

**Experimental Data Analysis Techniques for Validation of Tokamak Impurity  
Transport Simulations**

by

Mark Alan Chilenski

B.S., Aeronautical and Astronautical Engineering (2010)  
University of Washington

Submitted to the Department of Nuclear Science and Engineering  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Nuclear Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Author .....  
Department of Nuclear Science and Engineering  
October 13, 2016

Certified by .....  
Martin J. Greenwald  
Senior Research Scientist and Deputy Director, Plasma Science and Fusion Center  
Thesis Supervisor

Certified by .....  
Anne E. White  
Cecil and Ida Green Associate Professor, Department of Nuclear Science and Engineering  
Thesis Reader

Accepted by .....  
Ju Li  
Battelle Energy Alliance Professor of Nuclear Engineering  
Chair, Department Committee on Graduate Students

This page intentionally left blank.



# Experimental Data Analysis Techniques for Validation of Tokamak Impurity Transport Simulations

by  
Mark Alan Chilenski

Submitted to the Department of Nuclear Science and Engineering  
on October 13, 2016, in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy in Nuclear Science and Engineering

## Abstract

This thesis presents two new techniques for analyzing data from impurity transport experiments in magnetically confined plasmas, with specific applications to the Alcator C-Mod tokamak. The objective in developing these new techniques is to improve the quality of the experimental results used to test simulations of turbulent transport: better characterization of the uncertainty in the experimental results will yield a better test of the simulations. Transport codes are highly sensitive to the gradients of the background temperature and density profiles, so the first half of this thesis presents a new approach to fitting tokamak profiles using nonstationary Gaussian process regression. This powerful technique overcomes many of the shortcomings of previous spline-based data smoothing techniques, and can even handle more complicated cases such as line-integrated measurements, computation of second derivatives, and 2D fitting of spatially- and temporally-resolved measurements. The second half of this thesis focusses on experimental measurements of impurity transport coefficients. It is shown that there are considerable shortcomings in existing point estimates of these quantities. Next, a linearized model of impurity transport data is constructed and used to estimate diagnostic requirements for impurity transport measurements. It is found that spatial resolution is more important than temporal resolution. Finally, a fully Bayesian approach to inferring experimental impurity transport coefficient profiles which overcomes the shortcomings of the previous approaches through use of multimodal nested sampling is developed and benchmarked using synthetic data. These tests reveal that uncertainties in the transport coefficient profiles previously attributed to uncertainties in the temperature and density profiles are in fact entirely explained by changes in the spline knot positions. Appendices are provided describing the extensive work done to determine the derivatives of stationary and nonstationary covariance kernels and the open source software developed as part of this thesis work. The techniques developed here will enable more rigorous benchmarking of turbulent transport simulations, with the ultimate goal of developing a predictive capability.

Thesis Supervisor: Martin J. Greenwald  
Senior Research Scientist and Deputy Director, Plasma Science and Fusion Center

Thesis Reader: Anne E. White  
Cecil and Ida Green Associate Professor, Department of Nuclear Science and Engineering

## Acknowledgements

This thesis is the product of just over six years of work at the Massachusetts Institute of Technology Plasma Science and Fusion Center under the supervision of Dr. Martin Greenwald, who gave me just the right balance of guidance and freedom to stumble into the interesting results presented herein. This work would not have been possible without Martin's support and top-notch mentorship over the years. This work would have been equally impossible without the ideas and continual prodding to keep making progress provided by my thesis committee: Prof. Anne White (thesis reader), Prof. Youssef Marzouk, Dr. John Rice and Prof. Jeffrey Freidberg (committee chair).

An individual's ability to conduct research is dictated primarily by the quality of the team they are working with, and I am very fortunate to have been able to work with the outstanding scientists, engineers, technicians and graduate students at Alcator C-Mod and the Levitated Dipole Experiment. In particular, at the risk of causing offense through omission, I would like to acknowledge the invaluable contributions of Nathan Howard, Matt Reinke, Tom Toland, Ron Rosati, Bill Parkin, Willy Burke, Ed Fitzgerald, Mark Iverson, Ian Faust, John Walk, Chi Gao, Yuri Podpaly, Luis Delgado-Aparicio, Greg Wallace, Bob Granetz, Jerry Hughes, Amanda Hubbard, Earl Marmar, Steve Wolfe, Steve Wukitch, Yijun Lin, John Wright, Mark London, Darin Ernst, Josh Stillerman, Tom Fredian, Dan Brunner, Norman Cao, Leigh Ann Kesler, Brandon Sorbom, Eric Edlund, Ted Golfinopoulos, Bob Mumgaard, Jay Kesner, Darren Garnier and Mike Mauel.

Of course, I would not have gotten to this point without the help of the numerous teachers I have had over the years, from the elementary school teachers who nurtured my budding interest in science and engineering, to the members of the HIT-SI experiment at the University of Washington who got me started in plasma physics research, all the way to the many excellent professors I have learned from here at MIT.

Special thanks goes to those who have facilitated my numerous side-projects, including Anne White and Mike Short who supported my development of the educational Geiger counter and Fred Harris who gave me the incredible opportunity to redesign the Kresge Hall percussion studio and modernize MIT's percussion equipment collection.

I am very thankful for the love and support of my family. In particular I am grateful to my parents, John and Sharon Chilenski, who inspire me to always learn and to always strive to do the best job possible. I am also grateful for the support I have received from my extended family over the years, in particular the hospitality and advice extended to me by my aunt and uncle, Barbara and Paul Schneeman.

I am also thankful for my friends, who have helped to keep me (relatively) sane over the years. In particular, I will mention the names of Sherry Gong, Matthew D'Asaro, Genia Vogman, Isaac Statnekov, Jonathan Biderman, Laird Bennion, Andrea Carney, Anna Wong, Ian Faust, John Walk and Evan Davis.

Finally, I am grateful to the sponsors of this research, who have contributed the following prose to this document: This material is based upon work conducted using the Alcator C-Mod tokamak, a DOE Office of Science user facility. This material is based upon work supported by the US Department of Energy, Office of Science, Office of Fusion Energy Sciences under Award Number DE-FCO2-99ER54512. This material is based upon work supported in part by the US Department of Energy Office of Science Graduate Research Fellowship Program (DOE SCGF), made possible in part by the American Recovery and Reinvestment Act of 2009, administered by ORISE-ORAU under contract number DE-AC05-06OR23100. The XEUS and LOWEUS spectrometers were developed at the LLNL EBIT lab. Work at LLNL was performed under the auspices of the US DOE under contract DE-AC52-07NA-27344. Some of the computations using STRAHL were carried out on the MIT PSFC parallel AMD Opteron/Infiniband cluster Loki.

# Contents

<b>List of Figures</b>	<b>12</b>
<b>List of Tables</b>	<b>21</b>
<b>List of Code Listings</b>	<b>23</b>
<b>Notation</b>	<b>24</b>
<b>1 Background</b>	<b>25</b>
1.1 Nuclear fusion and impurity transport . . . . .	25
1.2 Verification, validation and uncertainty quantification of turbulent transport simulations . . . . .	26
1.3 Measuring impurity transport . . . . .	28
1.3.1 Methods of impurity injection . . . . .	31
1.3.2 Diagnostics . . . . .	32
1.3.3 Inference methods . . . . .	35
1.4 Notation and terminology from probability theory and Bayesian inference . . . . .	37
1.4.1 Random variables, probability densities and Bayes' rule . . .	37
1.4.2 Summary statistics . . . . .	40
1.4.3 Sample statistics . . . . .	42
1.4.4 The uncertainty propagation equation . . . . .	44
1.4.5 Useful probability distributions . . . . .	44
<b>2 Profile fitting with Gaussian process regression</b>	<b>48</b>
2.1 Introduction . . . . .	48
2.2 Uncertainty quantification and the need for advanced profile fitting	50
2.2.1 Uncertainty quantification with profile inputs . . . . .	50
2.2.2 Profile fitting with splines . . . . .	51
2.3 Profile fitting with Gaussian process regression . . . . .	54
2.3.1 Basic details of Gaussian process regression . . . . .	55
2.3.2 An intuitive picture of Gaussian process regression . . . . .	57

2.3.3	Full details of Gaussian process regression . . . . .	59
2.3.4	Explanation of the result of Gaussian process regression . . .	64
2.3.5	Prediction of gradients and their uncertainties . . . . .	66
2.3.6	Selection of a covariance kernel and its hyperparameters . . .	68
2.3.7	Computing mean profiles and their uncertainties from MCMC output . . . . .	72
2.3.8	Drawing samples for uncertainty propagation . . . . .	73
2.3.9	Extensions to handle non-local measurements and outputs . .	74
2.3.10	Gaussian process regression versus Bayesian integrated data analysis . . . . .	80
2.4	Verification of Gaussian process regression for tokamak profile fit- ting using synthetic data . . . . .	80
2.4.1	Basic case . . . . .	80
2.4.2	Effect of errors in the independent variable . . . . .	82
2.4.3	Comparison of various covariance kernels . . . . .	85
2.5	Application of Gaussian process regression to Alcator C-Mod profiles	96
2.5.1	Basic fit . . . . .	96
2.5.2	Fit including TCI data . . . . .	105
2.5.3	Computing second derivatives to investigate rotation reversals	114
2.5.4	2D fitting of sawtooth-free data . . . . .	121
2.5.5	Fitting of pedestal data using the mtanh mean function . . .	130
2.6	Application of Gaussian process regression to impurity transport measurements . . . . .	139
2.7	Summary of profile fitting with Gaussian process regression . . . . .	141
<b>3</b>	<b>Inference of experimental impurity transport coefficient profiles</b>	<b>145</b>
3.1	Introduction and motivation: issues with existing approaches . . . . .	145
3.2	Basic details of impurity transport . . . . .	148
3.3	Intuition from time-series characteristics . . . . .	150
3.4	Bayesian analysis of required tolerances . . . . .	159
3.4.1	Derivation of the model . . . . .	160
3.4.2	Determining the required tolerances on $\tau_{\text{imp}}$ , $t_r$ and $b_{0.75}$ . . .	163
3.4.3	Determining the number of spatial channels, sampling rate and noise level . . . . .	167
3.5	Inferring impurity transport coefficient profiles is an inverse problem	178
3.5.1	The structure of the forward model . . . . .	178
3.5.2	The spectral model . . . . .	180
3.5.3	The probability model . . . . .	183
3.5.4	Performing the inference . . . . .	187

3.6	Application to synthetic data . . . . .	191
3.6.1	Basic case with local impurity density measurements . . . . .	191
3.6.2	Basic case with line-integrated spectral measurements . . . . .	195
3.6.3	Resolving the rise does not matter . . . . .	196
3.6.4	Free knots produce non-identifiable posterior distributions . . . . .	196
3.6.5	The uncertainty attributable to $n_e$ and $T_e$ is minimal . . . . .	199
3.6.6	The model evidence allows rigorous selection of the model complexity . . . . .	212
3.6.7	Testing parameter estimation and model selection on more complicated synthetic data . . . . .	217
3.7	Summary, conclusions and next steps . . . . .	227
<b>4</b>	<b>Conclusions and future directions</b>	<b>230</b>
4.1	Contributions and conclusions . . . . .	230
4.1.1	Profile data analysis . . . . .	230
4.1.2	Inference of impurity transport coefficient profiles . . . . .	232
4.2	Future directions . . . . .	233
4.2.1	Profile data analysis . . . . .	233
4.2.2	Impurity transport . . . . .	233
	<b>Bibliography</b>	<b>234</b>
<b>A</b>	<b>Supporting material from chapter 2: Profile fitting with Gaussian process regression</b>	<b>310</b>
A.1	Material from section 2.4.3.2: Nonstationary covariance kernels applied to entire profiles . . . . .	310
A.2	Material from section 2.5.2: Fit including TCI data . . . . .	323
A.3	Material from section 2.5.3: Computing second derivatives to investigate rotation reversals . . . . .	327
A.4	Material from section 2.5.4: 2D fitting of sawtooth-free data . . . . .	330
<b>B</b>	<b>Covariance kernels</b>	<b>336</b>
B.1	The squared exponential covariance kernel . . . . .	337
B.2	The rational quadratic covariance kernel . . . . .	340
B.3	The Matérn covariance kernel . . . . .	344
B.3.1	Handling the behavior as $y \rightarrow 0$ . . . . .	350
B.4	Handling the edge: nonstationary covariance kernels . . . . .	358
B.4.1	The Gibbs covariance kernel . . . . .	358
B.4.2	Warped inputs . . . . .	363

B.5	Transformations of covariance kernels . . . . .	366
B.5.1	Sums of covariance kernels . . . . .	366
B.5.2	Products of covariance kernels . . . . .	367
<b>C</b>	<b>Supporting material from chapter 3: Inference of experimental impurity transport coefficient profiles</b>	<b>368</b>
C.1	Relative contributions of diffusion, convection and atomic physics . . . . .	368
C.2	Discrepancies between linearized and exact figures of merit . . . . .	371
C.3	Posterior distributions for complex synthetic data . . . . .	371
	<b>Software for data analysis</b>	<b>381</b>
<b>D</b>	<b>eqtools: an open-source Python package for handling magnetic equilibrium data</b>	<b>382</b>
D.1	Package overview . . . . .	382
D.2	Coordinate systems supported and the coordinate mapping routines	384
D.3	Computing magnetic fields and current densities . . . . .	386
D.3.1	The magnetic field . . . . .	386
D.3.2	The current density . . . . .	386
D.3.3	Tracing field lines . . . . .	388
D.4	Verification and benchmarking . . . . .	389
<b>E</b>	<b>gptools: an open-source Python package for general-purpose Gaussian process regression</b>	<b>394</b>
E.1	Package overview . . . . .	394
E.2	Computational preliminaries: keeping track of hyperparameters in composite objects . . . . .	397
E.3	Representing a Gaussian process computationally: the <code>GaussianProcess</code> class . . . . .	398
E.3.1	Attributes of the <code>GaussianProcess</code> class . . . . .	398
E.3.2	Creating a <code>GaussianProcess</code> and adding data . . . . .	401
E.3.3	Formulating the pieces necessary to make an inference . . . . .	402
E.3.4	Finding the MAP estimate . . . . .	405
E.3.5	Making a prediction and drawing random samples . . . . .	405
E.3.6	Using MCMC to sample the hyperparameters . . . . .	408
E.4	Representing a covariance kernel computationally: the <code>Kernel</code> class and the <code>gptools.kernel</code> subpackage . . . . .	409
E.4.1	The core <code>Kernel</code> classes . . . . .	409
E.4.2	The <code>gptools.kernel.noise</code> submodule . . . . .	415

E.4.3	The SquaredExponentialKernel class . . . . .	415
E.4.4	The RationalQuadraticKernel class . . . . .	416
E.4.5	The <code>gptools.kernel.matern</code> submodule . . . . .	416
E.4.6	The <code>gptools.kernel.gibbs</code> submodule . . . . .	418
E.4.7	The <code>gptools.kernel.warping</code> submodule . . . . .	418
E.5	The MeanFunction class and the <code>gptools.mean</code> submodule . . . . .	419
E.5.1	The <code>mtanh</code> mean function . . . . .	421
E.6	Representing prior distributions: the <code>JointPrior</code> abstract class and its subclasses . . . . .	422
E.6.1	The <code>JointPrior</code> abstract class . . . . .	422
E.6.2	Combining <code>JointPrior</code> instances . . . . .	423
E.6.3	Normal prior distributions . . . . .	424
E.6.4	Log-normal prior distributions . . . . .	424
E.6.5	Gamma prior distributions . . . . .	424
E.6.6	Prior distributions for ordered values . . . . .	425
E.7	Miscellaneous classes and functions from the <code>gptools.utils</code> submodule . . . . .	427
E.7.1	Correcting SciPy's Pochhammer symbol . . . . .	427
E.7.2	Finding the derivatives of $K_\nu(y^{1/2})$ . . . . .	427
E.7.3	Finding the derivatives of $y^{1/2}K_\nu(y^{1/2})$ . . . . .	427
E.7.4	Generating all of the partitions of a set . . . . .	428
E.8	The <code>gptools.splines</code> submodule . . . . .	428
E.8.1	Mathematical preliminaries . . . . .	428
E.8.2	Software implementation . . . . .	430
<b>F</b>	<b>profiletools: an open-source Python package for handling profile data</b> . . . . .	<b>433</b>
F.1	Package overview . . . . .	433
F.2	The <code>profiletools</code> data model and the <code>Profile</code> class . . . . .	434
F.2.1	Automatically removing suspicious data points . . . . .	436
F.2.2	Creating a Gaussian process . . . . .	436
F.2.3	Keeping track of transformed quantities: the <code>Channel</code> class . . . . .	436
F.3	Extensions to handle plasma profile data . . . . .	436
F.3.1	Converting coordinates . . . . .	439
F.3.2	Imposing constraints . . . . .	439
F.3.3	Computing gradient scale lengths . . . . .	440
F.3.4	Computing volume averages . . . . .	442
F.3.5	Computing peaking factors . . . . .	443
F.4	Functions for retrieving Alcator C-Mod profiles . . . . .	443
F.4.1	The electron density profile, $n_e$ . . . . .	443



F.4.2	The electron temperature profile, $T_e$ . . . . .	444
F.5	Averaging schemes and their interpretations . . . . .	445
F.5.1	Robust versus standard estimators . . . . .	447
F.5.2	Weighted versus non-weighted estimators . . . . .	447
F.5.3	Methods of averaging . . . . .	454
<b>G</b>	<b>bayesimp: an open-source Python package for Bayesian impurity trans-</b>	
	<b>port analysis</b> . . . . .	<b>462</b>
G.1	Package overview . . . . .	462
G.2	The bayesimp module . . . . .	462
G.2.1	The Run class . . . . .	463
G.2.2	Enabling parallel execution: the acquire_working_dir() and release_working_dir() functions . . . . .	472
G.3	The lines module . . . . .	473
G.4	Tutorial . . . . .	473
G.4.1	Using real experimental data . . . . .	474
G.4.2	Creating local synthetic data . . . . .	479
G.4.3	Creating line-integrated synthetic data . . . . .	482
G.4.4	Performing the inference . . . . .	484
<b>H</b>	<b>Software configuration</b> . . . . .	<b>486</b>

# List of Figures

1.1	Conceptual structure of a simulation code . . . . .	27
2.1	Typical analysis scheme . . . . .	51
2.2	Overview of sampling based (“Monte Carlo”) uncertainty propagation . . . . .	52
2.3	Illustration of the effect of the covariance kernel . . . . .	56
2.4	Results of fitting synthetic data with two different noise levels . . . . .	58
2.5	Illustration of GPR with one observation and one prediction . . . . .	60
2.6	Effect of $r/\ell$ on the shape of the joint prior PDF . . . . .	61
2.7	Synthetic data, GP mean and basis functions centered at each point . . . . .	63
2.8	Workflows using splines, GPR and IDA . . . . .	81
2.9	Results of fitting synthetic data . . . . .	83
2.10	Demonstration of asymptotic consistency with synthetic data . . . . .	84
2.11	Observed covariance matrix for the $r/a$ coordinates of the Thomson scattering diagnostic . . . . .	85
2.12	Results of fitting synthetic data both with and without the effect of errors in the independent variable . . . . .	86
2.13	Synthetic core data and fitted profiles for the squared exponential, rational quadratic and Matérn covariance kernels . . . . .	88
2.14	Root-mean-square errors and uncertainty estimates for the fits to synthetic core data using various stationary covariance kernels . . . . .	89
2.15	Matrix of univariate and bivariate marginal posterior distributions and MCMC sampler chains for the hyperparameters of the squared exponential, rational quadratic and Matérn covariance kernels fit to synthetic core data . . . . .	90
2.16	Synthetic data and fits performed using the Gibbs covariance kernel with tanh covariance length scale function . . . . .	97
2.17	Synthetic data and fits performed using the SE covariance kernel with 3-knot I-spline input warping function . . . . .	98
2.18	Root-mean-square errors and uncertainty estimates in the fits to synthetic data using various nonstationary covariance kernels . . . . .	99

2.19	Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1101014006, the L-mode shot used to demonstrate the impurity transport analysis . . . . .	101
2.20	Data and results for the (a) $n_e$ and (b) $T_e$ profiles for Alcator C-Mod shot 1101014006 . . . . .	106
2.21	Matrix of univariate and bivariate marginal posterior distributions for the hyperparameters of the fit to the $n_e$ profile, as computed with MCMC . . . . .	107
2.22	Univariate and bivariate marginal posterior distributions for the hyperparameters of the fit to the $T_e$ profile, as computed with MCMC . . . . .	108
2.23	Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1120907032, including the locations of the eight TCI chords in use . . . . .	109
2.24	Data and results for the $n_e$ profile with and without the TCI data included . . . . .	112
2.25	Data and results for the values of line-integrated density on the various TCI chords, where the fit is performed with and without the TCI data included . . . . .	113
2.26	Toroidal rotation profiles for shots 1120221011 (hollow rotation profile) and 1120221012 (peaked rotation profile) . . . . .	115
2.27	Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shots 1120221011 and 1120221012 . . . . .	117
2.28	Data and fitted $n_e$ profile for the two rotation reversal shots . . . . .	119
2.29	Data and fitted $T_e$ profile for the two rotation reversal shots . . . . .	120
2.30	Probabilities of various levels of difference between the second derivatives of the $n_e$ and $T_e$ profiles at $r/a = 0.5$ . . . . .	122
2.31	Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1110329013, the sawtooth-suppressed lower hybrid current drive shot used to demonstrate 2D fitting of profile data . . . . .	123
2.32	Data and 2D fitted profile for the sawtooth-free temperature data . . . . .	125
2.33	Profiles from the sawtooth-free data evaluated at various time points . . . . .	128
2.34	Data and fitted curves subject to three ways of combining the data from multiple time slices . . . . .	129
2.35	Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1110201035, the H-mode shot used to demonstrate fitting with and without the mtanh mean function . . . . .	132

2.36	Plots of the core and pedestal portions of the fits using the Gibbs covariance kernel with tanh covariance length scale function, the SE covariance kernel with mtanh mean function, and only the mtanh function . . . . .	133
2.37	Marginal posterior distributions for the hyperparameters of the Gibbs covariance kernel with tanh covariance length scale function for the fit to the H-mode density profile . . . . .	134
2.38	Marginal posterior distributions for the parameters of the mtanh mean function and the hyperparameters of the SE covariance kernel for the fit to the H-mode density profile . . . . .	135
2.39	Marginal posterior distributions for the parameters of the mtanh function fit to the H-mode density profile . . . . .	136
2.40	Schematic of the STRAHL analysis . . . . .	140
2.41	$D$ and $V$ profiles for spline fits, sampling from the MAP estimate and from hierarchical sampling with MCMC . . . . .	142
3.1	Basic impurity transport coefficient profiles used to generate the synthetic impurity transport data . . . . .	151
3.2	Temporal evolution of the impurity density $n_Z$ at several radii . . . . .	152
3.3	Temporal evolution of the impurity density profile . . . . .	152
3.4	Temporal evolution of the total impurity content $N_Z$ . . . . .	153
3.5	Effect of $D$ , $V$ on $n_Z(r, t)$ . . . . .	154
3.6	Impurity confinement time $\tau_{\text{imp}}$ as a function of $D$ and $V$ . . . . .	155
3.7	Core rise time $t_r$ as a function of $D$ and $V$ . . . . .	156
3.8	Peak core impurity density as a function of $D$ and $V$ . . . . .	156
3.9	Impurity density profile broadness $b_{0.75} = n_Z(0.75)/n_Z(0)$ as a function of $D$ and $V$ . . . . .	157
3.10	Normalized impurity confinement time $\tau_{\text{imp}}D/a^2$ as a function of $S = -aV/(2D)$ . . . . .	158
3.11	Impurity density profile broadness $b_{0.75}$ as a function of normalized peaking factor $S$ . . . . .	159
3.12	Contours of $\tau_{\text{imp}}$ , $t_r$ and $b_{0.75}$ for $D = 1 \text{ m}^2/\text{s}$ , $V = -10 \text{ m/s}$ . . . . .	161
3.13	Impurity confinement time $\tau_{\text{imp}}$ in the vicinity of $D = 1 \text{ m}^2/\text{s}$ , $V = -10 \text{ m/s}$ . . . . .	164
3.14	Rise time $t_r$ in the vicinity of $D = 1 \text{ m}^2/\text{s}$ , $V = -10 \text{ m/s}$ . . . . .	164
3.15	Impurity density profile broadness $b_{0.75}$ in the vicinity of $D = 1 \text{ m}^2/\text{s}$ , $V = -10 \text{ m/s}$ . . . . .	165
3.16	Isosurfaces of $\sigma_D = 0.1D_0$ and $\sigma_V = 0.1 V_0 $ in $(\sigma_{\tau_{\text{imp}}}, \sigma_{t_r}, \sigma_{b_{0.75}})$ space	166
3.17	Isosurfaces for $\sigma_D = 0.1D_0$ and $\sigma_V = 0.1 V_0 $ . . . . .	167

3.18	Uncertainty in $D$ and $V$ as a function of $\sigma_{t_r}$ and $\sigma_{b_{0.75}}$ for $\sigma_{\tau_{\text{imp}}} = 1 \mu\text{s}$	168
3.19	Uncertainty in the profile broadness $b_{0.75}$ as a function of the sampling interval $\Delta t$ and the relative noise level $u = \sigma_y/y$	170
3.20	Assumed Gaussian distribution $\mathcal{N}(b_{0.75,0}, 2u^2 b_{0.75,0}^2)$ from equation (3.41), exact distribution from equation (3.44) and normalized histogram of a Monte Carlo simulation of $b_{0.75} = n_Z(0.75)/n_Z(0)$	171
3.21	Normalized histograms of the mean and median of $N$ samples of $b_{0.75}$ along with $\mathcal{N}(b_{0.75,0}, 2u^2 b_{0.75,0}^2/N)$ , the approximate result from equation (3.43)	172
3.22	Uncertainty in $\tau_{\text{imp}}$ as a function of $\Delta t$ , $u$ and $N_s$	174
3.23	Uncertainty in $t_r$ as a function of $\Delta t$ and $u$ for $N_s = 1$	175
3.24	Uncertainty in $b_{0.75}$ as a function of $\Delta t$ and $u$	175
3.25	Uncertainty in $D$ as a function of $\Delta t$ , $u$ and $N_s$	176
3.26	Uncertainty in $V$ as a function of $\Delta t$ , $u$ and $N_s$	177
3.27	Diagram of the forward model and probability model used to infer the impurity transport coefficients $D$ and $V$ from spectroscopic measurements $\int \varepsilon dl$	179
3.28	Transmission of 50 $\mu\text{m}$ beryllium filter used in XTOMO 1 and 3	181
3.29	Synthetic spectrum weighted by the 50 $\mu\text{m}$ beryllium filter transmission	184
3.30	Eigenvalue spectra for the $n_e$ and $T_e$ profiles	186
3.31	Posterior distribution for $D$ and $V$ inferred from five local measurements	193
3.32	Marginalized posterior $D$ and $V$ profiles inferred from five local measurements	194
3.33	Spectrometer lines of sight for Alcator C-Mod shot 1101014006, the shot used as the basis for the synthetic impurity transport data	195
3.34	Posterior distribution for $D$ and $V$ inferred from line-integrated measurements representing the diagnostic situation on Alcator C-Mod	197
3.35	Posterior distribution for $D$ and $V$ inferred from line-integrated measurements which only have data from the decay portion	198
3.36	Example of non-identifiable posterior distribution resulting from the use of a free-knot spline	200
3.37	Marginalized $D$ and $V$ profiles for the case with one free internal knot and local measurements	201
3.38	Posterior distribution for $D$ , $V$ , $n_e$ , $T_e$ inferred from line-integrated spectroscopic measurements	202

3.39	Posterior $D$ and $V$ profiles inferred from line-integrated spectroscopic measurements and marginalized over $n_e$ and $T_e$ . . . . .	203
3.40	Prior and posterior $n_e$ and $T_e$ profiles . . . . .	204
3.41	Posterior correlation matrix for the inference with line-integrated spectroscopic measurements and $n_e$ and $T_e$ free to vary . . . . .	205
3.42	First three weighted eigenvectors for $n_e$ and $T_e$ . . . . .	206
3.43	Posterior correlation matrix for the inference with $n_e$ and $T_e$ free to vary and local charge state density measurements . . . . .	207
3.44	Recombination, ionization and photon emission (for the 0.32 nm He-like calcium line) rate coefficients with uncertainty envelopes consistent with the prior uncertainty in $n_e$ and $T_e$ . . . . .	208
3.45	Correlations between the local $D(r/a)$ and $V(r/a)$ values and the local $n_e(r/a)$ and $T_e(r/a)$ values . . . . .	210
3.46	Scatterplots of $D$ , $V$ versus $n_e$ , $T_e$ for the pairs of points with the highest correlations . . . . .	211
3.47	Correlation of $D$ and $V$ with the knot positions for the four internal knots . . . . .	212
3.48	Scatterplot of $D$ and $V$ versus knot location for the pairs of points with the highest correlations . . . . .	213
3.49	Demonstration of model selection using the model evidence . . . . .	214
3.50	Synthetic data, true curve and polynomial fits of various degrees . . . . .	215
3.51	Estimated log-evidence as a function of polynomial degree $d$ . . . . .	216
3.52	Evidence as a function of model complexity for the basic synthetic data . . . . .	217
3.53	Posterior $D$ and $V$ profiles for the four different levels of complexity . . . . .	218
3.54	Evidence as a function of model complexity for the synthetic data using complicated $D$ and $V$ profiles . . . . .	220
3.55	Posterior $D$ and $V$ profiles for various levels of complexity . . . . .	221
3.56	Number of forward model evaluations for MULTINEST to converge as a function of model complexity . . . . .	222
3.57	Synthetic data and predicted diagnostic signals for varying levels of complexity . . . . .	223
3.58	Posterior $D$ and $V$ profiles with various fixed knot locations for the five coefficient per profile case . . . . .	225
3.59	Posterior distribution for $D$ and $V$ for the seven coefficient per profile case . . . . .	226

A.1	Synthetic data and fits performed using the Gibbs covariance kernel with a covariance length scale function which is an exponential of a single Gaussian basis function . . . . .	312
A.2	Synthetic data and fits performed using the Gibbs covariance kernel with a covariance length scale function which is an exponential of four Gaussian basis functions . . . . .	313
A.3	Synthetic data and fits performed using the SE covariance kernel with beta-CDF input warping function . . . . .	314
A.4	Synthetic data and fits performed using the SE covariance kernel with 2-knot I-spline input warping function . . . . .	315
A.5	Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with tanh covariance length scale function . . . . .	316
A.6	Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with covariance length scale function which is an exponential of a single Gaussian basis function . . . . .	317
A.7	Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with covariance length scale function which is an exponential of a sum of four Gaussian basis functions . . . . .	318
A.8	Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with beta-CDF input warping function . . . . .	319
A.9	Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with 2-knot I-spline input warping function . . . . .	320
A.10	Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with 3-knot I-spline input warping function . . . . .	321
A.11	Marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using only the Thomson scattering data . . . . .	324
A.12	Marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using the core and edge Thomson scattering data combined with the TCI data . . . . .	325

A.13	Marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using only the edge Thomson scattering and TCI data . . . . .	326
A.14	Marginal posterior distributions for the hyperparameters of the fits to the profiles for Alcator C-Mod shots 1120221011 and 1120221012 . . . . .	328
A.15	Univariate and bivariate marginal posterior distributions and MCMC sampler chains for the hyperparameters of the 2D SE covariance function with added homoscedastic noise . . . . .	331
A.16	Marginal posterior distributions for the hyperparameters for the evaluation of the sawtooth-free $T_e$ profiles at various time points . . . . .	332
A.17	Marginal posterior distributions for the hyperparameters of the 1D sawtooth-free $T_e$ profiles with various averaging schemes . . . . .	334
B.1	One-dimensional squared exponential covariance kernel and a few of its derivatives . . . . .	341
B.2	Two-dimensional squared exponential covariance kernel and a few of its derivatives . . . . .	342
B.3	One-dimensional rational quadratic covariance kernel and its derivatives for $\alpha = 1/2, 2, 10$ and a comparison of $k_{\text{RQ}}$ for $\alpha \in \{1/2, 2, 10, \infty\}$ . . . . .	345
B.4	Two-dimensional rational quadratic covariance kernel and a few of its derivatives for $\alpha = 1/2$ . . . . .	346
B.5	One-dimensional Matérn covariance kernel and its derivatives for $\nu = 1/2, 3/2, 5/2$ and a comparison of $k_{\text{M}}$ for $\nu \in \{1/2, 3/2, 5/2, \infty\}$ . . . . .	349
B.6	Two-dimensional Matérn covariance kernel for $\nu = 1/2$ . . . . .	350
B.7	Two-dimensional Matérn covariance kernel and a few of its derivatives for $\nu = 3/2$ . . . . .	351
B.8	Two-dimensional Matérn covariance kernel and a few of its derivatives for $\nu = 5/2$ . . . . .	352
B.9	One-dimensional Gibbs covariance kernel $k_{\text{G}}(x_i, x_j)$ with tanh covariance length scale function and a few of its derivatives for various values of $x_i$ . . . . .	361
C.1	Contributions from diffusion and convection to the flux of He-like calcium . . . . .	369
C.2	Contributions from transport and atomic physics to the change in the He-like calcium density . . . . .	370
C.3	Discrepancy between the actual $\tau_{\text{imp}}$ obtained from the STRAHL outputs and the linearized $\hat{\tau}_{\text{imp}}$ . . . . .	371



C.4	Discrepancy between the actual $t_r$ obtained from the STRAHL outputs and the linearized $\hat{t}_p$ . . . . .	372
C.5	Discrepancy between the actual $b_{0.75}$ obtained from the STRAHL outputs and the linearized $\hat{b}_{0.75}$ . . . . .	372
C.6	Posterior distribution for $D$ and $V$ for the one coefficient per profile case . . . . .	373
C.7	Posterior distribution for $D$ and $V$ for the two coefficient per profile case . . . . .	374
C.8	Posterior distribution for $D$ and $V$ for the three coefficient per profile case . . . . .	375
C.9	Posterior distribution for $D$ and $V$ for the four coefficient per profile case . . . . .	376
C.10	Posterior distribution for $D$ and $V$ for the five coefficient per profile case with linearly-spaced knots . . . . .	377
C.11	Posterior distribution for $D$ and $V$ for the five coefficient per profile case with near-optimal knots . . . . .	378
C.12	Posterior distribution for $D$ and $V$ for the five coefficient per profile case with purposefully bad knots . . . . .	379
C.13	Posterior distribution for $D$ and $V$ for the six coefficient per profile case . . . . .	380
D.1	Structure of the <code>eqtools</code> package . . . . .	384
D.2	Vacuum and total toroidal magnetic field at midplane as a function of major radius for $t = 1$ s in Alcator C-Mod shot 1120907032 . . . . .	387
D.3	Magnetic field lines for Alcator C-Mod shot 1120907032 . . . . .	390
D.4	Current density field lines for Alcator C-Mod shot 1120907032 . . . . .	391
D.5	Absolute difference between calculations of a mapping between the $(R, Z)$ grid and poloidal flux $\psi$ for <code>eqtools</code> and the current IDL implementation of the mapping routine for the analysis EFIT of Alcator C-Mod shot 1120914027 . . . . .	392
E.1	Basic structure of <code>gptools</code> . . . . .	396
E.2	Class diagram for the <code>GaussianProcess</code> class . . . . .	399
E.3	Basic structure of the <code>kernel</code> subpackage . . . . .	410
E.4	Class diagram for the <code>Kernel</code> abstract class . . . . .	411
E.5	Class diagram for the <code>WarpingFunction</code> class . . . . .	419
E.6	Class diagram for the <code>MeanFunction</code> class . . . . .	420
E.7	Class diagram for the <code>JointPrior</code> abstract class . . . . .	422

F.1	Class diagram for the <code>Profile</code> class . . . . .	435
F.2	Class diagram for the <code>Channel</code> class . . . . .	437
F.3	Class diagram for the <code>BivariatePlasmaProfile</code> class . . . . .	438
F.4	Data from channel 14 of the edge Thomson scattering system from Alcatraz C-Mod shot 1110329013 used to illustrate the different averaging schemes . . . . .	459
F.5	Empirical distributions for the data shown in figure F.4 . . . . .	460
G.1	Basic structure of the <code>bayesimp</code> package . . . . .	463
G.2	<code>gpfit</code> window to fit the temperature profile, showing the completed fit	476
G.3	HIREX-SR inspector window . . . . .	477
G.4	XEUS inspector window . . . . .	478
G.5	XTOMO inspector window . . . . .	479

# List of Tables

1.1	Review of impurity transport research, broken down by machine . . .	29
1.2	Comparison of different injection methods . . . . .	33
1.3	Comparison of different diagnostics for impurity transport measurements . . . . .	34
1.4	Comparison of different methods of inferring transport coefficients	38
2.1	Stationary covariance kernels tested with synthetic data and the prior distributions for their hyperparameters . . . . .	87
2.2	Parameters used for MCMC sampling of the hyperparameters of the various stationary covariance kernels applied to synthetic data . . .	87
2.3	Summary statistics for the posterior distributions for the hyperparameters of the various stationary covariance kernels applied to synthetic core data . . . . .	92
2.4	Nonstationary covariance kernels tested with synthetic data and the prior distributions for their hyperparameters . . . . .	94
2.5	Parameters used for MCMC sampling of the hyperparameters of the various nonstationary covariance kernels applied to synthetic data .	95
2.6	Prior distributions used for the hyperparameters of the Gibbs covariance kernel with tanh covariance length scale function when fitting Alcator C-Mod shot 1101014006 . . . . .	102
2.7	Constraints imposed on the profile fits for Alcator C-Mod shot 1101014006 by the addition of artificial “observations” . . . . .	103
2.8	MCMC sampler autocorrelation times for each hyperparameter . . .	104
2.9	Median relative uncertainties over the region $0 \leq r/a \leq 1$ , given as percentages . . . . .	104
2.10	Summary statistics for the posterior distribution of the hyperparameters of the Gibbs covariance kernel with tanh covariance length scale function for Alcator C-Mod shot 1101014006 . . . . .	105
2.11	Prior distributions for the hyperparameters used when assessing the effect of including the TCI data in the fit . . . . .	110
2.12	Summary statistics for the posterior distributions of the hyperparameters for the fits performed with and without TCI/CTS included .	111

2.13	Median relative uncertainties over the region $0 \leq r/a \leq 1$ for the cases with and without TCI/CTS included in the fit . . . . .	111
2.14	Prior distributions used for the hyperparameters of the squared exponential covariance kernel when fitting the rotation reversal data . . . . .	118
2.15	Prior distributions for the hyperparameters used to fit the sawtooth-free data with 1D and 2D SE covariance kernels and additional homoscedastic noise . . . . .	124
2.16	Summary statistics for the posterior distributions of the hyperparameters for the 1- and 2D fits performed on the sawtooth free data . . . . .	127
2.17	Median relative uncertainties over the region $0 \leq r/a \leq 1$ for the 1- and 2D fits performed on the sawtooth-free data . . . . .	130
2.18	Parameters of the affine-invariant ensemble samplers used to fit the H-mode $n_e$ profile with and without the mtanh mean function . . . . .	137
2.19	Prior distributions for the parameters and hyperparameters of the mean functions and covariance kernels used when fitting the H-mode $n_e$ profile . . . . .	137
2.20	Summary statistics for the posterior distributions of the parameters and hyperparameters of the fits to the H-mode $n_e$ profile with and without the mtanh mean function . . . . .	138
2.21	Median relative uncertainties over the region $0 \leq r/a \leq 1$ for fits to the H-mode $n_e$ profile with and without the mtanh mean function . . . . .	138
3.1	Series limits for calcium . . . . .	183
3.2	Summary statistics for $D$ and $V$ under various conditions . . . . .	192
3.3	Comparison of uncertainty estimates from the linearized Bayesian analysis and MULTINEST . . . . .	192
A.1	Summary statistics of the posterior distributions for the hyperparameters of the various non-stationary fits to synthetic data . . . . .	322
A.2	Summary statistics of the posterior distributions for the hyperparameters of the fits for Alcator C-Mod shots 1120221011 and 1120221012 . . . . .	327
D.1	Time to convert all 180 time slices in milliseconds . . . . .	393
F.1	Results of the different averaging schemes applied to the data from channel 14 of the edge Thomson scattering system from Alcator C-Mod shot 1110329013 . . . . .	461
H.1	Software configuration in use at the time this thesis was written . . . . .	486

# List of Code Listings

D.1	Fetching the $q$ profile with MDSplus directly . . . . .	383
D.2	Fetching the $q$ profile with eqtools . . . . .	383
E.1	Basic gptools example with univariate data . . . . .	395
F.1	Example showing to use of profiletools to fit the $T_e$ profile . . . .	434
G.1	Settings file to analyze real experimental data . . . . .	474
G.2	Settings file to create local synthetic data . . . . .	480
G.3	Settings file to create line-integrated synthetic data . . . . .	482
G.4	Code to find the MAP estimate . . . . .	484
G.5	Code to run MULTINEST . . . . .	484

# Notation

In this thesis, the following conventions are used:

- Scalars are written in roman italic font:  $a, A, \alpha, \Gamma$ .
- Vectors are written in boldface roman italic font:  $\mathbf{a}, \mathbf{A}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}$ . Elements of vectors are indicated with subscripts. Indices start from one, so  $a_1, A_1, \alpha_1, \Gamma_1$  are the first elements of the vectors indicated.
- Matrices are written in upright boldface sans serif font:  $\mathbf{a}, \mathbf{A}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}$ . Elements of matrices are indicated with two subscripts.  $a_{1,2}, A_{1,2}, \alpha_{1,2}, \Gamma_{1,2}$  are the second elements in the first row of the matrices indicated.
- Higher-dimensional arrays (mostly restricted to describing computer code) are written in italic boldface sans serif font:  $\mathbf{a}, \mathbf{A}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}$ .
- Specific functions and operators are written in upright roman font:  $B(x, y)$  is the beta function,  $\Gamma(z)$  is the gamma function. Generic functions are written in roman italic font:  $y = f(x)$ .
- To indicate that vector  $\mathbf{v}$  is real-valued and  $D$ -dimensional, the notation  $\mathbf{v} \in \mathbb{R}^D$  is used.
- To indicate that matrix  $\mathbf{M}$  is real-valued and  $m \times n$ , the slight abuse of notation  $\mathbf{M} \in \mathbb{R}^{m \times n}$  is used. This is believed to be more concise than the traditional notation  $\mathbf{M} \in \text{Mat}_{m \times n}(\mathbb{R})$ .
- Because the notation “log” means many things to people in different fields, the notation “ln” is used for the natural logarithm.
- Because the variable  $\rho$  seems to mean something different to every single plasma physicist, it is only ever used to refer to the Pearson correlation coefficient or a generic coordinate.
- Source code is written in monospaced font.

# 1

## Background

### 1.1 Nuclear fusion and impurity transport

The ultimate goal of nuclear fusion research is to develop a reactor which uses nuclear fusion reactions for energy production [1]. The leading device working towards this goal is the *tokamak*, a toroidal chamber surrounded with magnetic field coils which hold the plasma away from the surrounding material surfaces [2]. While the plasma of a reactor is intended to be primarily composed of some fuel such as a mixture of deuterium and tritium, the contamination of the plasma by *impurities* is inevitable for several reasons:

- All of the fuel cycles under consideration produce non-reacting “ash”:  ${}^4\text{He}$  in the case of the D-T and D- ${}^3\text{He}$  fuel cycles and  ${}^1\text{H}$  in one branch of the D-D fuel cycle.
- The confined plasma is surrounded by physical structures, and where the boundary plasma impinges on these surfaces wall material is inevitably sputtered off and can then be transported into the hot core plasma.
- Impurities are sometimes purposefully seeded into the plasma both to help spread out the heat flux coming out of the edge plasma onto the material surfaces and to diagnose the conditions inside the plasma.

The transport of impurities plays a critical role in the objective of building a practical fusion reactor [3–5]:

- The presence of impurities negatively impacts the power balance in two ways: by diluting the fuel and by enhancing the power loss from both line radiation and bremsstrahlung. The accumulation of too many impurities can cause the radiated power to exceed the input power, in which case the plasma suffers a radiative collapse.

- The maintenance of a radiative mantle to spread out the heat flux coming out of the plasma depends on the ability to keep the seeded impurities at the edge where they are needed without excessively contaminating the core plasma.

These problems are exacerbated by the predicted and observed behavior that impurities experience an inwards pinch which scales with  $Z$ , meaning that heavy impurities from the tokamak wall will accumulate in the center of the plasma where they will have the most pernicious effect (see, for example, [6] for a comparison of experimental high- $Z$  impurity transport and neoclassical predictions). In addition to assessing the potential impact of impurities on future fusion devices (as well as developing and testing ways to mitigate those issues), measurements of impurity transport are very useful from a diagnostic perspective:

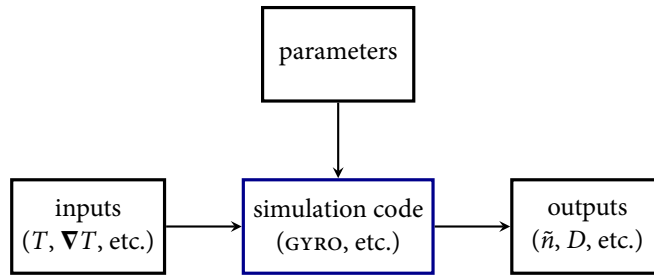
- Measurements of impurity transport provide an additional channel to test simulations of turbulent transport against. Testing how the transport trends with atomic mass and charge can provide a strong experimental test of theory.
- The transport properties of the main ion species can be difficult to measure directly, and trace amounts of light impurities (helium, tritium) can provide a proxy for diagnosing the main ion transport.

## 1.2 Verification, validation and uncertainty quantification of turbulent transport simulations

The use of impurity transport as an additional channel to test transport simulations is the main motivation of this thesis: the transport of particles, energy and momentum in magnetically-confined plasmas is not fully described by the neoclassical theory of collisional particle transport [7, 8]. Instead, it has been necessary to construct increasingly complicated gyrokinetic models of plasma turbulence such as the GYRO code to attempt to explain plasma transport [9, 10]. While there has been progress in recent years testing such codes against experimental measurements [11–30], the state of the art is far from being reliable enough to extrapolate beyond existing experimental conditions. This means that new devices must be designed using empirical scaling laws such as those given in [3]. Building large experiments without a predictive simulation capability is a risky proposition, and the pace of fusion development could be accelerated if new concepts could be reliably tested in computer simulations without having to make the considerable capital investment associated with building a new device.

The structure of a generic simulation code is given in [figure 1.1](#). The steps to





**Figure 1.1:** Conceptual structure of a simulation code. The quantities which go into the code are broken down into two categories: “inputs” are the physical conditions the code is to simulate, such as a given temperature  $T$  and temperature gradient  $\nabla T$ . (For a flux-driven code the inputs would be the fluxes and sources.) “Parameters” are the parameters of the underlying theoretical model which the code implements. A theory which is derived from first principles may have no such parameters, whereas a semi-empirical code may have many such parameters which have been estimated from existing data and therefore have associated uncertainties. The simulation code can then be used to produce outputs at many levels of detail such as the density fluctuation spectrum  $\tilde{n}$  and the diffusion coefficient  $D$ .

check that a simulation code is capable of explaining existing observations can be broken down into three components, often collectively referred to by the acronym VVUQ [31–38]:

**Verification** Checking that the simulation has implemented the underlying theory faithfully and that it solves the equations accurately.

