here at MIT.

First and foremost, I would like to thank my advisor, Martin Greenwald. From my start on the deuterium pellet injector to the final stages of thesis preparation, we have worked closely together, sharing a sense of humor and a common work ethic. Every aspect of our interaction, from heady discussions about the nature of turbulent eddies, to nit-picking advice on the best way to wire up a cable, has been most enjoyable and very educational. His comprehensive knowledge of tokamak transport, and his grasp of and enthusiasm for scientific computing, made him a mentor beyond par.

Martin also had the good sense to hire Tom Fredian and Josh Stillerman. I am grateful to both Tom and Josh for taking the time to answer my innumerable and sometimes inane questions. Their skill in software design has been an inspiration to me. To Tom in particular, I am indebted for the many hours he spent with me debugging the TRANSP system. That it works as well as it does is more a testament of Tom's ability than my own.

I would also like to thank Doug McCune for the time he spent helping to get TRANSP working at C-Mod, and for reviewing the TRANSP-related sections in this thesis. I have learned much from his skill at managing and developing that "unwieldy FORTRAN analysis code", and hope to continue our collaboration in the future.

Although the total integrated time we have spent talking is short compared to my stay at MIT, in recent months Bill Dorland and I have been constant phone companions. With his help I was able to work on an exciting new theory that contributed much to the depth of this thesis. I am also thankful for his patient answers to my many questions, and for his encouragement in the face of approaching deadlines.

Thanks are due to my thesis reader, Prof. Ian Hutchinson, for his many insightful suggestions and comments. I must also thank Prof. Peter Catto for his often-sought help, and for sitting on my thesis committee. His ability to teach kinetic theory is commendable. Thanks are also due to the other members of my committee, Professors Jeff Freidberg and Kevin Wenzel.

Throughout this thesis I have relied heavily on the work of the scientists, engineers, technicians, and graduate students of the C-Mod group. A collective "thanks" is in order. I would like to single out the diagnostic efforts of John Rice (especially for the last minute important numbers), Amanda Hubbard, Tom Hsu, Peter O'Shea, Jim Irby, Tom Luke, Reich Watterson, John Goetz, and Earl Marmor. I would also like to thank Frank Silva, Tom Toland, Bob Childs, Ed Fitzgerald, Steve Kochan, and all the guys in the machine shop for their help with the pellet injector.

I acknowledge Yuichi Takase for his help with the intricacies of running TRANSP, and Paul Bonoli for spending time with me sorting out the SPRUCE bug. I am grateful

to both Yuichi and Paul for teaching me about RF heating in the process. At one point or another I must have sought the advice of every scientist on C-Mod, but I particularly remember bugging Bob Granetz, Earl Marmor, Jim Terry, and Steve Wolfe. Working in the back of the control room across from “spectroscopy corner” has been educational as well. Thanks are due to Earl, JT, and of course the Juan brothers, who always had something for me when I needed it.

Not all of my time at C-Mod has been spent in the control room. Several mornings I was up before the sun driving across the state as the assistant to Mr. Magnet, Paul Thomas. Presenting the joys of science to 2nd graders was a very rewarding experience, and I thank Paul for the opportunity. Thanks are also due to Valerie Censabella and Paul Rivenberg for their work in the outreach program and for their companionship attending the APS “seminars” in Denver. Their efforts also allowed me to work on the C-Mod Jr. video game, which was both fun and educational. My trips to the library were always enjoyable because of Kathy Powers and Jason Thomas.

I’ve made many friends at MIT over the years; some have moved on, and now that I too am leaving, I realize that it is them I will remember most fondly about my time here. I thank John, Tom H., Chris K., Tom L., Mike, Artur, Dan L., Paul, Darren, and Peter for the shining examples they set as C-Mod graduate students. To those students that still face the thesis battle, including Jim R., Joe, Chris R., Jim W., Dimitri, Maxim, and Sanjay, I say thanks for the companionship, and wish you the best in your work to come. I cannot summarize my years of friendship with Robert Nachtrieb in one sentence. I can say that it has been a pleasure to have worked with someone as like-minded as he, even if it meant not having someone to argue with.

I’d also like to thank Sean, Garth, Bruce, Janet, John & Karen, Dave & Karen, Raymie, Cliff & Josee, and Don for their support from afar. To my adopted family, Lionel, Anne, Debbie, Rob, Paul, and Carol Bercovitch, I say thank you for sharing with me the warmth of your home. I will miss you.

My interest in science and in coming to MIT began with my family, and to them I dedicate this thesis in part. It has been a long journey from styrofoam solar systems and bees to tokamaks and transport. All along my mother Cheryl has offered love and support for which I may not have always shown the proper gratitude. I can only hope that I have made her proud. I am forever destined to be known as “Melanie’s brother” at the U. of M. To both her and my sister Candice I say thanks for being there through the years. I also thank my stepfather Gordon for his encouragement and support. A regrettable aspect of my coming to MIT has been missing my little sister Samantha grow up. I hope that in the years to come we can spend more time together. I am also grateful to my Baba and Zeida for their love and encouragement.

This thesis is also dedicated to Kristin Hood. She faced the hardship of separation on top of adjusting to a new life in an unfamiliar place all alone, and still had the strength to support me through my darkest hours. Though we will never be able to recover the time we were apart, it will eventually be lost in the eternity that we will spend together. I’m coming home, my Love.