**Validation** Checking the simulation against real data to make sure the theory (and hence the verified code which implements the theory) describes reality.

**Uncertainty quantification** Assessing the uncertainties in the code to figure out how much confidence there is in its results. There are two forms of uncertainty to consider:

- The code may have numerical approximations (as in the case of discretization error), or the computation itself may be random (as in the case of Monte Carlo sampling) or noisy (as in the case of simulations of turbulent transport).
- The inputs or parameters of the code may have associated uncertainties.

Note that input uncertainties and parameter uncertainties have been grouped together because they can be handled with similar techniques.

The uncertainty quantification step is important but often overlooked: a code which is strongly sensitive to its inputs can often be made to output a result close to the experimental observations for some reasonable combination of inputs, but such a comparison is not a strong statement on the validity of the code without an accompanying quantification of the range of possible outputs given the uncertainty in the inputs. Much effort has been expended on efficient quantification of uncertainty in codes, but comparatively little work has been done on refining the uncertainty estimates for the experimental data used for validation. The key contributions of this thesis are new techniques for handling experimental data which were developed with the intent of improving the quality of data used for vvuq of turbulent transport simulations. [Chapter 2](#) covers profile fitting and the estimation of the profile gradients which are a key input to all transport simulations and [chapter 3](#) covers the measurement of impurity transport.

### 1.3 Measuring impurity transport

This section reviews the techniques used to measure cross-field impurity transport in tokamaks; the specific contributions of this thesis to this topic are given in [chapter 3](#). The mathematical details of impurity transport are discussed in details in [section 3.2](#), so only a brief introduction is given here. Impurity behavior is governed by the radial continuity equation

$$\frac{\partial n_Z}{\partial t} = -\frac{1}{r} \frac{\partial}{\partial r} (r\Gamma_Z) + Q_Z, \quad (1.1)$$

where  $n_Z$  is the impurity density,<sup>1</sup>  $\Gamma_Z$  is the (radial) impurity flux and  $Q_Z$  is the source/sink term. The flux is often described using a diffusive and convective component:

$$\Gamma_Z = -D_Z \frac{\partial n_Z}{\partial r} + V_Z n_Z, \quad (1.2)$$

where  $D_Z$  is the diffusion coefficient and  $V_Z$  is the convective velocity. Collectively,  $D_Z$  and  $V_Z$  are referred to as the “transport coefficients.” To date, most of the impurity transport literature has attempted to summarize their observations in terms of transport coefficients.

---

1. In this context,  $n_Z$  may refer to either the density of a single charge state, or the total impurity density which has been summed over all charge states. More details regarding this distinction are given in [section 3.2](#).

Impurity transport has investigated in a wide variety of magnetic confinement devices for many years: see [table 1.1](#) for a partial review of the literature. In assembling the table, preference was given to papers on tokamaks, though other magnetic confinement devices were included because the diagnostic techniques are essentially the same between tokamaks, stellarators/heliotrons, reversed-field pinches (RFP) and spheromaks. Furthermore, preference was given to papers which discuss impurity transport in terms of transport coefficients, mostly leaving out those papers which only discuss phenomenology such as confinement times and profile shapes.

**Table 1.1:** Review of impurity transport research, broken down by machine.

Type	Name	Papers
Tokamak	Aditya	[39]
	Alcator A	[40, 41]
	Alcator C	[42, 43]
	Alcator C-Mod	[18, 19, 44–59]
	ASDEX	[60, 61]
	ASDEX Upgrade	[62–72]
	ATC	[73, 74]
	ATF	[75, 76]
	CDX-U	[77]
	DIII	[78]
	DIII-D	[79–85]
	FT	[86]
	FTU	[87–93]
	HL-1M	[94]
	HL-2A	[95–98]
	HT-6B	[99]
	HT-7	[100]
	ISX	[101]
	JET	[68, 71, 102–134]
	JIPP T-II	[135]
	JIPP T-IIU	[136]
	JT-6O	[137]
	JT-6OU	[138–144]
KSTAR	[145]	
MAST	[146–150]	
NSTX	[151–161]	

*(Continues on the next page.)*

**Table 1.1:** (Continued from the previous page.) Review of impurity transport research.

Type	Name	Papers
	PBX	[162–164]
	PDX	[165, 166]
	Phaedrus-T	[167]
	PLT	[168–172]
	TCV	[173–179]
	TdeV	[180–182]
	TEXT	[80, 183–188]
	TEXTOR	[6, 189–194]
	TFR	[195–199]
	TFTR	[200–211]
	Tore Supra	[212–225]
	UNITOR	[226]
Stellarator	CHS	[227–229]
	EXTRAP-T2	[230]
	EXTRAP-T2R	[231]
	Heliotron E	[232, 233]
	LHD	[229, 234–249]
	TJ-II	[250–257]
	W7-A	[258]
	W7-AS	[188, 259–263]
	W7-X	[264]
RFP	MST	[265]
	RFX	[116, 214, 266–268]
	RFX-mod	[269, 270]
Spheromak	S-1	[271]
	SPHEX	[272]

There are three critical aspects to measuring impurity transport:

1. The method used to inject impurities into the plasma.
2. The diagnostics used to track the impurities.
3. The method used to infer transport coefficients from the diagnostic measurements.

The following sections address the different options for each of these aspects.

### 1.3.1 Methods of impurity injection

There are multiple ways to inject impurities into the plasma, listed here roughly in order of complexity:

**Intrinsic impurities** Any magnetically confined plasma will be contaminated with material from the wall and residual gasses in the vacuum system. Note from the continuity equation that, in steady-state,

$$\frac{\partial n_Z / \partial r}{n_Z} = \frac{V_Z}{D_Z}, \quad (1.3)$$

which means that measurements of steady-state profiles from continuous wall erosion only give information on the ratio of  $V_Z$  and  $D_Z$ : transient impurity injections are needed to be able to uniquely measure  $D_Z$  and  $V_Z$ . These certainly do happen, but there is no way of controlling when or how big the injection is.

**Gas puff and supersonic molecular beam injection (SMBI) [273]** A plenum with a fast valve provides a way of transiently introducing a controlled amount of gas into the plasma. Unfortunately, the length of the gas puff is often comparable to the timescales of impurity transport, which makes inferring the transport coefficients difficult. SMBI is a gas injection technique originally designed for plasma fueling which can overcome the shortcomings of conventional gas puffing, and has been used for impurity transport studies on Tore Supra [220] and HL-2A [97]. Continuous impurity gas puffing is often used for ion temperature measurements and heat flux control, but as per the discussion above can only be used to obtain the ratio  $V_Z/D_Z$ .

**Laser blow-off (LBO) [74, 274]** A laser is used to ablate material from a thin film deposited on a slide held near the plasma. The source function is narrow enough to approximate a delta function, which simplifies the analysis. Because of the fine-grained control, fast injection rate and wide array of materials which can be injected, this is one of the most widely-used impurity injection techniques.

**Pellet and tracer pellet (TESPEL) [275]** Pellets have been widely used for plasma fueling. Impurity injection can be accomplished either by using a very small pellet made out of the desired material or by doping another pellet with the desired impurity. This approach is used on the LHD heliotron to allow the injected impurities to penetrate past the outer ergodic region.

A comparison of these techniques is given in [table 1.2](#), and an experimental comparison of several techniques is given in [\[220\]](#).

### 1.3.2 Diagnostics

A wide variety of diagnostics can be used to measure the injected impurities. Most of these are based on passively measuring the radiation from the impurities and can be broadly divided into spectrally-resolved instruments which can isolate the emission from single transitions of single charge states and broadband instruments which integrate the radiation from over a broad spectral range, capturing spectral line radiation, radiative recombination and bremsstrahlung emission. The spectrally-resolved category includes grating spectrometers in the visible through x-ray range and lower-resolution devices such as x-ray pulse height analyzers and multiply-filtered soft x-ray (SXR) photodiode arrays. The broadband category includes both SXR photodiode arrays and foil bolometers. The signal on the broadband instruments requires considerably more complicated atomic physics modeling to estimate the intensity reaching the detector from *all* transitions and charge states. Furthermore, intrinsic impurities can also affect the broadband emission, so spectrally-resolved measurements offer considerable advantages over broadband measurements. A key advantage of broadband instruments like SXR photodiode arrays, however, is that the broader wavelength range and simpler construction (both in the sense of not needing a small entrance slit to obtain good spectral resolution and having fewer elements such as gratings between the plasma and the detector) of such diagnostics permits a higher signal level and higher spatial and temporal resolution. In a completely different category of emissions connected to impurities, for the special case of tritium injected into a deuterium plasma, the fusion neutron emissivity can be used to monitor the tritium density [\[112, 122–125, 208–210\]](#).

All of these techniques integrate the intensity along a line sight, so some post-processing is necessary to obtain the local impurity density. It is also possible to actively interrogate the ion density of low- $Z$  impurities using charge exchange recombination spectroscopy (CXRS), which has the advantage of giving a local measurement of the impurity density.

On most experiments multiple diagnostics will be available, and the inference procedure used to find the transport coefficients must combine the data in a self-consistent manner. A comparison of these diagnostics as applied to impurity transport is given in [table 1.3](#).

**Table 1.2:** Comparison of different injection methods.

Method	Pros	Cons
Intrinsic	<ul style="list-style-type: none"> <li>• Always present, requires no special apparatus.</li> </ul>	<ul style="list-style-type: none"> <li>• No control over injection timing and size.</li> <li>• Limited selection of impurities.</li> <li>• Steady-state conditions cannot uniquely determine <math>D_Z</math> and <math>V_Z</math>.</li> </ul>
Gas puff	<ul style="list-style-type: none"> <li>• Plenum/valve system is multipurpose, available on most machines.</li> <li>• Easy to use gas mixtures in order to measure multiple impurities at once [130].</li> </ul>	<ul style="list-style-type: none"> <li>• Slow source function makes interpretation difficult.</li> <li>• Can only inject gasses.</li> <li>• Many gasses recycle which further complicates the analysis.</li> </ul>
Laser blow-off	<ul style="list-style-type: none"> <li>• Rapid (near-delta function) source function.</li> <li>• Fine control over injection size.</li> <li>• Can use a wide variety of materials, including medium- and high-Z metals.</li> </ul>	<ul style="list-style-type: none"> <li>• Difficult and expensive to fabricate slides, particularly with multiple impurity species.</li> <li>• Injection is impulsive: cannot produce a steady/slowly modulated impurity source.</li> </ul>
Pellet	<ul style="list-style-type: none"> <li>• Can deliver impurities to the core plasma.</li> <li>• Can use a wide variety of materials.</li> </ul>	<ul style="list-style-type: none"> <li>• Difficult to make pellets small enough to be non-perturbative.</li> <li>• Analysis complicated by need to model pellet ablation.</li> <li>• Injection is impulsive: cannot produce a steady/slowly modulated impurity source.</li> </ul>

**Table 1.3:** Comparison of different diagnostics for impurity transport measurements.

Diagnostic	Pros	Cons
Spectrally-resolved	<ul style="list-style-type: none"> <li>• Simpler atomic physics.</li> <li>• Can separate signals from injected versus intrinsic impurities.</li> </ul>	<ul style="list-style-type: none"> <li>• Often has lower spatial and temporal resolution than broadband diagnostics.</li> <li>• Lower signal levels compared to broadband diagnostics.</li> <li>• Line-integrated.</li> </ul>
Broadband	<ul style="list-style-type: none"> <li>• Can have very high spatial and temporal resolution.</li> <li>• Higher signal levels compared to spectrally-resolved diagnostics.</li> </ul>	<ul style="list-style-type: none"> <li>• Much more complicated atomic physics.</li> <li>• Cannot discriminate between the intentional injection and intrinsic impurities.</li> <li>• Line-integrated.</li> </ul>
Charge exchange	<ul style="list-style-type: none"> <li>• Local.</li> <li>• Insensitive to intrinsic impurities.</li> </ul>	<ul style="list-style-type: none"> <li>• Requires a neutral beam.</li> <li>• Often has lower spatial and temporal resolution than broadband diagnostics.</li> <li>• Not usable for high-Z impurities.</li> </ul>



### 1.3.3 Inference methods

The key part of the process of measuring impurity transport is to actually infer the transport coefficient profiles from the data. Depending on the specific details of the experiment at hand, there are several ways to do this. By far the most common approach is to use an impurity transport code combined with the appropriate synthetic diagnostic(s) to construct a *forward model* which predicts what the diagnostic signals would be given some specific profiles of  $D_Z$  and  $V_Z$ . Then, some sort of fitting procedure can be used to determine the  $D_Z$  and  $V_Z$  profiles which best reproduce the data. A simulation-based approach is needed because there is no closed-form solution to the impurity continuity equation for arbitrary  $D_Z$ ,  $V_Z$  profiles and source function, though there have been some attempts to construct Green's functions for simple  $D_Z$ ,  $V_Z$  profiles [186]. The most common codes for this application are STRAHL [276] and UTC-SANCO [107, 122, 123], with MIST [277] having been widely used in the older literature. The forward model approach has the advantage of providing complete flexibility when modeling any combination of diagnostics, and can even handle cases where the transport coefficients may be changing in time, such as in discharges with sawteeth.

As will be discussed in [chapter 3](#), this process requires extreme care to deliver useful results. Specifically, finding the best-fitting transport coefficients is a very difficult optimization problem with the potential for local minima and wide regions of poor fit with low gradients where the optimizer can get stuck. Even once the optimal solution(s) has been found, estimating the uncertainty in this solution is a difficult prospect given that high roundoff error in the forward model calls the use of finite differences to compute sensitivities into question. Furthermore, the fitting procedure requires a specific choice be made for the functional form for the  $D_Z$  and  $V_Z$  profiles. A poor choice can end up over-constraining the fitting process and preventing a reasonable solution from being found, but the procedure as implemented in the past provides no warning that this may be the case. [Chapter 3](#) presents a new forward model-based methodology which attempts to overcome these issues.

When high-resolution local impurity density measurements are available, the *flux-gradient* method may be employed. Consider the impurity continuity equation

in a region where the source/sink function is zero:

$$\frac{\partial n_Z}{\partial t} = -\frac{1}{r} \frac{\partial}{\partial r} (r\Gamma_Z) \quad (1.4)$$

$$\Gamma_Z = -\frac{1}{r} \int_0^r \frac{\partial n_Z}{\partial t} r' dr' = -D_Z \frac{\partial n_Z}{\partial r} + V_Z n_Z \quad (1.5)$$

$$\frac{\Gamma_Z}{n_Z} = -D_Z \frac{\partial n_Z / \partial r}{n_Z} + V_Z. \quad (1.6)$$

Therefore, if the impurity density  $n_Z$  has been measured with high enough spatial and temporal resolution to reconstruct local values of both the flux  $\Gamma_Z \propto \partial n_Z / \partial t$  and inverse gradient scale length  $\partial n_Z / \partial r / n_Z$  at a point, the transport coefficients  $D_Z$  and  $V_Z$  can be determined from a simple linear fit to the data obtained during the influx and decay of the impurity injection. A comparison of the flux-gradient and forward model approaches is given in [148, 149].

In addition to the very high temporal and spatial resolution needed to determine  $\partial n_Z / \partial t$  and  $\partial n_Z / \partial r$ , a key shortcoming of the flux-gradient approach is that [equation \(1.5\)](#) is only valid in regions where the source/sink function is zero. If measurements of the *total* impurity density (i.e., summed over all charge states) are available, it is reasonable to expect that this will hold true in most of the core of the plasma as the source/sink is localized to the edge. In the more common case that only a specific charge state has been measured (e.g., by an imaging spectrometer or CXRS), however, the source/sink term consists of ionization and recombination to adjacent charge states and can end up being non-negligible throughout the plasma. To handle this effect requires knowledge of the charge state distribution, which is itself a function of the transport coefficients. This is exacerbated when using broadband measurements such as SXR photodiode arrays because the signal observed is a complicated function of the distribution of all charge states. This has been overcome in some cases by alternating between using an impurity transport code to form estimates of the transport-dependent charge state distributions and accompanying SXR power loss coefficients and then using these estimates to infer local values of the transport coefficients [70, 72]. In order to overcome some of the difficulties of both the forward model and flux-gradient approaches, one approach is to use the flux-gradient results as initial guesses for the forward model, as was done in [113].

In long, stationary discharges *harmonic analysis of gas puff modulation* may be used [61, 139]. This technique is also relevant to the measurement of main ion and electron transport [278–280]. Harmonic analysis exploits the relationship between a modulated source function (typically accomplished by sinusoidally varying the gas puff rate) and the Fourier transform of the local impurity density measurements. Provided sufficiently precise measurements and a sufficiently long discharge, this

allows the reconstruction of the local transport coefficients. This method has similar requirements as the flux-gradient approach: the transport coefficients must be constant in time and the analysis requires assumptions be made regarding the charge state distribution. A comparison of the forward model, flux-gradient and harmonic analysis techniques is given in [table 1.4](#), and some additional discussion is given in [\[281\]](#).

## 1.4 Notation and terminology from probability theory and Bayesian inference

### 1.4.1 Random variables, probability densities and Bayes' rule

This section presents a basic review of the key terminology of probability theory, statistics and Bayesian inference. There are many excellent books and review papers on this topic including [\[282–286\]](#), so this section is purposefully brief and focusses solely on what is needed to understand the statistical content of this thesis.

Consider a *random variable*  $X$  which may take on specific values  $x$  according to some probability law. Here, we are careful to distinguish between the random variable  $X$  and its specific value  $x$ , consistent with the notation of [\[284\]](#). In later chapters this distinction is relaxed when the intent is clear from context.

A *probability density function* (PDF) describes the probability of a random/uncertain quantity taking a given value. Specifically, the probability that random variable  $X$  with PDF  $f_X(x)$  lies between  $a$  and  $b$  is

$$\mathbb{P}(a < X < b) = \int_a^b f_X(x) dx. \quad (1.7)$$

Because  $f_X(x)$  is a “density,” the term *probability mass* is often used to describe the result of integrating the PDF over some region. A probability distribution may also be described by its *cumulative density function* (CDF):

$$F_X(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^x f_X(u) du. \quad (1.8)$$

Clearly,  $f_X(x) = dF_X(x)/dx$ . The CDF is monotonic and has range  $[0, 1]$ , which also makes it useful for some non-statistical applications. If  $F_X^{-1}(y)$  exists (i.e.,  $F_X(x)$  is monotonic and one-to-one) then, given a uniformly-distributed variable  $u \sim \mathcal{U}(0, 1)$ ,  $x = F_X^{-1}(u)$  is distributed according to  $f_X(x)$ . As (pseudo)random samples from the uniform distribution can be obtained through many well-established algorithms, this gives a convenient way of producing samples from any distribution for which  $F_X^{-1}(u)$  can be computed [\[287–292\]](#).

**Table 1.4:** Comparison of different methods of inferring transport coefficients.

Method	Pros	Cons
Forward model	<ul style="list-style-type: none"> <li>• Can handle arbitrary source functions.</li> <li>• Can handle temporal variation of transport coefficients.</li> <li>• Can work with any combination of diagnostics and injection methods.</li> </ul>	<ul style="list-style-type: none"> <li>• Very difficult optimization problem.</li> <li>• Choice of basis functions can have strong influence on answer.</li> </ul>
Flux-gradient	<ul style="list-style-type: none"> <li>• Allows reconstruction of local transport coefficients without assuming a functional form for <math>D_Z</math> and <math>V_Z</math>.</li> <li>• Simple analysis to go from fluxes/gradients to transport coefficients.</li> </ul>	<ul style="list-style-type: none"> <li>• High spatial and temporal resolution needed to be able to measure fluxes and gradients.</li> <li>• Only valid in source-free regions: most diagnostic situations require iteration with impurity transport code to handle transport dependence of charge state distribution.</li> <li>• Requires transport coefficients be constant in time.</li> </ul>
Harmonic analysis	<ul style="list-style-type: none"> <li>• Collecting data over a long time period enables very low perturbation of the plasma.</li> </ul>	<ul style="list-style-type: none"> <li>• Generally only applicable to gas puffs. (Other modulated sources could be used, but would require complicated analysis to handle the higher harmonics associated with non-sinusoidal source functions.)</li> <li>• Requires very long stationary discharge.</li> <li>• Requires transport coefficients be constant in time.</li> </ul>

A *joint PDF* describes the probability that multiple variables lie in a given region. For instance, given random quantities  $X$  and  $Y$  with joint PDF  $f_{X,Y}(x, y)$ , the probability that  $a < X < b$  and  $c < Y < d$  is

$$\mathbb{P}(a < X < b \wedge c < Y < d) = \int_a^b \int_c^d f_{X,Y}(x, y) dy dx. \quad (1.9)$$

It is clear that  $f_{X,Y}(x, y) = f_{Y,X}(y, x)$ . The *marginal PDF* of  $X$  is the result of integrating the joint PDF over all possible values of  $Y$ :

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) dy. \quad (1.10)$$

The *conditional PDF* of  $X$  given a specific observed value for  $Y$  is

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x, y)}{f_Y(y)}. \quad (1.11)$$

Rearranging, the joint distribution can then be written as

$$f_{X,Y}(x, y) = f_{X|Y}(x|y)f_Y(y). \quad (1.12)$$

But, a similar expression can be obtained starting from the conditional distribution for  $Y$  given an observation of  $X$ :

$$f_{X,Y}(x, y) = f_{Y|X}(y|x)f_X(x). \quad (1.13)$$

Equating [equation \(1.12\)](#) and [equation \(1.13\)](#) and rearranging gives *Bayes' rule*:

$$f_{Y|X}(y|x) = \frac{f_{X|Y}(x|y)f_Y(y)}{f_X(x)}. \quad (1.14)$$

This is the fundamental equation of Bayesian inference: it lets you infer the value of  $Y$  given observations of  $X$ . For instance,  $Y$  might be the parameter of a model and  $X$  might be observations of the data to which the model is to be fit. In the context of inference, each of the terms in Bayes' rule has a specific interpretation and corresponding name:

- $f_{X|Y}(x|y)$  is called the *likelihood*. This is the probability that the observed value of  $X$  will be observed given some value of the parameter  $Y$ . In the context of inference, the data are taken to be given and the parameter is what is to be determined, so this is often written as a function of  $y$ :  $f_{X|Y}(x|y) = L(y)$ .

- $f_Y(y)$  is called the *prior distribution* (or *prior*). This distribution represents any information that happens to be available about  $Y$  before the observations  $X$  are included. In the context of the profile fitting task discussed in [chapter 2](#), the prior distribution for the covariance length scale is chosen to represent the assumption that tokamak profiles are macroscopically smooth.
- $f_X(x)$  is called the *evidence*. In the context of inference it simply acts as a normalization constant. In the context of model selection it represents the probability of observing the data given the specific model of which  $Y$  is a parameter, and can be used to compare different models with different sets of parameters.
- $f_{Y|X}(y|x)$  is called the *posterior distribution* (or *posterior*). This is the desired result of the inference: the probability distribution for the parameter  $Y$  given the observation of  $X$ .

[Equation \(1.14\)](#) shows that the inference (i.e., the posterior distribution of  $Y$  given  $X$ ) is the result combining the information in the likelihood (which is dictated by the data  $X$ ) and the prior distribution (which represents any knowledge about  $Y$  before the data have been considered). If a large quantity of high-quality data is available the likelihood will be strongly peaked and overwhelm the influence of the prior distribution. In the presence of sparse, noisy data the influence of the prior distribution can play a critical role in keeping the inference from being unphysical.

### 1.4.2 Summary statistics

A PDF contains all of the information there is to know about a random variable (or, in the case of a joint PDF, a collection of random variables). But this is often more information than is necessary. Various *summary statistics* are in common use to summarize the details of a PDF. This section defines the ones used in this thesis (as well as alternate forms often encountered in the literature), and the next section discusses how to compute them for sampled data. In all cases, the notation for a given summary statistic may have a subscript added to indicate which variable it refers to when it is not clear from context. In other words,  $\mu$  is the general notation for the mean and  $\mu_X$  refers to the mean of random variable  $X$  when this is not clear from context. Furthermore, a conditional summary statistic is indicated in a similar manner to conditional probability distributions:  $\mu_{X|Y} = \mathbb{E}[X|Y] = \int_{-\infty}^{\infty} x f_{X|Y}(x|y) dx$  is the (conditional) mean of  $X$  given  $Y = y$ .

The *mode*  $m$  is the value with the highest probability:

$$m = \arg \max_x f_X(x). \quad (1.15)$$

A distribution for which there are multiple local maxima in the PDF is said to be *multimodal*. The *median*  $M = \mathbb{M}[X]$  is the point which bisects the PDF:

$$\mathbb{P}(X \leq \mathbb{M}[X]) = \mathbb{P}(X \geq \mathbb{M}[X]) = \frac{1}{2}. \quad (1.16)$$

The *mean*  $\mu = \mathbb{E}[X]$  is the centroid of the PDF:

$$\mu = \mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x) dx. \quad (1.17)$$

The *variance*  $\sigma^2 = \text{var}[X]$  represents the width of the PDF:

$$\sigma^2 = \text{var}[X] = \mathbb{E}[(X - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 f_X(x) dx. \quad (1.18)$$

The variance has units of  $X^2$ , so the *standard deviation*  $\sigma \equiv \sqrt{\sigma^2}$  is more useful for communicating the width of the PDF. The *covariance*  $\sigma_{XY} = \text{cov}[X, Y]$  represents the extent to which two variables vary together:

$$\sigma_{XY} = \text{cov}[X, Y] = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]. \quad (1.19)$$

Note that  $\sigma_{XX} = \text{var}[X] = \sigma_X^2$ , a potentially confusing side-effect of this notation. The *covariance matrix*  $\mathbf{\Sigma}_{XY} = \text{cov}[\mathbf{X}, \mathbf{Y}] \in \mathbb{R}^{m \times n}$  between two random vectors  $\mathbf{X} \in \mathbb{R}^m$ ,  $\mathbf{Y} \in \mathbb{R}^n$  is the result of evaluating the covariance between all pairs of values in  $\mathbf{X}$  and  $\mathbf{Y}$ :

$$\mathbf{\Sigma}_{XY} = \text{cov}[\mathbf{X}, \mathbf{Y}] \quad (1.20)$$

$$\Sigma_{ij} = \text{cov}[X_i, Y_j]. \quad (1.21)$$

Sometimes the covariance is expressed instead in terms of the (*Pearson*) *correlation coefficient*  $\rho_{XY} = \text{corr}[X, Y]$ :

$$\rho_{XY} = \text{corr}[X, Y] = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}. \quad (1.22)$$

The *correlation matrix*  $\mathbf{C}_{XY} = \text{corr}[\mathbf{X}, \mathbf{Y}] \in \mathbb{R}^{m \times n}$  between two random vectors  $\mathbf{X} \in \mathbb{R}^m$  is then obtained from

$$C_{ij} = \text{corr}[X_i, Y_j] = \frac{\text{cov}[X_i, Y_j]}{\sigma_{X_i} \sigma_{Y_j}}. \quad (1.23)$$

The median can also be referred to as the *second quartile*  $Q_2$  of the PDF. The *first quartile*  $Q_1$  is the point such that

$$\mathbb{P}(X \leq Q_1) = \frac{1}{4}. \quad (1.24)$$

The *third quartile*  $Q_3$  is the point such that

$$\mathbb{P}(X \leq Q_3) = \frac{3}{4}. \quad (1.25)$$

The *interquartile range*  $\text{IQR} = Q_3 - Q_1$  is one way of representing the width of the PDF. To represent the range that a random variable is expected to lie within, the *95% interval*  $[l, u]$  is used. This is the interval containing 95% of the probability mass:

$$\mathbb{P}(l \leq X \leq u) = 0.95. \quad (1.26)$$

The simplest way of obtaining  $l$  and  $u$  is the *equal-tailed interval*:

$$\mathbb{P}(X \leq l) = \mathbb{P}(X \geq u) = \frac{1 - 0.95}{2} = 0.025. \quad (1.27)$$

This has the disadvantage that it may end up excluding the mode of very skew distributions (such as the exponential distribution, where the mode is at the left-hand edge of the PDF's support) and could include large portions with small probability density for multimodal distributions. As such, some authors [286] recommend using the *95% highest density region*: the set of values which encompass 95% of the probability mass and for which the density in the region is never lower than outside the region. Note that this may end up consisting of multiple disconnected intervals if the PDF is multimodal. Because of this complexity, the equal-tailed interval is used throughout this thesis; this is why 95% posterior intervals for parameter estimates sometimes do not include the posterior mode. Note that the interquartile range is the width of the 50% equal-tailed interval.

### 1.4.3 Sample statistics

#### 1.4.3.1 Definitions

This section considers how to estimate the summary statistics discussed above when some set of  $N$  observations  $\{x^{(i)}\}_{i=1}^N$  has been collected. The notation  $x^{(i)}$  to refer to the  $i^{\text{th}}$  sample is used to avoid confusion with the notation  $x_i$  for the  $i^{\text{th}}$  component of vector  $\mathbf{x}$ . The PDF can be approximated by forming the *histogram* of the data: the grid on which the data were taken is divided into  $k$  bins, and the number of samples lying



within a given bin is tallied up to give the (unnormalized) value of the histogram in that bin. The *sample mode* is then the bin with the highest tally. The *sample median* is obtained by sorting the data and finding the point in the middle. (For an even number of samples, the mean of the middle two values is used.) The *sample first quartile*, *sample third quartile*, *sample 2.5% percentile* and *sample 97.5% percentile* can be obtained in a similar manner, which then allows the *sample interquartile range* and *sample 95% equal-tailed interval* to be computed. The *sample mean* is given by

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x^{(i)}. \quad (1.28)$$

The *sample variance* is given by

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x^{(i)} - \bar{x})^2, \quad (1.29)$$

where the factor in front<sup>2</sup> is  $1/(N-1)$  and not  $1/N$  to account for the bias introduced by using the data to compute  $\bar{x}$ . The *sample standard deviation* is then simply  $s = \sqrt{s^2}$ . The *sample covariance* between samples  $\{x^{(i)}\}_{i=1}^N$ ,  $\{y^{(i)}\}_{i=1}^N$  of scalar random variables  $X$ ,  $Y$  is

$$s_{XY} = \frac{1}{N-1} \sum_{i=1}^N (x^{(i)} - \bar{x})(y^{(i)} - \bar{y}), \quad (1.30)$$

and the *sample covariance matrix* for samples  $\{\mathbf{x}^{(i)}\}_{i=1}^N$ ,  $\{\mathbf{y}^{(i)}\}_{i=1}^N$  of vector random variables  $\mathbf{X}$ ,  $\mathbf{Y}$  is

$$\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}^{(i)} - \bar{\mathbf{x}})(\mathbf{y}^{(i)} - \bar{\mathbf{y}})^T \quad (1.31)$$

$$S_{jk} = \frac{1}{N-1} \sum_{i=1}^N (x_j^{(i)} - \bar{x}_j)(y_k^{(i)} - \bar{y}_k). \quad (1.32)$$

### 1.4.3.2 Robust versus non-robust estimators

Note that the sample mean and sample variance can both be made arbitrarily large by just a single outlying sample  $x^{(i)}$  which is far from the other points. This means that the sample mean and sample variance are *non-robust estimators*. Outliers may be

2. This can also be seen as the result of assuming a specific prior distribution for  $\sigma^2$  and applying Bayes' rule to condition on the data [283].

present as a result of diagnostic malfunctions or other events which are not included in the probability model. In order to account for this possibility without having to manually remove such outliers (a statistically questionable process), it is useful to use *robust estimators* which require a significant fraction of the data to be bad before they break down. The sample median and sample interquartile range are useful robust replacements for the sample mean and sample standard deviation, respectively. The software developed as part of this thesis supports a wide variety of schemes for averaging data which are described in detail in [section F.5](#).

#### 1.4.4 The uncertainty propagation equation

Given a quantity  $y = y(x_1, x_2, \dots)$  which is computed as a function of other uncertain values  $x_1, x_2, \dots$ , the variance in  $y$  can be approximated using the *uncertainty propagation equation* [282]:

$$\sigma_y^2 = \sigma_{x_1}^2 \left( \frac{\partial y}{\partial x_1} \right)^2 + \sigma_{x_2}^2 \left( \frac{\partial y}{\partial x_2} \right)^2 + \dots + 2\sigma_{x_1 x_2} \left( \frac{\partial y}{\partial x_1} \right) \left( \frac{\partial y}{\partial x_2} \right) + \dots \quad (1.33)$$

This is based on a linearization of the function  $y$  and hence should not be expected to be reliable for highly nonlinear functions. That being said, it seemed to perform adequately for all of the cases encountered in this thesis.

#### 1.4.5 Useful probability distributions

This section gives the explicit forms of the probability distributions in this thesis. Most of this information comes from table A.1 of [286].

##### 1.4.5.1 The uniform distribution

The (*continuous*) *uniform distribution* is a univariate probability distribution which is constant within its support  $[a, b]$ . The PDF is

$$f_X(x) = \mathcal{U}(x; a, b) = \begin{cases} 1/(b-a), & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases} \quad (1.34)$$

This distribution represents the maximum state of ignorance when it is only known that a random variable lies within a given interval  $[a, b]$ . Some relevant features of the uniform distribution are

$$\mu = \frac{b-a}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}. \quad (1.35)$$

### 1.4.5.2 The normal distribution

The (*univariate*) *normal* or *Gaussian distribution* is a symmetric univariate probability distribution which has support  $(-\infty, \infty)$ . The PDF is

$$f_X(x) = \mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right), \quad (1.36)$$

where  $\mu = m = M$  is the mean (and mode and median) and  $\sigma$  is the standard deviation. The interquartile range of the normal distribution is

$$\text{IQR} = F_X^{-1}(0.75) - F_X^{-1}(0.25) = \sigma(\Phi^{-1}(0.75) - \Phi^{-1}(0.25)) \approx 1.349\sigma, \quad (1.37)$$

where  $\Phi^{-1}(x)$  is the inverse CDF of the standard normal  $\mathcal{N}(0, 1)$ , also known as the *probit function*. Therefore, the robust estimator for the interquartile range can be used to construct an estimate of the standard deviation, assuming that the underlying distribution is in fact normal. This distribution represents the maximum state of ignorance when only the mean and variance of a random variable are known.

### 1.4.5.3 The log-normal distribution

The *log-normal distribution* describes a variable whose logarithm is normally distributed and therefore has support  $[0, \infty)$ . The PDF is<sup>3</sup>

$$\ln X \sim \mathcal{N}(\mu, \sigma^2) \quad (1.38)$$

$$f_X(x) = \ln \mathcal{N}(x; \mu, \sigma^2) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right). \quad (1.39)$$

Because of how SciPy [293] implements this PDF, it is most convenient to express this in terms of the median of  $x$ :  $M = \mathbb{M}[X] = e^\mu$ :

$$f_X(x) = \ln \mathcal{N}(x; \mu, \sigma^2) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x/M))^2}{2\sigma^2}\right). \quad (1.40)$$

Some relevant features of the log-normal distribution are

$$\mathbb{E}[X] = \exp\left(\mu + \frac{\sigma^2}{2}\right), \quad \text{var}[X] = e^{2\mu+2\sigma^2}(e^{\sigma^2} - 1) \quad (1.41)$$

$$m = e^{\mu-\sigma^2}, \quad M = e^\mu. \quad (1.42)$$

3. Notice an unfortunate side effect of the conventional notation:  $\ln \mathcal{N}(x; \mu, \sigma^2)$  would seem to imply that the natural logarithm of the normal PDF is taken, when in fact it refers to the function given in equation (1.39).

#### 1.4.5.4 The gamma and exponential distributions

The *gamma distribution* is a univariate probability distribution which has support  $[0, \infty)$ . This distribution's PDF is often expressed in one of several different forms, the one used in this thesis is

$$f_X(x; \alpha, \beta) = \text{Gamma}(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad (1.43)$$

where  $\alpha > 0$  is the “shape” parameter,  $\beta > 0$  is the “rate” parameter and  $\Gamma(\alpha)$  is the gamma function [294, 295]. By varying these two parameters, a wide variety of shapes can be obtained. The *exponential distribution* is a special case of the gamma distribution:

$$f_X(x; \beta) = \text{Exp}(x; \beta) = \text{Gamma}(x; 1, \beta) = \beta e^{-\beta x}. \quad (1.44)$$

Some relevant features of the gamma distribution are

$$\mu = \frac{\alpha}{\beta}, \quad \sigma^2 = \frac{\alpha}{\beta^2}, \quad m = \frac{\alpha - 1}{\beta} \text{ (for } \alpha \geq 1). \quad (1.45)$$

For  $\alpha < 1$ , the PDF goes to infinity as  $x \rightarrow 0$  so the mode is undefined.

While the gamma distribution is very useful because it ensures a variable is always positive and its two parameters let a wide variety of shapes be obtained, it is somewhat cumbersome to connect physical insights about the typical values a parameter assumes to specific values for  $\alpha$  and  $\beta$ . In order to obtain a simpler parameterization which is valid for  $\alpha \geq 1$ , substitute  $\alpha = \sigma^2 \beta^2$  into the expression for the mode to obtain

$$m = \frac{\sigma^2 \beta^2 - 1}{\beta} \quad (1.46)$$

$$\sigma^2 \beta^2 - m\beta - 1 = 0 \quad (1.47)$$

$$\beta = \frac{m \pm \sqrt{m^2 + 4\sigma^2}}{2\sigma^2} = \frac{m + \sqrt{m^2 + 4\sigma^2}}{2\sigma^2} \quad (1.48)$$

$$\alpha = m\beta + 1, \quad (1.49)$$

where the positive root was chosen to satisfy  $\beta > 0$ . Therefore, the gamma distribution with  $\alpha \geq 1$  can be specified in terms of  $m$  and  $\sigma$ .

#### 1.4.5.5 The multivariate normal distribution

The *multivariate normal distribution* (MVN) is a very useful multivariate probability distribution [296–298]. To indicate a random vector  $\mathbf{X} \in \mathbb{R}^m$  follows the multivariate

normal distribution with mean  $\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}]$  and (positive definite) covariance matrix  $\boldsymbol{\Sigma} = \text{cov}[\mathbf{X}, \mathbf{X}]$ , the notation  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is used. The PDF is

$$f_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{m/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\top} \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right). \quad (1.50)$$

Marginal distributions are simply obtained by taking the appropriate “slices” from  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ : given random vectors  $\mathbf{X} \in \mathbb{R}^m$ ,  $\mathbf{Y} \in \mathbb{R}^n$  with joint PDF

$$f_{\mathbf{X}, \mathbf{Y}}(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{\mathbf{X}} \\ \boldsymbol{\mu}_{\mathbf{Y}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{X}} & \boldsymbol{\Sigma}_{\mathbf{XY}} \\ \boldsymbol{\Sigma}_{\mathbf{YX}} & \boldsymbol{\Sigma}_{\mathbf{Y}} \end{bmatrix}\right), \quad (1.51)$$

where  $\boldsymbol{\Sigma}_{\mathbf{XY}} = \boldsymbol{\Sigma}_{\mathbf{YX}}^{\top} = \text{cov}[\mathbf{X}, \mathbf{Y}]$ , the marginal distribution for  $\mathbf{X}$  is simply

$$f_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}}, \boldsymbol{\Sigma}_{\mathbf{X}}). \quad (1.52)$$

The conditional distribution is rather more complicated, but is given in a very useful form in [297]:

$$f_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}) = \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{XY}}\boldsymbol{\Sigma}_{\mathbf{Y}}^{-1}(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}}), \boldsymbol{\Sigma}_{\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{XY}}\boldsymbol{\Sigma}_{\mathbf{Y}}^{-1}\boldsymbol{\Sigma}_{\mathbf{YX}}\right). \quad (1.53)$$

# 2

## Profile fitting with Gaussian process regression

### 2.1 Introduction

As discussed in [section 1.2](#), the primary motivation of this thesis is to improve the quality of data used for the validation of transport simulations. The level of turbulence is highly sensitive to the *gradients* in the temperature and density profiles, which means these are very important inputs to most codes. Having reliable estimates of the temperature and density profiles is still important when testing flux-driven codes which take fluxes as inputs and output temperature and density profiles, as the uncertainties in the profiles must be known in order to compare the outputs to the experimental observations. A key contribution of this thesis is the extensive development of procedures and software for fitting tokamak profiles with Gaussian process regression (GPR) [286, 297, 299–304]. This technique provides rigorous uncertainty estimates on the profiles and their gradients, and allows the fit to be performed automatically without the type of bias which can be introduced by manual intervention.

Some of this work was published in [305], from which much of this chapter is drawn. The extensive work which was performed on stationary and nonstationary covariance kernels and their derivatives is described in [appendix B](#). The actual software developed for GPR (`gptools`) is described in [appendix E](#) and [306, 307]. The software to simplify analysis of Alcator C-Mod profiles (`profiletools`) is described in [appendix F](#) and [308, 309]. Supporting material from this chapter which distracts from the main discussion but gives additional details on the results is given in [appendix A](#).

A situation which is ubiquitous in plasma physics and many other fields is that a quantity of interest is computed by a complicated, computationally expensive code

whose inputs are not single values but rather entire profiles of quantities given as functions of space, time and possibly other independent variables. In plasma physics, examples of these derived quantities include heat fluxes and particle diffusivities. As many quantities of interest and processes such as transport depend strongly on the *gradient* of a measured profile (and hence depend strongly on the fine details of the fitted curve), it is critical that the process of taking noisy, discrete observations and turning them into a smooth curve be done in a rigorous, statistically principled way. This is particularly true in the context of validation of gyrokinetic codes [14]: if a statistically meaningful comparison between the code and experiment is to be performed, then the high sensitivity of turbulent transport to profile gradients means that experimental data must be analyzed very carefully to deliver valid uncertainty estimates for the gradient scale lengths, as well as other derived experimental quantities to be compared, such as heat and particle fluxes. Furthermore, for the results of the analysis to be complete they must include an estimate of their uncertainty, so it is desirable that the fit be performed in a way that enables the uncertainty to be propagated through the model with a minimal number of code runs. Splines [310, 311], the traditional tool for this profile fitting and sampling task, have a number of shortcomings with respect to these objectives that will be discussed in [section 2.2.2](#). It will be shown that improvements in the quality of results, rate of convergence and level of automation of the data analysis workflow can be obtained by instead fitting profiles and producing samples using GPR. Because of these advantages, there has been considerable interest in the use of Gaussian processes in plasma physics which has grown over the time that this thesis work was conducted. Papers using Gaussian processes in plasma physics contexts include [305, 312–318]. As a profile fitting approach, GPR is very general, and can be applied in any situation where it is necessary to fit a smooth curve to noisy, discrete observations – even if the profile is a function of many independent variables. The sampling workflow presented in this chapter is also quite general, and can be applied to any code that takes entire profiles as inputs.

The rest of this chapter is organized as follows: [section 2.2](#) discusses uncertainty quantification, profile fitting with splines and the need for advanced profile fitting techniques. [Section 2.3](#) gives the mathematical details of GPR starting from a very basic example and building to cover all of the key steps in the analysis, including some extensions which were made to handle line-integrated (or otherwise transformed) quantities in the inference. [Section 2.4](#) applies GPR to synthetic data in order to verify that GPR produces reasonable, consistent results. This section includes a comparison of various covariance kernels, including several novel approaches to handling the nonstationarity intrinsic to tokamak profiles. [Section 2.5](#) applies GPR to real Alcator

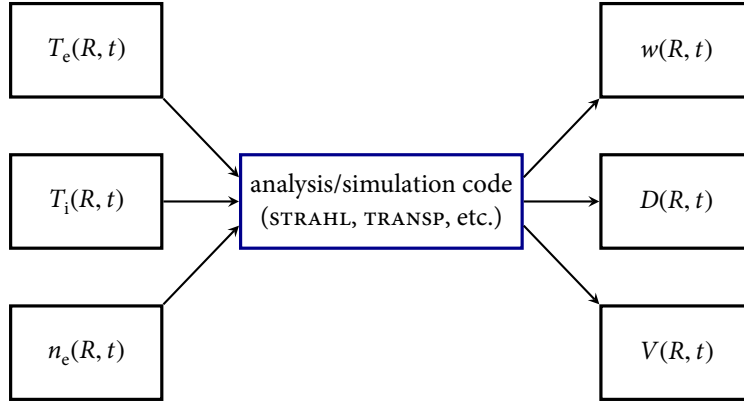
C-Mod data. This includes basic fits to L-mode  $n_e$  and  $T_e$  profiles (section 2.5.1), a fit to an I-mode  $n_e$  profile which incorporates line-integrated TCI data (section 2.5.2), fits to L-mode  $n_e$  and  $T_e$  profiles which include predictions of second derivatives in order to test theories regarding momentum transport and rotation reversals (section 2.5.3), a 2D fit (in space and time) to the  $T_e$  profile of a sawtooth-free, lower hybrid-driven discharge (section 2.5.4), and a fit to an H-mode  $n_e$  profile using the popular mtanh function as the Gaussian process' mean function (section 2.5.5). In section 2.6 GPR is used in an analysis workflow built around the STRAHL code [18, 19, 55, 276] to obtain experimental estimates of impurity transport coefficients  $D$  and  $V$  from measurements of impurity brightness, electron temperature and electron density profiles. This measurement is of interest as impurity transport is critical in determining the power balance of a confined plasma [2], and acts as an additional channel for comparison when testing transport codes [18, 19]. As will be shown in chapter 3, however, this simplistic impurity transport analysis has a number of shortcomings and should only be taken as an illustration of the potential use of GPR to propagate uncertainties through a more complicated model rather than a rigorous result. Finally, section 2.7 summarizes the key results of this work.

## 2.2 Uncertainty quantification and the need for advanced profile fitting

### 2.2.1 Uncertainty quantification with profile inputs

The situation this thesis is concerned with is shown schematically in figure 2.1: a code takes as inputs one or more profiles and computes one or more output quantities from these profiles. Furthermore, even if the required input is the local value of a gradient, the entire profile must still be analyzed to obtain this result from the discrete experimental measurements of the profile. In order to fully specify the result of the code it is necessary to compute not just a point estimate of the output but also to provide an estimate of the uncertainty in the output and its sensitivity to the input parameters. This task is most often accomplished with techniques such as Monte Carlo sampling: a series of input samples is prepared by randomly perturbing the input profiles according to their respective uncertainty estimates. These samples are then run through the code to produce an ensemble of possible realizations of the outputs. Computing the relevant summary statistics of this ensemble then gives the estimate of the value and its uncertainty. This workflow is shown schematically in figure 2.2. To carry this workflow out in practice when the inputs are noisy, discrete observations it is necessary to have a fitting procedure which takes the observations





**Figure 2.1:** Typical analysis scheme: the analysis code requires complete profiles of quantities that are only measured at discrete points in space and time. The outputs can in general also be functions of space and time. Here,  $w$  refers to an arbitrary profile output from the code. Transport coefficients  $D$  and  $V$  are explicitly specified as outputs as they are the quantities of interest for the analysis in both this chapter and [chapter 3](#). This figure is an updated version of one which originally appeared in [305].

and produces an estimate of the underlying smooth curve (and potentially its derivatives) and the accompanying uncertainty in a way that perturbed samples can be extracted.

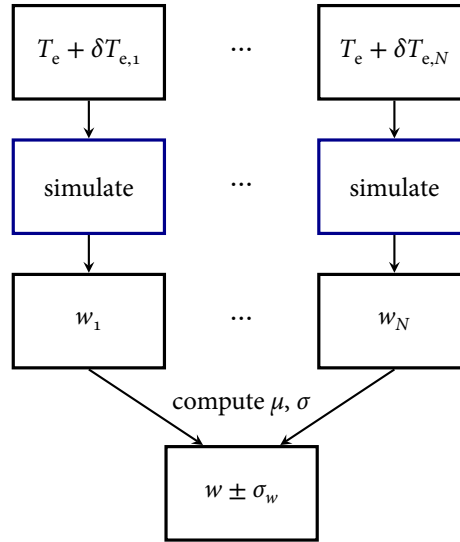
Furthermore, models for turbulence-induced transport are highly sensitive to the gradient scale lengths, given here as the normalized (against the minor radius  $a$ ) inverse scale length for  $T_e$  (but which could in general be for  $T_e$ ,  $T_i$ ,  $n_e$ , etc.):

$$\frac{a}{L_{T_e}} = a \frac{|\nabla T_e|}{T_e} \approx a \frac{|\partial T_e / \partial R|}{T_e}, \quad (2.1)$$

where  $R$  refers to the mapped outboard midplane major radius. Because this depends on the derivative  $\partial/\partial R$ , it is inevitably highly sensitive to the fine details of the profile. Superficially similar profiles can have comparable goodness-of-fit, but wildly different gradients. This sensitivity means it is essential to fit the discrete observations using a mathematically principled procedure, avoiding the temptation to pick the properties of the smoothing curve “by eye.”

### 2.2.2 Profile fitting with splines

A very common approach at present is to use a spline to fit a smooth curve to experimental data ([319] and the references therein give a mathematical perspective



**Figure 2.2:** Overview of sampling based (“Monte Carlo”) uncertainty propagation. The fitted profile  $T_e$  is perturbed  $N$  times by random (or otherwise intelligently selected [291]) amounts  $\delta T_e$ . This produces  $N$  possible realizations of the output quantity  $w$ . The relevant summary statistics are computed from this ensemble to give the final estimate of the quantity and its uncertainty.  $N$  must be selected such that these estimates are sufficiently accurate. This figure is an updated version of one which originally appeared in [305].

with some connections to plasma physics, [320] shows a more recent application including gradient scale lengths). Splines have the advantage of being thoroughly explored in a large body of literature and routines for performing spline fitting are readily available in most programming languages commonly used for scientific data analysis. There is, however, a number of drawbacks with splines that the use of GPR bypasses. In order to highlight the advantages of GPR, a brief outline of the mathematical properties of splines is given in section 2.2.2.1, followed by a discussion of their shortcomings in section 2.2.2.2.

### 2.2.2.1 Mathematical details of splines

The mathematical details of splines are discussed in detail in [310, 311, 321, 322] and the references therein. These references form the basis for this section, with other references given as needed. A (univariate) spline is a piecewise polynomial of degree  $d$  which has continuous derivatives up to order  $d - 1$ . Discontinuities in the  $d^{\text{th}}$  derivative are allowed to exist at a finite number of locations referred to as “knots.”

An interpolating spline is the curve satisfying these properties that is further required to pass through specified values at each of the knots. But, given noisy data, forcing the curve to go through all of the observations will inevitably result in a curve with too much unphysical structure. There are two general approaches to produce a curve that smoothes rather than interpolates the data. A smoothing spline is the spline of degree  $d = 2m - 1$  with knots located at each of the observations that minimizes

$$\frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_a^b (f^{(m)}(x))^2 dx, \quad (2.2)$$

where  $y_i$  is the observed value at location  $x_i$  (where  $i \in \{1, 2, \dots, n\}$ ),  $f(x)$  is the spline function and  $\lambda > 0$  is the smoothing parameter. This expression represents a tradeoff between the mean square error (first term) and the complexity of the curve (second term). The smoothing parameter sets the priority of this tradeoff: for small  $\lambda$  complex curves that lie close to the data are preferred, whereas large  $\lambda$  will drive the solution to smoother curves that are allowed to lie farther away from the data points. The other approach is to use a reduced set of knots and minimize the sum of squared residuals

$$\frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 \quad (2.3)$$

directly. In this case, the number of knots acts as the smoothing parameter and repositioning the knots allows the fit to have more or less spatial structure in some regions compared to others. This type of smoothing can be seen as a sum over piecewise polynomial basis functions  $S_j$  with weights  $C_j$ :

$$f(x) = \sum_j C_j S_j(x). \quad (2.4)$$

The B-spline basis functions are a particularly popular choice on account of their favorable computational and mathematical properties [321, 322]. With this approach the knot positions can be used as an additional parameter to help better fit the data, a situation referred to as a free-knot spline [323].

#### 2.2.2.2 Difficulties with spline fitting

With splines, selection of how flexible/complex the curve should be is a difficult problem. Lee [324] presents and compares a number of approaches for performing this operation, but the general theme is that this is a rather involved process, with

Dierckx [311] admitting that the positioning of knots often becomes a matter of (manual) trial and error. Holland [14] comments on manual choice of spline properties as a potentially substantial source of systematic error in tokamak profile fits. Free-knot splines additionally suffer from the so-called “lethargy property” which means that there will be many local minima to contend with when optimizing the knot positions [323, 325]. As will be seen, the approach adopted here selects the properties of the fitted curve using basic statistical procedures.

A further problem arises when attempting to fit data which depend on more than one independent variable. The most common choice when using splines on multivariate data is the tensor product spline [311], but this has the disadvantage of requiring that the knots fill a rectangular grid, which can present problems depending on the nature of the data to be fit. A further problem encountered is that most readily available implementations only support bivariate data. In contrast, the GPR approach can work on data of arbitrary dimension with little to no modification.

Confidence intervals for spline fits are discussed widely in the literature, including [326–331], though the most common software packages fall short of providing an implementation of these extra steps. Literature regarding uncertainties in derivatives of splines appears to be far more scarce, but includes [332–334]. There has been some work to provide confidence bands on the gradients of plasma profiles estimated using so-called exponential splines in [335–337]. A simple approach which is widespread in plasma physics is to perform Monte Carlo sampling to obtain uncertainty estimates for the fit and its gradients, such as was done in [320]. In contrast to the mathematical constructions in the preceding references or the brute force application of Monte Carlo sampling, the Gaussian process regression approach adopted here is based directly on the properties of the multivariate normal distribution, and therefore permits an intuitive interpretation of the variance of the fitted curve and its derivatives.

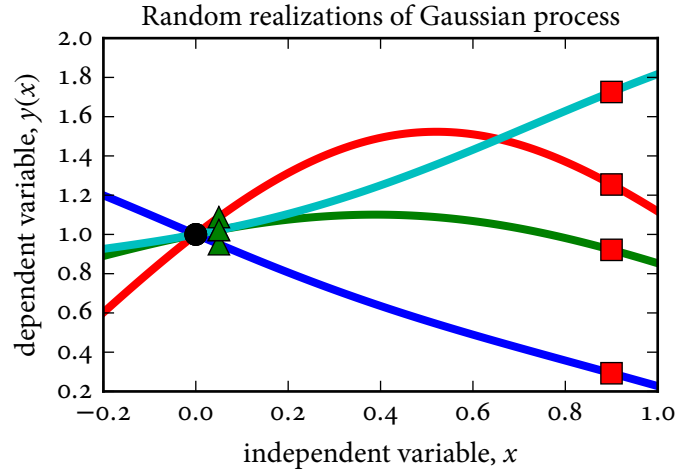
### 2.3 Profile fitting with Gaussian process regression

This section follows the development, notation and nomenclature of [297], with other references given as needed. For this work, the Python package `gptools` [306, 307] was implemented to provide support for GPR with gradient constraints and predictions. The `gptools` software is described in detail in [appendix E](#). The extensive work on derivatives of covariance kernels performed as part of this thesis is described in [appendix B](#).

### 2.3.1 Basic details of Gaussian process regression

Gaussian process regression (GPR) is a general-purpose Bayesian nonparametric regression technique [286, 297, 299–304]. Here, nonparametric refers to the fact that the observations must be used in order to make a prediction and that a specific functional form is not assumed: the form of the fit is left exceedingly flexible so that the data themselves can give rise to the correct shape in a statistically rigorous manner. As discussed in [297, 310], there is in fact a very deep mathematical connection between GPR and splines: smoothing splines are simply a special case of GPR with a specific choice of prior distribution. The difference is that GPR is cast in a statistical framework that makes interpretation of the fit, its gradients and the associated uncertainties far more straightforward. Furthermore, GPR can be used to yield a low-dimensional representation of the profile uncertainty which enables the use of efficient uncertainty propagation techniques, such as sparse quadrature [338], which can dramatically reduce the number of code runs necessary to propagate the uncertainty through a computationally expensive analysis code. Gaussian process regression has been in use in one form or another for many years under the term “kriging” [339], though the term Gaussian process regression is preferred here as it emphasizes the full statistical framework in which the approach is cast. This section necessarily assumes a basic understanding of statistics and data analysis, a review of which was given in section 1.4 and details of which are available in [282–286].

The essence of Gaussian process regression is that all observations and predictions are related through a multivariate normal distribution with a given mean function  $m(\mathbf{x})$  and covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j) \equiv \text{cov}[y(\mathbf{x}_i), y(\mathbf{x}_j)]$ . Here,  $\mathbf{x} \in \mathbb{R}^D$  is a  $D$ -dimensional vector corresponding to a single input location and  $\text{cov}[a, b]$  is the covariance between quantities  $a$  and  $b$ . For example,  $\mathbf{x}$  could consist of the  $R$ ,  $Z$ ,  $\varphi$  and  $t$  values at which a measurement was made, in which case  $D = 4$ . The mean function can encode any prior knowledge regarding the typical value or underlying shape of the data, but a zero mean prior distribution (i.e.,  $m(\mathbf{x}) = 0$ ) was found to perform well for the work presented here. The covariance kernel plays a key role in determining the smoothness of the fit: it determines how the correlation between points drops off with distance, as illustrated in figure 2.3. For a function to be a valid covariance kernel, it must give rise to a symmetric positive definite covariance matrix for all possible inputs. A covariance kernel is said to be *stationary* if it only depends on  $\mathbf{x}_i, \mathbf{x}_j$  through the quantity  $\boldsymbol{\tau} = \mathbf{x}_i - \mathbf{x}_j$ , and is furthermore said to be *isotropic* if it only depends on  $\mathbf{x}_i, \mathbf{x}_j$  through  $r = |\boldsymbol{\tau}|$ . A stationary covariance kernel has the same behavior anywhere along a given axis: the curve behaves the same at  $x = 0$  as it does at  $x = 10$ . This is similar to the common meaning of stationarity in the physical sciences, where it often refers to a process which is not changing in



**Figure 2.3:** Illustration of the effect of the covariance kernel. Under the assumption that the underlying true curve to be reconstructed is smooth, adjacent points should be very close in value but distant points can differ substantially. The covariance kernel determines how this correlation drops off with distance. Shown are four random draws from a Gaussian process with a squared exponential covariance kernel (equation (2.5)) with  $\sigma_f = 1$  and  $\ell = 1$ , conditioned on the single observation  $y = 1$  at  $x = 0$ . In other words, each curve represents a possible realization of the profile consistent with the observation and the selected covariance kernel. This choice of covariance kernel causes the values at  $x = 0.05$  (green triangles) to be close to the observed value  $y(0) = 1$  (black circle). But, the values at  $x = 0.9$  (red squares) are much less correlated with the observation at  $x = 0$  and hence exhibit a much wider spread across the four samples shown. This figure is an updated version of one which originally appeared in [305]. This figure was produced using [make\\_covariance\\_demo.py](#).

time, but in the present work the concept is applied primarily to spatial coordinates. An isotropic covariance kernel has the same behavior anywhere in the input space: the curve behaves the same at  $\mathbf{x} = [100, 0]$  as at  $\mathbf{x} = [0, 10]$ .

A very common and useful choice is the squared exponential (SE) covariance kernel

$$k_{\text{SE}}(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2\ell^2}\right), \quad (2.5)$$

where  $\ell$  is the covariance length scale which sets how fast the correlation drops off and  $\sigma_f^2$  is the signal variance which sets the extent of variation in the fitted curve. The SE covariance kernel is isotropic and encodes the assumption that the underlying

curve to be predicted is smooth (specifically, infinitely differentiable) and has a constant covariance length scale throughout its domain. It is very important to note that the covariance length scale  $\ell$  is *not* in any way the same thing as the gradient scale length: even if  $\ell$  is constant throughout the domain, the gradient scale length can still vary. The covariance length scale simply sets the (constant) distance the spatial correlation decays over. Further details of the SE and other covariance kernels are given in [appendix B](#).

A key feature advantage of GPR is that the covariance length scale and other properties of the covariance kernel are automatically inferred from the data, without requiring manual intervention to tell the fit how smooth or wiggly it should be. To illustrate this, [figure 2.4](#) shows the results of smoothing noisy synthetic data with a Gaussian process with an SE covariance kernel. In both cases the exact same data points were used. For one case, the uncertainties are small and the fit automatically infers a short  $\ell$  which permits lots of spatial structure to get the curve to go near each observation. For the other case, the uncertainties are large and the fit infers a long  $\ell$  which does not permit much spatial structure. The residuals are higher, but the uncertainty in the fitted curve is also higher, commensurate with the higher uncertainties on the data points.

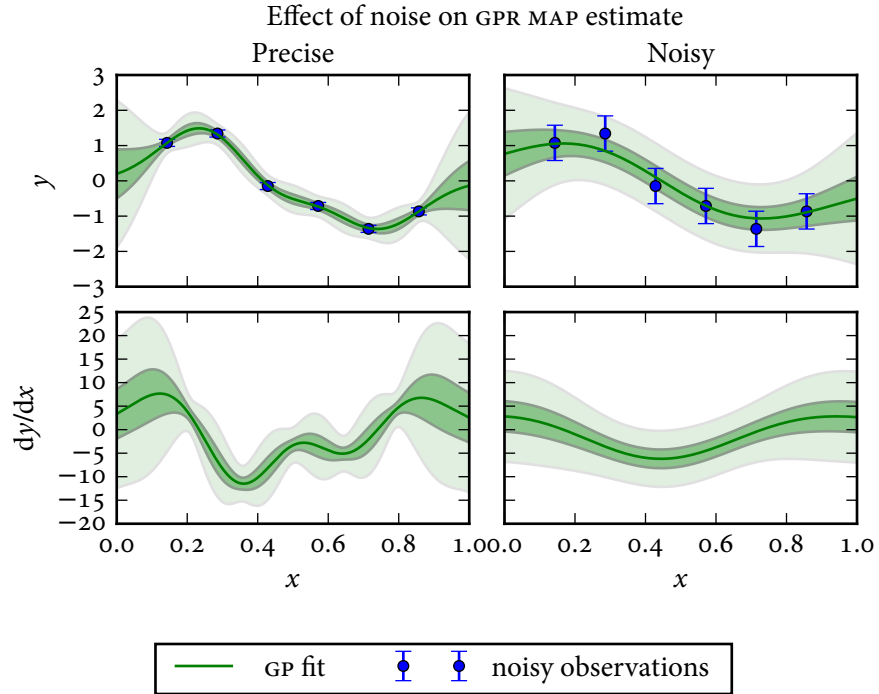
### 2.3.2 An intuitive picture of Gaussian process regression

Before presenting the full mathematical details, it is useful to consider the univariate ( $D = 1$ ) case with one observation and one prediction in order to obtain an intuitive picture of how GPR works. Given one location  $x$  at which a noise-free observation  $y$  has been made and one location  $x_*$  at which a prediction  $y_*$  is to be made, the joint prior probability density function (PDF) is then the bivariate normal

$$f_{Y,Y_*}(y, y_*) = \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x_*) \end{bmatrix}, \begin{bmatrix} k(x, x) & k(x, x_*) \\ k(x_*, x) & k(x_*, x_*) \end{bmatrix}\right) \quad (2.6)$$

$$= \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(x, x) & k(x, x_*) \\ k(x_*, x) & k(x_*, x_*) \end{bmatrix}\right), \quad (2.7)$$

where the notation  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  refers to the multivariate normal distribution with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  (see [section 1.4.5.5](#)) and in the last step a zero mean function has been used. This is the distribution *before* any observations have been included: it encodes prior assumptions regarding smoothness, bounds, etc. In the context of plasma physics,  $y$  could be the electron temperature  $T_e$  and  $x$  the normalized minor radius  $r/a$ , for example. To make this quantitative, take  $k$  to be the squared exponential given in [equation \(2.5\)](#) with  $\sigma_f = 1$  and  $r/\ell = |x - x_*|/\ell = 1$ ,



**Figure 2.4:** Results of fitting synthetic data with two different noise levels. The Gaussian process fits are shown with  $\pm 1\sigma$  and  $\pm 3\sigma$  uncertainty envelopes. The fitting procedure automatically selects the allowed complexity of the curve given the noise estimates for the observations. Note that only the MAP estimate (see section 2.3.6) is shown, so the uncertainties in the gradient may be underestimated (see the discussion in section 2.5.1). This figure was produced using `make_noise_demo.py`.

which gives the joint prior PDF

$$f_{Y, Y_*}(y, y_*) = \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & e^{-1/2} \\ e^{-1/2} & 1 \end{bmatrix}\right). \quad (2.8)$$

This is shown along with the marginal prior PDFs

$$f_Y(y) = \int_{-\infty}^{\infty} f_{Y, Y_*}(y, y_*) dy_* = \mathcal{N}(0, 1) \quad (2.9)$$

$$f_{Y_*}(y_*) = \int_{-\infty}^{\infty} f_{Y, Y_*}(y, y_*) dy = \mathcal{N}(0, 1) \quad (2.10)$$

in figure 2.5. (As indicated in the previous equations, the marginal PDF for  $y$  is the result of integrating the joint distribution over all possible values of  $y_*$ . In equa-



tion (2.9), for example,  $y_*$  is said to have been marginalized out of the distribution.) The effect of varying  $r/\ell$  is shown in figure 2.6.

Now consider the situation once a noise-free observation of a specific value for  $y$  has been made. The PDF of  $y_*$  conditioned on this observation is then

$$f_{Y_*|Y}(y_*|y) = \frac{f_{Y,Y_*}(y = y, y_*)}{f_Y(y = y)} \quad (2.11)$$

$$= \mathcal{N}\left(\frac{k(x_*, x)}{k(x, x)}y, k(x_*, x_*) - \frac{(k(x_*, x))^2}{k(x, x)}\right). \quad (2.12)$$

For instance, for  $y = 1$  and the parameters used above, this becomes

$$f_{Y_*|Y}(y_*|y = 1) = \mathcal{N}(e^{-1/2}, 1 - e^{-1}). \quad (2.13)$$

This is shown as the dashed curve in the top plot of figure 2.5. As is evident from both the figure and equation (2.13), the effect of including the information  $y = 1$  is to shift the expected value of  $y_*$  from  $\mathbb{E}[y_*] = 0$  to  $\mathbb{E}[y_*|y = 1] = e^{-1/2}$  and to lower the variance from  $\text{var}[y_*] = 1$  to  $\text{var}[y_*|y = 1] = 1 - e^{-1}$ . Hence, the prediction at  $x_*$  with  $\pm 1\sigma$  uncertainty interval is  $y_* = e^{-1/2} \pm (1 - e^{-1})^{1/2} = 0.6 \pm 0.8$ . As will be seen in subsequent sections, using more observations would reduce the uncertainty even more, as would be expected.

If instead a noisy observation  $z = y + \varepsilon$  is made, where the noise is  $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ , then a joint prior distribution between  $z$  and  $y_*$  is used instead:

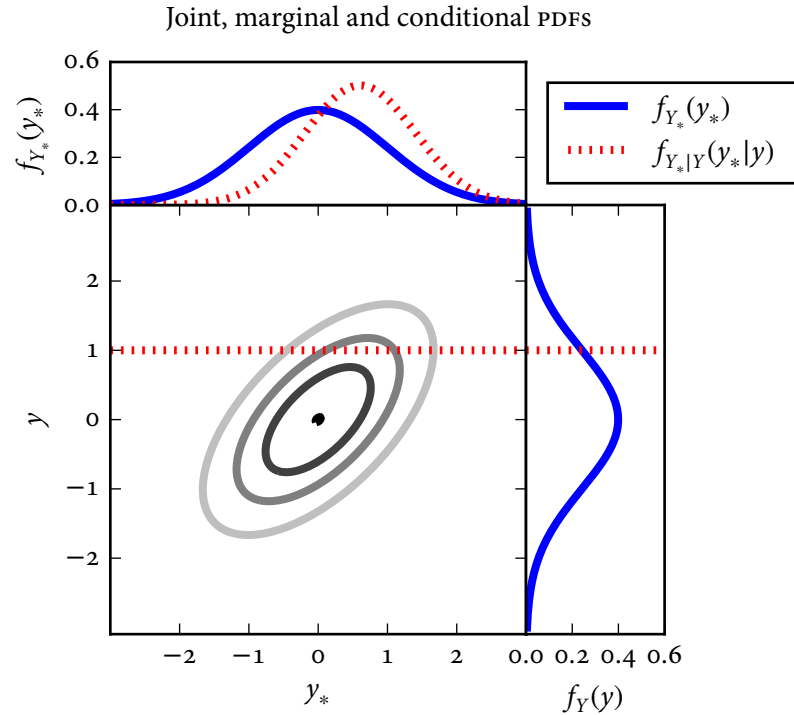
$$f_{Z,Y_*}(z, y_*) = \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(x, x) + \sigma_n^2 & k(x, x_*) \\ k(x_*, x) & k(x_*, x_*) \end{bmatrix}\right). \quad (2.14)$$

The analysis is then the same as before, with the one change that in equation (2.7) through equation (2.13)  $k(x, x)$  is replaced with  $k(x, x) + \sigma_n^2$ .

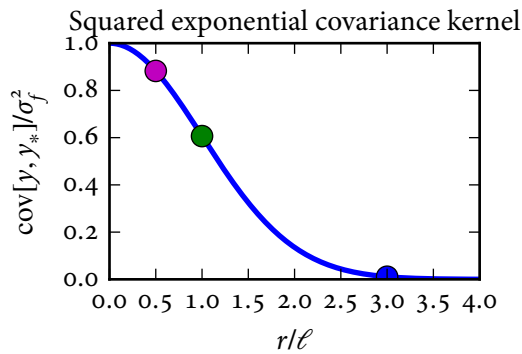
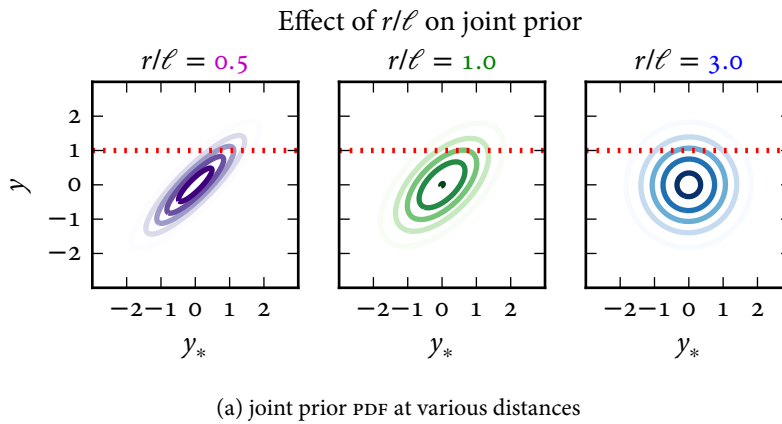
### 2.3.3 Full details of Gaussian process regression

The objective of profile fitting is to take  $n$  (potentially noisy) observations collected into the vector  $\mathbf{y}$  at locations which have been gathered into the  $D \times n$  matrix  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$  (where each  $\mathbf{x}_i$  is a  $D$ -dimensional column vector) and use them to make  $n_*$  predictions of the values of the underlying smooth curve collected into the vector  $\mathbf{y}_*$  at locations in the  $D \times n_*$  matrix  $\mathbf{X}_*$ . In the plasma context  $\mathbf{y}$  could be, for instance, the electron temperature  $T_e$  measured as a function of radius, in which case  $\mathbf{X}$  would be a (row) vector of radial locations. The key step of GPR is to assume that  $\mathbf{y}$  has the joint prior PDF

$$f_Y(\mathbf{y}) = \mathcal{N}(\mathbf{m}(\mathbf{X}), \mathbf{K}(\mathbf{X}, \mathbf{X}) + \mathbf{\Sigma}_n), \quad (2.15)$$



**Figure 2.5:** Joint prior probability density function  $f_{Y, Y_*}(y, y_*)$  (grey contours), with the marginal PDFs  $f_Y(y)$ ,  $f_{Y_*}(y_*)$  (solid blue curves), the conditional PDF  $f_{Y_*|Y}(y_*|y = 1)$  (dashed red curve), and the observation  $y = 1$  (horizontal dashed red line). The covariance matrix was constructed from a squared exponential covariance kernel with  $\sigma_f = 1$  and  $r/\ell = 1$ . The tilted ellipse shape of the contours is indicative of the correlation between  $y$  and  $y_*$ : the values for  $y$  and  $y_*$  are expected to be related (see figure 2.6). The effect of conditioning on the observation  $y = 1$  is to shift the distribution for  $y_*$  towards 1 and to make the distribution narrower: the shift from the blue to the red curve in the upper plot represents the change in the state of knowledge about  $y_*$  after making the observation  $y = 1$ . This figure is an updated version of one which originally appeared in [305]. This figure was produced using `make_single_var_demo.py`.



**Figure 2.6:** Effect of  $r/\ell$  on the shape of the joint prior PDF for  $r/\ell = 0.5, 1.0$  and  $3.0$ . The values were computed using an SE covariance kernel with  $\sigma_f = 1$ , which is shown in the lower plot. The smaller  $r/\ell$  is (equivalently, the closer  $x$  is to  $x_*$ ) the more correlated  $y$  and  $y_*$  are, thereby causing a more dramatic elongation of the tilted elliptical contours. As  $r/\ell$  increases,  $y$  and  $y_*$  become less correlated and the contours become circular. Also shown is the observation  $y = 1$  as the **red horizontal dashed line**. As  $r/\ell$  increases, the observation is less informative and therefore the conditional PDF will be wider. This can also be thought of in terms of the smoothing effect of  $\ell$ : the larger  $\ell$  is, the smaller  $r/\ell$  will be for any given value of  $r$ . Hence, a larger  $\ell$  leads to a smoother curve by driving down the variance over a larger region around each observation. This figure is an updated version of one which originally appeared in [305]. This figure was produced using `make_r_l_demo.py` and `make_r_l_scale.py`.

where the notation  $\mathbf{m}(\mathbf{X})$  indicates the  $n$  element vector formed by evaluating  $m(\mathbf{x})$  at each of the columns of  $\mathbf{X}$ ,  $\mathbf{K}(\mathbf{X}, \mathbf{X})$  indicates the  $n \times n$  matrix formed by evaluating  $k(\mathbf{x}_i, \mathbf{x}_j)$  between each of the possible pairs of columns in  $\mathbf{X}$ , and  $\boldsymbol{\Sigma}_n$  is the noise covariance matrix of the observations. In general  $\boldsymbol{\Sigma}_n$  could include correlated noise, but in this application a diagonal matrix was used to model uncorrelated heteroscedastic Gaussian noise. While it is possible to include powerful constraints in the prior distribution/evaluation of the model itself [340], for this work it was found that the simple zero mean prior distribution as given in [297] was sufficient, such that the joint prior PDF between the observations  $\mathbf{y}$  and the predictions  $\mathbf{y}_*$  is

$$f_{Y, Y_*}(\mathbf{y}, \mathbf{y}_*) = \mathcal{N}\left(\mathbf{0}_{n+n_*}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right), \quad (2.16)$$

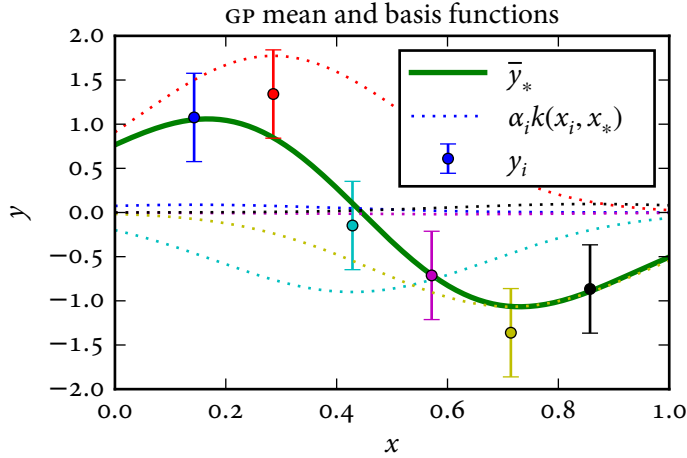
where  $\mathbf{X}_*$ ,  $\mathbf{y}_*$  are the locations and values of the predictions, respectively, and  $\mathbf{0}_{n+n_*}$  is the  $(n + n_*)$ -dimensional zero vector. (Note that  $\mathbf{X}_* \in \mathbb{R}^{D \times n_*}$ ,  $\mathbf{y}_* \in \mathbb{R}^{n_*}$ , where there are  $n_*$  points at which a prediction is to be made.)

Note that the assumed prior distribution for  $\mathbf{y}$  and  $\mathbf{y}_*$  in equation (2.16) does not explicitly restrict the prediction  $\mathbf{y}_*$  to have a specific functional form (though refer to equation (2.19) for the form which is implicitly assumed). This choice of prior distribution merely establishes the spatial structure of the covariance between any given pair of points (whether observations or predictions). While the use of a mean function as in equation (2.15) establishes the general shape of the curve (particularly when only a small number of observations is available), a sufficient number of observations can pull the predicted curve away from an oversimplified or otherwise incorrect mean function. In particular, most of the work in this thesis used a zero mean function, but the resulting curve was far from zero wherever there were nonzero observations.

What is of interest to make predictions is the conditional PDF of  $\mathbf{y}_*$  given the observations  $\mathbf{y}$ . From equation (1.53), this is

$$f_{Y_*|Y}(\mathbf{y}_*|\mathbf{y}) = \mathcal{N}\left(\mathbf{K}(\mathbf{X}_*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)^{-1}\mathbf{y}, \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*)\right). \quad (2.17)$$

The conditional mean then gives the prediction and the diagonal elements of the conditional covariance matrix give the variance in the prediction. As this can be evaluated at any point  $\mathbf{x}_*$ , a Gaussian process is said to represent a distribution over functions. Note that inversion of an  $n \times n$  symmetric positive definite matrix is required to find the solution, which leads to an asymptotic complexity of  $\mathcal{O}(n^3)$ .



**Figure 2.7:** Synthetic data, GP mean and basis functions centered at each point. The data are the same as in figure 2.4. The heavy green solid curve is the mean  $\bar{y}_*$  and the dotted, colored curves are the weighted basis functions  $\alpha_i k(x_i, \mathbf{x}_*)$ . The points are color-coded to match the basis function which is centered on that point. As shown by equation (2.19), the mean is the sum of all six of these basis functions. This figure was produced using `make_noise_demo.py`.

The mean of equation (2.17) merits further inspection:

$$\bar{\mathbf{y}}_* = \mathbb{E}[\mathbf{y}_* | \mathbf{y}] = \mathbf{K}(\mathbf{X}_*, \mathbf{X}) (\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)^{-1} \mathbf{y} \quad (2.18)$$

$$\bar{y}_*(\mathbf{x}_*) = \sum_{i=1}^n \alpha_i k(x_i, \mathbf{x}_*), \quad (2.19)$$

where the weights  $\alpha_i$  are linear combinations of the measurements:

$$\boldsymbol{\alpha} = (\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)^{-1} \mathbf{y}. \quad (2.20)$$

The conditional mean as a function of  $\mathbf{x}_*$  is a weighted sum of  $n$  copies of the covariance kernel, with each copy centered at an observation as shown in figure 2.7. This makes the connection between GPR and splines obvious: if  $k$  were an appropriately selected polynomial basis function, this would be equivalent to the spline given in equation (2.4) with the knots centered at each observation, though with the added benefits alluded to previously in section 2.2.2 and the additional flexibility of being able to select from a wider variety of basis functions in order to obtain whatever properties might be required for the task at hand.

### 2.3.4 Explanation of the result of Gaussian process regression

The end result of Gaussian process regression is the multivariate normal posterior distribution given in [equation \(2.17\)](#) with mean vector

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{y}_* | \mathbf{y}] = \mathbf{K}(\mathbf{X}_*, \mathbf{X}) (\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)^{-1} \mathbf{y}, \quad (2.21)$$

and covariance matrix

$$\boldsymbol{\Sigma} = \text{cov}[\mathbf{y}_*, \mathbf{y}_* | \mathbf{y}] = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X}) (\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}_*). \quad (2.22)$$

Here and in [equation \(2.17\)](#),  $f_{Y_*|Y}(\mathbf{y}_* | \mathbf{y})$  is the probability density function (PDF) for the predictions  $\mathbf{y}_*$  at locations  $\mathbf{X}_*$  conditioned on the observations  $\mathbf{y}$  at locations  $\mathbf{X}$ , the notation  $\mathbf{K}(\mathbf{A}, \mathbf{B})$  means the result of evaluating the covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$  between all possible pairs of locations in  $\mathbf{A}$  and  $\mathbf{B}$ , and  $\boldsymbol{\Sigma}_n$  is the noise covariance matrix. The mean given in [equation \(2.21\)](#) is then used as the estimate of the profile and the diagonal elements of the covariance matrix given in [equation \(2.22\)](#) represent the uncertainty on the fit. The term ‘‘posterior distribution’’ refers to the fact that this is the distribution that has been conditioned on the observations, and is in contrast to the ‘‘prior distribution’’ which is the distribution before observations have been included. The prior distribution encodes any prior knowledge regarding the form of the solution.

For this construction, the noise is assumed to be (approximately) Gaussian, but can be correlated and/or heteroscedastic (i.e., each observation can have a different variance). The approach used in this work also assumes that systematic errors have been calibrated out and/or are handled in some external uncertainty quantification step. It is possible, however, to construct a more complicated model which includes systematic effects – refer to the discussion of Bayesian integrated data analysis (IDA) in [section 2.3.10](#).

To put it more concretely, the following procedure would be used to smooth a profile with  $n = 15$  measurements of the electron temperature  $T_e$  sampled at radial locations  $r$ . The data are assumed to have independent, zero-mean Gaussian noise with standard deviation  $\sigma_{T_e}$  which has been estimated separately, and only a single time slice or averaged profile is analyzed (i.e., univariate smoothing,  $D = 1$ ). The smooth curve is evaluated at  $n_* = 100$  radial locations  $r_*$ .

1. Assemble all of the  $T_e$  measurements into the 15 element vector

$$\mathbf{y} = [T_{e,1}, T_{e,2}, \dots, T_{e,15}]^T.$$

2. Assemble the noise variance  $\sigma_{T_e}^2$  along the diagonal of the  $15 \times 15$  matrix

$$\mathbf{\Sigma}_n = \begin{bmatrix} \sigma_{T_{e,1}}^2 & 0 & \dots & 0 \\ 0 & \sigma_{T_{e,2}}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \sigma_{T_{e,15}}^2 \end{bmatrix}. \quad (2.23)$$

3. Evaluate the covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$  between all pairs of points the observations were taken at to form the  $15 \times 15$  matrix

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(r_1, r_1) & k(r_1, r_2) & \dots & k(r_1, r_{15}) \\ k(r_2, r_1) & k(r_2, r_2) & \dots & k(r_2, r_{15}) \\ \vdots & \vdots & \ddots & \vdots \\ k(r_{15}, r_1) & \dots & \dots & k(r_{15}, r_{15}) \end{bmatrix}. \quad (2.24)$$

4. Evaluate the covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$  between all pairs of points the smoothed curve is to be evaluated at to form the  $100 \times 100$  matrix

$$\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) = \begin{bmatrix} k(r_{*,1}, r_{*,1}) & k(r_{*,1}, r_{*,2}) & \dots & k(r_{*,1}, r_{*,100}) \\ k(r_{*,2}, r_{*,1}) & k(r_{*,2}, r_{*,2}) & \dots & k(r_{*,2}, r_{*,100}) \\ \vdots & \vdots & \ddots & \vdots \\ k(r_{*,100}, r_{*,1}) & \dots & \dots & k(r_{*,100}, r_{*,100}) \end{bmatrix}. \quad (2.25)$$

5. Evaluate the covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$  between all pairs of points the observations were taken at and the smoothed curve is to be evaluated at to form the  $15 \times 100$  matrix

$$\mathbf{K}(\mathbf{X}, \mathbf{X}_*) = \begin{bmatrix} k(r_1, r_{*,1}) & k(r_1, r_{*,2}) & \dots & k(r_1, r_{*,100}) \\ k(r_2, r_{*,1}) & k(r_2, r_{*,2}) & \dots & k(r_2, r_{*,100}) \\ \vdots & \vdots & \ddots & \vdots \\ k(r_{15}, r_{*,1}) & \dots & \dots & k(r_{15}, r_{*,100}) \end{bmatrix} \quad (2.26)$$

and its transpose  $\mathbf{K}(\mathbf{X}_*, \mathbf{X}) = \mathbf{K}(\mathbf{X}, \mathbf{X}_*)^\top$ .

6. Construct the posterior mean vector by substituting these quantities into equation (2.21):

$$\boldsymbol{\mu} = \mathbf{K}(\mathbf{X}_*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \mathbf{\Sigma}_n)^{-1} \mathbf{y}. \quad (2.27)$$

This is the prediction of the smooth curve.

7. Construct the posterior covariance matrix by substituting these quantities into equation (2.22):

$$\boldsymbol{\Sigma} = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*). \quad (2.28)$$

This matrix contains the variance in the individual predictions along the diagonal and the covariances between the predictions at different points in the off-diagonal elements.

All of these steps are implemented in open-source software which was developed as part of this work [306, 307].

### 2.3.5 Prediction of gradients and their uncertainties

One of the features that makes GPR very well-suited to plasma profile analysis is that the covariance matrix given in equation (2.22) can be constructed to include not just the value of the fit but also the gradients of the fit – both for the observations and for the predictions [297, 341–344]. This means that it is trivial both to add a zero slope constraint at the magnetic axis and to obtain values *and uncertainty estimates* for the gradients. The relationship between a Gaussian process and its derivatives is

$$\text{cov} \left[ y_i, \frac{\partial y_j}{\partial x_{j_d}} \right] = \frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial x_{j_d}} \quad (2.29)$$

$$\text{cov} \left[ \frac{\partial y_i}{\partial x_{i_c}}, \frac{\partial y_j}{\partial x_{j_d}} \right] = \frac{\partial^2 k(\mathbf{x}_i, \mathbf{x}_j)}{\partial x_{i_c} \partial x_{j_d}}, \quad (2.30)$$

where the notation  $\partial/\partial x_{j_d}$  refers to a derivative with respect to the  $d^{\text{th}}$  component of the input  $\mathbf{x}_j$  to the covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$ . Repeated application of these equations allows derivatives of arbitrary order to be included. By constructing the joint distribution between the observed values  $y$ , observed values  $\partial y/\partial x_d$  (and possibly higher order derivatives), predicted values  $y_*$  and predicted values  $\partial y_*/\partial x_{*d}$  (and possibly higher order derivatives) it is possible to make a simultaneous prediction of the underlying smooth curve, its derivative(s) *and* the uncertainty in both the value and its derivative(s).

Another application is to use derivative information to incorporate symmetry and boundary constraints. In the work presented here, an artificial zero slope “observation” at the magnetic axis was used to implement a symmetry constraint. While such constraints can be included through transformations on the prior distribution itself [345], the approach adopted here was found to perform well in practice. (Note



that [345] uses finite differences to approximate gradients, whereas the approach used here allows the gradient to be constrained and predicted directly without requiring such approximations.)

### 2.3.5.1 The appropriate coordinate systems to impose constraints in

There is a disturbing amount of confusion as to which of the flux surface coordinates (see section D.2 for a discussion of the more common systems in use) in which it is appropriate to impose a zero slope constraint at the magnetic axis. The following discussion, stemming from a conversation with J. Freidberg [346], attempts to explore this question and to justify both the use of  $r/a$  as the coordinate to perform fits in and the use of a zero slope constraint at the magnetic axis in this coordinate system.

Take some quantity  $w$  to be a flux function:  $w$  is constant on a given flux surface, but varies from flux surface to flux surface. The magnetic axis is the center of the nested, closed flux surfaces, which is defined to have coordinates  $(R_0, Z_0)$ . Consider the infinitesimal flux surface that encircles the magnetic axis. Clearly, for an infinitesimal displacement of  $\pm\epsilon$  in major radius  $R$  away from the magnetic axis, either sign of displacement will end up on the same flux surface. Therefore, it is clear that

$$\left. \frac{dw}{dR} \right|_{R=R_0} = 0 \quad (2.31)$$

is an appropriate constraint, as is

$$\left. \frac{dw}{dZ} \right|_{Z=Z_0} = 0, \quad (2.32)$$

but outboard midplane major radius is a far more useful flux surface label in practice. These two constraints could in fact be seen as a definition of the magnetic axis. Converting to minor radius  $r = R - R_0$  preserves the condition in equation (2.31), as does normalizing the minor radius to  $r/a = (R - R_0)/a$  where  $a$  is the minor radius of the last closed flux surface (LCFS). Now consider  $w$  expressed as a function of some other flux surface label  $\rho$  which could be, for example, normalized poloidal flux  $\psi_n$ . The derivative from equation (2.31) can then be rewritten as

$$\frac{dw}{dR} = \frac{dw}{d\rho} \frac{d\rho}{dR}. \quad (2.33)$$

But, because we have taken  $\rho$  to be a flux surface quantity itself,  $d\rho/dR|_{R=R_0}$  is guaranteed to be zero. Therefore, the constraint equation (2.31) will typically be satisfied *independent* of the value of  $dw/d\rho$  at the magnetic axis as long as there is nothing pathological about the coordinate used or the curve fit to the data.

But now consider the class of normalized coordinates which are constructed to act like  $r/a$ , specifically the square root of the normalized poloidal flux  $\rho_{\text{pol}} = \psi_n^{1/2}$  and the square root of the normalized toroidal flux  $\rho_{\text{tor}} = \varphi_n^{1/2}$ . These coordinates are often used because the fluxes go like the area, or  $R^2$ , and hence their square roots go like  $R$ . Now consider the derivative from [equation \(2.31\)](#) again:

$$\frac{dw}{dR} = \frac{dw}{d\rho_{\text{pol}}} \frac{d\rho_{\text{pol}}}{dR} = \frac{dw}{d\rho_{\text{pol}}} \frac{d\rho_{\text{pol}}}{d\psi_n} \frac{d\psi_n}{dR}. \quad (2.34)$$

In the middle step it can already be seen that a problem might be expected because if  $\rho_{\text{pol}} \propto R$  then  $d\rho_{\text{pol}}/dR \propto 1$ . Substituting  $d\rho_{\text{pol}}/d\psi_n = 1/(2\psi_n^{1/2})$  gives

$$\frac{dw}{dR} = \frac{1}{2\psi_n^{1/2}} \frac{dw}{d\rho_{\text{pol}}} \frac{d\psi_n}{dR}. \quad (2.35)$$

The problem is that  $\psi_n$  goes to zero at the magnetic axis, and hence the first term will blow up. Noting that  $\psi_n \propto R^2$ , the first term goes like  $1/R$  and the third term goes like  $R$ . Therefore, to ensure that  $dw/dR|_{R=R_0} = 0$  is satisfied, we must have  $dw/d\rho_{\text{pol}}|_{\rho_{\text{pol}}=0} = 0$ . The same reasoning applies to  $\rho_{\text{tor}}$ .

Another coordinate in common use is the normalized flux surface volume,  $V_n = V/V_a$  where  $V$  is the volume enclosed by the flux surface of interest and  $V_a$  is the volume enclosed by the LCFS. The volume of a given flux surface with centroid  $R_c$  and area  $A$  is given by Pappus' second centroid theorem [\[347, 348\]](#) as  $V = 2\pi R_c A$ . Assuming circular flux surfaces and ignoring the Shafranov shift, the centroid is  $R_c = R_0$  and the area is  $A = \pi r^2 = \pi(R - R_0)^2$ , so  $V = 2\pi^2 R_0 (R - R_0)^2$ . Therefore,  $V_n \propto R^2$  (and not  $R^3$  as might be intuitively expected), and the same arguments as for  $\psi_n$  and  $\varphi_n$  hold.

Because it permits direct application of [equation \(2.31\)](#) and avoids any of the issues discussed above, the results presented in this thesis primarily come from fitting the profile as a function of  $r/a$  with the constraint  $dw/d(r/a)|_{r/a=0} = 0$  imposed through the addition of a synthetic data point to the Gaussian process.

### 2.3.6 Selection of a covariance kernel and its hyperparameters

The SE covariance kernel given in [equation \(2.5\)](#) has two hyperparameters  $\sigma_f$  and  $\ell$  that determine the properties of the fit; other choices of covariance kernel may have more hyperparameters. The term *hyperparameter* is used because we are referring to parameters that determine the prior distribution rather than the shape of the fitted curve directly. It is also instructive to recall at this point that the hyperparameters are *not* the parameters of a parametric model that the data are reduced into:

a specific functional form is not assumed, and the observations must be used to make predictions. Rather, the hyperparameters dictate the structure of the spatial covariance between any given pair of points (whether observations or predictions) which, together with the observations, determine the prediction. In other words, given a specific, arbitrary choice for  $\sigma_f$  and  $\ell$ , the conditioned PDF in [equation \(2.17\)](#) will yield a curve that is most consistent with the observations *given that particular choice of hyperparameters*. What now remains is to pick the hyperparameters (and covariance kernel) that are most consistent with the data. Note that this is a different question than asking which hyperparameters fit the data with the smallest residual: with the SE covariance kernel, for example, one could always make the error small by taking  $\ell$  to be very small, but then the model would be fitting the noise. There are several possible approaches to carry out the selection of hyperparameters discussed under the topic of model comparison and selection in [\[286, 297\]](#). Three levels of sophistication are considered here: maximum likelihood estimation, maximum a posteriori estimation and marginalization over the hyperparameters.

The simplest approach presented here is the maximum likelihood (ML) estimate. The ML estimate is a point estimate for the hyperparameters consisting of the values of the hyperparameters that maximize the probability of the observed data  $\mathbf{y}$  given the hyperparameters  $\boldsymbol{\theta}$ . This is simply the marginal PDF for  $\mathbf{y}$  as in [equation \(2.9\)](#) (but now given for the general case) with the dependence on the vector of hyperparameters  $\boldsymbol{\theta} \equiv [\sigma_f, \ell, \dots]$  made explicit:

$$f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{m}(\mathbf{X}|\boldsymbol{\theta}), \mathbf{K}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \boldsymbol{\Sigma}_n), \quad (2.36)$$

where the notations  $\mathbf{m}(\mathbf{X}|\boldsymbol{\theta})$  and  $\mathbf{K}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta})$  respectively refer to the  $n$  element mean vector and  $n \times n$  covariance matrix constructed using the mean function  $m$  and covariance kernel  $k$ , respectively, with the specific hyperparameters  $\boldsymbol{\theta}$ . Explicitly, the ML estimate is

$$\hat{\boldsymbol{\theta}}^{\text{ML}} = \arg \max_{\boldsymbol{\theta}} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}). \quad (2.37)$$

In practice what is used is the natural logarithm of the likelihood:

$$\ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) = -\frac{1}{2} \mathbf{y}^\top (\mathbf{K}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \boldsymbol{\Sigma}_n)^{-1} \mathbf{y} - \frac{1}{2} \ln |\mathbf{K}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \boldsymbol{\Sigma}_n| - \frac{n}{2} \ln 2\pi, \quad (2.38)$$

where a zero-mean prior distribution has been assumed in going from [equation \(2.36\)](#) to this result. Each of these terms permits a simple interpretation [\[297\]](#):

- The first term is the only one that depends on the observations  $\mathbf{y}$  and is related to how well the model fits the data.

- The second term depends only on the determinant of the covariance matrix, and is related to the complexity of the model.
- The final term only depends on the number of observations  $n$  and is a normalization constant which does not depend on the hyperparameters  $\boldsymbol{\theta}$  and hence does not affect the optimization.

The next level of sophistication is to include prior information on the hyperparameters in order to obtain the posterior PDF for the hyperparameters. This prior information, encoded in the (hyper)prior distribution  $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ , can readily be included in [equation \(2.38\)](#) using Bayes' rule to give the posterior PDF for the hyperparameters:

$$f_{\boldsymbol{\theta}|Y}(\boldsymbol{\theta}|\mathbf{y}) = \frac{f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{f_Y(\mathbf{y})}. \quad (2.39)$$

The maximum a posteriori (MAP) estimate for the hyperparameters is a point estimate consisting of the most likely values of the hyperparameters  $\boldsymbol{\theta}$  given the observations  $\mathbf{y}$ , or

$$\hat{\boldsymbol{\theta}}^{\text{MAP}} = \arg \max_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}|Y}(\boldsymbol{\theta}|\mathbf{y}). \quad (2.40)$$

Note that the term in the denominator of [equation \(2.39\)](#) is simply a normalizing constant which is independent of  $\boldsymbol{\theta}$ , so the end result is that the expression to be maximized over  $\boldsymbol{\theta}$  is simply [equation \(2.38\)](#) with the extra factor  $\ln f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$  added in.

In order to use a gradient-based optimizer or advanced Markov chain Monte Carlo (MCMC) sampling schemes such as Hamiltonian Monte Carlo (HMC) [286, 349, 350], it is necessary to have the derivative of [equation \(2.39\)](#) with respect to each hyperparameter  $\theta_i \in \boldsymbol{\theta}$ :

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \ln f_{\boldsymbol{\theta}|Y}(\boldsymbol{\theta}|\mathbf{y}) &= \frac{\partial}{\partial \theta_i} (\ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) + \ln f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) - \ln f_Y(\mathbf{y})) \\ &= -\frac{1}{2} \mathbf{y}^T \frac{\partial}{\partial \theta_i} (\mathbf{K}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \boldsymbol{\Sigma}_n)^{-1} \mathbf{y} \\ &\quad - \frac{1}{2} \frac{\partial}{\partial \theta_i} \ln |\mathbf{K}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \boldsymbol{\Sigma}_n| + \frac{\partial}{\partial \theta_i} \ln f_{\boldsymbol{\theta}}(\boldsymbol{\theta}). \end{aligned} \quad (2.41)$$

$$\quad (2.42)$$

Note from appendix A.3.1 of [297] the following two identities:

$$\frac{\partial}{\partial \theta_i} \mathbf{K}^{-1} = -\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_i} \mathbf{K}^{-1}, \quad \frac{\partial}{\partial \theta_i} \ln |\mathbf{K}| = \text{tr} \left( \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_i} \right). \quad (2.43)$$

Substituting,

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \ln f_{\Theta|Y}(\theta|y) &= \frac{1}{2} y^\top (\mathbf{K}(\mathbf{X}, \mathbf{X}|\theta) + \Sigma_n)^{-1} \frac{\partial}{\partial \theta_i} (\mathbf{K}(\mathbf{X}, \mathbf{X}|\theta) + \Sigma_n) (\mathbf{K}(\mathbf{X}, \mathbf{X}|\theta) + \Sigma_n)^{-1} y \\ &\quad - \frac{1}{2} \text{tr} \left( (\mathbf{K}(\mathbf{X}, \mathbf{X}|\theta) + \Sigma_n)^{-1} \frac{\partial}{\partial \theta_i} (\mathbf{K}(\mathbf{X}, \mathbf{X}|\theta) + \Sigma_n) \right) + \frac{\partial}{\partial \theta_i} \ln f_{\Theta}(\theta), \end{aligned} \quad (2.44)$$

where  $\Sigma_n$  has been retained inside the derivatives because it may itself depend on some hyperparameters.

It must be noted that both the ML and MAP approaches are point estimates: they select a single value of the hyperparameters given the data and possibly some prior information. The posterior distribution for  $\theta$  (equation (2.39)) can, however, have substantial variance, leading to uncertainty in the fit that is not captured with a point estimate like ML or MAP gives. What is better is to employ a fully Bayesian approach and marginalize (integrate) the predictive distribution over the hyperparameters:

$$\begin{aligned} f_{Y_*|Y}(\mathbf{y}_*|y) &= \int f_{Y_*,\Theta|Y}(\mathbf{y}_*, \theta|y) d\theta \\ &= \int f_{Y_*|Y,\Theta}(\mathbf{y}_*|y, \theta) f_{\Theta|Y}(\theta|y) d\theta, \end{aligned} \quad (2.45)$$

where the second line follows from the definition of conditional probability (equation (1.11)), the term  $f_{Y_*|Y,\Theta}(\mathbf{y}_*|y, \theta)$  is equation (2.17) with the conditioning on the hyperparameters  $\theta$  made explicit and  $f_{\Theta|Y}(\theta|y)$  is as in equation (2.39). This integration was efficiently carried out in practice using Markov chain Monte Carlo (MCMC) integration [286, 351, 352], specifically the affine-invariant ensemble sampler discussed in [353, 354]. This algorithm uses an ensemble of many “walkers” (typically on the order of several hundred) which in effect perform a random walk guided by the posterior distribution to yield a collection of samples  $\{\theta^{(i)}\}$  of the hyperparameters distributed according to  $f_{\Theta|Y}(\theta|y)$  which can then be used to evaluate integrals like equation (2.45). For the affine-invariant ensemble sampler the random walk is governed by a single dimensionless parameter, the proposal width  $a > 1$ . This eliminates the need to tune the proposal distribution, beyond selecting a value for the single parameter  $a$  which gives acceptable convergence and mixing between multiple modes. This formulation can also be used to account for uncertainties in the independent variable  $\mathbf{X}$  by noting that the result of equation (2.45) is implicitly conditioned on  $\mathbf{X}$  and then marginalizing out the values of  $\mathbf{X}$ , but this was not done in the present work.

The choice between different types of covariance kernels is a more complicated question. In the simplest case, the log-likelihood given in equation (2.38) can be

recognized as being implicitly conditioned on a choice of covariance kernel. Then, the best covariance kernel can be selected by picking the one that maximizes the likelihood, as was done in chapter 5 of [297]. The most general approach is to use Bayesian model comparison [285, 286] and choose between covariance kernels based on the *evidence*, which is the probability of the data given the covariance kernel (marginalized over the covariance kernel's hyperparameters):

$$f_{Y|K}(y|k) = \int f_{Y|\theta,K}(y|\theta, k) f_{\theta|K}(\theta|k) d\theta, \quad (2.46)$$

where the conditioning on the kernel  $k$  has now been made explicit.

### 2.3.7 Computing mean profiles and their uncertainties from MCMC output

Given a set of  $m$  samples  $\{\theta^{(i)}\}$ , the marginalized mean profile is computed using the law of iterated expectations:

$$\mathbb{E}[y_*|y] = \mathbb{E} [\mathbb{E}[y_*|y, \theta]] = \frac{1}{m} \sum_{i=1}^m \mathbb{E}[y_*|y, \theta^{(i)}], \quad (2.47)$$

where  $\mathbb{E}[y_*|y, \theta^{(i)}]$  is the posterior mean given in equation (2.21) evaluated with the given vector of hyperparameters  $\theta^{(i)}$ . The variance in the marginalized estimate of the profile is computed using the law of total variance:

$$\begin{aligned} \text{var}[y_*|y] &= \text{var} [\mathbb{E}[y_*|y, \theta]] + \mathbb{E} [\text{var}[y_*|y, \theta]] \\ &= \frac{1}{m-1} \sum_{i=1}^m (\mathbb{E}[y_*|y, \theta^{(i)}] - \mathbb{E}[y_*|y])^2 + \frac{1}{m} \sum_{i=1}^m \text{var}[y_*|y, \theta^{(i)}], \end{aligned} \quad (2.48)$$

where  $\text{var}[y_*|y, \theta^{(i)}]$  is the diagonal of the posterior covariance matrix given in equation (2.22) evaluated with the given vector of hyperparameters  $\theta^{(i)}$ .

The uncertainties in the normalized inverse gradient scale lengths are computed using the uncertainty propagation equation [282]:

$$\frac{a}{L_y} \approx -a \frac{\partial y / \partial R}{y} = -\frac{y'}{y} \quad (2.49)$$

$$\text{var} \left[ \frac{a}{L_y} \right] = \text{var}[y] \left( \frac{y'}{y^2} \right)^2 + \text{var}[y'] \left( -\frac{1}{y} \right)^2 + 2 \text{cov}[y, y'] \left( \frac{y'}{y^2} \right) \left( -\frac{1}{y} \right), \quad (2.50)$$

where  $y' = \partial y / \partial (r/a)$ . The covariance  $\text{cov}[y, y']$  is computed for a given set of hyperparameters by using [equation \(2.29\)](#) when computing the relevant off-diagonal elements of the covariance matrix given in [equation \(2.22\)](#). The marginalized covariance is computed from the MCMC samples using the law of total covariance:

$$\text{cov}[\mathbf{y}_*, \mathbf{y}'_* | \mathbf{y}] = \mathbb{E} [\text{cov}[\mathbf{y}_*, \mathbf{y}'_* | \mathbf{y}, \boldsymbol{\theta}]] + \text{cov} [\mathbb{E}[\mathbf{y}_* | \mathbf{y}, \boldsymbol{\theta}], \mathbb{E}[\mathbf{y}'_* | \mathbf{y}, \boldsymbol{\theta}]]. \quad (2.51)$$

While [equation \(2.49\)](#) is nonlinear with respect to  $y$  and hence the uncertainty propagation equation might not be expected to deliver reliable estimates, it was generally found to be fairly accurate over  $0 \leq r/a \lesssim 1$  when compared to a brute force Monte Carlo estimate of  $a/L_y$ .

### 2.3.8 Drawing samples for uncertainty propagation

One of the main goals of adopting an improved approach to fit plasma profiles is to be able to produce inputs for an uncertainty propagation technique such as Monte Carlo (or other more efficient techniques like Latin hypercube sampling [\[355\]](#), quasi Monte Carlo [\[291\]](#) or sparse quadrature [\[338\]](#)). Specifically, for many of the codes used to analyze plasma data, what is needed is not a random draw of a single scalar quantity but rather a random realization of the entire profile  $\mathbf{y}_*$  at the  $n_*$  points in  $\mathbf{X}_*$ . This is exceptionally straightforward with GPR, as the result ([equation \(2.17\)](#)) is simply a multivariate normal distribution for the values of the profile  $\mathbf{y}_*$  at the points  $\mathbf{X}_*$ . This section discusses well-established techniques to efficiently produce random samples from the multivariate normal distribution, or otherwise compute the expectation of a code output given a multivariate normal distribution on the inputs.

The standard recipe for producing a random draw  $\tilde{\mathbf{y}}_*$  from the  $n_*$ -dimensional multivariate normal distribution  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is to produce through standard means a vector  $\mathbf{u}$  of  $n_*$  independent, standard normal variables (i.e.,  $\mathbf{u} \sim \mathcal{N}(\mathbf{0}_{n_*}, \mathbf{I}_{n_*})$  where  $\mathbf{0}_{n_*}$  is the  $n_*$ -dimensional zero vector and  $\mathbf{I}_{n_*}$  is the  $n_* \times n_*$  identity matrix), then find

$$\tilde{\mathbf{y}}_* = \mathbf{A}\mathbf{u} + \boldsymbol{\mu}, \quad (2.52)$$

where  $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\top$  [\[296–298\]](#). A common, computationally efficient choice for how to decompose  $\boldsymbol{\Sigma}$  is the Cholesky decomposition  $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$ , where  $\mathbf{L}$  is lower triangular. But for the application of advanced uncertainty propagation methods such as quasi Monte Carlo [\[291\]](#) or sparse quadrature [\[338\]](#), large increases in the convergence rate can be gained by reducing the dimension of the parameter space that must be explored. When using the Cholesky decomposition the dimension of the space to

be sampled is equal to the number of points the curve is evaluated at,  $n_*$ . Instead, consider the eigendecomposition:

$$\mathbf{\Sigma} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{1/2}(\mathbf{Q}\mathbf{\Lambda}^{1/2})^T, \quad (2.53)$$

where in the last step the fact that  $\mathbf{\Sigma}$  is guaranteed to be symmetric and hence have an orthogonal matrix of eigenvectors was used. Hence, we can take  $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}^{1/2}$ . In practice, the eigenvalues drop off quite rapidly and can therefore be truncated to produce draws while sampling in a space with much lower dimension than the number of points the curve is to be evaluated at.

If MCMC is being used to marginalize over the hyperparameters, then sampling must take place hierarchically [286]: first, a sample  $\tilde{\boldsymbol{\theta}} \sim f_{\boldsymbol{\theta}|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y})$  is drawn from equation (2.39) using MCMC. Then, using equation (2.52), a sample  $\tilde{\mathbf{y}}_*$  is drawn from  $f_{\mathbf{Y}_*|\mathbf{Y},\boldsymbol{\theta}}(\mathbf{y}_*|\mathbf{y}, \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}})$ . Performing such sampling repeatedly then gives an ensemble of possible realizations  $\{\tilde{\mathbf{y}}_*^{(i)}\}$  to be used as inputs in the next step of the analysis workflow.

### 2.3.9 Extensions to handle non-local measurements and outputs

#### 2.3.9.1 Mathematical framework

The previous sections have only considered GPR when all measurements are *local*: the diagnostic collects the data at a single point in space. But, many diagnostics measure the value integrated along a line of sight instead; notable examples relevant to this work include the two-color interferometer (TCI) [356–359], x-ray imaging crystal spectrometer (XICS, specifically HIREX-SR on Alcator C-Mod) [360] and soft x-ray diode arrays (SXR, specifically the XTOMO systems on Alcator C-Mod) [361]. It is of great interest to be able to include such data in the same profile fit as local measurements so that all of the measurements contribute to reducing the uncertainty in the fit. Work similar to this was introduced in [345], but they assume a zero mean function and their expressions unfortunately contain several typos. Li et al. [313] uses Gaussian processes to invert soft x-ray data from w7-AS, including a fairly sophisticated nonstationary covariance kernel. Langenberg et al. [317] has applied GPR to invert synthetic data representing the spectroscopy system planned for w7-x. Boyle [343] gives exact expressions for integrals of Gaussian processes and comments on the issues inherent to this approach, but does not give an approximate formulation like the one used here.

A line-integrated quantity  $z$  (for instance, the line-integrated density obtained



from an interferometer) can be approximated by a weighted sum:

$$z \equiv \int y \, dl \approx \sum_{i=1}^{n_Q} w_i y_i(\mathbf{x}_i), \quad (2.54)$$

where the quadrature weights  $w_i$  and corresponding quadrature points  $\mathbf{x}_i$  are chosen with one of several rules, such as the trapezoid rule or Simpson's rule. Furthermore, some collection  $\mathbf{z} \in \mathbb{R}^n$  of  $n$  such quantities can be represented as the matrix product

$$\mathbf{z} = \mathbf{T}\mathbf{y}, \quad (2.55)$$

where  $\mathbf{y} \in \mathbb{R}^{n_Q}$  is the vector containing the local quantity at each of the quadrature points  $\mathbf{X} \in \mathbb{R}^{D \times n_Q}$  and  $\mathbf{T} \in \mathbb{R}^{n \times n_Q}$  is the transformation matrix. Note that local measurements can be included in  $\mathbf{z}$  simply by including a row in  $\mathbf{T}$  which only contains one non-zero entry. The derivation below will work for any linearly-transformed quantity (or quantity which can be approximated as such), line integrals are used here simply because they are most relevant to this work. This derivation will also include a non-zero mean function, so as to present the complete form of GPR supported by the `gptools` software [306, 307].

Given  $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , a fundamental property of the multivariate normal distribution is that

$$\mathbf{z} = \mathbf{T}\mathbf{y} \sim \mathcal{N}(\mathbf{T}\boldsymbol{\mu}, \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}^\top). \quad (2.56)$$

Start by assuming a noise-free Gaussian process joint prior distribution for  $\mathbf{y}$  and  $\mathbf{y}_*$  with mean function  $m(\mathbf{x})$  and covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$ :

$$f_{Y, Y_*}(\mathbf{y}, \mathbf{y}_*) = \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right). \quad (2.57)$$

Then we have the following joint prior distribution for transformed observations  $\mathbf{z} = \mathbf{T}\mathbf{y}$  and transformed predictions  $\mathbf{z}_* = \mathbf{T}_*\mathbf{y}_*$  (if local predictions are desired,  $\mathbf{T}_*$  is simply the identity matrix,  $\mathbf{I}_{n_*}$ ):

$$f_{\mathbf{Z}, \mathbf{Z}_*}(\mathbf{z}, \mathbf{z}_*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{T}m(\mathbf{X}) \\ \mathbf{T}_*m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_* \end{bmatrix} \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \begin{bmatrix} \mathbf{T}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_*^\top \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Sigma}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\right), \quad (2.58)$$

where  $\mathbf{0}$  is the zero matrix and  $\boldsymbol{\Sigma}_n \in \mathbb{R}^{n \times n}$  is now the Gaussian noise on the *transformed* observations  $\mathbf{z}$ . Using [equation \(1.53\)](#), the posterior distribution for  $\mathbf{z}_*$  conditioned on  $\mathbf{z}$  is

$$f_{\mathbf{z}_*|\mathbf{z}}(\mathbf{z}_*|\mathbf{z}) = \mathcal{N}\left(\mathbf{T}_* \mathbf{m}(\mathbf{X}_*) + \mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}) \mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} (\mathbf{z} - \mathbf{Tm}(\mathbf{X})), \right. \\ \left. \mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \mathbf{T}_*^\top - \mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}) \mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} \mathbf{TK}(\mathbf{X}, \mathbf{X}_*) \mathbf{T}_*^\top\right). \quad (2.59)$$

Note that here  $\mathbf{X} \in \mathbb{R}^{D \times n_Q}$  and  $\mathbf{X}_* \in \mathbb{R}^{D \times n_{Q^*}}$  are the locations of the quadrature points for the observed and predicted quantities, respectively. Again, the mean of this distribution is the prediction for the smoothed and transformed quantities  $\mathbf{z}_*$  and the covariance matrix contains the uncertainty in this prediction. If all that is being predicted is the profile of the local quantity  $\mathbf{y}_*$ , then  $\mathbf{T}_* = \mathbf{I}_{n_*}$  and this simplifies to

$$f_{\mathbf{y}_*|\mathbf{z}}(\mathbf{y}_*|\mathbf{z}) = \mathcal{N}\left(\mathbf{m}(\mathbf{X}_*) + \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}) \mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} (\mathbf{z} - \mathbf{Tm}(\mathbf{X})), \right. \\ \left. \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}) \mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} \mathbf{TK}(\mathbf{X}, \mathbf{X}_*)\right). \quad (2.60)$$

Furthermore, if  $\mathbf{T} = \mathbf{I}_n$  and  $m(\mathbf{x}) = 0$ , this reduces to [equation \(2.17\)](#), as expected.

The final result necessary to use [equation \(2.59\)](#) is the log-likelihood, which is simply

$$\ln f_{\mathbf{z}|\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{\theta}) = -\frac{1}{2} (\mathbf{z} - \mathbf{Tm}(\mathbf{X}|\boldsymbol{\theta}))^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) \mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} (\mathbf{z} - \mathbf{Tm}(\mathbf{X}|\boldsymbol{\theta})) \\ - \frac{1}{2} \ln |\mathbf{TK}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) \mathbf{T}^\top + \boldsymbol{\Sigma}_n| - \frac{n}{2} \ln 2\pi. \quad (2.61)$$

The three terms still have the same interpretation as was given for the simpler form given back in [section 2.3.6](#). If  $m(\mathbf{x}) = 0$  and  $\mathbf{T} = \mathbf{I}_n$ , this reduces to [equation \(2.38\)](#).

The derivative of this expression with respect to the hyperparameters is

$$\begin{aligned}
\frac{\partial}{\partial \theta_i} f_{Z|\theta}(z|\theta) &= \frac{1}{2} \left( \mathbf{T} \frac{\partial}{\partial \theta_i} \mathbf{m}(\mathbf{X}|\theta) \right)^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} (z - \mathbf{Tm}(\mathbf{X}|\theta)) \\
&\quad - \frac{1}{2} (z - \mathbf{Tm}(\mathbf{X}|\theta))^\top \frac{\partial}{\partial \theta_i} (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} (z - \mathbf{Tm}(\mathbf{X}|\theta)) \\
&\quad + \frac{1}{2} (z - \mathbf{Tm}(\mathbf{X}|\theta))^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} \left( \mathbf{T} \frac{\partial}{\partial \theta_i} \mathbf{m}(\mathbf{X}|\theta) \right) \\
&\quad - \frac{1}{2} \frac{\partial}{\partial \theta_i} \ln |\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n| \tag{2.62} \\
&= \frac{1}{2} \left( \mathbf{T} \frac{\partial}{\partial \theta_i} \mathbf{m}(\mathbf{X}|\theta) \right)^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} (z - \mathbf{Tm}(\mathbf{X}|\theta)) \\
&\quad + \frac{1}{2} (z - \mathbf{Tm}(\mathbf{X}|\theta))^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} \\
&\quad \quad \cdot \frac{\partial}{\partial \theta_i} (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n) (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} (z - \mathbf{Tm}(\mathbf{X}|\theta)) \\
&\quad + \frac{1}{2} (z - \mathbf{Tm}(\mathbf{X}|\theta))^\top (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} \left( \mathbf{T} \frac{\partial}{\partial \theta_i} \mathbf{m}(\mathbf{X}|\theta) \right) \\
&\quad - \frac{1}{2} \operatorname{tr} \left( (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n)^{-1} \frac{\partial}{\partial \theta_i} (\mathbf{TK}(\mathbf{X}, \mathbf{X}|\theta)\mathbf{T}^\top + \boldsymbol{\Sigma}_n) \right). \tag{2.63}
\end{aligned}$$

Note that the result only requires the inversion of an  $n \times n$  matrix, where  $n$  is the number of transformed observations: the term which previously dominated the computational complexity is still simply  $\mathcal{O}(n^3)$ , and is independent of the number of quadrature points used. But forming the covariance matrices  $\mathbf{K}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n_Q \times n_Q}$ ,  $\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \in \mathbb{R}^{n_{Q_*} \times n_{Q_*}}$  and  $\mathbf{K}(\mathbf{X}_*, \mathbf{X}) \in \mathbb{R}^{n_{Q_*} \times n_Q}$  requires the covariance kernel to be evaluated between each pair of input and output quadrature points. Furthermore, the product  $\mathbf{TK}(\mathbf{X}, \mathbf{X})\mathbf{T}^\top$  has complexity  $\mathcal{O}(nn_Q^2 + n^2n_Q)$ , the product  $\mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \mathbf{T}_*^\top$  has complexity  $\mathcal{O}(n_*n_{Q_*}^2 + n_*^2n_{Q_*})$  and the product  $\mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top$  has complexity  $\mathcal{O}(n_*n_{Q_*}n_Q + n_*n_Qn)$ . Since the  $n_Q^2$  and  $n_{Q_*}^2$  terms can become large and the covariance kernel can sometimes be computationally expensive to evaluate, this means it is beneficial to keep the number of quadrature points required for both the inputs and the outputs small. This can be accomplished both by using a coarse grid as well as by using the same quadrature points (but with different weights) for each chord.

### 2.3.9.2 Handling TCI data

As part of this work, a tool was written to allow the TCI data to be combined with the Thomson scattering data to help constrain the density profile. This code uses the trapezoid rule with equally-spaced points along each chord to select the quadrature points and compute the quadrature weights. Because of the inefficient selection of quadrature points, the code takes a prohibitively long time to run.<sup>1,2</sup> This implementation is included in the `profiletools` software described in [appendix F](#) and [308, 309]. The software also has the ability to automatically set the weight for any quadrature point which lies in the shadow of the limiter to zero, which helps increase the stability of the fit by preventing it from adopting unphysically high values at the edge of the fitting domain in order to get better agreement with the TCI data. The results of applying this tool to Alcator C-Mod data are given in [section 2.5.2](#).

### 2.3.9.3 Computing volume averages

This framework can also be used to compute the volume averages of profile quantities, given here for the electron density  $n_e$ :

$$\langle n_e \rangle_{\text{vol}} = \frac{1}{V_a} \int n_e dV, \quad (2.64)$$

where  $V_a$  is the total volume enclosed by the LCFS. Noting that the volume  $V_r$  enclosed by a flux surface is a flux surface label and defining the normalized flux surface volume as  $V_n = V_r/V_a$ , this can then be written as:

$$\langle n_e \rangle_{\text{vol}} = \int n_e(V_n) dV_n. \quad (2.65)$$

If  $n_e$  is expressed as a function of a different flux surface label, this can be transformed according to

$$\langle n_e \rangle_{\text{vol}} = \int n_e(x) \frac{dV_n}{dx} dx, \quad (2.66)$$

1. The analysis combining Thomson scattering and TCI data presented in [section 2.5.2](#) takes about ten wall-clock minutes on a workstation with two 2.66 GHz Intel® Xeon® x5650 processors which supports a total of 24 threads to find the MAP estimate and about 45 wall-clock minutes to run the MCMC sampler for a single profile. The MAP estimate is run in parallel with one starting point per thread, which means that adding additional threads will not help performance. The MCMC sampler will benefit from additional threads.
2. There was an attempt to implement a more efficient version using the TRIPPY code [362] as part of an undergraduate research project. The use of TRIPPY allows the code to compute the quadrature weights for each chord using the same quadrature points, thereby reducing the run time by a factor equal to the number of chords. Unfortunately the code is not robust to changes in the TCI chord locations, and so only works reliably on recent shots.

where  $x$  is some arbitrary flux surface label. This integral can then be discretized as

$$\langle n_e \rangle_{\text{vol}} \approx \sum_{i=1}^{n_Q} n_e(x_i) \left. \frac{dV_n}{dx} \right|_{x=x_i} w_i, \quad (2.67)$$

where the weights  $w_i$  could either be  $\Delta x_i$  or weights generated by some more complicated rule such as the trapezoid rule or Simpson's rule. With  $\langle n_e \rangle_{\text{vol}}$  cast in the same form as equation (2.54), the volume integral can be included in the prediction simply by placing the weights  $dV_n/dx|_{x=x_i} w_i$  in a row of the output transformation matrix  $\mathbf{T}_*$  when evaluating equation (2.59).

Note that the predictive covariance matrix of equation (2.59) will include both the variance (i.e., uncertainty) of  $\langle n_e \rangle_{\text{vol}}$  as well as its covariance with the other predictions. Therefore, this formulation permits one to predict the peaking factor

$$f_p = \frac{n_e(\psi_n = 0.2)}{\langle n_e \rangle_{\text{vol}}} \quad (2.68)$$

relevant to the work on electron density peaking in H-mode plasmas in [68, 363–368], complete with an uncertainty estimate which accounts for the potential covariance between  $n_e(\psi_n = 0.2)$  and  $\langle n_e \rangle_{\text{vol}}$ . (Here,  $\psi_n = (\psi - \psi_o)/(\psi_a - \psi_o)$  is the normalized poloidal flux, where  $\psi$  is the poloidal flux,  $\psi_o$  is the poloidal flux at the magnetic axis and  $\psi_a$  is the poloidal flux at the LCFS.) Using the uncertainty propagation equation [282], the variance in  $f_p$  is approximately

$$\begin{aligned} \text{var}[f_p] \approx & \frac{\text{var}[n_e(\psi_n = 0.2)]}{\langle n_e \rangle_{\text{vol}}^2} + \frac{\text{var}[\langle n_e \rangle_{\text{vol}}] (n_e(\psi_n = 0.2))^2}{\langle n_e \rangle_{\text{vol}}^4} \\ & - 2 \frac{\text{cov}[n_e(\psi_n = 0.2), \langle n_e \rangle_{\text{vol}}] n_e(\psi_n = 0.2)}{\langle n_e \rangle_{\text{vol}}^3}. \end{aligned} \quad (2.69)$$

Note that this expression is based on a linearization, and hence would be expected to break down if  $\langle n_e \rangle_{\text{vol}} \ll n_e(\psi_n = 0.2)$ . While in general one would in fact expect  $\langle n_e \rangle_{\text{vol}} < n_e(\psi_n = 0.2)$ , these two quantities will still be of the same order of magnitude in any realistic case.

The ability to compute volume averages when fitting the profile also permits the calculation of the global effective collisionality

$$\nu_{\text{eff}} = 0.1 \frac{Z_{\text{eff}} \langle n_{e, [10^{19} \text{ m}^{-3}]} \rangle_{\text{vol}} R_{o, [\text{m}]}}{\langle T_{e, [\text{keV}]} \rangle_{\text{vol}}^2}, \quad (2.70)$$

but no attempt has been made to quantify the covariance between  $\langle n_e \rangle_{\text{vol}}$  and  $\langle T_e \rangle_{\text{vol}}$ . (Here, the subscripts in brackets indicate the units a quantity must be given in,  $R_o$  is

the major radius of the magnetic axis and  $Z_{\text{eff}} = \sum_j Z_j^2 n_j / \sum_j Z_j n_j$  is the effective ion charge [1] where  $Z_j$  is the charge number of ion species  $j$  and  $n_j$  is the ion density of species  $j$ .)

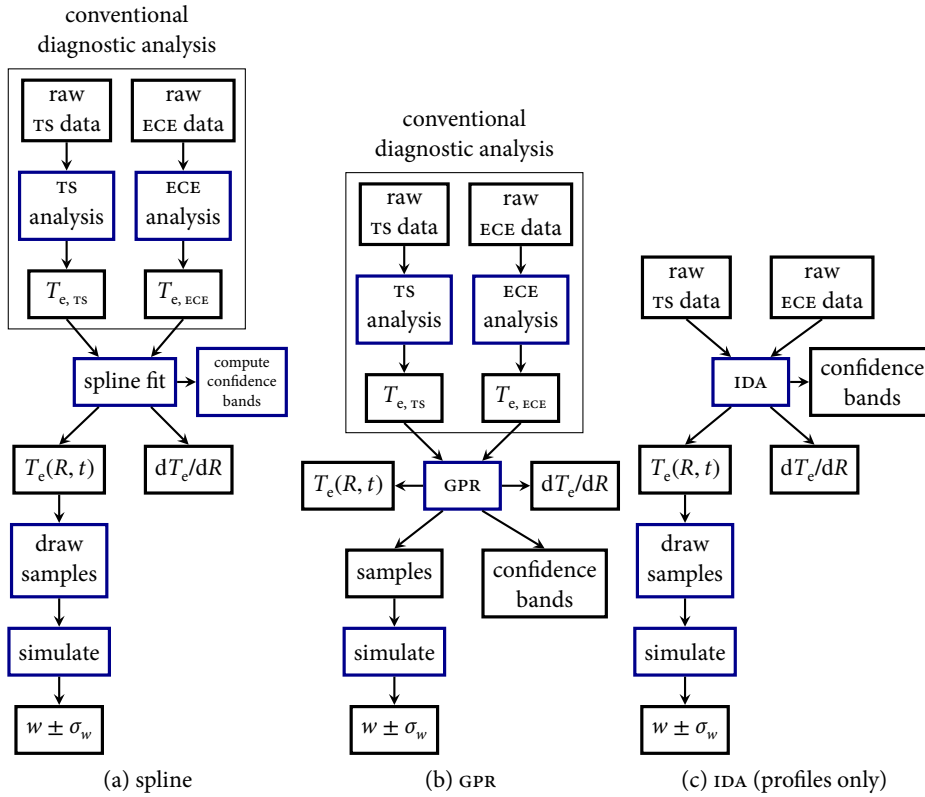
### 2.3.10 Gaussian process regression versus Bayesian integrated data analysis

It is worth comparing the present results to the work that has been done on Bayesian integrated data analysis (IDA) to combine multiple data sources and even interpretive models into a single self-consistent analysis [285, 312, 315, 369–394]. This type of integrated analysis approach has in fact been done in a way that incorporates Gaussian processes on MAST [316]. While both techniques share the fact that they are built within a Bayesian statistical framework, they differ substantially in their details and how they fit into an analysis workflow. Essentially, IDA constructs a joint probability distribution that can encompass many diverse diagnostics and other information. Such an analysis can even start from the level of more or less raw data and then infers a joint posterior distribution for the desired quantities and profiles given these observations. Note that the prior distribution for the profiles to be inferred through IDA could be a Gaussian process, in which case IDA inherits all of the advantages of GPR discussed in this thesis, plus the statistical advantages of ensuring consistency between all of the diagnostics. The role of GPR in the present work, however, is to simply replace the profile fitting, data fusion and sample generation steps of a traditional analysis workflow while still using the existing procedures for turning the raw data into discrete measurements. In this way, GPR can be more readily deployed in cases where trusted data analysis codes are already in place, though it does not have some of the very powerful capabilities that the more complicated IDA approach offers. Simplified workflows using splines, GPR and one form of IDA are shown in figure 2.8.

## 2.4 Verification of Gaussian process regression for tokamak profile fitting using synthetic data

### 2.4.1 Basic case

In order to verify the suitability of the Gibbs covariance kernel given in equation (B.73) with the tanh covariance length scale function given in equation (B.78) for fitting tokamak profile data, a synthetic profile was constructed using a cubic spline. To represent the Thomson scattering system on Alcator C-Mod, the measurements came from



**Figure 2.8:** Examples of simplified workflows for obtaining some quantity  $w$  and its uncertainty from an input  $T_e$  profile using the traditional spline-based approach, GPR and IDA. With splines the computation of the fit, determining the uncertainty on the fit and the drawing of perturbed samples are all typically separate operations. Doing the fit with GPR replaces these three operations, but otherwise leaves the workflow intact. Applying IDA to just the analysis of the raw profile data replaces the diagnostic analysis steps, but leaves the process of computing the output quantity  $w$  itself untouched. It is also possible to perform a fully integrated analysis to get from raw data to the desired output quantities (such as has been done to estimate  $Z_{\text{eff}}$  [390–392] and has been used with interpretive transport codes [377]), in which case even these steps are absorbed into the IDA step. These figures are updated versions of ones which originally appeared in [305].

two systems with equally-spaced channels: a core system covering  $0 < r/a < 0.97$  and a noisier edge system covering  $0.97 < r/a < 1.07$ . An equal number of uniformly-spaced points was used for each system. Zero-mean Gaussian noise with a relative uncertainty of  $\sigma/y = 0.03$  was added to the core data points and additive Gaussian noise with an absolute uncertainty of  $\sigma = 0.1$  was added to the edge data points. The fits were performed using the same prior distributions and similar core/edge constraints as were used on the experimental data in [section 2.5.1](#). [Figure 2.9](#) shows the results in the core and edge regions for 15, 30 and 60 points for each system. For reference, the Thomson scattering system on Alcator C-Mod has 15 core channels and 18 edge channels.

In order to assess the consistency of the fit as more points are added and the reasonableness of the error estimates from the fit, this procedure was repeated five times for each of  $n \in \{3, 7, 15, 30, 60, 120\}$  points in each system. The root-mean-square (RMS) error between the posterior mean of the fit and the true curve

$$e_{\text{RMS}} = \sqrt{\mathbb{E}[(y - y_{\text{true}})^2]} \quad (2.71)$$

was computed in each case along with the RMS uncertainty estimate

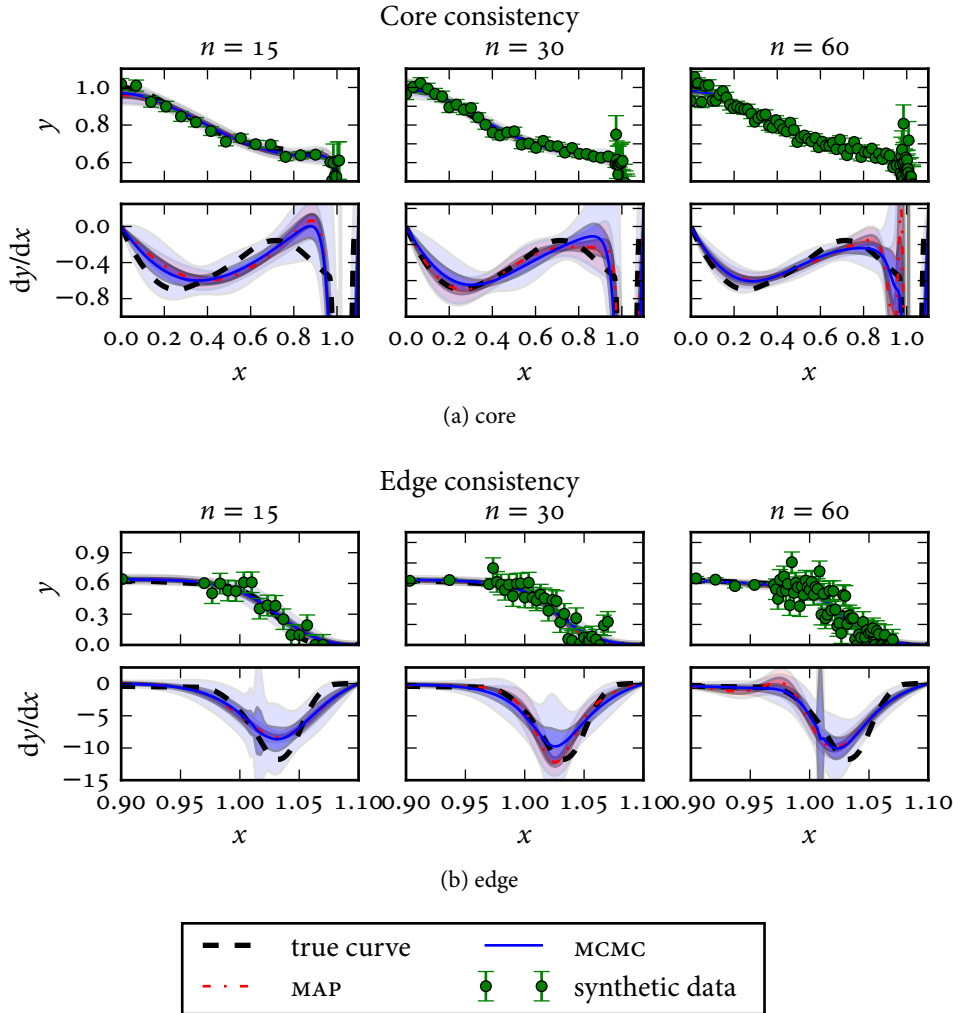
$$\sigma_{\text{RMS}} = \sqrt{\mathbb{E}[\sigma^2]} \quad (2.72)$$

from the fit. The results are shown in [figure 2.10](#). As expected, as more points are added the RMS discrepancy between the fit and the true curve goes down. Furthermore, the uncertainty estimate from the fit tracks this trend. Reducing the noise was observed to accelerate the convergence, as would be expected.

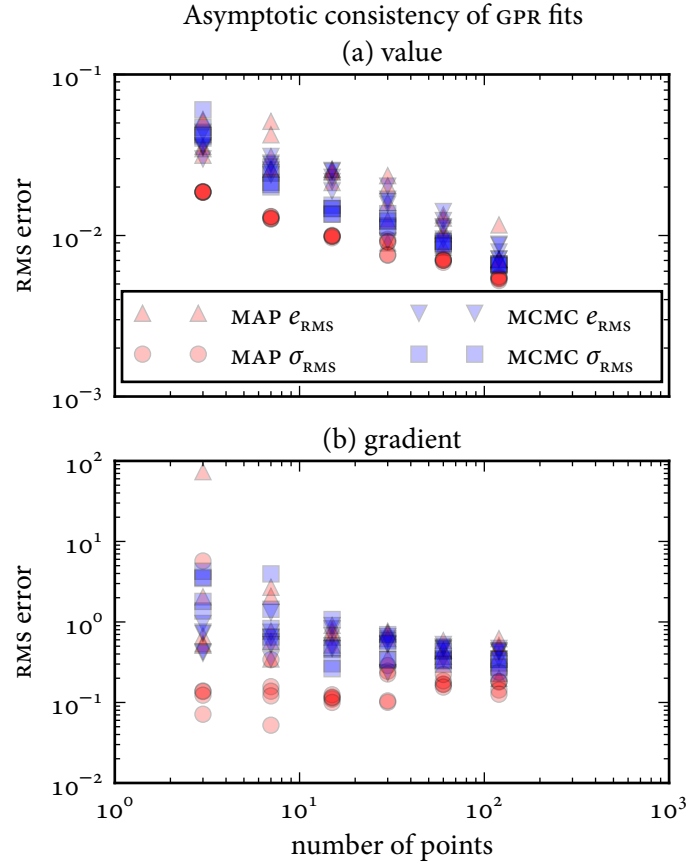
#### 2.4.2 Effect of errors in the independent variable

The analysis presented here ignores the errors in the independent variable ( $r/a$ , etc.), despite the fact that there is both observed variation in the magnetic equilibrium over the time window observed and unknown systematic errors in the magnetic equilibrium reconstruction. In order to assess the impact of this neglect of errors in the independent variable another synthetic data set was constructed with data points at the locations where the Thomson scattering systems measure and the  $x$  coordinate perturbed with Gaussian noise whose covariance matrix matched that of the observed variation of the coordinate mapping over the time window analyzed. The observed covariance matrix is shown in [figure 2.11](#). As can be seen, the variances are very small: the magnetic equilibrium was quite steady over the period considered. As such, it is likely that systematic errors dominated the actual coordinate mapping, but no attempt has been made to estimate these errors. In order to illustrate the

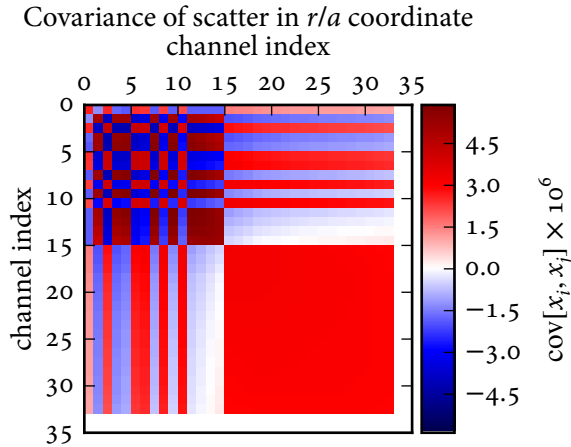




**Figure 2.9:** Results of fitting synthetic data for  $n \in \{15, 30, 60\}$  channels in each system (core, edge). Examination of many realizations of the measurement noise shows that even with the lowest number of points (comparable to the Thomson scattering system on Alcator C-Mod) the fit does a reasonable job of capturing the true curve within its error bars. Obviously the quality of the fit improves as more points are added. The true curve is shown as the black dashed line, the synthetic data are the green points with  $\pm 1\sigma$  error bars, the MAP estimate is shown as the red dash-dot curve and the MCMC estimate is shown as the blue solid curve. The fits are shown with  $\pm 1\sigma$  and  $\pm 3\sigma$  uncertainty envelopes. These figures are updated versions of ones which originally appeared in [305]. These figures were produced using `synthetic_test_consistency.py`.



**Figure 2.10:** Demonstration of asymptotic consistency for the (a) value and (b) gradient of the fit as the number of points for each system covers  $n \in \{3, 7, 15, 30, 60, 120\}$ . Five realizations of the synthetic measurement noise were used at each value of  $n$ . The points are plotted with 25% transparency so that the density of the scatterplot can be seen. The values connected to the MAP estimate are shown in red and the values connected with the MCMC estimate are shown in blue. The RMS discrepancy between the fit and the true curve goes down as more points are added and the RMS uncertainty from the fit matches this trend. As expected, the MAP estimate under-predicts the uncertainties in both the value and the gradient. These figures are updated versions of ones which originally appeared in [305]. These figures were produced using `synthetic_test_consistency.py`.



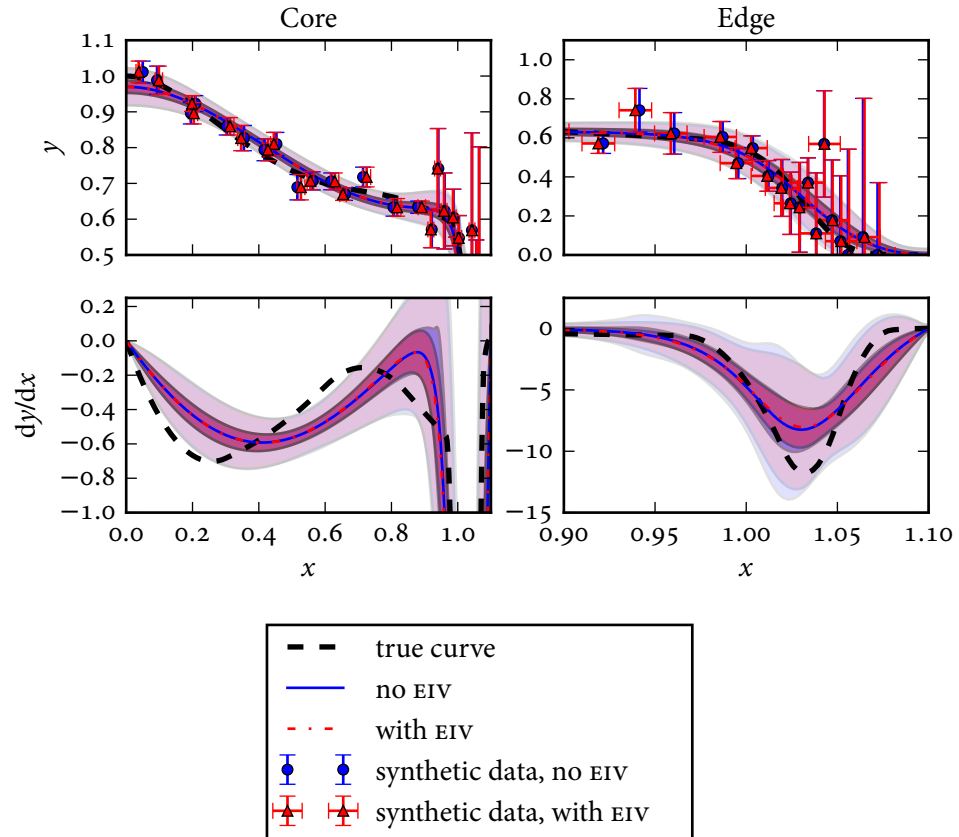
**Figure 2.11:** Observed covariance matrix for the  $r/a$  coordinates of the Thomson scattering diagnostic. The data from the range  $0.965 \text{ s} \leq t \leq 1.365 \text{ s}$  of shot 1101014006 were used. Elements 0 through 14 are the core system and elements 15 through 32 are the edge system. The correlation is so high and uniform in the portion of the matrix corresponding to the edge system because of how close the measurement locations are to each other. The regions of positive versus negative correlation arise because the core Thomson scattering system measures both above and below the magnetic axis. Note the exceedingly small values indicated on the color bar: the typical standard deviation of any given point is only about  $2 \times 10^{-3}$ . This figure was produced using `synthetic_test_EIV.py`.

potential effect (or lack thereof) of the errors in  $r/a$ , five times the observed variation was used. The actual correlations were preserved, however, since the magnetic equilibrium must vary in a self-consistent manner. The results are shown in [figure 2.12](#). Even with the inflated noise level, the results are very similar both with and without the added noise. Therefore, it was concluded that the effect of random noise in the coordinate mapping of the Thomson scattering diagnostic is negligible.

## 2.4.3 Comparison of various covariance kernels

### 2.4.3.1 Stationary covariance kernels applied to core data

To test the effect of the various stationary covariance kernels discussed in [appendix B](#), the core portion of the 15 points/region case of the synthetic data discussed in [section 2.4.1](#) was fit using the squared exponential (SE, [equation \(2.5\)](#), [section B.1](#)), rational quadratic (RQ, [section B.2](#), [equation \(B.22\)](#)) and Matérn ([section B.3](#), [equation \(B.23\)](#)).



**Figure 2.12:** Results of fitting synthetic data both with (red dash-dot, triangles) and without (blue solid, circles) the effect of errors in the independent variable. The notation “EIV” in the legend refers to “errors-in-variables.” The (co)variance in  $r/a$  was scaled up by a factor of five from what was actually observed in the experiment, and the horizontal error bars shown on the red triangles correspond to this scaled standard deviation. Even at this level, the effect of this error on the fit is minimal. The true curve is shown as the dashed black curve. The colors of the data points correspond to the curve which was fit to them. This figure was produced using [synthetic\\_test\\_EIV.py](#).

**Table 2.1:** Stationary covariance kernels tested with synthetic data and the prior distributions for their hyperparameters. The data are taken to be dimensionless, so all hyperparameters are dimensionless.

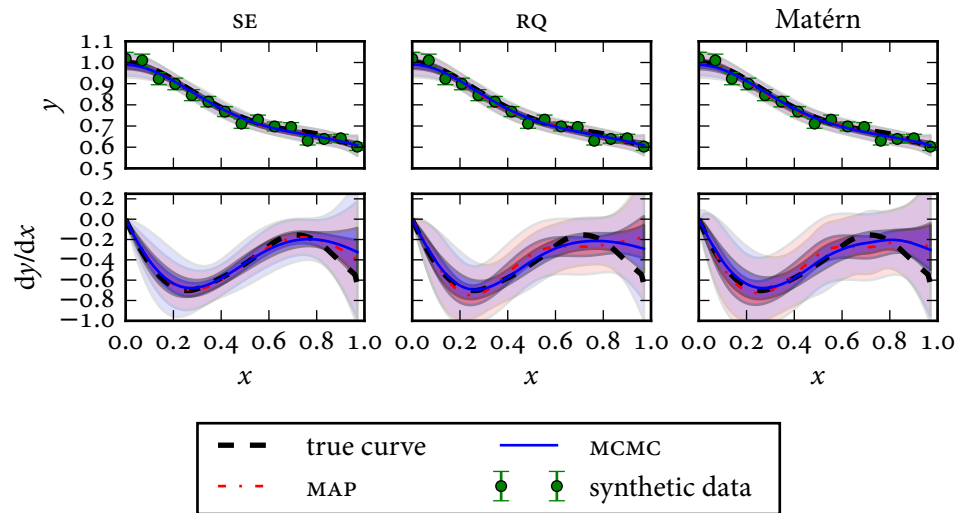
Case	Parameter	Prior distribution
SE	$\sigma_f$	$\mathcal{U}(0, 10)$
	$\ell$	$\mathcal{U}(0, 5)$
RQ	$\sigma_f$	$\mathcal{U}(0, 10)$
	$\alpha$	$\mathcal{U}(0.01, 10)$
	$\ell$	$\mathcal{U}(0, 5)$
Matérn	$\sigma_f$	$\mathcal{U}(0, 10)$
	$\nu$	$\mathcal{U}(1.01, 50)$
	$\ell$	$\text{Gamma}(13, 12)$ ( $m = 1, \sigma = 0.3$ )

**Table 2.2:** Parameters used for MCMC sampling of the hyperparameters of the various stationary covariance kernels applied to synthetic data. “Walkers” is the number of Markov chains run in parallel, “Burn” is the number of samples discarded from the start of each chain, and “Thin” is the factor by which each chain’s remaining samples were thinned before computing profiles.

Case	Proposal width, $a$	Walkers	Samples per walker	Burn	Thin
SE	2	200	300	200	100
RQ	6	200	400	300	100
Matérn	2	200	400	300	100

tion (B.32)) covariance kernels. The RQ covariance kernel represents a mixture of SE covariance kernels with different length scales, the distribution of which is determined by an additional hyperparameter  $\alpha$ . As  $\alpha \rightarrow \infty$  the RQ covariance kernel reduces to the SE covariance kernel. The Matérn covariance kernel represents a Gaussian process which is only differentiable up to order  $n < \nu$ , where  $\nu$  is an additional hyperparameter. As  $\nu \rightarrow \infty$  the Matérn covariance kernel reduces to the SE covariance kernel. Refer to the relevant sections in [appendix B](#) for more details on these covariance kernels.

The hyperparameters  $\alpha$  for the RQ and  $\nu$  for the Matérn covariance kernels were treated as free hyperparameters with uninformative prior distributions. The prior distributions used are given in [table 2.1](#) and the parameters used for the affine-invariant ensemble sampler are given in [table 2.2](#). The gamma prior for  $\ell$  of the Matérn covari-

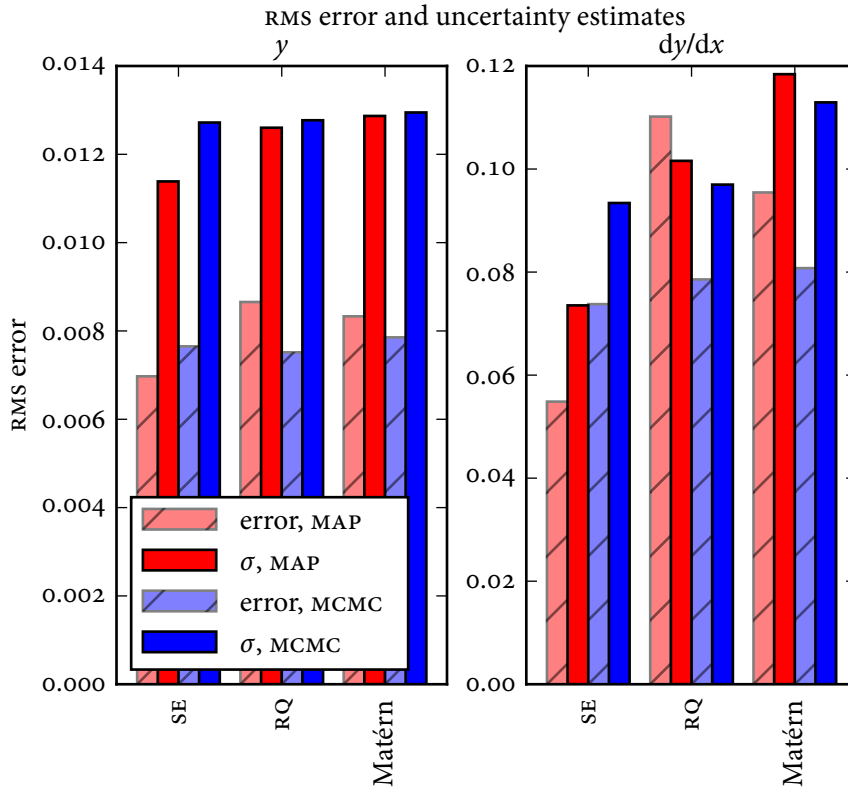


**Figure 2.13:** Synthetic core data and fitted profiles for the squared exponential, rational quadratic and Matérn covariance kernels. The true curve is shown as the black dashed line. All fits were evaluated with both the MAP (red dash-dot line) and MCMC (blue solid line) techniques. This figure was produced using `synthetic_test_stationary.py`.

ance kernel was found to be necessary to keep the optimizer from getting stuck in a region with  $\ell = 0$  and failing to find a valid solution. The uniform prior on  $\nu$  was used to enforce  $\nu > 1$  so that the fit would be differentiable and the uniform prior on  $\alpha$  was used to enforce  $\alpha > 0$ . The fitted profiles are given in figure 2.13, the posterior distributions of the hyperparameters are shown in figure 2.15, summary statistics for the posterior distributions of the hyperparameters are given in table 2.3 and a plot showing the RMS error and uncertainty estimates is given in figure 2.14. All three covariance kernels did a reasonable job of fitting the core profile. The RMS errors and standard deviations in figure 2.14 show that the SE covariance kernel produced both the smallest RMS error and the most reliable uncertainty estimate for both the MAP and MCMC estimates.

As shown in figure 2.13, all three covariance kernels obtained similar posterior distributions for  $\sigma_f$ . The SE covariance kernel favored a shorter covariance length scale than either the RQ or Matérn covariance kernels, but the posterior distributions still encompass the same order of magnitude.

The results for  $\alpha$  in the RQ covariance kernel and  $\nu$  in the Matérn covariance kernel are interesting. In both cases, the relevant hyperparameter has a posterior



**Figure 2.14:** Root-mean-square errors and uncertainty estimates for the fits to synthetic core data using various stationary covariance kernels. The “RMS error” (lighter, cross-hatched bars) is as given in equation (2.71) and the “RMS  $\sigma$ ” (darker bars) is as given in equation (2.72). Loosely speaking, the closer the “ $\sigma$ ” bar is to the corresponding “error” bar, the more accurate the fit’s uncertainty estimate is. The smaller the “error” bar is, the more accurate the fit itself is. For both the MAP (red) and MCMC (blue) estimates, the SE covariance kernel had a smaller RMS error than the RQ and Matérn covariance kernels. This figure was produced using `synthetic_test_stationary.py`.

mode at some low value ( $\alpha = 0.037$  and  $\nu = 2.2$ , respectively), and a distribution which is more or less uniform over the (broad) prior bounds. This indicates that the fits with different values of  $\alpha$  or  $\nu$  are equally likely above some threshold. This calls the interpretation of the MCMC estimate into question: the wider the posterior distribution on  $\alpha$  or  $\nu$  is, the less of an effect the peak at low values has on the fitted curve. Therefore, if there is in fact complete prior ignorance of  $\alpha$  or  $\nu$ , then the

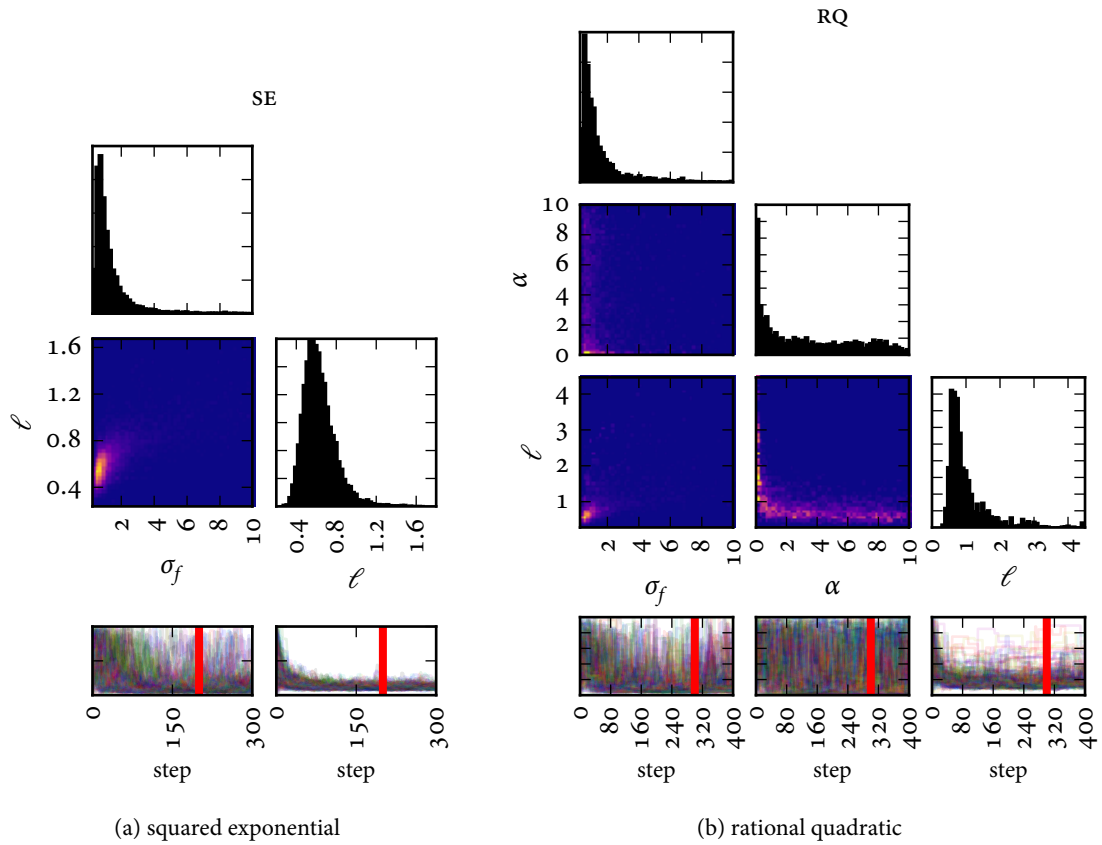
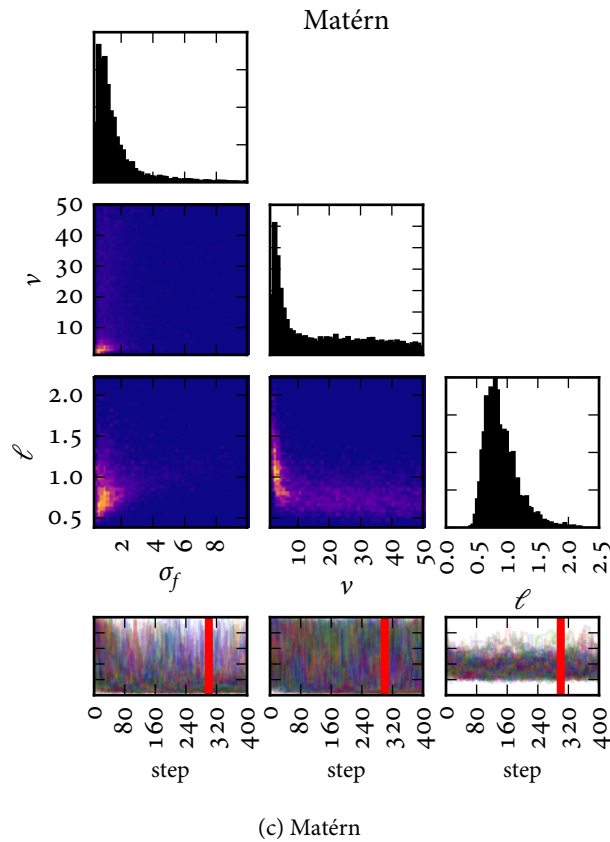


Figure 2.15: (Continues on the facing page.)





**Figure 2.15:** (Continued from the facing page.) Matrices of univariate (on the diagonal) and bivariate marginal posterior distributions and MCMC sampler chains (at the bottom) for the hyperparameters of the (a) squared exponential, (b) rational quadratic and (c) Matérn covariance kernels fit to synthetic core data. These plots are essentially 1- and 2D projections of posterior distributions  $f_{\theta|Y}(\theta|y)$  as given in equation (2.39). The vertical red line indicates the burn-in period; only the samples to the right of the line were used. To keep the figure legible the vertical scales for the sampler chains are not shown: what matters is that each chain has settled into a stationary behavior. These figures were produced using `synthetic_test_stationary.py`.

**Table 2.3:** Summary statistics for the posterior distributions for the hyperparameters of the various stationary covariance kernels applied to synthetic core data. The mode is the MAP estimate. The mean and 95% interval for  $\alpha$  and  $\nu$  are not very useful estimators, given that the distribution for each of those appears to consist of a spike at some low value and a roughly uniform distribution over the rest of the bounds.

Case	Parameter	Mode	Mean	95% interval
SE	$\sigma_f$	0.581	1.40	[0.399 , 5.68]
	$\ell$	0.533	0.633	[0.391 , 1.03]
RQ	$\sigma_f$	0.759	1.56	[0.384 , 6.53]
	$\alpha$	0.0366	3.38	[0.0224, 9.30]
	$\ell$	1.74	1.02	[0.453 , 2.95]
Matérn	$\sigma_f$	0.562	1.62	[0.413 , 6.17]
	$\nu$	2.20	17.2	[1.78 , 47.4 ]
	$\ell$	1.10	0.902	[0.538 , 1.55]

marginalized fit should consist almost entirely of samples with arbitrarily large  $\alpha$  or  $\nu$ , and hence is equivalent to using an SE covariance kernel. Note that the plots of the posterior distributions for the hyperparameters in [figure 2.15](#) support this conclusion: the bivariate marginal distributions between  $\alpha$  and  $\ell$  of the RQ covariance kernel and between  $\nu$  and  $\ell$  of the Matérn covariance kernel both have “L” shapes which favor longer covariance length scales at smaller values of  $\alpha$  or  $\nu$ , but then for higher  $\alpha$  or  $\nu$  favor shorter values covering a range similar to the range of values that  $\ell$  for the SE covariance kernel takes. Therefore, if the RQ or Matérn covariance kernels are to be used, either a fixed value of  $\alpha$  or  $\nu$  should be chosen, or a stronger prior distribution for  $\alpha$  or  $\nu$  must be used. Because of the observation that the MCMC estimate converges to the SE covariance kernel for both the RQ covariance kernel with an uninformative prior distribution for  $\alpha$  and the Matérn covariance kernel with an uninformative prior distribution for  $\nu$  and the observation that the SE covariance kernel did a better job of fitting the synthetic data, the SE covariance kernel was used whenever a stationary covariance kernel was required in the remainder of this thesis.

#### 2.4.3.2 Nonstationary covariance kernels applied to entire profiles

In order to observe the effects of the various nonstationary covariance kernels discussed in [section B.4](#), a synthetic data set equivalent to the 15 point/region case in [section 2.4.1](#) was fit with a variety of nonstationary covariance kernels, including

both the Gibbs covariance kernel (section B.4.1, equation (B.73)) with a variety of covariance length scale functions and the SE covariance kernel (equation (2.5), section B.1) with a variety of input warping functions (section B.4.2). The nonstationary covariance kernels tested and the prior distributions over the hyperparameters used are given in table 2.4. Each case required slightly different settings for the MCMC sampler; the parameters are given in table 2.5. The covariance kernels tested are as follows:

**Gibbs covariance kernel + tanh covariance length scale function** (See section B.4.1, equation (B.73), section B.4.1.1 and equation (B.78).) This is the nonstationary covariance kernel used for most of this thesis. The covariance length scale function consists of a long- $\ell$  region in the core joined to a short- $\ell$  region in the edge using a tanh function. The prior distribution for the hyperparameters is very similar to what is used elsewhere in this thesis. Note that the tanh function is used only to set the way the covariance length scale varies in space, not to set an explicit functional form. See section 2.5.5 for a fit which explicitly incorporates the popular modified hyperbolic tangent (“mtanh”) function.<sup>3</sup>

**Gibbs covariance kernel + exponential of one Gaussian** (See section B.4.1, equation (B.73), section B.4.1.2.1 and equation (B.80).) This is similar to the approach used in [395]. The covariance length scale is equal to  $\ell_o$  far from the data and is modified by the exponential of a single Gaussian basis function near the data. The distant covariance length scale  $\ell_o$  has a prior distribution similar to what is used for the core covariance length scale of other covariance kernels. The center  $\mu_1$  and width  $\sigma_1$  of the one Gaussian is allowed to vary throughout the region where there are data. The Gaussian prior distribution for the amplitude of the Gaussian  $\beta_1$  is again inspired by [395]. The width was selected to keep the value of  $\ell(x)$  from varying too far from  $\ell_o$ .

**Gibbs covariance kernel + exponential of four Gaussians** (See section B.4.1, equation (B.73), section B.4.1.2.1 and equation (B.80).) This case is the most similar to the approach in [395]. The covariance length scale is again equal to  $\ell_o$  far from the data but here is modified by the exponential of four Gaussians with fixed, uniformly spaced centers between  $\mu_1 = 0$  and  $\mu_4 = 1.1$ . The widths were all fixed at  $1.1/3 = 0.367$ .

**SE covariance kernel + beta-CDF input warping function** (See equation (2.5), section B.1, section B.4.2, section B.4.2.2 and equation (B.97).) This case is equiv-

3. The mtanh-based fit was not tested here as it works best on data which are expressed as a function of  $\psi_n$  and hence does not need a zero slope constraint at the origin.

**Table 2.4:** Nonstationary covariance kernels tested with synthetic data and the prior distributions for their hyperparameters. The data are taken to be dimensionless, so all hyperparameters are dimensionless.

Parameter	Prior distribution
<i>Gibbs covariance kernel + tanh covariance length scale function</i>	
$\sigma_f$	$\mathcal{U}(0, 10)$
$\ell_1$	Gamma(13, 12) ( $m = 1, \sigma = 0.3$ )
$\ell_2$	Gamma(5.8, 9.7) ( $m = 0.5, \sigma = 0.25$ )
$\ell_w$	Exp(10) ( $m = 0, \sigma = 0.1$ )
$x_o$	Gamma(402, 401) ( $m = 1, \sigma = 0.05$ )
<i>Gibbs covariance kernel + exponential of one Gaussian</i>	
$\sigma_f$	$\mathcal{U}(0, 4)$
$\ell_o$	Gamma(13, 12) ( $m = 1, \sigma = 0.3$ )
$\mu_1$	$\mathcal{U}(0, 1.1)$
$\sigma_1$	$\mathcal{U}(0, 2.2)$
$\beta_1$	$\mathcal{N}(0, 1)$
<i>Gibbs covariance kernel + exponential of four Gaussians</i>	
$\sigma_f$	$\mathcal{U}(0, 4)$
$\ell_o$	Gamma(13, 12) ( $m = 1, \sigma = 0.3$ )
$\beta_1$	$\mathcal{N}(0, 1)$
$\beta_2$	$\mathcal{N}(0, 1)$
$\beta_3$	$\mathcal{N}(0, 1)$
$\beta_4$	$\mathcal{N}(0, 1)$
<i>SE covariance kernel + beta-CDF input warping function</i>	
$\sigma_f$	$\mathcal{U}(0, 40)$
$\ell$	Gamma(13, 12) ( $m = 1, \sigma = 0.3$ )
$\alpha$	Gamma(2.6, 1.6) ( $m = 1, \sigma = 1$ )
$\beta$	Exp(2) ( $m = 0, \sigma = 0.5$ )
<i>SE covariance kernel + 2-knot I-spline input warping function</i>	
$\sigma_f$	$\mathcal{U}(0, 10)$
$C_1$	$\mathcal{U}(10^{-3}, 10)$
$C_2$	$\mathcal{U}(0, 10)$
$C_3$	$\mathcal{U}(0, 10)$
<i>SE covariance kernel + 3-knot I-spline input warping function</i>	
$\sigma_f$	$\mathcal{U}(0, 10)$
$t_2$	$\mathcal{U}(0, 1.1)$
$C_1$	$\mathcal{U}(10^{-3}, 10)$
$C_2$	$\mathcal{U}(0, 10)$
$C_3$	$\mathcal{U}(0, 10)$
$C_4$	$\mathcal{U}(0, 10)$

**Table 2.5:** Parameters used for MCMC sampling of the hyperparameters of the various nonstationary covariance kernels applied to synthetic data.

Case	Proposal width, $a$	Walkers	Samples per walker	Burn	Thin
Gibbs+tanh	6	200	500	400	100
Gibbs+exp(1 Gaussian)	16	200	2000	1900	100
Gibbs+exp(4 Gaussians)	8	200	1500	1400	100
SE+beta-CDF warp	8	200	500	400	100
SE+2-knot I-spline warp	4	200	500	400	100
SE+3-knot I-spline warp	8	200	1500	1400	100

alent to the approach in [396]. The inputs are warped using the cumulative distribution function (CDF) of the beta distribution. The shape parameters  $\alpha$  and  $\beta$  have prior distributions which tend to favor  $\alpha \gtrsim 1$  and  $\beta < 1$ . This is intended to compress the core and stretch the edge. The gamma and exponential distributions used were found to offer better control than the log-normal prior distributions used in [396]. Before being passed to the beta-CDF warping function the inputs are linearly mapped from the interval  $[0, 1.1]$  to the interval  $[10^{-6}, 1 - 10^{-6}]$ .

**SE covariance kernel + 2-knot I-spline input warping function** (See equation (2.5), section B.1, section B.4.2, section B.4.2.2 and equation (B.101).) This case warps the inputs using cubic ( $d = 3$ ) I-spline basis functions with internal knots at  $x = 0$  and  $x = 1.1$ . There is a total of three spline coefficients  $C_i$ . The uniform prior distributions were chosen to keep the value of the effective covariance length scale near one. When finding the MAP estimate,  $C_1$  tended to get stuck in a non-physical mode at  $C_1 = 0$ , so the prior distribution was modified to exclude this point. The covariance length scale of the underlying SE covariance kernel was held fixed at  $\ell = 1$  because the effective covariance length scale is entirely determined by the spline coefficients.

**SE covariance kernel + 3-knot I-spline input warping function** (See equation (2.5), section B.1, section B.4.2, section B.4.2.2 and equation (B.101).) This case is the same as above but with an additional internal knot whose position  $t_2$  is a free hyperparameter.

The Gibbs covariance kernel was also tested with covariance length scale functions consisting of several B-spline basis functions (section B.4.1.2.2, equation (B.87)) but the non-informative priors used were insufficient to deliver a credible fit, so the

results are not shown here.

Plots of the results of the two most promising fits are given in [figure 2.16](#) and [figure 2.17](#), plots of the other fits are given in [figure A.1](#) through [figure A.4](#). A plot of the root-mean-square (RMS) error and estimated uncertainty in the fits is given in [figure 2.18](#). Plots of the posterior distributions for the hyperparameters are given in [figure A.5](#) through [figure A.10](#) and summary statistics for the posterior distributions for the hyperparameters are given in [table A.1](#).

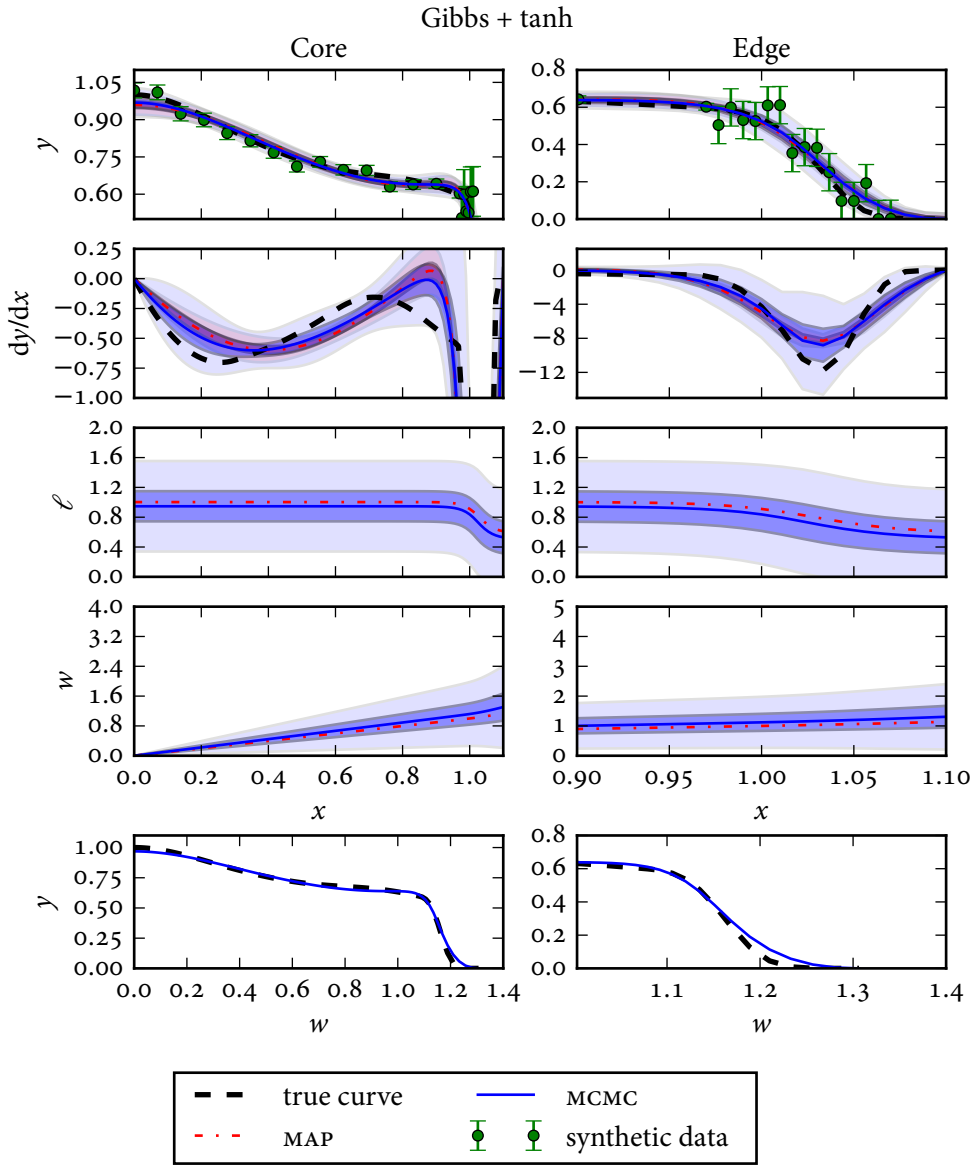
As indicated by [figure 2.18](#), the Gibbs covariance kernel with tanh covariance length scale function used for the bulk of this thesis delivers the most accurate RMS uncertainty estimates and nearly the smallest actual RMS error when evaluated using MCMC. Also of note is the SE covariance kernel with 3-knot I-spline input warping function. This covariance kernel delivered the best RMS error on  $y$  and second-best RMS error on  $dy/dx$ . Its mean effective covariance length scale function for this case (shown in [figure 2.17](#)) has a particularly appealing interpretation: there is a moderate covariance length scale in the core, where the fit needs to change somewhat rapidly to satisfy the zero-slope constraint at the magnetic axis. Then there is a region of long covariance length scale just outside of midradius, corresponding to the region where the profile is mostly flat. Finally, there is a region at the edge with very short covariance length scale in order to accommodate the rapid change at the edge. It is possible that this covariance kernel can be made more useful through appropriate selection of the prior distribution for the hyperparameters.

## 2.5 Application of Gaussian process regression to Alcator C-Mod profiles

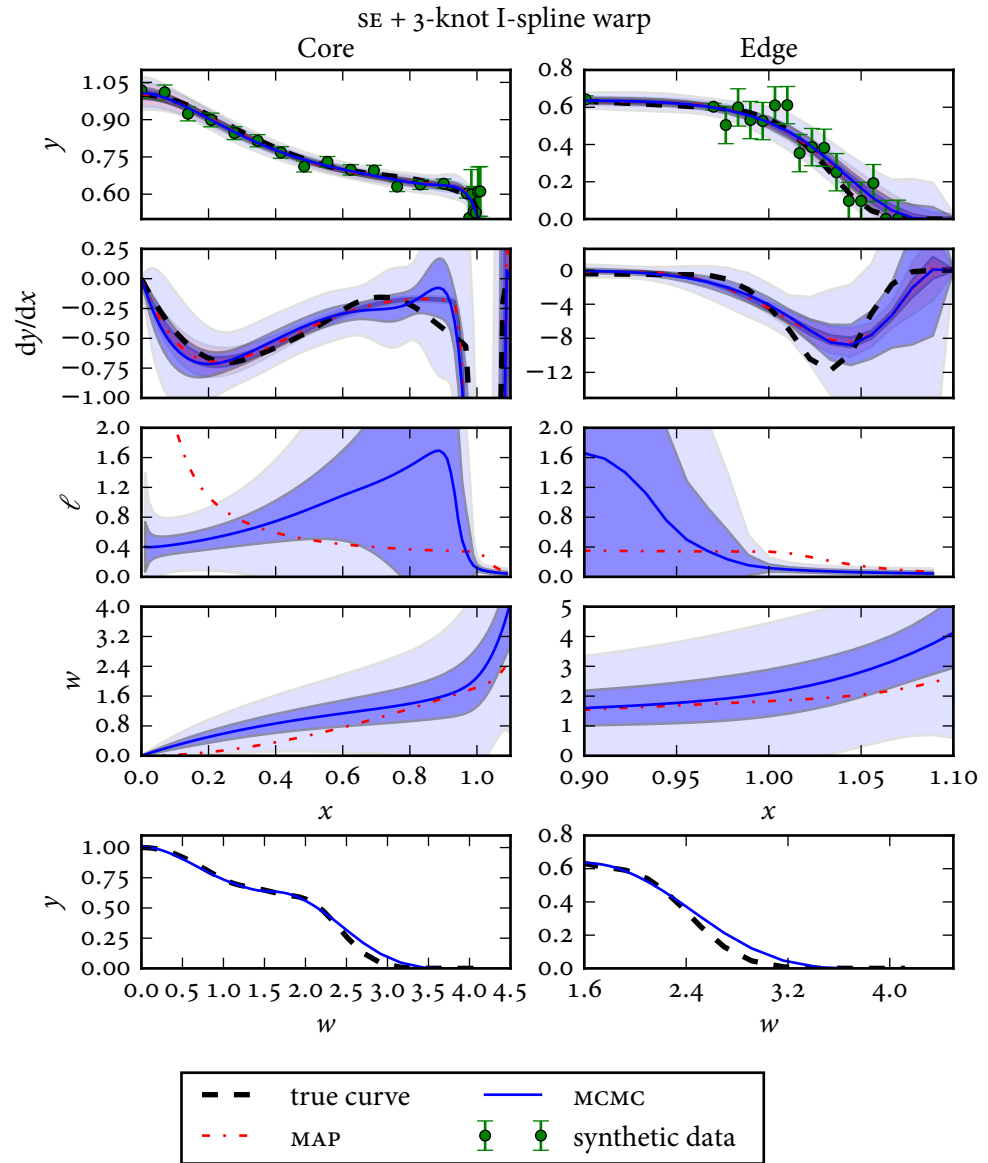
### 2.5.1 Basic fit<sup>4</sup>

This section and [section 2.6](#) will focus on data from Alcator C-Mod [[397](#)] shot number 1101014006, an L-mode discharge with  $I_p = 800$  kA,  $B_T = 5.4$  T and 1 MW of ion cyclotron range of frequencies (ICRF) heating power. This shot was chosen because it was used for previous impurity transport and gyrokinetic validation work [[18](#), [19](#), [55](#)], and the impurity transport coefficient profiles will be re-analyzed using the profiles obtained in this section in [section 2.6](#) and [chapter 3](#). In order to avoid H-mode, this discharge was operated in the upper single null configuration such that the grad- $B$  drift was away from the active x-point. Under these conditions, on-axis parame-

4. This is an updated version of the analysis originally presented in [[305](#)]. The analysis has been revised to be consistent with the current best practices for profile fitting, which have evolved since that paper was written.

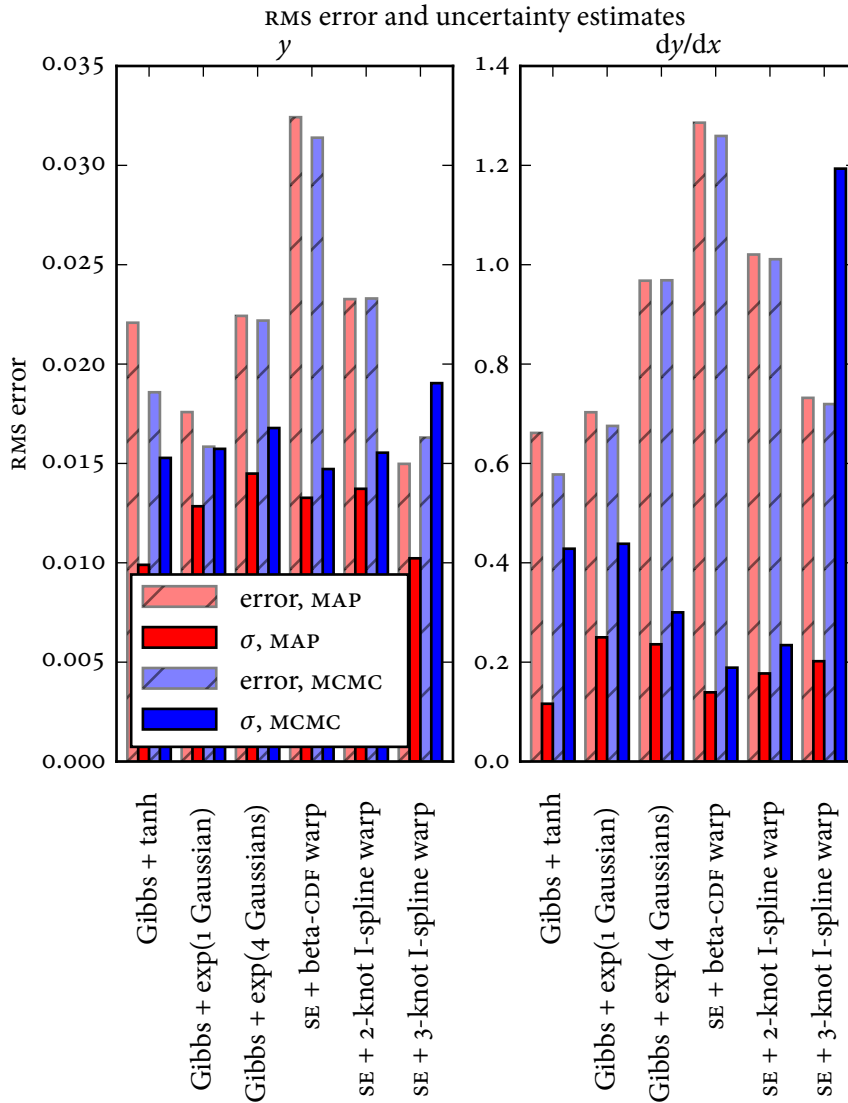


**Figure 2.16:** Synthetic data and fits performed using the Gibbs covariance kernel with tanh covariance length scale function. The true curve is shown as the black dashed line, the synthetic data as the green data points, the MAP estimate as the red dash-dot line and the MCMC estimate as the blue solid line. The effective input warping function  $w$  was computed from equation (B.94). The warped profile in the bottom row shows the result of plotting the true curve (black dashed line) and the mean of the MCMC estimate (blue solid line) as a function of the mean of the MCMC estimate for  $w$ . The effective warping function slightly stretches out the edge to make the data appear stationary. This figure was produced using `synthetic_test_nonstationary.py`.



**Figure 2.17:** Synthetic data and fits performed using the SE covariance kernel with 3-knot I-spline input warping function. This case is comparable to the Gibbs + tanh case in terms of goodness-of-fit. The effective length scale function  $l$  was computed from [equation \(B.93\)](#). The effective length scale function from the MCMC estimate has a particularly nice interpretation: it is moderate in the core where the profile needs to roll over to accommodate the zero-slope constraint, goes to a high value where the profile levels out just outside of midradius, then gets very short to fit the edge. This case also shows the most profound effect of the warping of any of the cases tested: the core has been visibly compressed and the edge stretched. This figure was produced using `synthetic_test_nonstationary.py`.

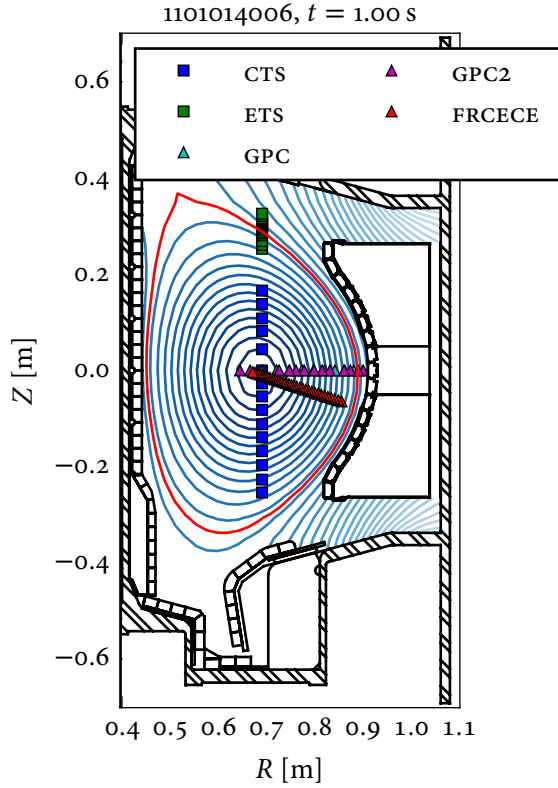




**Figure 2.18:** Root-mean-square errors and uncertainty estimates in the fits to synthetic data using various nonstationary covariance kernels. The “RMS error” is as given in equation (2.71) and the “RMS  $\sigma$ ” is as given in equation (2.72). The Gibbs covariance kernel with tanh covariance length scale function delivered the most accurate uncertainty estimates for both  $y$  and  $dy/dx$  when evaluated using MCMC. The SE covariance kernel with 3-knot I-spline input *overestimated* the errors in both  $y$  and  $dy/dx$  when evaluated with MCMC. All of the other configurations tested dramatically underestimated the uncertainty in the gradient with the SE covariance kernel with beta-CDF input warping function showing particularly abysmal performance on both  $y$  and  $dy/dx$ . This figure was produced using `synthetic_test_nonstationary.py`.

ters of  $n_{e,0} = 1.5 \times 10^{20} \text{ m}^{-3}$  and  $T_{e,0} = 2.5 \text{ keV}$  were obtained over a steady period around 0.4 s long. In this section, we analyze the background  $n_e$  and  $T_e$  profiles from this L-mode using Gaussian process regression and then proceed to obtain profiles of the inverse gradient scale lengths with statistically rigorous uncertainty estimates. Having valid estimates of these uncertainties is critical for comparing to gyrokinetic codes, and the Gaussian process framework makes propagating the uncertainty in the profiles through the analysis code to determine the experimental impurity transport coefficients very efficient, as is demonstrated in [section 2.6](#) and [section 3.6.5](#).

Alcator C-Mod has an extensive diagnostic suite which is described in [\[398\]](#). Two Thomson scattering (TS) systems are used to measure the  $n_e$ ,  $T_e$  profiles in the core and the edge (core TS (CTS) and edge TS (ETS)), and three separate electron cyclotron emission (ECE) systems (two grating polychromators (GPC and GPC2) and a high-resolution heterodyne system (FRCECE [\[399\]](#))) are used to further constrain the core  $T_e$  profile. The equilibrium shape and diagnostic locations are shown in [figure 2.19](#). In addition to extensive bench calibrations [\[400\]](#), the TS density measurements are cross-calibrated with the ECE measurements and a multichannel two-color far-infrared interferometer (TCI). During the stationary period of the discharge the  $n_e$  and  $T_e$  profiles were fairly constant with the exception of sawtooth oscillations. This study is concerned primarily with obtaining sawtooth-averaged estimates of transport in the steady state period, so all signals were averaged over the 0.4 s period from  $1.365 \text{ s} \leq t \leq 0.965 \text{ s}$ . The mean and standard deviation of the TS measurements were weighted according to the diagnostic uncertainty estimates. This had the effect of reducing the influence of noisy/outlying data points, particularly in the edge. The diagnostic uncertainty estimates for the ECE measurements are not as detailed as those of TS, so conventional (unweighted) estimators were used for the sample mean and standard deviation of the ECE diagnostics. In all cases, histograms and probability (“Q-Q”) plots [\[401\]](#) were examined to verify the suitability of the assumption that the noise is approximately Gaussian. Note that while horizontal error bars are shown in [figure 2.20](#) to give a representation of the variability in the equilibrium mapping, these uncertainties were *not* included in the analysis. In general, these error bars are smaller than the width of a given data point as shown in these figures. This coupled with the shallow slope throughout the core means that uncertainties in the independent variable are only likely to play a significant role in the edge (where the profile gets much steeper), and so should not affect the calculation of core transport in the present chapter. Furthermore, the uncertainty in the diagnostic mapping is expected to be dominated by systematic effects, but no attempt was made to quantify these effects. Refer to [section 2.4.2](#) for a discussion of



**Figure 2.19:** Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1101014006, the L-mode shot used to demonstrate the impurity transport analysis. In the legend, “CTS” is the core Thomson scattering system; “ETS” is the edge Thomson scattering system; and “GPC,” “GPC2” and “FRCECE” are the three electron cyclotron emission diagnostics. This figure was produced using `make_diagnostic_figure.py`.

the effect of scatter in the equilibrium mapping on the fit to synthetic data.

The Gibbs covariance kernel given in [equation \(B.73\)](#) with the hyperbolic tangent covariance length scale function given in [equation \(B.78\)](#) was used to smooth both the temperature and density profiles expressed as functions of normalized outboard midplane minor radius  $r/a = (R - R_o)/a$ . The prior distributions for the hyperparameters used for the temperature and density profiles are given in [table 2.6](#). The uniform prior distribution for  $\sigma_f$  was selected to keep the MCMC sampler away from unphysical values while still keeping the information content of the prior distribution weak. Uniform prior distributions were tested for the other hyperparameters, but

**Table 2.6:** Prior distributions used for the hyperparameters of the Gibbs covariance kernel with tanh covariance length scale function when fitting Alcator C-Mod shot 1101014006. For the gamma and exponential distributions the mode  $m$  and the standard deviation  $\sigma$  are also given.

Quantity	$\sigma_f$	$\ell_1$	$\ell_2$	$\ell_w$	$x_0$
$n_e$	$\mathcal{U}(0, 30 \times 10^{20} \text{ m}^{-3})$	Gamma(11, 10) $m = 1,$ $\sigma = 0.33$	Gamma(6, 10) $m = 0.5,$ $\sigma = 0.24$	Exp(10) $m = 0,$ $\sigma = 0.1$	Gamma(203, 200) $m = 1.01,$ $\sigma = 0.071$
$T_e$	$\mathcal{U}(0, 30 \text{ keV})$	Gamma(11, 10) $m = 1,$ $\sigma = 0.33$	Gamma(6, 10) $m = 0.5,$ $\sigma = 0.24$	Exp(10) $m = 0,$ $\sigma = 0.1$	Gamma(203, 200) $m = 1.01,$ $\sigma = 0.071$

were found to occasionally let the MCMC sampler stray into unphysical regions of the hyperparameter space. As such, the remaining prior distributions were selected to be more informative while still giving the problem enough flexibility that the inference is relatively insensitive to the specific details of the prior distribution:

- The gamma prior distribution for  $\ell_1$  was selected to have a mode of 1 and to favor core covariance length scales between 0.5 and 1.75.
- The gamma prior distribution for  $\ell_2$  was selected to have a mode of 0.5 and to favor edge covariance length scales between 0.25 and 1.
- The exponential prior distribution for  $\ell_w$  was chosen to have a mean of 0.1 and to favor short transition widths.
- The gamma prior distribution for  $x_0$  was chosen to have a mode of 1.01 and to favor transition locations close to the LCFS.

As mentioned in [section 2.3.5](#), artificial “observations” can be added to the data  $\mathbf{y}$  to enforce symmetry and other constraints. A zero slope point at  $r/a = 0$  was used to impose a symmetry constraint at the magnetic axis. Value and slope constraints were added outside of the approximate location of the limiter at midplane,  $r/a = 1.1$ . These constraints are listed in [table 2.7](#). Note that the constraints at the edge are given with uncertainties. This is an advantage of this formulation in that it allows a constraint to be specified as being approximate (in the sense of having a Gaussian distribution), such that the data can drive the mean higher or lower at that location if necessary.

**Table 2.7:** Constraints imposed on the profile fits for Alcator C-Mod shot 1101014006 by the addition of artificial “observations.” The slope constraint at the magnetic axis was set as a precise value, the edge values outside of the midplane location of the limiter were set with the indicated  $\pm 1\sigma$  uncertainty.

Quantity	$r/a$	$y$	$y'$
$n_e$ [ $10^{20} \text{ m}^{-3}$ ]	0	—	0
	[1.1, 1.2, 1.3, 1.4]	$0.00 \pm 0.01$	$0.0 \pm 0.1$
$T_e$ [keV]	0	—	0
	[1.1, 1.2, 1.3, 1.4]	$0.00 \pm 0.01$	$0.0 \pm 0.1$

The MAP estimate was found using the sequential quadratic programming routine in SciPy [293, 402]. The optimizer was started at 24 points randomly distributed in the parameter bounds in order to ensure the global maximum was found. The MAP estimates of the hyperparameters are given in the third column of table 2.10, and are shown as the red curves in figure 2.20.

Marginalization over the hyperparameters was carried out using the Python package emcee [354] which implements the affine-invariant ensemble sampler described in [353]. As described in section 2.3.6, this algorithm uses an ensemble of many “walkers” that perform a random walk guided by the posterior distribution for the hyperparameters. The end result is a collection of samples  $\{\theta^{(i)}\}$  for the hyperparameters which can then be used to evaluate equation (2.45). An ensemble of 200 such walkers was used to draw samples from the posterior distribution for the hyperparameters as given in equation (2.39). Each walker was started at a point randomly distributed according to the prior distribution of the hyperparameters. For the  $n_e$  profile each walker was run for 500 samples while the  $T_e$  profile required 1000 samples to reach steady-state. Discarding (“burning”) the first 400 samples from the  $n_e$  chains and 900 samples from the  $T_e$  chains was found to be more than sufficient for the walkers to forget their initial states and settle into a steady-state distribution equal to the posterior distribution. It was found that walkers could get trapped in unphysical modes, so the width of the MCMC proposal distribution was increased to  $a = 4$  for the  $n_e$  profile and  $a = 32$  for the  $T_e$  profile such that all modes had acceptable mixing. The average fraction of proposed steps which were accepted was 28% for  $n_e$  and 6.7% for  $T_e$ . The acceptance fraction for  $T_e$  is somewhat low as a result of having to increase the width of the proposal distribution to allow the walkers to escape from unphysical modes. The autocorrelation times for the unthinned chains for each parameter are given in table 2.8. This yielded far more

**Table 2.8:** MCMC sampler autocorrelation times for each hyperparameter. This is the number of steps required for the correlation between samples in a given chain to decay significantly, and determines the effective sample size.

Quantity	$\sigma_f$	$\ell_1$	$\ell_2$	$\ell_w$	$x_0$
$n_e$	60	72	78	51	31
$T_e$	88	80	69	91	64

**Table 2.9:** Median relative uncertainties over the region  $0 \leq r/a \leq 1$ , given as percentages. The MCMC estimate has a higher uncertainty than the MAP estimate because it fully accounts for the uncertainty in the hyperparameters.

Quantity	Technique	$y$	$y'$	$a/L_y$
$n_e$	spline	1.2	5.9	5.3
	MAP	1.1	5.6	5.6
	MCMC	1.2	6.9	7.0
$T_e$	spline	2.1	12	10
	MAP	1.1	1.6	1.7
	MCMC	1.2	4.2	4.5

samples than is necessary to obtain the uncertainty in the profiles, so the chains were thinned by a factor of 100 before computing any profiles, which is longer than the observed autocorrelation times and has the result of eliminating almost all of the correlation between samples.

The fitted profiles are shown as the red and blue curves in [figure 2.20](#), the bivariate and univariate marginal posterior distributions for the hyperparameters are given in [figure 2.21](#) and [figure 2.22](#), and summary statistics for the posterior distributions of the hyperparameters are given in [table 2.10](#). The MAP and marginalized MCMC estimates yielded very similar mean curves, but with substantially different uncertainty estimates, particularly on  $T_e'$  and  $a/L_{T_e}$ . These differences can be seen in [figure 2.20](#) and are summarized in [table 2.9](#), which gives the median relative uncertainties  $\mathbb{M}[\sigma_{y_*|y}/\mu_{y_*|y}]$  in the quantities of interest over the region  $0 \leq r/a \leq 1$ . The difference in the uncertainties on the gradient between the MAP and MCMC results is very important for applications that are strongly sensitive to gradients: in order to obtain credible estimates of gradients it is necessary to fully account for any uncertainty in the hyperparameters by marginalizing them out using MCMC. This situation has an analogue with the traditional use of splines: using the MAP esti-

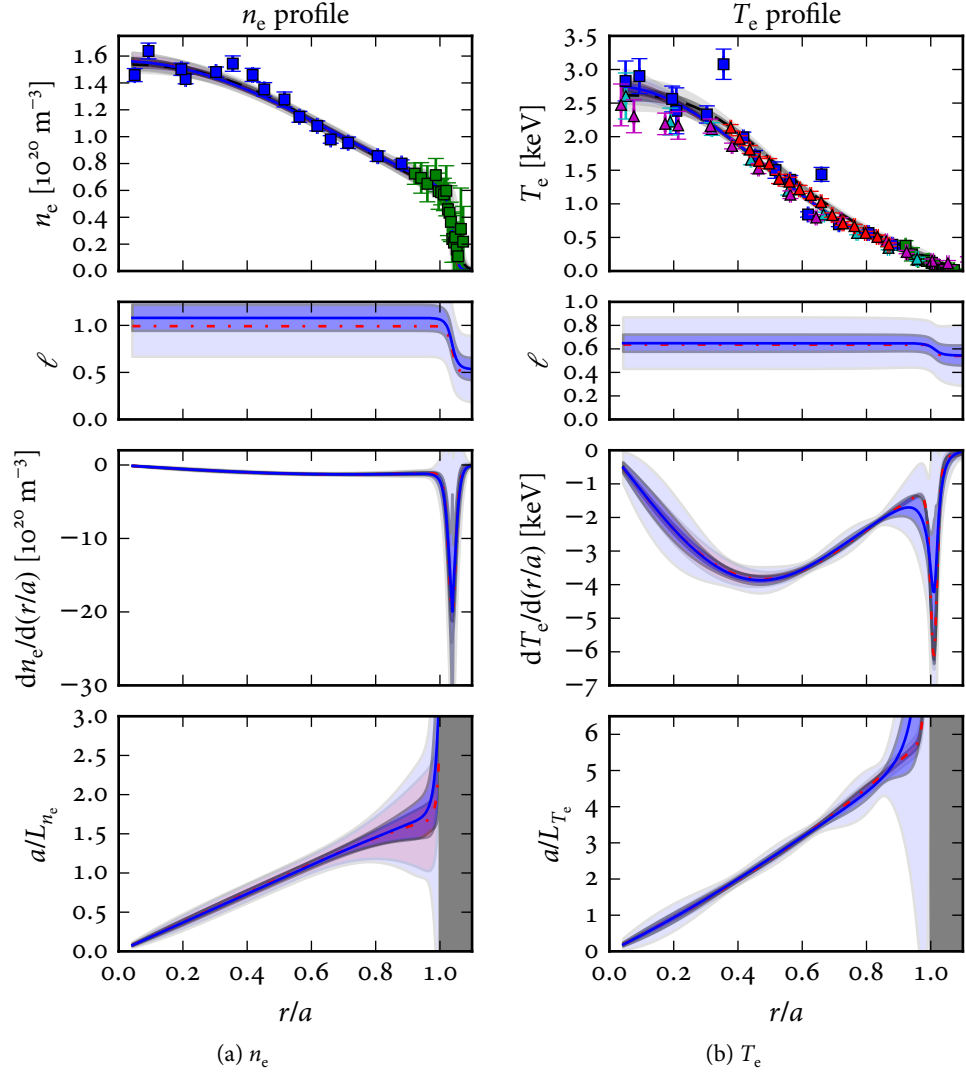
**Table 2.10:** Summary statistics for the posterior distributions of the hyperparameters of the Gibbs covariance kernel with tanh covariance length scale function for Alcator C-Mod shot 1101014006. The “Mode” column is the MAP estimate.

Quantity	Parameter [units]	Mode	Mean	95% interval
$n_e$	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	0.826	1.41	[0.590 , 3.59 ]
	$\ell_1$	0.992	1.08	[0.845 , 1.37 ]
	$\ell_2$	0.489	0.532	[0.334 , 0.803 ]
	$\ell_w$	0.0171	0.0192	[0.00539, 0.0377]
	$x_0$	1.04	1.04	[1.03 , 1.05 ]
$T_e$	$\sigma_f$ [keV]	1.42	2.32	[0.961 , 6.06 ]
	$\ell_1$	0.637	0.649	[0.486 , 0.790 ]
	$\ell_2$	0.541	0.547	[0.400 , 0.765 ]
	$\ell_w$	0.0142	0.0345	[0.00749, 0.137 ]
	$x_0$	1.01	1.01	[0.995 , 1.05 ]

mate is equivalent to simply picking one “best” location for the spline knots and/or smoothing parameter, when this can in fact end up making the curve too restrictive to properly capture the full uncertainty in the gradients. For the density profile, the uncertainty from the spline fit is comparable to the MAP estimate, confirming this hypothesis. For the temperature profile, however, the uncertainty of the spline fit is in fact higher than the MCMC estimate. But, as shown in [figure 2.20](#), the spline fits the data less well than either the MAP or MCMC estimate, and so should not be trusted. To get an estimate of the uncertainty on just the value of a quantity it appears to be sufficient to use the much simpler MAP estimate for the hyperparameters. The choice of which level of sophistication to use depends on how sensitive the end use is to gradients; it is preferable to use the computationally cheap MAP approach of handling the hyperparameters when possible.

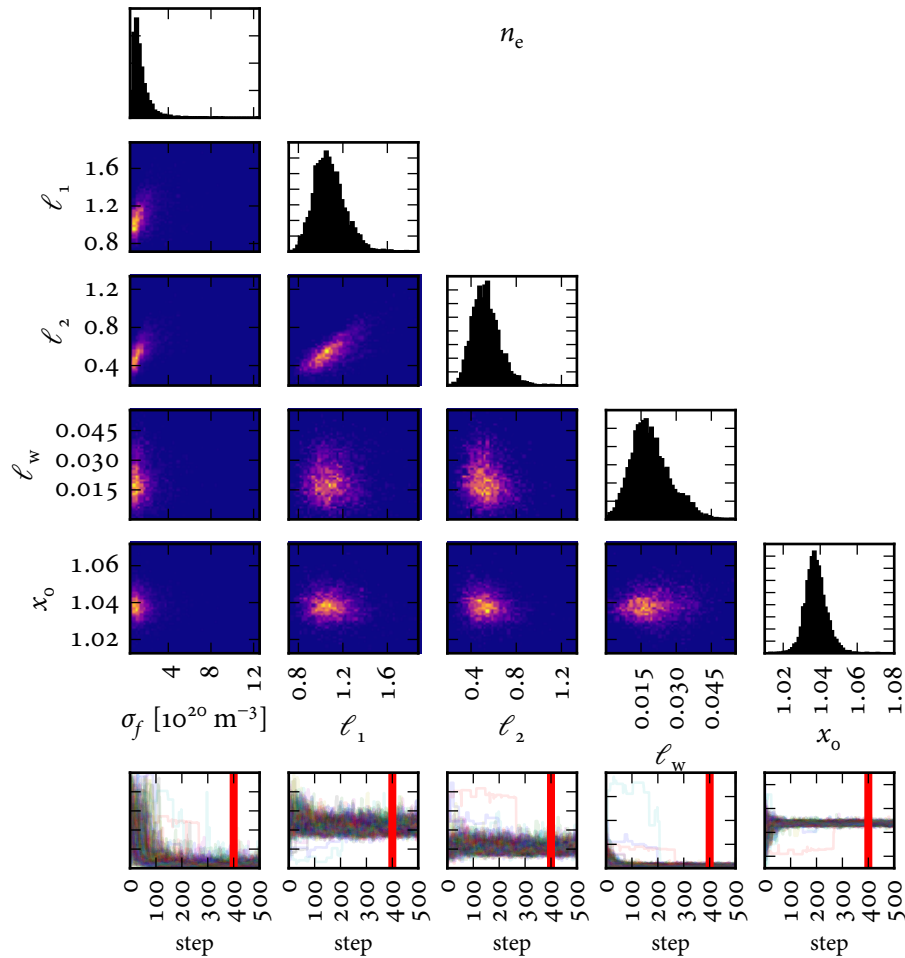
### 2.5.2 Fit including TCI data

In order to demonstrate the results of [section 2.3.9](#), the density profile for a high-density I-mode discharge (Alcator C-Mod shot 1120907032) was fit using the TCI data together with the core and edge Thomson scattering data. This discharge had  $I_p = 1.1 \text{ MA}$ ,  $B_T = 5.8 \text{ T}$  and 5 MW of ICRF heating power. The equilibrium shape and diagnostic locations are shown in [figure 2.23](#). The data were averaged over the constant density time period  $0.8 \text{ s} \leq t \leq 0.9 \text{ s}$ . The outermost point from the core Thomson scattering system was not used because it was systematically much higher

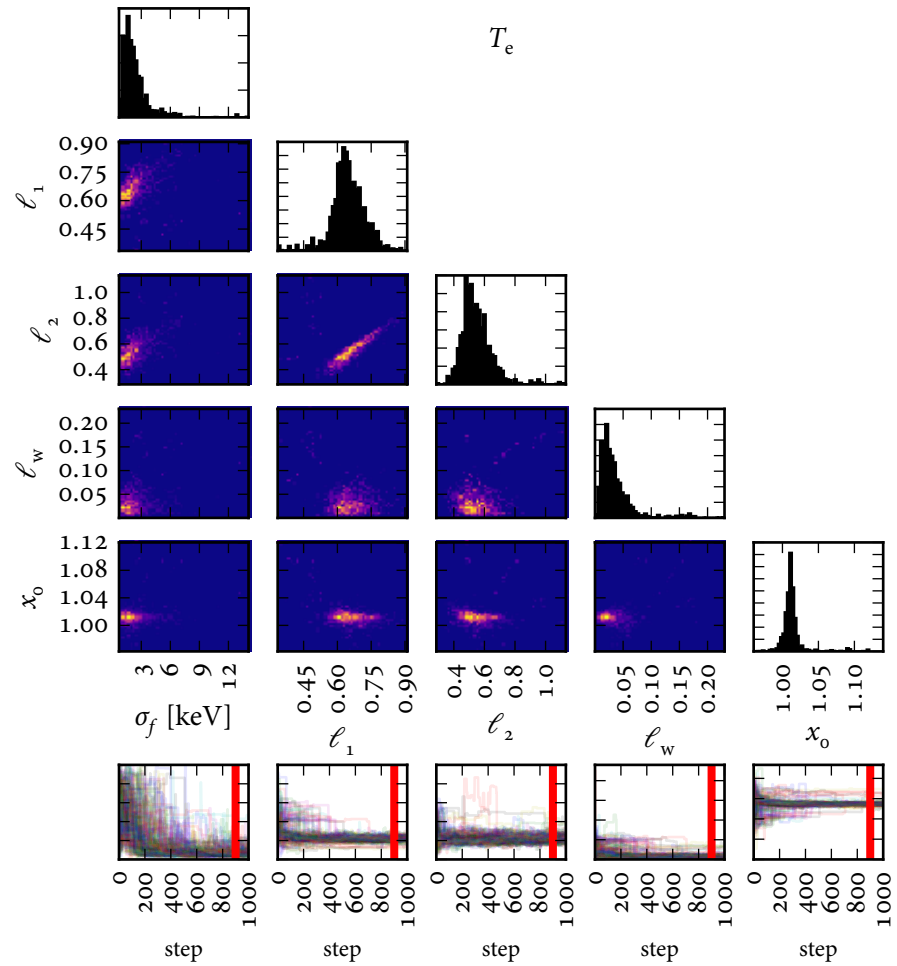


**Figure 2.20:** Data and results for the (a)  $n_e$  and (b)  $T_e$  profiles for Alcator C-Mod shot 1101014006. The error bars are  $\pm 1\sigma$ . Horizontal error bars are, in general, smaller than the width of the plotted points. On the fitted results, the inner dark band is  $\pm 1\sigma$  and the lighter band is  $\pm 3\sigma$ . The result of the MAP estimate is shown as the red dash-dot curve and the result of the marginalization with MCMC is shown as the solid blue curve. In the top panel the spline samples used in the previous work are shown as the black dashed curve; the spline results are *not* shown in the other subplots. Note that the plot of  $a/L$  is cut off at  $r/a = 1$  because the calculation is not trusted outside of  $0 \leq r/a \leq 1$ . All three curves for  $n_e$  overlay very closely, and the largest discrepancies are near the edge. For  $T_e$ , however, there is a much more pronounced discrepancy. This figure was produced using `ne_profiletools.py` and `Te_profiletools.py`.

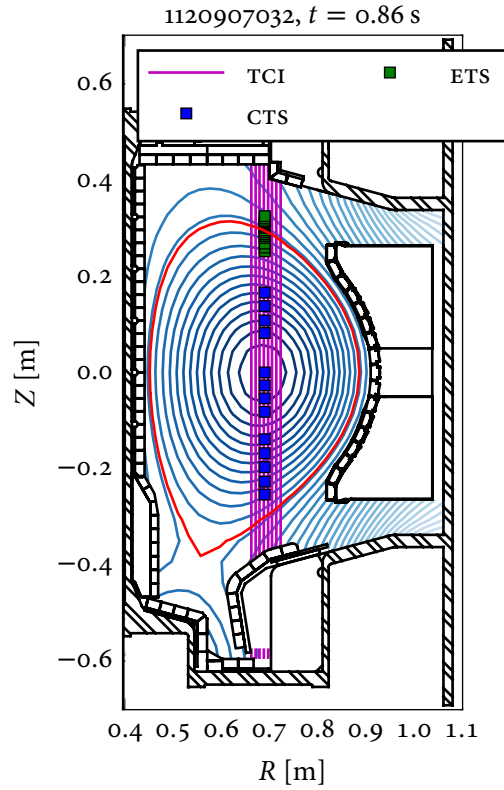




**Figure 2.21:** Matrix of univariate (on the diagonal) and bivariate marginal posterior distributions for the hyperparameters of the fit to the  $n_e$  profile, as computed with MCMC. These plots are essentially 1- and 2D projections of the 5D distribution  $f_{\theta|Y}(\theta|y)$  given in equation (2.39). The bivariate marginal distributions yield information on the correlation between hyperparameters: for instance, the tilted and elongated shape of the bivariate marginal distribution between  $\ell_1$  and  $\ell_2$  means that if the core covariance length scale is shorter, the edge covariance length scale will also tend to be shorter. The bottom row of plots shows the sampler chains, which give an indication as to whether or not the MCMC sampling has converged. This figure was produced using `ne_profiletools.py`.



**Figure 2.22:** Univariate and bivariate marginal posterior distributions for the hyperparameters of the fit to the  $T_e$  profile, as computed with MCMC, presented as in figure 2.21. This figure was produced using `Te_profiletools.py`.



**Figure 2.23:** Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1120907032, including the locations of the eight TCI chords in use. This is the I-mode discharge used to demonstrate fitting the density profile with and without the TCI and CTS diagnostics. Note that the TCI chords were tightly spaced in the center of the plasma for this shot, which makes reconstruction of the profile from TCI data alone impossible. In the legend, “TCI” is the two-color interferometer, “CTS” is the core Thomson scattering system and “ETS” is the edge Thomson scattering system. This figure was produced using `make_diagnostic_figure.py`.

than the surrounding points in both the core and edge systems and caused the fit to have an unphysical bump at the edge. One hundred equally spaced quadrature points were used along each of the eight TCI chords and the weights for quadrature points which were in the shadow of the limiter were set to zero. The prior distribution for the hyperparameters used is given in [table 2.11](#). An exact zero slope constraint was used at the magnetic axis and an edge constraint similar to that given in [table 2.7](#) was imposed by adding four points linearly spaced between  $r_{\text{GH}}/a$  and  $1.25r_{\text{GH}}/a$  (where

**Table 2.11:** Prior distributions for the hyperparameters used when assessing the effect of including the TCI data in the fit.

$\sigma_f$	$\ell_1$	$\ell_2$	$\ell_w$	$x_0$
$\mathcal{U}(0, 8.73 \times 10^{20} \text{ m}^{-3})$	Gamma(13, 12)	Gamma(5.8, 9.7)	Exp(10)	Gamma(102, 101)
	$m = 1.0,$ $\sigma = 0.3$	$m = 0.5,$ $\sigma = 0.25$	$m = 0,$ $\sigma = 0.1$	$m = 1.0,$ $\sigma = 0.1$

$r_{\text{GH}}/a$  is the normalized midplane minor radius of the GH limiter) where the value and slope were both required to be approximately zero. The hyperparameters were sampled using an affine-invariant ensemble sampler with 200 walkers and a proposal distribution width of  $a = 8$ . The sampler was run for 500 samples, the first 400 samples were discarded and the remaining samples were thinned by a factor of 100 before computing profiles.

Summary statistics for the posterior distributions with and without the TCI and core Thomson scattering data included are given in [table 2.12](#), the fitted profiles are given in [figure 2.24](#) and the values on the TCI chords are given in [figure 2.25](#). The marginal posterior distributions of the hyperparameters are given in [figure A.11](#) through [figure A.13](#). In the figures and tables, the case “CTS+ETS” refers to the use of both the core and edge Thomson scattering data, “CTS+ETS+TCI” refers to the use of the core and edge Thomson scattering data together with the TCI data and “ETS+TCI” refers to the use of just the edge Thomson scattering data and the TCI data.

Between the CTS+ETS case and the CTS+ETS+TCI case the shape of the fit does not change substantially outside of the error bars, nor do the posterior distributions for the hyperparameters differ significantly. The relative uncertainties of the fits are given in [table 2.13](#). Including the TCI data slightly reduces the uncertainty in the values in the core, and has only a very minor effect on the uncertainty in the gradients in the core: because the chords are all concentrated on the core and the Thomson scattering diagnostic has already been cross-calibrated with the TCI data, these data do not provide a very strong constraint on the fit. It may be possible to adjust the relative weights of the Thomson scattering data and the TCI data in the fit to obtain better agreement with the TCI chords and hence make better use of the TCI data, but this has not yet been attempted.

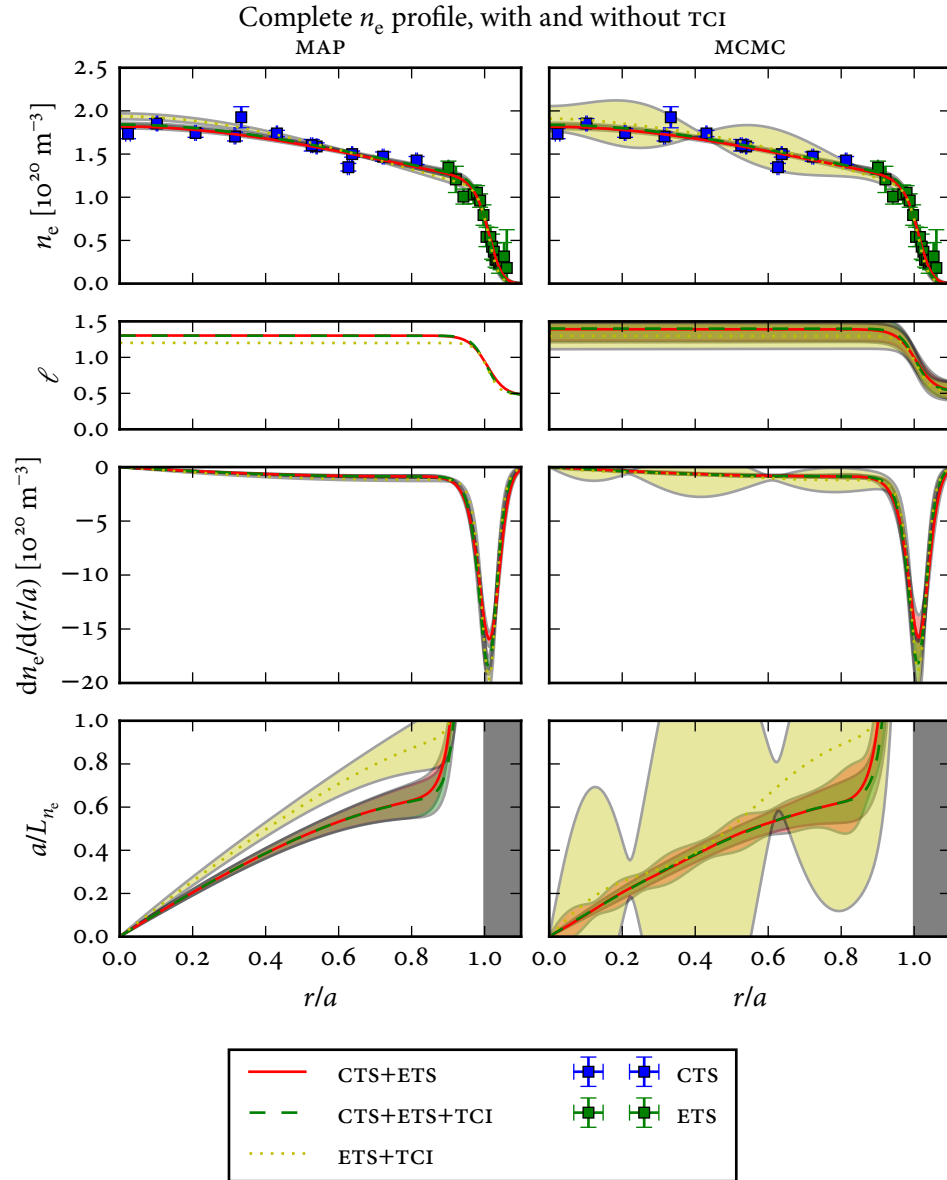
A further improvement would be to perform an integrated analysis (see [section 2.3.10](#)) of the Thomson scattering and TCI data together in order to infer the calibrations and profiles simultaneously with the Gaussian process acting as the prior distribution for the profile shape, but this level of sophistication has not yet

**Table 2.12:** Summary statistics for the posterior distributions of the hyperparameters for the fits performed with and without TCI/CTS included. All three cases arrived at very similar posterior distributions.

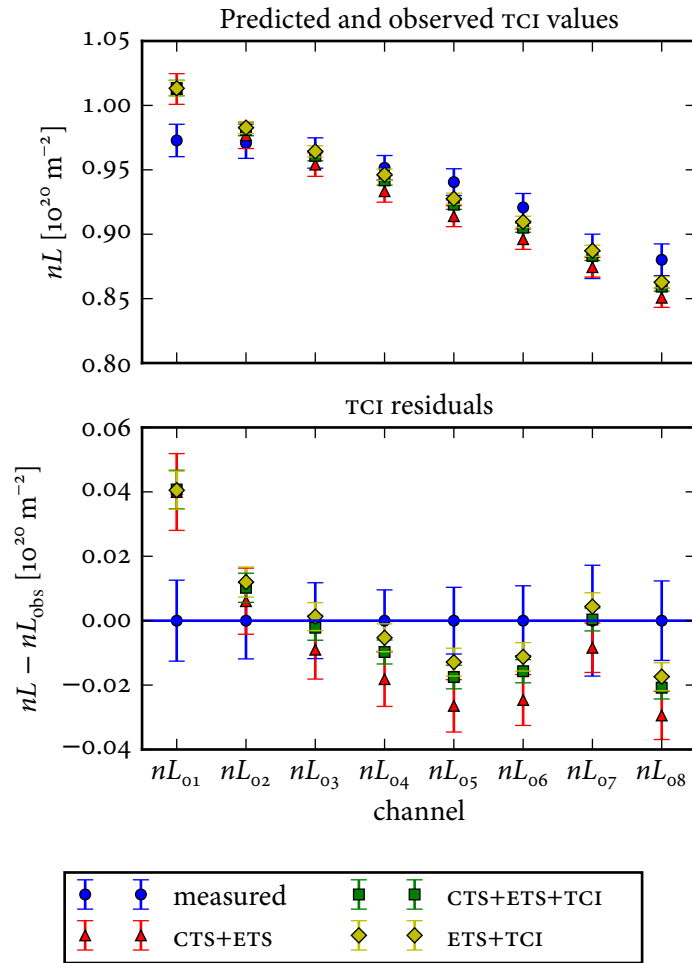
Case	Parameter [units]	Mode	Mean	95% interval
CTS+ETS	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	1.07	1.75	[0.732 , 4.45 ]
	$\ell_1$	1.30	1.39	[1.10 , 1.73 ]
	$\ell_2$	0.464	0.529	[0.318 , 0.808 ]
	$\ell_w$	0.0494	0.0548	[0.0353, 0.0736]
	$x_0$	1.01	1.01	[0.989 , 1.02 ]
CTS+ETS+TCI	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	1.09	1.75	[0.741 , 4.12 ]
	$\ell_1$	1.30	1.40	[1.11 , 1.76 ]
	$\ell_2$	0.477	0.529	[0.327 , 0.778 ]
	$\ell_w$	0.0416	0.0451	[0.0307, 0.0632]
	$x_0$	1.01	1.00	[0.989 , 1.02 ]
ETS+TCI	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	1.10	1.73	[0.736 , 4.14 ]
	$\ell_1$	1.20	1.30	[0.969 , 1.66 ]
	$\ell_2$	0.478	0.524	[0.310 , 0.809 ]
	$\ell_w$	0.0342	0.0420	[0.0244, 0.0569]
	$x_0$	1.01	1.01	[0.988 , 1.02 ]

**Table 2.13:** Median relative uncertainties over the region  $0 \leq r/a \leq 1$  for the cases with and without TCI/CTS included in the fit. Relative uncertainties are given as percentages. The uncertainties are higher when the core Thomson scattering data are not present. The addition of the TCI data to the core and edge Thomson scattering data only decreases the uncertainty very slightly.

Case	Method	$n_e$	$dn_e/d(r/a)$	$a/L_{n_e}$
CTS+ETS	MAP	1.1	7.9	7.6
	MCMC	1.3	18	18
CTS+ETS+TCI	MAP	0.95	7.8	7.5
	MCMC	1.1	12	12
ETS+TCI	MAP	1.4	9.0	8.4
	MCMC	9.8	94	92



**Figure 2.24:** Data and results for the  $n_e$  profile with (green dashed) and without (red solid) the TCI data included, presented as in figure 2.20. Only the  $\pm 1\sigma$  uncertainty envelopes are shown. The TCI data only have a fairly weak effect on the fit in the case where both core and edge Thomson scattering are available. The TCI data enable the profile to be roughly reconstructed in the case where the core Thomson scattering data are absent (yellow dotted), despite the very tight chord spacing shown in figure 2.23. The MAP (on the left) and MCMC (on the right) estimates have very different uncertainty envelopes for the ETS+TCI case: while the TCI data enable the shape of the mean profile to be reconstructed in the absence of the CTS data, the uncertainties are *much* higher, especially for the gradient and gradient scale length. This figure was produced using `ne_TCI_test.py`.



**Figure 2.25:** Data and results for the values of line-integrated density on the various TCI chords, where the fit is performed with (green points) and without (red points) the TCI data included. The upper figure shows the actual values, the lower figure shows the residuals. Including the TCI data brings the fitted values closer to the observations, but does not arrive at complete agreement. It also makes the uncertainties on the predicted TCI values much smaller. When the core Thomson scattering data are not present (yellow points) the fitted values lie even closer to the observations. The persistent discrepancy on the innermost and outermost chords may indicate calibration issues, or the need for finer structure in the very core of the profile. Note that these points were all evaluated using the MCMC estimate; the values and uncertainties on the MAP estimate were basically the same. This figure was produced using `ne_TCI_test.py`.

been attempted on Alcator C-Mod data. (Integrated analysis of Thomson scattering, interferometer and soft x-ray data has been implemented on w7-AS for single spatial points [285, 372, 373, 375, 378], but the interferometer is simply used to set an upper bound on the density. Integrated analysis of Thomson scattering, electron cyclotron emission, interferometer and lithium ion beam emission spectroscopy data has been implemented on ASDEX Upgrade [386, 388, 389], but the use of B-splines leads to uncertainty envelopes which indicate that an unphysical degree of spatial variation is permitted by the prior distribution. Integrated analysis of Thomson scattering, interferometry, reflectometry and helium ion beam data has been implemented on TJ-II [394]. Integrated analysis of Thomson scattering data and a single interferometer chord has been implemented on MAST [316]. Of these, only the MAST analysis uses a Gaussian process prior distribution.)

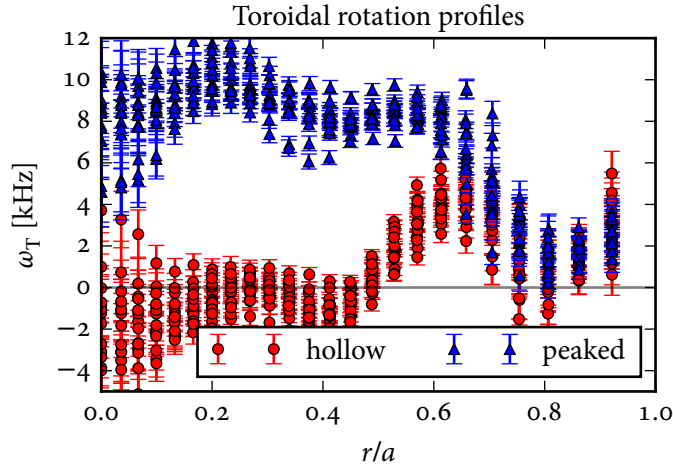
Another potential application of this analysis is to make use of shots where the core Thomson scattering data are corrupted or otherwise unavailable but the edge Thomson scattering and TCI data are both available. In this case, GPR lets the edge Thomson scattering data act as a constraint on the TCI data, which would otherwise not have enough information about the edge of the plasma to be successfully inverted using conventional tomographic techniques [359]. This case corresponds to the “ETS+TCI” case in the previous tables and figures. The rough shape of the core profile is recovered, but obviously the fit is not ideal. This case exhibits a dramatic difference between the MAP and MCMC uncertainty estimates: for the MAP estimate the uncertainties are slightly higher, which makes sense given that the local core Thomson scattering measurements are assumed to be a much stronger constraint than the line-integrated TCI measurements. But for the MCMC estimate the uncertainties are *much* higher – the median relative uncertainty is as high as 94% for  $dn_e/d(r/a)$ ! This highlights the importance of using MCMC to fully capture the posterior uncertainty when very little data are available. While this analysis demonstrates that combining the TCI data with the ETS data allows reconstruction of the rough shape of the mean curve, the uncertainties are so high as to limit the usefulness of this approach for rigorous analysis. If more chords were available, or if the existing chords were moved farther out in the plasma, more detail in the core could be inferred when the core Thomson scattering data are not reliable.

### 2.5.3 Computing second derivatives to investigate rotation reversals

#### 2.5.3.1 Motivation

Explaining the dramatic change of intrinsic rotation with slight changes in density is an open question [26]: White et al. present a pair of discharges where the turbulent





**Figure 2.26:** Toroidal rotation profiles for shots 1120221011 (red circles, hollow rotation profile) and 1120221012 (blue triangles, peaked rotation profile). The data shown cover  $0.9 \text{ s} \leq t \leq 1.2 \text{ s}$ . This figure was produced using `plot_rotation.py`.

drive terms are essentially the same within error bars, the only noticeable difference being that the electron density is slightly higher in one discharge. Despite this apparent similarity, the toroidal rotation profile is dramatically different between the two discharges: peaked and strongly co-current in the lower density case and hollow and slightly counter-current on axis in the higher density case. The rotation profiles for these shots are shown in figure 2.26, the electron density profiles are shown in figure 2.28 and the electron temperature profiles are shown in figure 2.29.

Ida et al. [403] and the references therein discuss the connection between the ion temperature gradient and the intrinsic toroidal rotation. Beyond this, there are theoretical reasons to believe that the *second* derivative of the profiles plays a role in momentum transport [404–414]. M. Barnes gave a particularly intuitive description of two of the ways this can occur [410]:

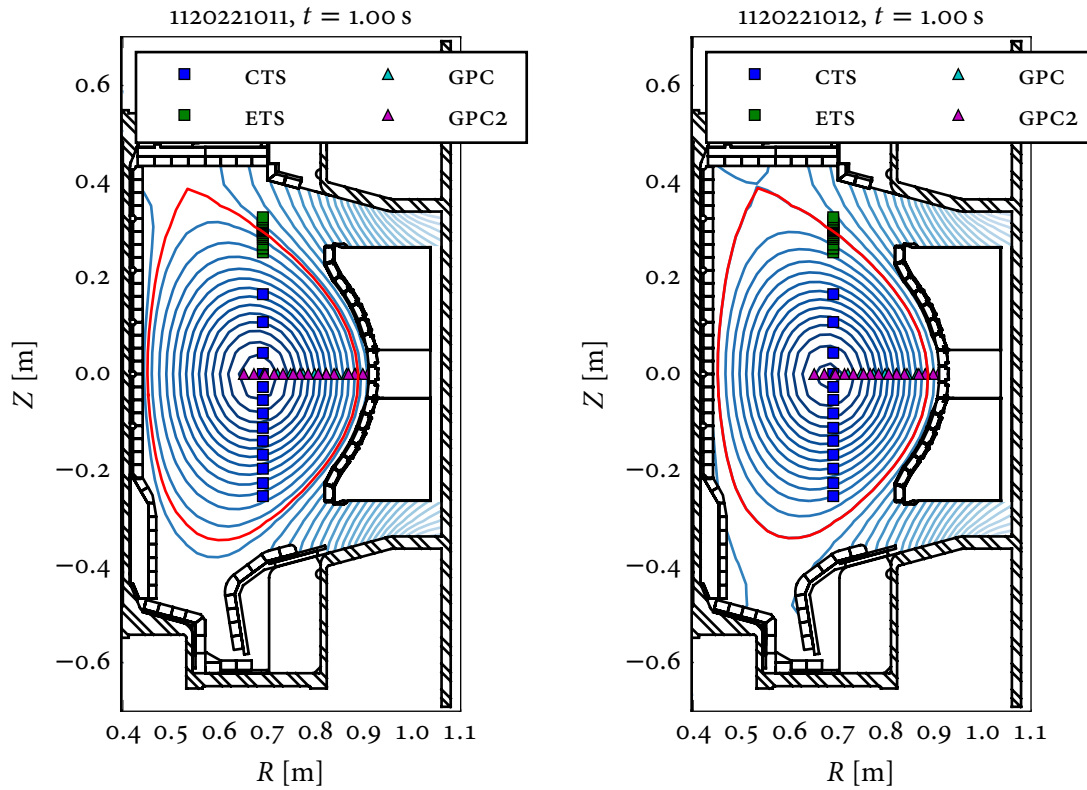
1. The strength of the turbulence depends on the profile gradient. Therefore, the second derivative of the profile corresponds to a gradient in the strength of the turbulence. Consider two ensembles of trapped electrons which start from the same place at the outboard midplane of an up-down symmetric plasma which have opposite parallel velocities. Supposing the toroidal plasma current is in the same direction as the toroidal magnetic field, the electrons for which  $v_{\parallel} > 0$  drift radially outwards when undergoing a banana orbit while the electrons for which  $v_{\parallel} < 0$  drift radially inwards (see figure 14.14 of [1] for

an illustration). In the absence of turbulence (and ignoring other collisions), the electrons all eventually return to the same poloidal location, forming a pair of closed banana orbits. With turbulence of constant strength (i.e., the second derivatives of the profiles are zero) the losses due to turbulence for both populations of electrons are equal, and no net momentum transport occurs. But if there is stronger turbulent transport on one side than on the other then more of the electrons going one direction will be lost and there will be net transport of momentum. Therefore, the second derivatives of the plasma profiles must be non-zero for there to be net momentum transport.

2. When there are spatial gradients in the density and temperature profiles there are diamagnetic corrections to the distribution function of the particles which induce a diamagnetic flow. The strength of this flow depends on the first derivatives of the density and temperature profiles. When there is no initial rotation, this diamagnetic flow cancels with the  $\mathbf{E} \times \mathbf{B}$  flow. But there can be different diffusion coefficients for these two flows. So, if the second derivatives of the profiles are non-zero, there will be a gradient in the diamagnetic flow which can cause the diamagnetic flow to diffuse at a different rate than the  $\mathbf{E} \times \mathbf{B}$  flow, thereby creating a net diffusion of momentum.

### 2.5.3.2 Performing the fit

In the past it has been widely believed that it is impossible to test a hypothesis depending on second derivatives because of how large the uncertainties are expected to grow with each order of derivative. As noted in [section 2.3.5](#), derivatives of arbitrary order and their uncertainties can be predicted when using GPR. The profile data from the same shots used in [26] were re-fit using GPR and the second derivatives and their uncertainties were computed. The shots of interest are Alcator C-Mod shots 1120221011 (higher density, hollow rotation profile) and 1120221012 (lower density, peaked rotation profile). Both of these L-mode discharges had  $I_p = 800$  kA,  $B_T = 5.4$  T and were heated with 1.2 MW of ICRF heating power. For each shot the data were averaged over the period  $0.9 \text{ s} \leq t \leq 1.2 \text{ s}$ . Data from the core and edge Thomson scattering systems were used for both the  $n_e$  and  $T_e$  profiles. Note that the innermost Thomson scattering point was removed from the fit because it was observed to exhibit a systematically low density throughout the run day and hence caused an unphysical dip in the core density profile fit. In addition, the data from two ECE systems (GPC and GPC2) were used for the  $T_e$  profiles. The equilibrium shape and diagnostic locations are shown in [figure 2.27](#). Only the data from inside the LCFS were used in the fit, so a stationary squared exponential kernel was used.



(a) 1120221011 (higher density, hollow rotation profile)

(b) 1120221012 (lower density, peaked rotation profile)

**Figure 2.27:** Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shots (a) 1120221011 and (b) 1120221012, the pair of discharges used to investigate rotation reversals. In the legend, “CTS” is the core Thomson scattering system, “ETS” is the edge Thomson scattering system, and “GPC” and “GPC2” are the two grating polychromator electron cyclotron emission diagnostics. These figures were produced using `make_diagnostic_figure.py`.

This eliminated the need for a constraint at the limiter, but the  $y' = 0$  constraint at the magnetic axis was still used. Very simple uniform prior distributions were used for the hyperparameters, which are given in [table 2.14](#). In order to ensure a complete accounting of the uncertainty, the hyperparameters were marginalized using MCMC with an affine-invariant ensemble sampler [353, 354]. The sampler was run with 200 walkers for 600 samples, 200 samples were burned and the remaining samples were

**Table 2.14:** Prior distributions used for the hyperparameters of the squared exponential covariance kernel when fitting the rotation reversal data.

Quantity	$\sigma_f$	$\ell$
$n_e$	$\mathcal{U}(0, 50 \times 10^{20} \text{ m}^{-3})$	$\mathcal{U}(0, 15)$
$T_e$	$\mathcal{U}(0, 50 \text{ keV})$	$\mathcal{U}(0, 25)$

thinned by a factor of 200. The width of the proposal distribution was kept at the default  $a = 2$ . The computed profiles are given in [figure 2.28](#) and [figure 2.29](#), the marginal posterior distributions for the hyperparameters are shown in [figure A.14](#) and summary statistics for the posterior distributions of the hyperparameters are given in [table A.2](#). The posterior distributions for the hyperparameters are very similar between the two shots for both  $n_e$  and  $T_e$ .

[Figure 2.28](#) and [figure 2.29](#) show the unnormalized second derivative profiles as well as two possible normalizations:

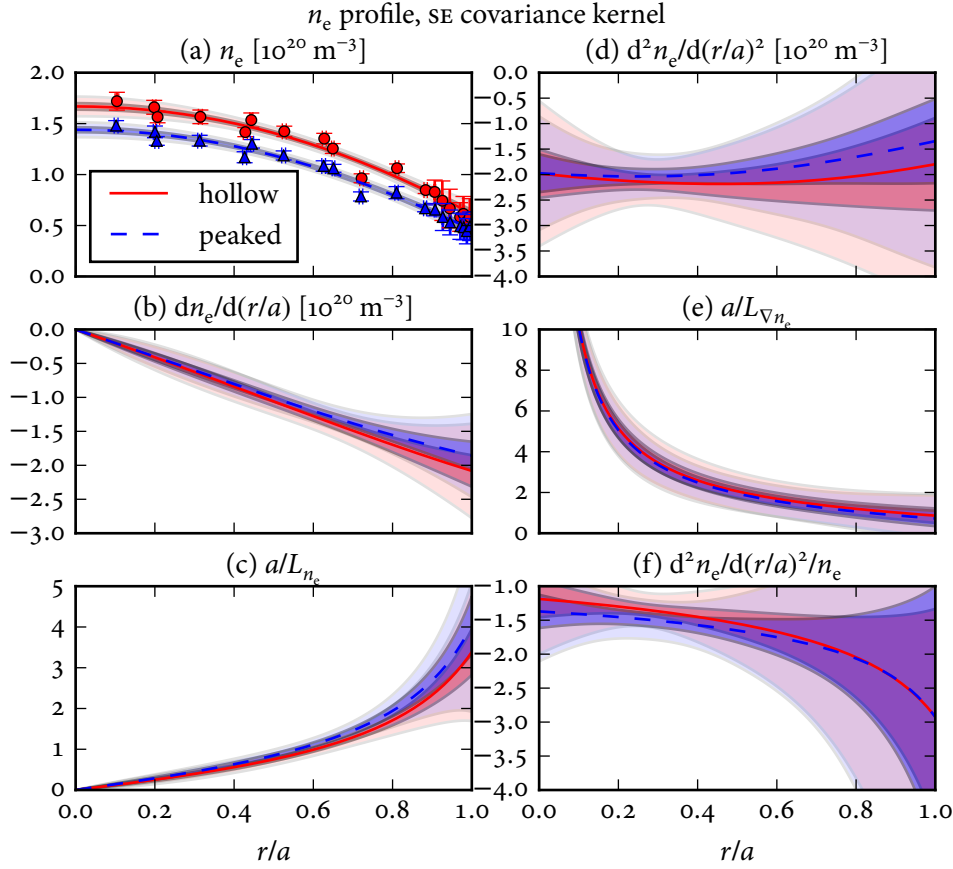
$$\frac{a}{L_{\nabla n_e}} = \frac{a d^2 n_e / dr^2}{dn_e / dr} \quad (2.73)$$

$$\frac{a^2 d^2 n_e / dr^2}{n_e}. \quad (2.74)$$

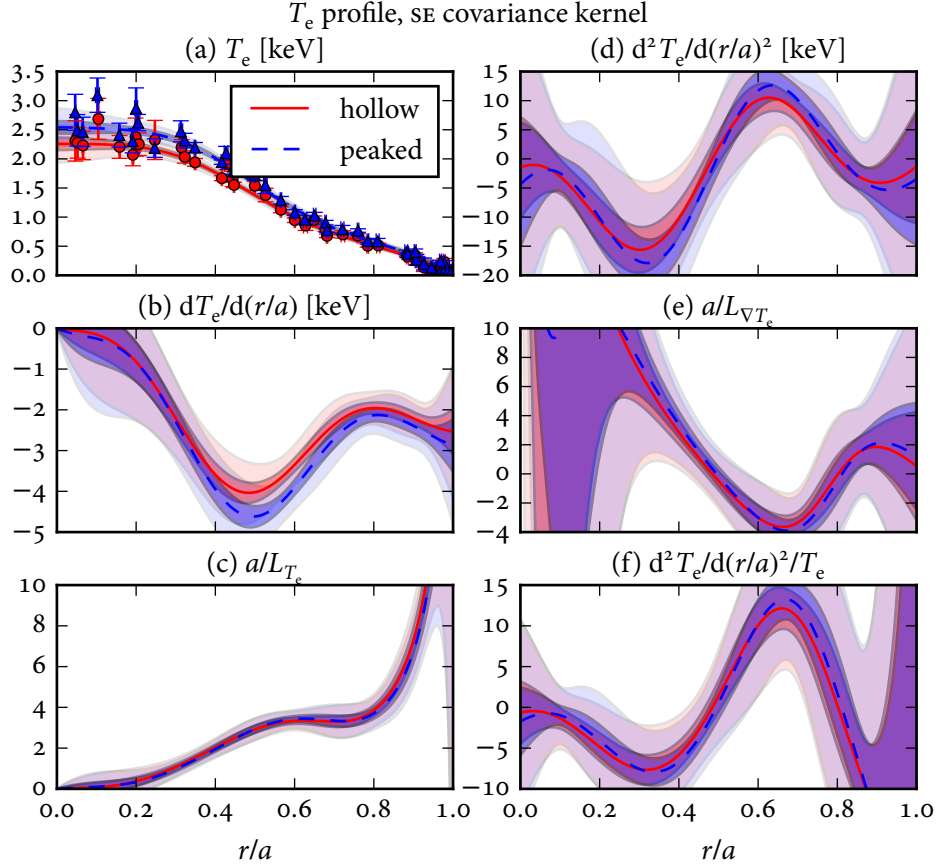
The first version was chosen by analogy with the gradient scale length, [equation \(2.1\)](#), but is made useless in the core by the fact that the first derivative  $dn_e/dr$  must go to zero on axis. Therefore, the second version, chosen purely from dimensional analysis, is likely to be more useful in practice.

### 2.5.3.3 Quantifying the profile differences

In order to test theories which attempt to explain momentum transport in terms of differences in the second derivatives it is necessary to have an objective way of characterizing the difference between two profiles in the presence of very large, overlapping uncertainty envelopes. One approach could be to apply multivariate hypothesis testing as described in [\[415\]](#). In this case, the test would be deciding whether or not the data are sufficient to reject the null hypothesis that the mean vectors are equal. This oversimplified choice between “mean vectors match” and “mean vectors do not match” has been noted as a key shortcoming of frequentist hypothesis testing [\[286\]](#): what really matters is the probability of the profiles being different *enough* to cause a change in the rotation profile. The Bayesian framework allows a more useful test to be made: we can actually use the results of the GPR fit to



**Figure 2.28:** Data and fitted  $n_e$  profile for the two rotation reversal shots. **Red circles and solid curves** are from shot 1120221011 (higher density, hollow rotation profile) and **blue triangles and dashed curves** are from shot 1120221012 (lower density, hollow rotation profile). For all curves, the dark band is  $\pm 1\sigma$  and the light band is  $\pm 3\sigma$ . The subplots are: (a) the averaged data and the fitted core profiles, (b) the first derivative of the profile, (c) the normalized inverse gradient scale length of the profile, (d) the second derivative of the profile, (e) the inverse normalized gradient scale length of the gradient (see equation (2.73)), and (f) the normalized second derivative (see equation (2.74)). There is no significant difference in the first or second derivatives between these two discharges. This figure was produced using `ne_rotation.py`.



**Figure 2.29:** Data and fitted  $T_e$  profile for the two rotation reversal shots, presented as in figure 2.28. This figure was produced using `Te_rotation.py`.

quantify how likely is it that the profiles differ by a given amount. For a given pair of quantities to compare  $y_1, y_2$  we have  $y_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$  and  $y_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ . In this context,  $y_1$  could be the second derivative at  $r/a = 0.5$  of the hollow rotation case and  $y_2$  could be the second derivative at  $r/a = 0.5$  of the peaked rotation case. The distribution for the difference of these two quantities is then

$$\Delta = y_1 - y_2 \sim \mathcal{N}(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2). \quad (2.75)$$

The probability that  $\Delta$  is larger than some value  $c$  is then simply

$$\mathbb{P}(\Delta > c) = 1 - F_\Delta(c) = 1 - \Phi\left(\frac{c - (\mu_1 - \mu_2)}{\sigma_1^2 + \sigma_2^2}\right), \quad (2.76)$$

and the probability that  $\Delta$  is less than some value  $-c$  is

$$\mathbb{P}(\Delta < -c) = F_{\Delta}(-c) = \Phi\left(\frac{-c - (\mu_1 - \mu_2)}{\sigma_1^2 + \sigma_2^2}\right), \quad (2.77)$$

where  $F_{\Delta}(\Delta)$  is the CDF of  $\Delta$  and  $\Phi(x)$  is the CDF of the standard normal distribution. Combining these gives

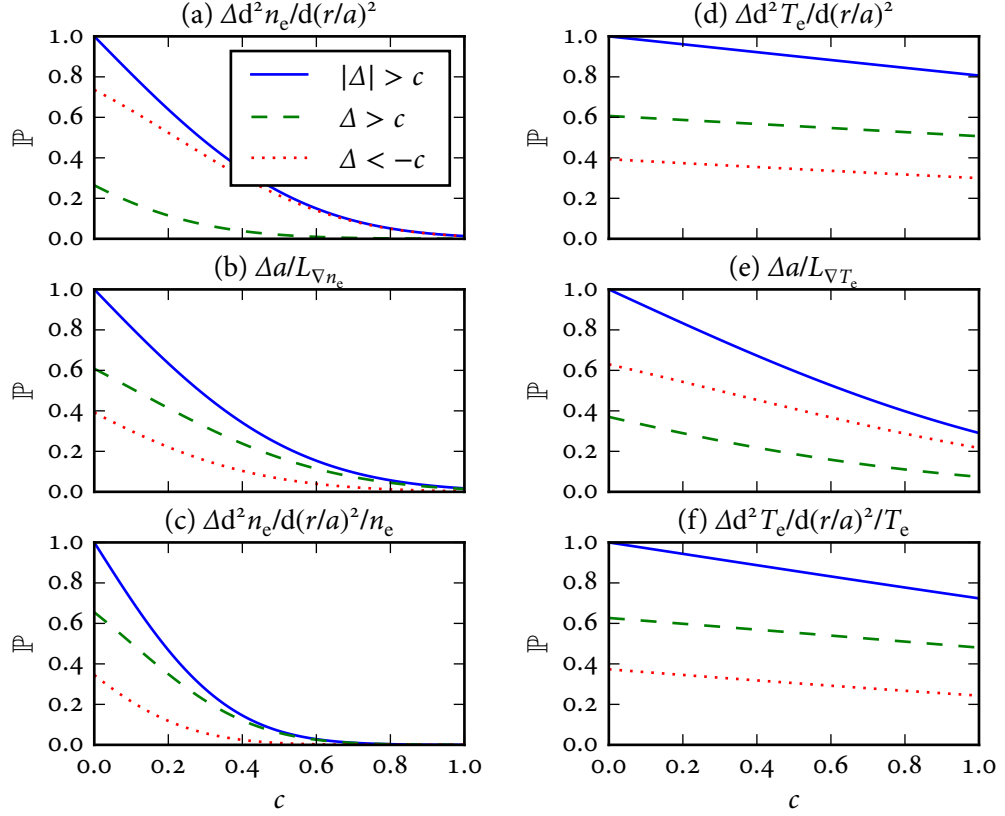
$$\mathbb{P}(|\Delta| > c) = \mathbb{P}(\Delta > c) + \mathbb{P}(\Delta < -c) = 1 - F_{\Delta}(c) + F_{\Delta}(-c). \quad (2.78)$$

Once quantitative theoretical predictions are available for these shots, [figure 2.30](#) will enable them to be checked against these results. Specifically, for a given change in the (normalized or non-normalized) second derivative profile needed to account for the change in rotation, the probability that the profiles differ by at least this much can be read off the figure. Of particular interest to the present discussion are the values of  $\mathbb{P}(\Delta > 0)$ , which can be interpreted as  $\mathbb{P}(y_1 > y_2)$ . In all of the normalized cases this probability is only about 60%, which is generally not taken to be statistically significant: therefore the present data do not appear to support any claims that there are substantial differences between the second derivative profiles, thus ruling out subtle differences in the profile shape as the cause of the dramatic change in rotation between these two shots.

#### 2.5.4 2D fitting of sawtooth-free data

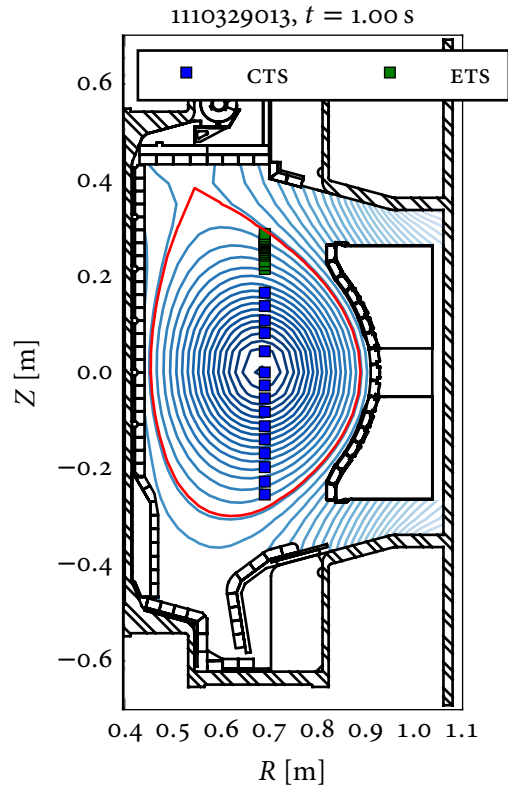
Because `gptools` includes support for data of arbitrary dimension, it is possible to fit profile data as a function of space *and* time. Handling sawteeth will require the development of a very complicated covariance kernel to represent the fast, periodic variations in the signal, a task which is not in the scope of this thesis. Therefore, a sawtooth-suppressed discharge with lower hybrid current drive (LHCD) was used to demonstrate this capability.<sup>5</sup> Alcator C-Mod shot 1110329013 is free of sawtooth activity over the period  $1.0 \text{ s} \leq t \leq 1.4 \text{ s}$ . This shot has  $I_p = 430 \text{ kA}$ ,  $B_T = 5.4 \text{ T}$ , with 800 kW of LHCD power. Only the  $T_e$  data from the core and edge Thomson scattering systems were used, as the ECE systems were contaminated with non-thermal emission and hence cannot deliver a reliable temperature measurement. The equilibrium shape and diagnostic locations are given in [figure 2.31](#). The fit was performed using a 2D SE covariance kernel. A zero slope constraint was imposed at the magnetic axis by adding an artificial  $dT_e/d(r/a) = 0$  point at  $r/a = 0$  for each Thomson time point. An edge constraint similar to what was used for the 1D fits was imposed by

5. Refer to section 5.4.3 of [297] for an example of a 1D GPR fit with a covariance kernel very similar to what would be needed to perform a time-dependent fit of sawtooth data.



**Figure 2.30:** Probabilities of various levels of difference between the second derivatives of the  $n_e$  (left) and  $T_e$  (right) profiles at  $r/a = 0.5$ . The threshold  $c$  has units of  $10^{20} \text{ m}^{-3}$  for plot (a), keV for plot (d) and is dimensionless for the others. Similar figures can be made at any value of  $r/a$ ;  $r/a = 0.5$  was selected because it is in a region with appreciable rotation change but is away from the pathologies which  $a/L_{\nabla n_e}$  and  $a/L_{\nabla T_e}$  experience at the origin. The **blue solid lines** show the probability that the absolute value of the difference exceeds the threshold  $c$ , and so must go to one as  $c$  goes to zero. The **green dashed lines** show the probability that the difference is greater than  $c$  (i.e., the hollow rotation case has a higher value than the peaked rotation case), and the **red dotted lines** show the probability that the difference is less than  $-c$  (i.e., the hollow rotation case has a lower value than the peaked rotation case). These figures were produced using `ne_rotation.py` and `Te_rotation.py`.





**Figure 2.31:** Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1110329013, the sawtooth-suppressed lower hybrid current drive shot used to demonstrate 2D fitting of profile data. In the legend, “CTS” is the core Thomson scattering system and “ETS” is the edge Thomson scattering system. This figure was produced using `make_diagnostic_figure.py`.

adding four artificial approximately zero slope and value points at and just outside of the limiter location at each Thomson time point. It was discovered that the fit tended to end up with an unreasonably short spatial covariance length scale, and that nearly singular covariance matrices would tend to occur. This seemed to indicate that the diagnostic uncertainty estimates were not an adequate measure of the actual noise, so an additional homoscedastic noise component  $\sigma_n^2$  was added to the diagnostic uncertainty estimates. The noise level  $\sigma_n$  was inferred along with the hyperparameters of the covariance kernel. This was not necessary in the other fits presented previously in this section because those fits used the sample mean and standard deviation over long time windows to compute the value and uncertainty of

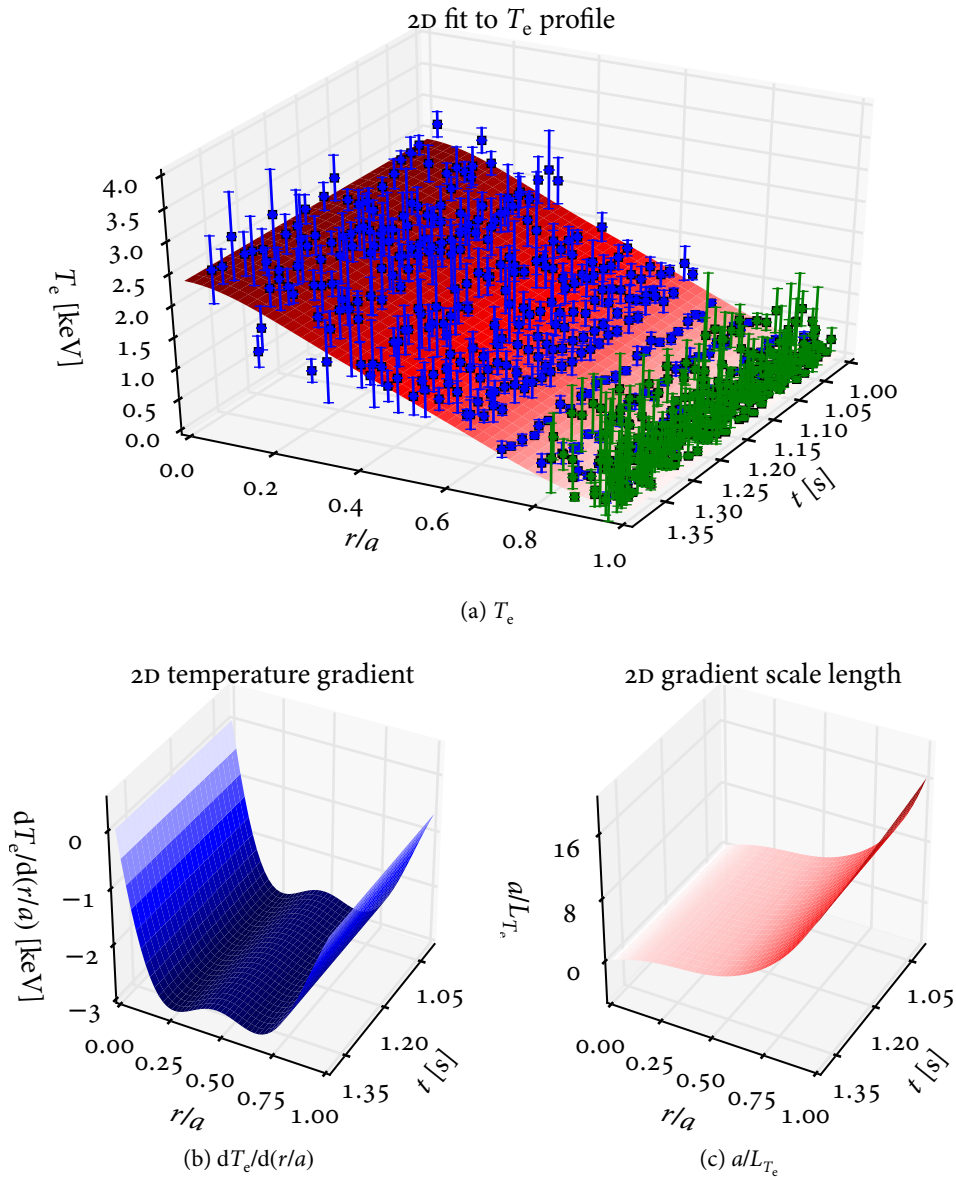
**Table 2.15:** Prior distributions for the hyperparameters used to fit the sawtooth-free data with 1D and 2D SE covariance kernels and additional homoscedastic noise.

Hyperparameter [units]	2D fit	1D fit
$\sigma_f$ [keV]	$\mathcal{U}(0, 10 \text{ keV})$	$\mathcal{U}(0, 10 \text{ keV})$
temporal $\ell_t$ [s]	$\mathcal{U}(0, 100 \text{ s})$	N/A
spatial $\ell$	$\text{Gamma}(27, 26)$ $m = 1.0, \sigma = 0.2$	$\text{Gamma}(27, 26)$ $m = 1.0, \sigma = 0.2$
$\sigma_n$ [keV]	$\mathcal{U}(0, 1 \text{ keV})$	$\mathcal{U}(0, 1 \text{ keV})$

a given channel, and therefore were not fully reliant on the diagnostic uncertainty estimates. In addition, the prior distribution for the spatial covariance scale length was made stronger than what was used for the fits presented previously in this section. The prior distributions used for the hyperparameters are given in [table 2.15](#).

To perform the inference, an affine-invariant ensemble sampler with 200 walkers and a proposal distribution width of  $a = 4$  was run for 250 samples, the first 150 samples were burned and the remaining samples were thinned by a factor of 100. The resulting posterior distribution for the hyperparameters is shown in [figure A.15](#), summary statistics of the posterior distribution are given in [table 2.16](#) and the profiles are given in [figure 2.32](#). Notice that when the temporal covariance length scale is longer than about 1 s the data over this 0.4 s window are so strongly correlated that all time variation is fairly negligible. Therefore, the fact that the posterior distribution for the temporal covariance length scale shown in [figure A.15](#) and summarized in [table 2.16](#) is concentrated entirely at large values indicates that the time variation over this portion of the discharge is very small. The distribution is not very informative about the temporal covariance length scale as 0.4 s of data is insufficient to decide between such long covariance length scales.

In order to make the effect of including all of the data in a bivariate fit more clear, the profile was evaluated at several time slices and compared to a fit using a 1D SE covariance function. For the 1D fits, no averaging was performed as just a single time slice's points were used. The diagnostic uncertainty estimates were used, and the fit was again allowed to infer additional homoscedastic noise. The profiles are given in [figure 2.33](#) and the posterior distributions for the hyperparameters are given in [figure A.16](#). The 2D fit infers extra structure in the gradient around midradius, and generally has a smaller uncertainty than the 1D fit. The relative uncertainties of the various fits are given in [table 2.17](#). The tighter uncertainty envelopes and extra structure in the 2D fit make sense because the 2D fit incorporates far more data and,



**Figure 2.32:** Data and 2D fitted profile for the sawtooth-free temperature data. In all three figures, the surfaces are colored red for positive values and blue for negative values. In (a) the blue points are the core Thomson scattering system and the green points are the edge Thomson scattering system. These figures were produced using LH\_2d.py.

as long as the assumption of a Gaussian process prior distribution for both the spatial and temporal variation is appropriate, all of the data can contribute to constraining the fit at a given time slice.

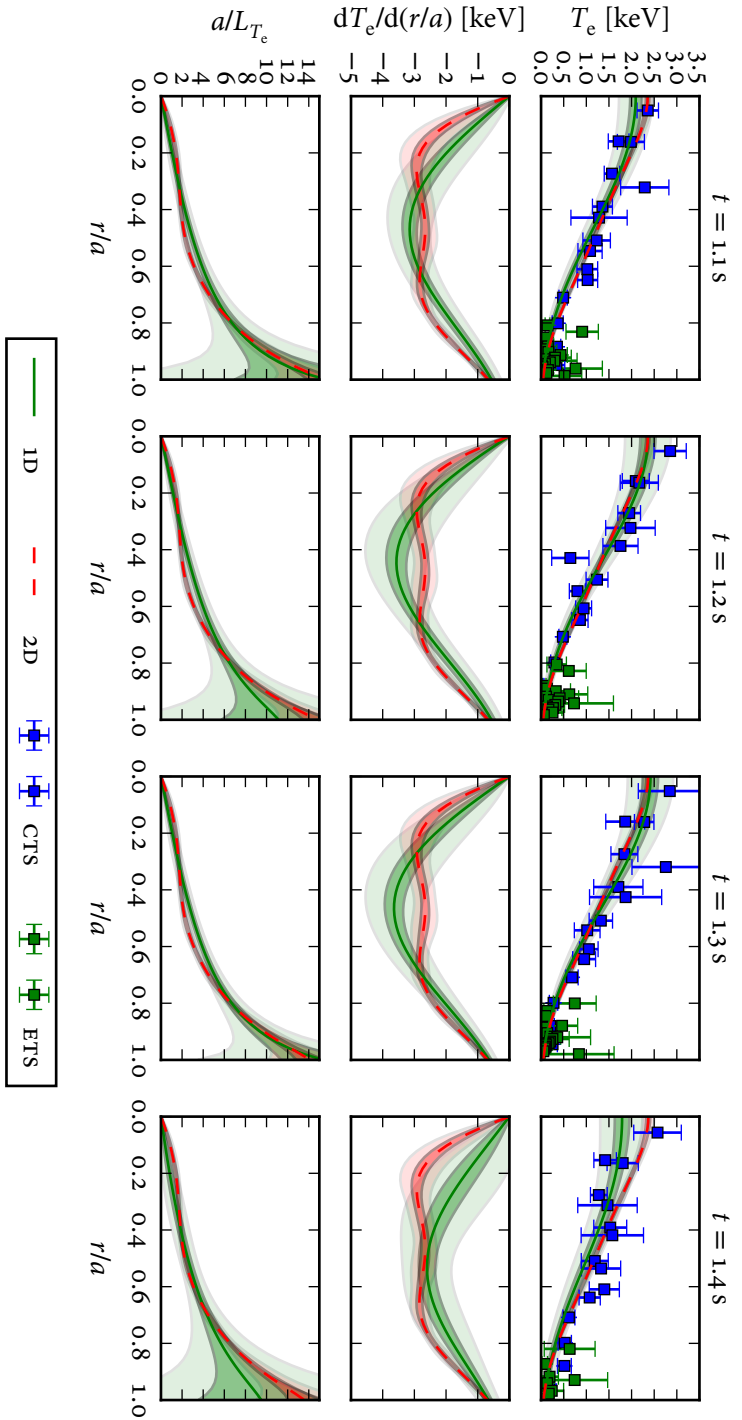
The difference between the uncertainties and structure of the 1D and 2D fits raises an interesting question regarding how to average the data over steady-state time windows like this. To this end, three ways of combining the multiple time slices were tested. First, all of the points from  $1.0 \text{ s} \leq t \leq 1.4 \text{ s}$  were kept in the fit. This ends up being about as computationally expensive as performing the 2D fit, and corresponds to the assumption that all of the time slices are perfectly correlated (consistent with the very long temporal covariance length scales inferred). Two types of averaging<sup>6</sup> were then tested: first, the weighted mean (where the diagnostic uncertainty estimates were used to determine the weights) of all of the data points in a given channel was used for the value and the weighted sample standard deviation was used for the uncertainty estimate. Second, the weighted mean was used and the uncertainty in this mean was computed using the error propagation equation and the diagnostic uncertainty estimates. As discussed in [section F.5](#), these represent two possible interpretations of the data and fit. The first will arrive at a finite uncertainty estimate  $\sigma$  for a given channel as more and more data are added, and is consistent with the assumption that there is some fundamental variability in  $T_e$  which the uncertainty estimate of the fit should represent. The second approach will go to zero as the number of samples increases, as it is the uncertainty in the mean which goes like  $\sigma/\sqrt{n}$ , where  $\sigma$  is the standard deviation in a given channel and  $n$  is the number of samples. This is consistent with the assumption that the only variation from time slice to time slice is from Gaussian noise in the measurement, and hence the variation can be averaged out through the accumulation of sufficient samples.

The fitted curves are given in [figure 2.34](#) and the posterior distributions for the hyperparameters inferred are given in [figure A.17](#). Interestingly, while the second averaging approach (referred to as “averaged,  $\sigma/\sqrt{n}$ ” in [table 2.16](#) and [table 2.17](#)) delivers a slightly tighter uncertainty envelope on  $T_e$  than the first approach (referred to as “averaged” in the tables), it has roughly the same envelopes for  $adT_e/dr$  and  $a/L_{T_e}$ . [Table 2.16](#) indicates that this may be a result of the fact that a higher level of additional homoscedastic noise  $\sigma_n$  was inferred for the  $\sigma/\sqrt{n}$  case. The fact that the inference seems to require a higher level of homoscedastic noise for this case again indicates that the diagnostic error bars have been underestimated.

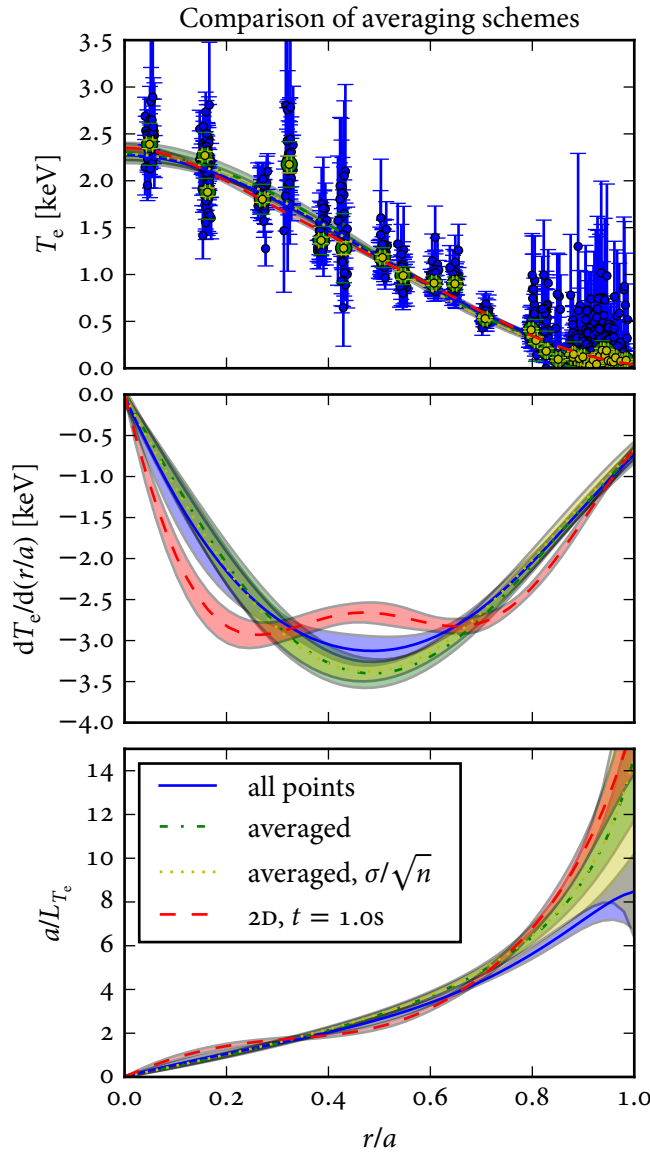
6. Refer to [section F.5](#) for more details on averaging schemes.

**Table 2.16:** Summary statistics for the posterior distributions of the hyperparameters for the 1- and 2D fits performed on the sawtooth free data.

Case	Parameter [units]	Mode	Mean	95% interval
2D SE	$\sigma_f$ [keV]	1.39	2.37	[ 0.896 , 6.38 ]
	temporal $\ell_t$ [s]	64.1	60.7	[15.9 , 97.9 ]
	spatial $\ell$	0.413	0.436	[ 0.360 , 0.531 ]
	$\sigma_n$ [keV]	0.0283	0.0282	[ 0.0221 , 0.0347]
1D SE, all points	$\sigma_f$ [keV]	1.36	3.84	[ 1.02 , 9.17 ]
	spatial $\ell$	0.403	0.768	[ 0.383 , 1.10 ]
	$\sigma_n$ [keV]	0.0648	0.0672	[ 0.0582 , 0.0769]
1D SE, averaged	$\sigma_f$ [keV]	1.56	3.21	[ 0.985 , 8.43 ]
	spatial $\ell$	0.696	0.780	[ 0.573 , 1.02 ]
	$\sigma_n$ [keV]	$5.92 \times 10^{-24}$	0.0101	[ 0.000390, 0.0379]
1D SE, averaged, $\sigma/\sqrt{n}$	$\sigma_f$ [keV]	1.68	3.38	[ 1.03 , 8.53 ]
	spatial $\ell$	0.718	0.801	[ 0.599 , 1.04 ]
	$\sigma_n$ [keV]	0.0919	0.0965	[ 0.0713 , 0.131 ]
1D SE, $t = 1.1$ s	$\sigma_f$ [keV]	1.45	3.20	[ 0.973 , 8.42 ]
	spatial $\ell$	0.688	0.787	[ 0.580 , 1.04 ]
	$\sigma_n$ [keV]	0.0197	0.0351	[ 0.00265 , 0.0905]
1D SE, $t = 1.2$ s	$\sigma_f$ [keV]	1.70	3.59	[ 1.07 , 8.74 ]
	spatial $\ell$	0.689	0.783	[ 0.582 , 1.03 ]
	$\sigma_n$ [keV]	0.0456	0.0547	[ 0.0236 , 0.102 ]
1D SE, $t = 1.3$ s	$\sigma_f$ [keV]	1.63	3.41	[ 1.05 , 8.53 ]
	spatial $\ell$	0.684	0.763	[ 0.567 , 1.01 ]
	$\sigma_n$ [keV]	$2.16 \times 10^{-9}$	0.00966	[ 0.000200, 0.0388]
1D SE, $t = 1.4$ s	$\sigma_f$ [keV]	1.27	3.07	[ 0.817 , 8.31 ]
	spatial $\ell$	0.675	0.776	[ 0.522 , 1.04 ]
	$\sigma_n$ [keV]	0	0.0412	[ 0.000639, 0.174 ]



**Figure 2.33:** Profiles from the sawtooth-free data evaluated at various time points. The red dashed curves are from the Gaussian process with a 2D SE covariance kernel applied to the entire data set and the green solid curves are from a Gaussian process with a 1D covariance kernel applied to a single time slice. Some time frames such as  $t = 1.4$  s have a very large error bar on the innermost data point, which can lead to an under-prediction of the core  $T_e$  when fitting just that frame of data. By using the temporal correlation with adjacent times, the 2D fit is not afflicted by this issue. Because the 2D fit uses so many more data points (and has such a long temporal covariance length scale) it ends up with more structure in the core and tighter uncertainty envelopes. These figures were produced using `LH_2d.py`.



**Figure 2.34:** Data and fitted curves subject to three ways of combining the data from multiple time slices. Only the  $\pm 1\sigma$  envelopes are shown. The schemes shown are as follows: “all points” (blue solid) refers to simply including all of the time points in the 1D Gaussian process, and lets the Gaussian process do the averaging. Note that there are many outliers at the edge which go away with averaging. “Averaged” (green dash-dot) uses the weighted mean and weighted standard deviation within a given channel. “Averaged,  $\sigma/\sqrt{n}$ ” (black dotted) uses the weighted mean and computes the uncertainty in the mean through uncertainty propagation. Since both of the averaging schemes use the same weighted mean, the green and black data points overlay, with the difference being that the “averaged,  $\sigma/\sqrt{n}$ ” scheme (black points) produces tighter error bars than the “averaged” scheme (green points). Also shown is the 2D fit at  $t = 1$  s (red dashed). All three of the averaging schemes produce comparable results, whereas the 2D fit infers more structure in the gradient. These figures were produced using `LH_2d.py`.

**Table 2.17:** Median relative uncertainties over the region  $0 \leq r/a \leq 1$  for the 1- and 2D fits performed on the sawtooth-free data. Relative uncertainties are given as percentages.

$t$ [s]	Case	$T_e$	$dT_e/d(r/a)$	$a/L_{T_e}$
[1.0, 1.4]	all points	1.72	5.02	5.05
	averaged	3.96	6.09	6.05
	averaged, $\sigma/\sqrt{n}$	3.01	6.35	6.74
1.1	2D SE	1.49	4.27	4.32
	1D SE	5.92	9.60	9.63
1.2	2D SE	1.45	4.25	4.26
	1D SE	6.88	10.5	8.81
1.3	2D SE	1.53	4.27	4.33
	1D SE	6.68	9.14	7.65
1.4	2D SE	1.73	4.32	4.50
	1D SE	8.98	13.9	19.2

### 2.5.5 Fitting of pedestal data using the mtanh mean function

The pedestals of H- and I-mode plasmas are often characterized using a modified hyperbolic tangent function [416, 417]:

$$z = \frac{x_0 - x}{\delta} \quad (2.79)$$

$$\text{mtanh}(\alpha, z) = \frac{(1 + \alpha z)e^z - e^{-z}}{e^z + e^{-z}} \quad (2.80)$$

$$y = \frac{h+b}{2} + \frac{h-b}{2} \text{mtanh}(\alpha, z) = \frac{h+b}{2} + \frac{h-b}{2} \frac{(1 + \alpha z)e^z - e^{-z}}{e^z + e^{-z}}, \quad (2.81)$$

where  $y$  is the quantity to be fit,  $x$  is the coordinate the fit is to be performed against,  $x_0$  is the midpoint of the pedestal,  $\delta$  is the half-width of the pedestal,  $\alpha$  sets the slope of the profile inside of the pedestal,  $h$  is the height of the pedestal and  $b$  is the foot of the pedestal. The gradient of the profile is [417]:

$$\frac{dy}{dx} = -\frac{h-b}{2\delta} \frac{1 + \alpha(1 + 2z + e^{2z})/4}{\cosh^2 z}. \quad (2.82)$$

Using the full formulation of GPR given in section 2.3.9 it is possible to include this mean function to introduce non-stationarity to allow the edge to be fit simultaneously



with the core. This is often conceptualized as using a zero-mean Gaussian process to fit the residuals from fitting the mean curve to the data, though it will be shown in this section that care is required when interpreting the parameters of the mean function obtained in this manner.

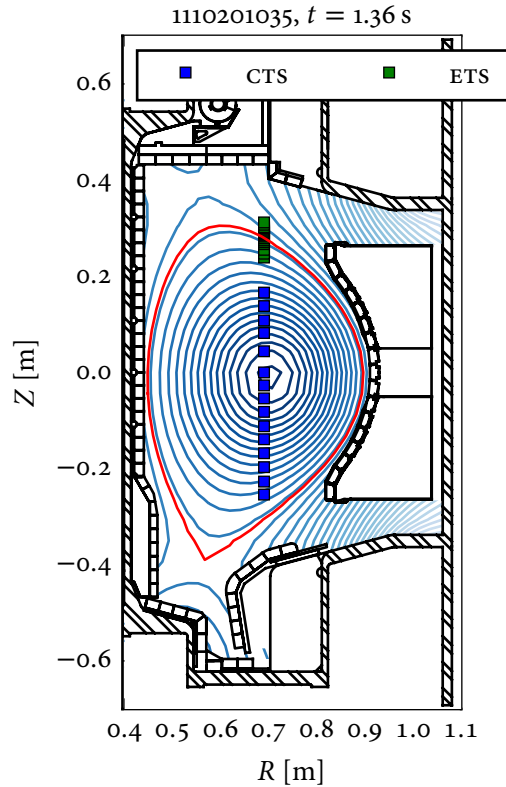
To demonstrate this capability, the density profile of an H-mode plasma (Alcator C-Mod shot 1110201035) was fit using an SE covariance kernel with the mtanh mean function<sup>7</sup> given in [equation \(2.81\)](#). In addition, the fit was repeated using the Gibbs covariance kernel from [equation \(B.73\)](#) with the tanh covariance length scale function from [equation \(B.78\)](#), and using the mtanh function alone. Note that `gptools` can be used for Bayesian nonlinear least-squares fitting: when  $k(\mathbf{x}_i, \mathbf{x}_j) = 0$  and  $\mathbf{T} = \mathbf{I}$ , [equation \(2.61\)](#) reduces to

$$\ln f_{Y|\Theta} = -\frac{1}{2}(\mathbf{y} - \mathbf{m}(\mathbf{X}|\boldsymbol{\theta}))^\top \boldsymbol{\Sigma}_n^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{X}|\boldsymbol{\theta})) - \frac{1}{2} \ln |\boldsymbol{\Sigma}_n| - \frac{n}{2} \ln 2\pi, \quad (2.83)$$

which is simply the log-likelihood for a multivariate normal  $\mathcal{N}(\mathbf{m}(\mathbf{X}|\boldsymbol{\theta}), \boldsymbol{\Sigma}_n)$  with (potentially correlated and/or heteroscedastic) observational noise  $\boldsymbol{\Sigma}_n$  and mean function  $\mathbf{m}(\mathbf{x}|\boldsymbol{\theta})$  – exactly the likelihood distribution used for fitting of the function  $m(\mathbf{x}|\boldsymbol{\theta})$  to the data  $\mathbf{y}(\mathbf{X})$ .

In order to align the temperature pedestal with the LCFS, the edge Thomson scattering points were shifted down by  $\Delta Z = -13$  mm such that  $T_e(\psi_n = 1) \approx 100$  eV [418]. The data from both the core and edge Thomson scattering systems were averaged over the constant-density period  $1.35 \text{ s} \leq t \leq 1.5 \text{ s}$ . Over this time window the shot had  $I_p = 500$  kA,  $B_T = 5.4$  T and was heated with 4 MW of ICRF power. The equilibrium shape and diagnostic locations are shown in [figure 2.35](#). The fit was performed as a function of normalized poloidal flux  $\psi_n$  in order to obtain the pedestal half-width  $\delta$  in terms of the conventional units of normalized poloidal flux. In line with the discussion of [section 2.3.5.1](#), a zero-slope constraint was *not* included at  $\psi_n = 0$ . Furthermore, because the mtanh fit includes a parameter for the pedestal foot, constraints on the value and slope at the limiter were only included for the fit using the Gibbs covariance kernel with tanh covariance length scale function. All of the fits were performed using MCMC, but different sampler proposal distribution widths were found to be necessary for each fit. The parameters used for the MCMC samplers are given in [table 2.18](#) and the prior distributions used for the parameters of the mtanh mean function and the hyperparameters of the covariance kernels are given in [table 2.19](#). The fits are shown in [figure 2.36](#), plots of the posterior distributions for the parameters and hyperparameters are given in [figure 2.37](#) through [figure 2.39](#), summary statistics for the posterior distributions

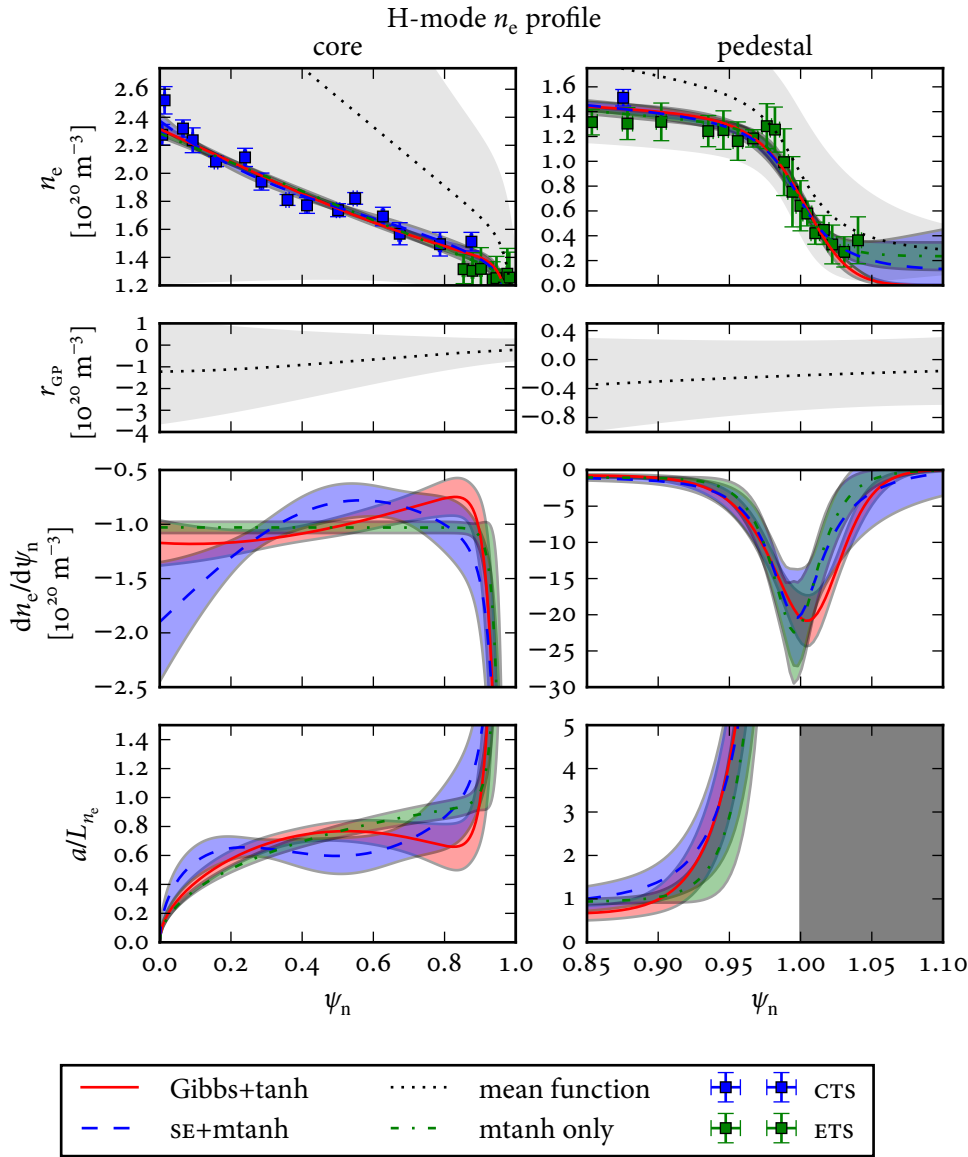
7. Note that in the rest of this section the term “mtanh mean function” will be taken to refer to the full profile function  $y(z(x))$  given in [equation \(2.81\)](#), not the  $\text{mtanh}(\alpha, z(x))$  function itself.



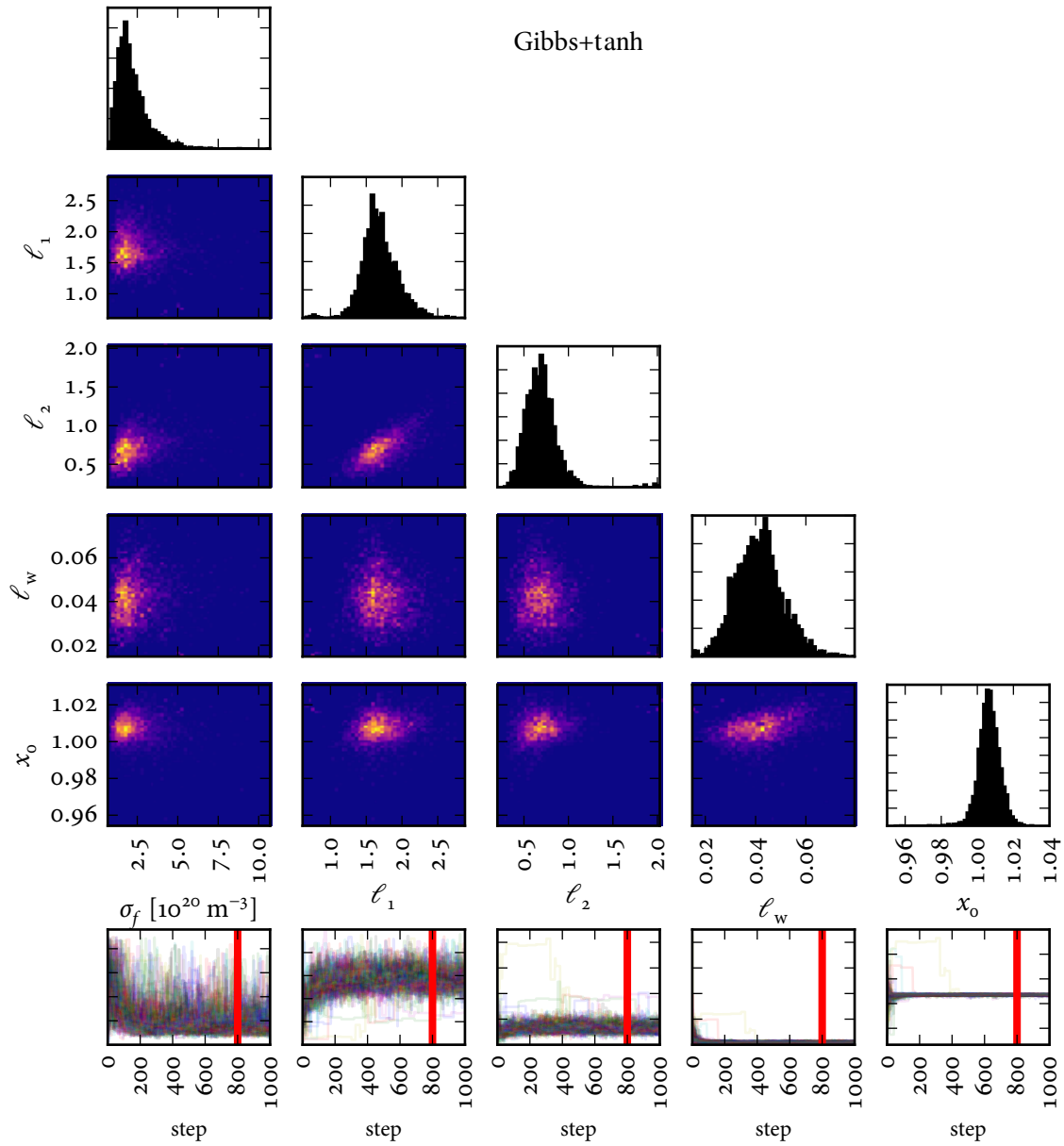
**Figure 2.35:** Magnetic equilibrium reconstruction and diagnostic locations for Alcator C-Mod shot 1110201035, the H-mode shot used to demonstrate fitting with and without the  $\text{mtanh}$  mean function. In the legend, “CTS” refers to the core Thomson scattering system and “ETS” refers to the edge Thomson scattering system. This figure was produced using `make_diagnostic_figure.py`.

for the parameters and hyperparameters are given in [table 2.20](#) and summaries of the uncertainties in the fits are given in [table 2.21](#). As illustrated in [table 2.21](#) and [figure 2.36](#), the fits with the Gibbs covariance kernel with  $\text{tanh}$  covariance length scale function and the  $\text{SE}$  covariance kernel with  $\text{mtanh}$  mean function both fit the data quite well and give comparable uncertainty estimates. While the  $\text{mtanh}$  function on its own fits the pedestal well, it is too inflexible to fit the core data well. This leads to both a poor fit to the core data as well as an underestimation of the uncertainty in the fit.

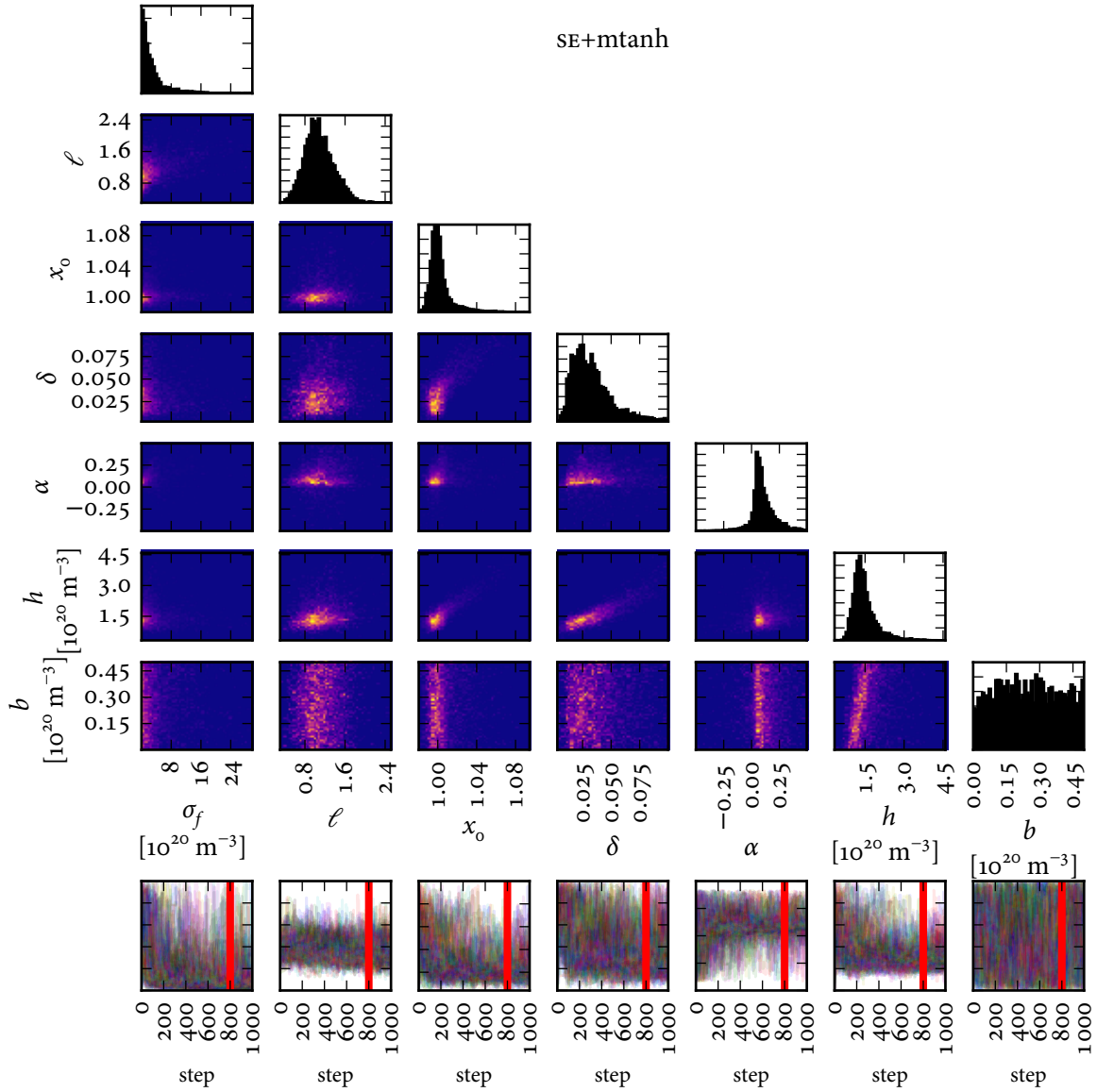
The results given in [table 2.20](#), [figure 2.38](#) and [figure 2.39](#) indicate one cautionary note about this technique: while all three techniques yielded acceptable fits to the



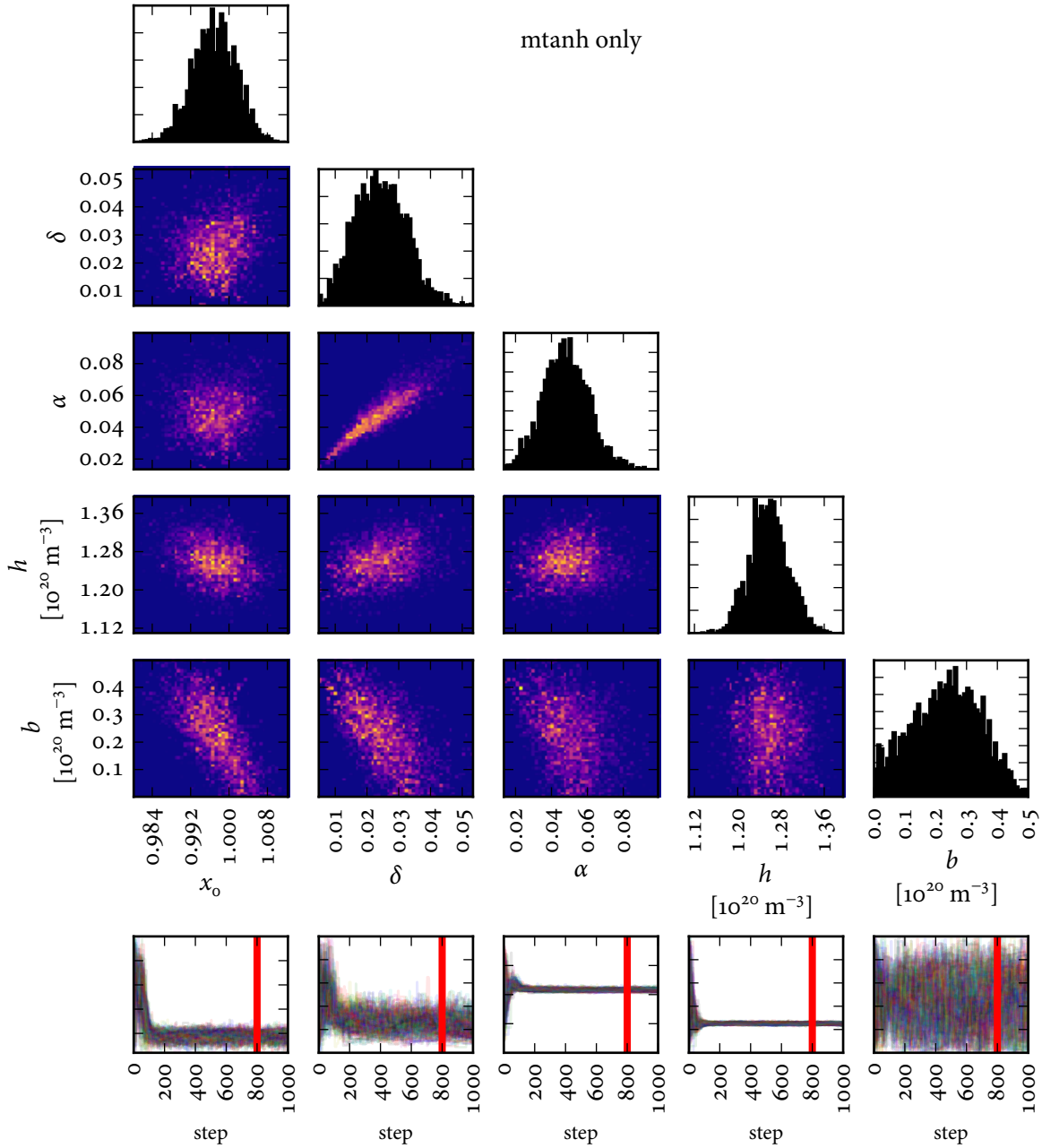
**Figure 2.36:** Plots of the core and pedestal portions of the fits using the Gibbs covariance kernel with tanh covariance length scale function (red solid), the SE covariance kernel with mtanh mean function (blue dashed), and only the mtanh function (green dash-dot). To make the plot more legible, only the  $\pm 1\sigma$  uncertainty envelopes are shown. In each column, the second panel from the top shows the residual from the fit using the SE covariance kernel with the mtanh mean function. The residual and the mean function are strongly correlated, which is why the uncertainty envelopes on the blue SE+mtanh curves are so much smaller than the uncertainty envelopes on the black mean function and GP residual curves. All three cases provide similar fits, though the fit using only the mtanh function misses a considerable amount of structure in the core, as would be expected given its simplicity in that region. This figure was produced using `ne_mtanh.py`.



**Figure 2.37:** Marginal posterior distributions for the hyperparameters of the Gibbs covariance kernel with tanh covariance length scale function for the fit to the H-mode density profile. This figure was produced using `ne_mtanh.py`.



**Figure 2.38:** Marginal posterior distributions for the parameters of the mtanh mean function and the hyperparameters of the SE covariance kernel for the fit to the H-mode density profile. When the Gaussian process with SE covariance kernel is present to fit for the residuals, the pedestal foot  $b$  is completely unconstrained and has a marginal posterior distribution which very strongly resembles the uniform prior distribution used. In this case  $h$  and  $b$  are positively correlated. This figure was produced using `ne_mtanh.py`.



**Figure 2.39:** Marginal posterior distributions for the parameters of the mtanh function fit to the H-mode density profile. Without the Gaussian process to fit the residuals, the pedestal foot  $b$  is now much better constrained by the data, and the clear positive correlation with  $h$  seen in figure 2.38 is no longer evident. Note that the posterior distribution for  $b$  is clearly non-Gaussian, which means that conventional fitting techniques which estimate the uncertainties in the parameters from the inverse Hessian of the log-likelihood may underestimate the uncertainty in  $b$ . This figure was produced using `ne_mtanh.py`.

**Table 2.18:** Parameters of the affine-invariant ensemble samplers used to fit the H-mode  $n_e$  profile with and without the mtanh mean function. “Burn” is the number of samples discarded from the start of each chain, “Thin” is the factor by which each chain’s remaining samples were thinned before computing profiles.

Case	Proposal width, $a$	Walkers	Samples per walker	Burn	Thin
Gibbs+tanh	16	200	1000	800	100
SE+mtanh	4	200	1000	800	100
mtanh only	16	200	1000	800	100

**Table 2.19:** Prior distributions for the parameters and hyperparameters of the mean functions and covariance kernels used when fitting the H-mode  $n_e$  profile.

(Hyper)parameter	Prior distribution
Signal variance, $\sigma_f$	$\mathcal{U}(0, 11.1 \times 10^{20} \text{ m}^{-3})$ for Gibbs+tanh, $\mathcal{U}(0, 30 \times 10^{20} \text{ m}^{-3})$ for SE+mtanh
Core covariance length scale, $\ell, \ell_1$	Gamma(13, 12) ( $m = 1.0, \sigma = 0.3$ )
Gibbs+tanh edge covariance length scale, $\ell_2$	Gamma(5.8, 9.7) ( $m = 0.5, \sigma = 0.25$ )
Gibbs+tanh transition width, $\ell_w$	Exp(10) ( $m = 0, \sigma = 0.1$ )
Gibbs+tanh pedestal center, $x_o$	Gamma(102, 101) ( $m = 1.0, \sigma = 0.1$ )
mtanh pedestal midpoint, $x_o$	$\mathcal{U}(0.98, 1.1)$
mtanh pedestal half-width, $\delta$	$\mathcal{U}(0, 0.1)$
mtanh core slope factor, $\alpha$	$\mathcal{U}(-0.5, 0.5)$
mtanh pedestal height, $h$	$\mathcal{U}(0, 5 \times 10^{20} \text{ m}^{-3})$
mtanh pedestal foot, $b$	$\mathcal{U}(0, 0.5 \times 10^{20} \text{ m}^{-3})$

data, the use of the SE covariance kernel changed the posterior distribution for the mtanh fit’s parameters relative to the case where only the mtanh function is used to fit the data. This is further illustrated by the dotted black curves in [figure 2.36](#), which show the mean and standard deviation of the marginalized mean curve<sup>8</sup>  $m(\mathbf{x}_*|\mathbf{y}) = \int m(\mathbf{x}_*|\boldsymbol{\theta})f_{\boldsymbol{\theta}|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}$  and the marginalized Gaussian process residual  $r_{\text{GP}}(\mathbf{x}_*|\mathbf{y}) = \int (y_*(\mathbf{x}_*|\mathbf{y}, \boldsymbol{\theta}) - m(\mathbf{x}_*|\boldsymbol{\theta}))f_{\boldsymbol{\theta}|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}$ . The marginalized mean curve is systematically higher than the data, and is drawn down to produce a good fit by the systematically negative residual. Furthermore, the very large variance in each of the

8. The uncertainties in the marginalized mean curve and Gaussian process residual were computed from the MCMC results using the same technique with the laws of iterated expectations and total covariance as described in [section 2.3.7](#).

**Table 2.20:** Summary statistics for the posterior distributions of the parameters and hyperparameters of the fits to the H-mode  $n_e$  profile with and without the mtanh mean function.

	Parameter [units]	Mode	Mean	95% interval
Gibbs+tanh	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	1.54	2.22	[ 0.982 , 4.94 ]
	$\ell_1$	1.62	1.69	[ 1.26 , 2.25 ]
	$\ell_2$	0.655	0.695	[ 0.413 , 1.07 ]
	$\ell_w$	0.0395	0.0414	[ 0.0242 , 0.0614]
	$x_o$	1.01	1.01	[ 0.995 , 1.02 ]
	SE+mtanh	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	$5.91 \times 10^{-22}$	3.25
$\ell$		1	1.09	[ 0.577 , 1.70 ]
$x_o$		0.997	1.00	[ 0.989 , 1.04 ]
$\delta$		0.0176	0.0329	[ 0.0107 , 0.0790]
$\alpha$		0.0386	0.0985	[ -0.128 , 0.355 ]
$h$ [ $10^{20} \text{ m}^{-3}$ ]		1.25	1.48	[ 0.864 , 2.73 ]
$b$ [ $10^{20} \text{ m}^{-3}$ ]		0.292	0.251	[ 0.0187 , 0.489 ]
mtanh only		$x_o$	0.997	0.997
	$\delta$	0.0176	0.0243	[ 0.00976, 0.0411]
	$\alpha$	0.0386	0.0481	[ 0.0228 , 0.0744]
	$h$ [ $10^{20} \text{ m}^{-3}$ ]	1.25	1.26	[ 1.19 , 1.33 ]
	$b$ [ $10^{20} \text{ m}^{-3}$ ]	0.292	0.233	[ 0.0230 , 0.432 ]

**Table 2.21:** Median relative uncertainties over the region  $0 \leq r/a \leq 1$  for fits to the H-mode  $n_e$  profile with and without the mtanh mean function. Relative uncertainties are given as percentages.

Case	$n_e$	$dn_e/d\psi_n$	$a/L_{n_e}$
Gibbs+tanh	1.2	12	12
SE+mtanh	1.4	19	20
mtanh only	0.99	5.0	5.4

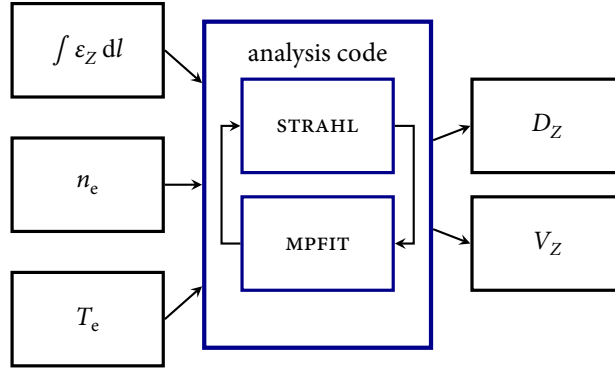


components is cancelled out by their covariance when they are added together. Thus, using a Gaussian process with an mtanh mean function will yield an acceptable fitted curve, but not necessarily a useful estimate of the pedestal parameters themselves: the approach presented here combining the mtanh mean function with the SE covariance kernel is an effective way of performing a nonstationary fit to the data, but not an effective way of estimating the specific pedestal parameters.

Note that the MAP estimate given in the “Mode” column of [table 2.20](#) has the same parameters for the mtanh function for both the SE covariance kernel with mtanh mean function and mtanh function only cases, but that  $\sigma_f$  is vanishingly small. This corresponds to the posterior mode being the pure mtanh function. While this means that there are only very small residuals between the mtanh curve and the data and hence supports the use of the mtanh function to produce summaries of H-mode pedestals, it also illustrates that the MAP estimate on its own is insufficient to produce reliable uncertainty estimates when using a parametric mean function: when  $\sigma_f$  is small, all of the posterior uncertainty is in the parameters of the mean function.

## 2.6 Application of Gaussian process regression to impurity transport measurements

This section considers the propagation of the profile uncertainties obtained in [section 2.5.1](#) through the analysis workflow used to obtain experimental impurity transport coefficients in Alcator C-Mod. This type of sampling can be extended to any analysis code that needs profile inputs, such as a power balance code used to compute experimental heat fluxes [419]. The approach used to obtain the impurity transport coefficients is described in detail in [18, 19, 55]. In the shot analyzed in [section 2.5.1](#) (Alcator C-Mod shot 1101014006), calcium (a non-intrinsic, non-recycling impurity) was injected four times during the stationary part of the discharge using a multi-pulse laser blow-off impurity injector [54, 55]. The temporal and spatial evolution of the He-like calcium was measured using the HIREX-SR x-ray imaging crystal spectrometer [360, 420] and a line-integrated view of the Li-like calcium was measured with the XEUS extreme ultraviolet spectrometer [421]. The STRAHL code [276] takes as input the  $n_e$  and  $T_e$  profiles plus guesses for the transport coefficient profiles  $D$  and  $V$  from the assumed impurity flux  $\Gamma_Z = -D\nabla n_Z + Vn_Z$  and yields as output the time evolution of the impurity density profile  $n_Z(R, t)$ . A synthetic diagnostic is used to obtain the line-integrated emissivity from this result which is then compared to the measured time evolution for He-like calcium observed with HIREX-SR and Li-like calcium observed with XEUS. The guesses for  $D$  and  $V$  are then iterated upon using the MPFIT Levenberg-Marquardt minimizer [422, 423] to find the choices



**Figure 2.40:** Schematic of the STRAHL analysis. Realizations of the  $n_e$  and  $T_e$  profiles are fed into the “analysis code” along with information on the impurity density  $n_Z$  in the form of the line-integrated measurements  $\int \epsilon_Z dl$  from several spectroscopic diagnostics. The MPFIT optimizer iterates upon the  $D_Z$  and  $V_Z$  profiles fed into STRAHL to find the best  $D_Z, V_Z$  for a given realization of  $n_e, T_e$ . The inputs  $n_e, T_e$  are then randomly sampled in the manner indicated in figure 2.2 to yield an ensemble of realizations of  $D_Z, V_Z$  from which the mean profiles and their uncertainties can be computed.

that produce emission time histories that best match the experimental observations. This process is shown schematically in figure 2.40. As noted in [18], the results are believed to be most sensitive to the uncertainties in the  $n_e$  and  $T_e$  profiles. Therefore, to quantify the uncertainty in the output  $D$  and  $V$  profiles, the code is run multiple times with random samples of the  $n_e$  and  $T_e$  profiles, in the manner discussed in section 2.2 and shown schematically in figure 2.2. Note, however, that there are substantial shortcomings with this approach which are covered in chapter 3. As such, this section should be viewed more as an illustration of propagation of uncertainties from profiles fit with GPR than a rigorous result on impurity transport.

The previous work fit the data using splines and obtained random samples by manually re-fitting the data after perturbing the points according to their uncertainties, a process which required considerable manual intervention. The present work improves on this through the use of GPR. The shape of the spline fits has already been shown in figure 2.20, and is mostly similar to that of the GPR fits. Sampling from the GPR fit was conducted in two ways. The simplest approach tested is to take the MAP estimate  $\hat{\theta}^{\text{MAP}}$  for the hyperparameters, then draw samples from  $f_{Y_*|Y, \theta}(y_*|y, \theta = \hat{\theta}^{\text{MAP}})$  according to equation (2.52). The more thorough approach tested is the fully Bayesian hierarchical sampling scheme using MCMC described at the end of section 2.3.8. In either case, the sampling would sometimes yield samples

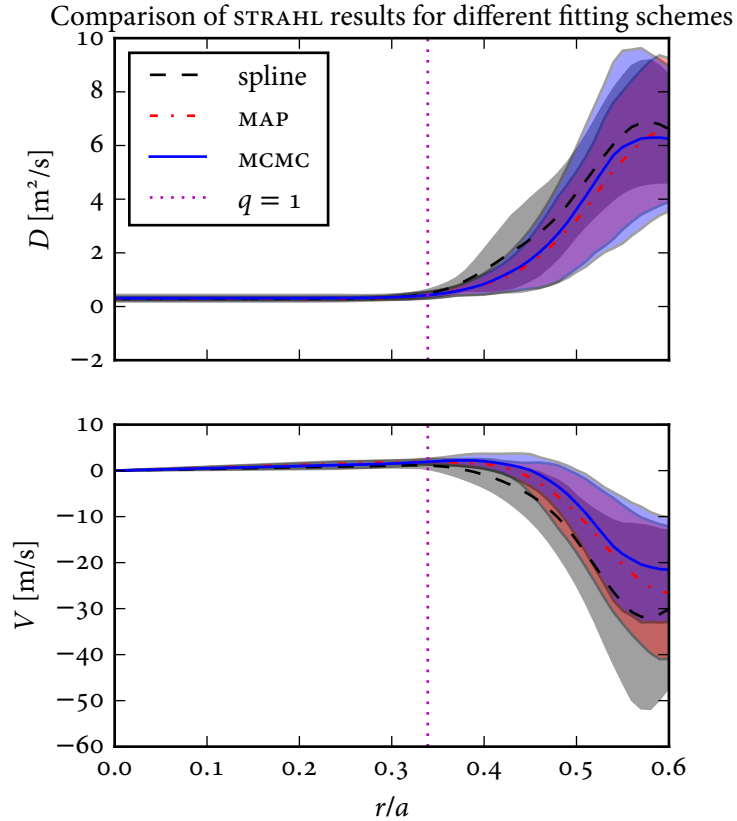
that exhibited nonphysical behavior such as nonmonotonicity or negativity. Therefore, each of the samples was checked at each of the evaluation points and the sample was thrown out if  $y < 0$  or  $y' > 0$  at any point within  $0 \leq r/a \leq 1$ . In either case, 65 samples that satisfied the constraints were obtained and propagated through STRAHL. It is important to note that, once appropriate prior distributions for the hyperparameters have been selected, this entire process proceeds in a completely automated manner – the number of samples run to obtain the accuracy desired from the Monte Carlo study is limited only by how much computer time the user is willing to devote to the STRAHL analysis. This is in contrast with the spline-based approach, where each sample required laborious hand-tuning of the spline parameters to produce an acceptable fit to each set of perturbed data points.

The resulting  $D$  and  $V$  profiles are given in [figure 2.41](#). The mean  $D$  profiles from the two GPR-based approaches show a slight offset from the previous spline result, but the uncertainty envelopes basically overlap over most of the region of interest. The  $V$  profile, however, shows a stronger systematic difference between the GPR-based approaches and the previous spline result. While this difference is not substantially outside of the  $\pm 1\sigma$  error bars, it is believed to be a result of the fact that the GPR-based  $T_e$  profile has a mean which is, on average, about 11% lower than the mean spline profile.

It is of interest to note that the MAP and MCMC treatments of the hyperparameters yielded approximately the same results for both the means and the uncertainties of  $D$  and  $V$ . This can be expected from the small change in uncertainty for the values of  $n_e$  and  $T_e$  noted in [table 2.9](#) and the fact that only the value and not the gradient of these background profiles enters the calculation. Therefore, for this case it is possible to use the simpler MAP calculation, which enables the use of advanced sampling strategies such as Latin hypercube sampling [355], quasi-Monte Carlo sampling [291] or sparse quadrature [338] to further improve the rate of convergence, though these have yet to be applied to this problem.

## 2.7 Summary of profile fitting with Gaussian process regression

The chapter has presented the use of Gaussian process regression (GPR) for fitting smooth curves to noisy, discrete observations of plasma profiles and then subsequently propagating the uncertainty in the fitted curve through an analysis code. Fits with various stationary and nonstationary covariance kernels were performed on synthetic data and two promising new nonstationary covariance kernels were identified: the Gibbs covariance kernel with tanh covariance length scale function



**Figure 2.41:**  $D$  and  $V$  profiles for spline fits (black dashed), sampling from the MAP estimate (red dash-dot) and from hierarchical sampling with MCMC (blue solid). The uncertainty envelopes are  $\pm 1\sigma$ . The profile is only shown over  $0 < r/a < 0.6$  because the results are not trusted outside of this region. This is an updated version of a figure which originally appeared in [305]. This figure was produced using `strahl_compare_post_new.py`.

and the SE covariance kernel with 3-knot I-spline input warping function. Profile fitting with real data from Alcator C-Mod demonstrated the basic fit including rigorous uncertainty estimates on the gradients and gradient scale lengths, the use of TCI data to constrain the fit, the inference of second derivative profiles to test theories of momentum transport, the ability to fit data as a function of both space and time and the capability to combine GPR with a parametric mean function in order to infer H-mode pedestal parameters. While the uncertainty propagation example shown here involved propagation of the uncertainty in the background  $n_e$ ,  $T_e$  profiles through

an analysis code to obtain impurity transport coefficients, this approach is extremely general and can deliver benefits in any situation where gradients or profile fits are needed, particularly within the context of gyrokinetic validation. This approach was shown to have considerable advantages over the more traditional use of splines in the context of fitting profiles and propagating uncertainty through analysis and simulation codes in the following respects:

- The fit proceeds automatically using standard statistical procedures instead of manual, subjective hand-tuning.
- This flexible, non-parametric approach does not overly constrain the shape of the fit.
- It is trivial to apply this approach to multivariate data of arbitrary dimension.
- Line-integrated or otherwise transformed data can be combined with local measurements in the fit in a statistically rigorous manner.
- The method provides an estimate of the uncertainty on the fitted value *and* derivative(s) without additional work.
- It is straightforward to draw random samples and easy to reduce the dimensionality of the space to be sampled in order to apply advanced techniques to improve the rate of convergence of uncertainty propagation.

Two approaches for handling the hyperparameters that dictate the nature of the fit were compared: the MAP estimate provides a point estimate for the hyperparameters and is faster and simpler to work with, while the use of MCMC to marginalize over the hyperparameters provides a more rigorous accounting of the uncertainty hiding in the hyperparameters of the fit. These two approaches give similar results for the uncertainty in the value of the fit, but differ substantially for the uncertainty in the gradient. Therefore, it is necessary to use the more complicated MCMC-based marginalization when working with processes that are strongly sensitive to gradients. These two approaches were applied to the task of inferring the impurity transport coefficients  $D$  and  $V$  from experimental data, and yielded results that were comparable to what was obtained previously using splines. But the new results were obtained in a far more automated manner and demonstrated far more convincing convergence. It was verified that the results for  $D$  and  $V$  do not depend on the gradients of the background profiles, and hence the use of the simpler MAP estimate is sufficient. Open-source software to perform GPR with gradient constraints and predictions has been developed and is available for use by anyone needing to fit

smooth curves, estimate uncertainties in gradients and efficiently produce samples for use in uncertainty propagation [306, 307]. Further use of the GPR-based fitting and sampling approaches presented here has the potential to improve the quality and trustworthiness of uncertainty estimates on both profile fits and code outputs, while simultaneously reducing the time for analysis both by reducing the amount of manual intervention necessary to produce fits and by improving the convergence of uncertainty propagation calculations.

# 3

## Inference of experimental impurity transport coefficient profiles

### 3.1 Introduction and motivation: issues with existing approaches

The measurement of impurity transport coefficients is a very challenging inference problem, with attempts at performing these measurements going back many decades (see [section 1.3](#) for a review of the relevant literature). This thesis builds on the work at Alcator C-Mod which has attempted to infer mid- $Z$  (i.e., calcium,  $Z = 20$ ) transport in L-mode plasmas [[18](#), [19](#), [55](#), [305](#)]. The approach used in the previous work (both at Alcator C-Mod and other facilities) suffered from a number of issues which this chapter attempts to highlight, explain and rectify. In particular, the following issues are present in the previous work:

- The previous work claims that the transport coefficients are unconstrained outside of  $r/a \gtrsim 0.6$  because of low signal strength on the x-ray imaging crystal spectrometer (XICS) chords viewing outside of that radius, but the computed uncertainties in that region are *not* consistent with this assumption. This is likely a result of some combination of the following three issues:
  1. The observations from the core where there is high signal strength might help to constrain the transport coefficients in the outer region more than expected.
  2. The basis functions for the  $D$  and  $V$  profiles may be too inflexible to allow the Monte Carlo sampling procedure described in [section 2.6](#) to capture the full uncertainty in the inferred transport coefficients. This is exacerbated by the fact that the previous approach did not provide a

rigorous procedure for selecting the appropriate level of complexity in the basis functions.

3. The Monte Carlo sampling procedure previously used to propagate the uncertainty in  $n_e$  and  $T_e$  may be missing a considerable amount of uncertainty in the parameters of the  $D$  and  $V$  profiles. This is very likely as the former procedure did not account for the width of the posterior distribution when maximizing the likelihood of the data, thereby effectively ignoring the uncertainties in the spectroscopic measurements and the possibility that there are multiple sets of parameters which might describe the data equally well.
- Based on the intuitive notion that a high time resolution is necessary to separate  $D$  from  $V$ , the previous analysis combined data from three injections at different phases with respect to the XICS sampling rate to yield a higher effective time resolution. Because each injection is at a different phase with respect to the sawteeth, this approach makes it impossible to properly account for the effects of sawteeth. At best this procedure will yield some form of “sawtooth-averaged” transport coefficients. But unless the measured transport coefficients are to be compared to a simulation which includes the effects of sawteeth and is post-processed in a similar manner, this measurement will be questionable for use in validating simulations of impurity transport. It will be shown in [section 3.4](#) and [section 3.6.3](#) that the intuition that high time resolution is necessary is incorrect, and 6 ms sampling of a single injection should be sufficient as long as a reasonable *spatial* resolution is obtained.
  - The optimization problem to find the parameters describing  $D$  and  $V$  is very poorly behaved in at least two ways:
    1. It was found that roundoff error in the forward model requires very large step sizes when computing derivatives using finite differences. This calls the use of derivative-based optimizers such as the Levenberg-Marquardt algorithm used for the previous work into question, both in terms of using the derivatives to guide the search and in terms of using the estimated inverse Hessian matrix to approximate the parameter uncertainties.
    2. Finding a good initial guess for the optimizer is difficult: many points in the parameter space will fail to make much progress, implying that there are large areas of the parameter space which give an equally poor fit to the data such that the optimizer cannot get out of these plateaus. For points that do find an acceptable solution, different starting points can



yield drastically different profiles for  $D$  and  $V$ , but comparable goodness-of-fit.

- The previous work ascribed most of the uncertainty in the  $D$  and  $V$  profiles to the uncertainties in  $n_e$  and  $T_e$ . But the present analysis indicates very low sensitivity to those quantities and instead indicates that the uncertainty seen before is exclusively a function of the random spline knot positions used during the Monte Carlo sampling.

The rest of the chapter is organized as follows: [section 3.2](#) reviews the basic details of impurity transport. [Section 3.3](#) describes the effects of  $D$  and  $V$  on the spatiotemporal evolution of the impurity density profile following an impurity injection in order to identify the key characteristics which help to constrain the transport coefficients. [Section 3.4](#) uses these results in a linearized Bayesian analysis in order to approximate the diagnostic requirements to reconstruct  $D$  and  $V$  accurate to a given tolerance. [Section 3.5](#) casts the inference of the transport coefficient profiles as an inverse problem and presents the mathematical machinery necessary to solve the problem. [Section 3.6](#) then applies the approach described in [section 3.5](#) to multiple synthetic datasets which were constructed to illustrate various features of this inference problem. Specifically, [section 3.6.1](#) proves the concept using local measurements of the total impurity density and [section 3.6.2](#) proves the concept using line-integrated measurements equivalent to the diagnostic situation on Alcator C-Mod. [Section 3.6.3](#) then uses the line-integrated synthetic data to demonstrate that simple transport coefficient profiles can be inferred using data only from the long decay period of the signals. [Section 3.6.4](#) demonstrates the pathologies which can arise from the use of free knot splines. [Section 3.6.5](#) demonstrates the surprising result that the background  $n_e$  and  $T_e$  profiles have little effect on the inferred  $D$  and  $V$  profiles and explains the previous results in terms of the sensitivity of the solution to the knot locations. [Section 3.6.6](#) demonstrates the use of the model evidence to select the model complexity in a very simple case, and [section 3.6.7](#) applies the whole framework to synthetic data which matches the complexity inferred by the previous work. Finally, [section 3.7](#) summarizes the work and suggests future research directions based on the results presented. Supporting material from this chapter which distracts from the main discussion but gives additional details on the results is given in [appendix c](#) and the `bayes_imp` impurity transport analysis code developed as part of this work is described in [appendix g](#).

### 3.2 Basic details of impurity transport

This section describes the basic details of impurity transport as modeled with the STRAHL code [276]. Consider the transport of an impurity with atomic number  $Z$ . The density of charge state  $i$  (where  $i = 0$  corresponds to the neutral impurities and  $i = Z$  corresponds to the fully-stripped impurities) is denoted by  $n_{Z,i}$ , the density of all charge states is denoted by  $n_Z = \sum_{i=0}^Z n_{i,Z}$ , and the density of impurity ions is denoted by  $n_Z^+ = \sum_{i=1}^Z n_{i,Z}$ . The transport of a given charge state<sup>1</sup>  $i > 0$  obeys the continuity equation

$$\frac{\partial n_{Z,i}}{\partial t} = -\nabla \cdot \Gamma_{Z,i} + Q_{Z,i}, \quad (3.1)$$

where  $\Gamma_{Z,i}$  is the flux and  $Q_{Z,i}$  is the source/sink term. The source/sink term includes ionization, recombination and charge exchange into/out of the charge state:

$$Q_{Z,i} = -n_e n_{Z,i} S_{Z,i} - n_e n_{Z,i} \alpha_{Z,i} - n_H n_{Z,i} \alpha_{cx,Z,i} + n_e n_{Z,i-1} S_{Z,i-1} + n_e n_{Z,i+1} \alpha_{Z,i+1} + n_H n_{Z,i+1} \alpha_{cx,Z,i+1}, \quad (3.2)$$

where  $S_{Z,i}$  is the rate coefficient for ionization out of charge state  $i$  (and hence into charge state  $i + 1$ ),  $\alpha_{Z,i}$  is the rate coefficient for recombination out of charge state  $i$  (and hence into charge state  $i - 1$ ),  $n_H$  is the neutral hydrogen density, and  $\alpha_{cx,Z,i}$  is the rate coefficient for charge exchange out of state  $i$  (and hence into charge state  $i - 1$ ). The neutral source/sink term  $Q_{Z,0}$  also includes the injected particle source. As shown in [276], if poloidal and toroidal asymmetries can be neglected<sup>2</sup> (i.e., the charge state densities  $n_{Z,i}$  are constant on flux surfaces), then equation (3.1) can be expressed as a 1D radial continuity equation:

$$\frac{\partial n_{Z,i}}{\partial t} = -\frac{1}{r_{\text{vol}}} \frac{\partial}{\partial r_{\text{vol}}} (r_{\text{vol}} \Gamma_{Z,i,r_{\text{vol}}}) + Q_{Z,i}, \quad (3.3)$$

where the use of the volumetric minor radius

$$r_{\text{vol}} = \sqrt{\frac{V_r}{2\pi^2 R_0}} \quad (3.4)$$

1. Technically  $n_{Z,0}$  will obey equation (3.1) but not the flux surface-averaged results given later in this discussion.
2. Note that poloidal asymmetries of high- [424] and mid-Z [425, 426] impurities have been observed in Alcator C-Mod, among other machines [427–430]. The assumption of poloidal symmetry is used in this thesis in order to keep the analysis tractable, but this should be noted as a potential source of error, and a future improvement which can be made in this analysis.

allows the flux surface-averaged transport in shaped plasmas to be described correctly. (Here,  $V_r$  is the volume enclosed by the flux surface.)

The impurity flux is typically taken to have a diffusive and a convective component:

$$\Gamma_{Z,i} = -D_{Z,i} \nabla n_{Z,i} + V_{Z,i} n_{Z,i} \quad (3.5)$$

$$\Gamma_{Z,i,r_{\text{vol}}} = -D_{Z,i} \frac{\partial n}{\partial r_{\text{vol}}} + V_{Z,i} n_{Z,i}, \quad (3.6)$$

where  $D_{Z,i}$  is the diffusion coefficient and  $V_{Z,i}$  is the convective velocity. A positive  $V_{Z,i}$  corresponds to a radially outwards flux. For simplicity, the following assumes that the transport coefficients are the same for all charge states, so the subscript  $i$  is omitted.<sup>3</sup> Therefore, the impurity transport equation for charge state  $i$  is

$$\frac{\partial n_{Z,i}}{\partial t} = -\frac{1}{r_{\text{vol}}} \frac{\partial}{\partial r_{\text{vol}}} \left( -r_{\text{vol}} D_Z \frac{\partial n_{Z,i}}{\partial r_{\text{vol}}} + r_{\text{vol}} V_Z n_{Z,i} \right) + Q_{Z,i}. \quad (3.7)$$

This represents a coupled set of  $Z$  partial differential equations. The STRAHL code solves this set of equations to yield the temporal and spatial evolution of the charge state density profiles  $n_{Z,i}(r, t)$ . The evolution of the total impurity density profile can be obtained by summing equation (3.7) over the charge states  $i \geq 1$ . In this case, the ionization/recombination from adjacent charge states balances, leaving only the ionization/recombination into/out of the neutral impurities:

$$\begin{aligned} \frac{\partial n_Z^+}{\partial t} = & -\frac{1}{r_{\text{vol}}} \frac{\partial}{\partial r_{\text{vol}}} \left( -r_{\text{vol}} D_Z \frac{\partial n_Z^+}{\partial r_{\text{vol}}} + r_{\text{vol}} V_Z n_Z^+ \right) \\ & - n_e n_{Z,1} \alpha_{Z,1} - n_H n_{Z,1} \alpha_{\text{cx},Z,1} + n_e n_{Z,0} S_{Z,0}. \end{aligned} \quad (3.8)$$

Consider the radial transport equation, where the subscripts have been omitted for simplicity:

$$\frac{\partial n}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left( -rD \frac{\partial n}{\partial r} + rVn \right) + Q. \quad (3.9)$$

Now consider the steady-state ( $\partial n/\partial t = 0$ ) behavior and take the particle source/sink

3. While the neoclassical transport coefficients are functions of the charge state  $i$ , the transport of mid- $Z$  impurities such as calcium is thought to be dominated by  $\mathbf{E} \times \mathbf{B}$  turbulence such that the transport will be mostly independent of  $i$ . See, for example, [431] for a  $Z$  scaling of impurity profiles arising from the combination of neoclassical and turbulent transport: the scaling levels off around  $Z \approx 20$ .

to be zero throughout the plasma volume:

$$0 = \frac{\partial}{\partial r} \left( -rD \frac{\partial n}{\partial r} + rVn \right) \quad (3.10)$$

$$0 = \int_0^r \frac{\partial}{\partial r'} \left( -r'D \frac{\partial n}{\partial r'} + r'Vn \right) dr' \quad (3.11)$$

$$0 = -rD \frac{\partial n}{\partial r} + rVn \quad (3.12)$$

$$\frac{\partial n / \partial r}{n} = \frac{V}{D}. \quad (3.13)$$

Therefore, the steady-state impurity density profile only yields information on the ratio  $V/D$ . This necessitates the use of time-resolved measurements of transient sources to separate the contributions of  $V$  and  $D$  to the flux.

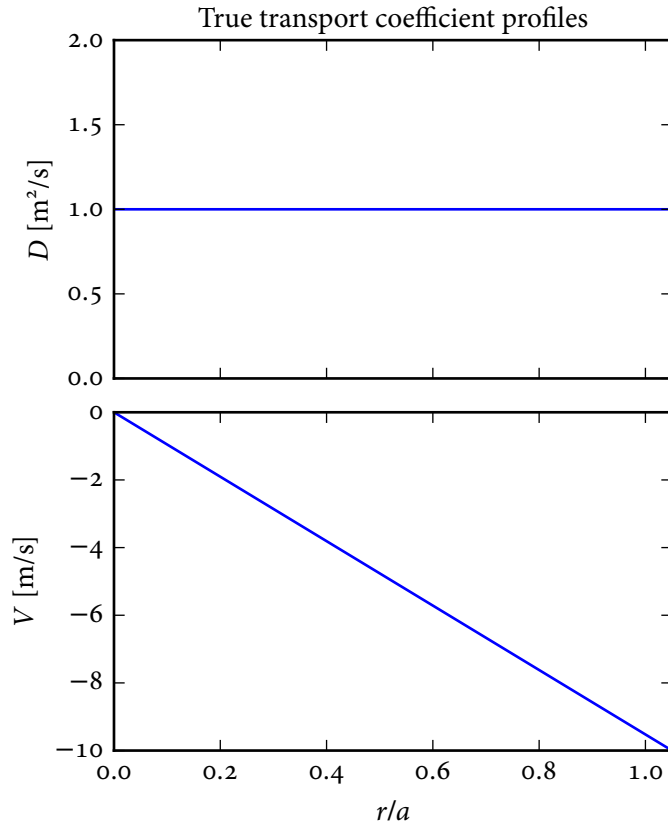
### 3.3 Intuition from time-series characteristics

In order to illustrate the effects of diffusion and convection on impurity transport, STRAHL was run many times with various values of  $D$  and  $V$ . The  $n_e$  and  $T_e$  profiles and the magnetic equilibrium were taken from Alcator C-Mod shot 1101014006, an L-mode discharge used for previous impurity transport work [18, 55, 305]. The source function was taken to be a narrow (in space and time) pulse of calcium to model a delta function. The transport coefficient profiles used were simply

$$D_Z = D, \quad V_Z = V \frac{a}{a_c} \frac{r}{a}, \quad (3.14)$$

where  $V$  is the convective velocity at the edge of the computational domain  $r = a_c$  (just outside of the LCFS), see [figure 3.1](#). Additional insight into the behavior of the impurity density given these simple transport coefficient profiles is given in [section C.1](#).

In general, a transient impurity injection exhibits two phases: a rapid rise of the plasma impurity content when the impurity flux is inwards followed by a slower decay of the impurity content while the impurity flux is outwards. This is shown in [figure 3.2](#), which gives the time histories of the impurity density at several radial locations for the basic values  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$  (these profiles are shown in [figure 3.1](#)). Notice that there is a finite delay between the time of the injection  $t_{\text{inj}}$  and the time the impurity density peaks, and that this delay is different between the core and the edge of the plasma: there is a finite time for the impurities to reach the core. This is also illustrated by [figure 3.3](#), which shows the evolution of the impurity density profile. Therefore, one possible figure of merit to describe these signals is



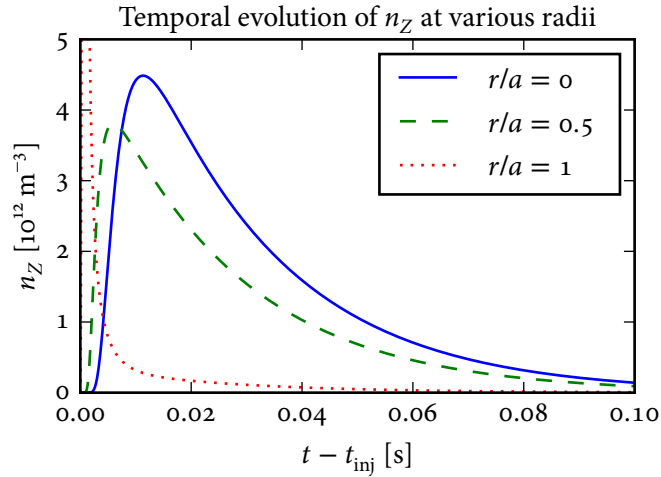
**Figure 3.1:** Basic impurity transport coefficient profiles used to generate the synthetic impurity transport data. This figure was produced using `make_time_res_plots.py`.

the core rise time, denoted by  $t_r$ . While in signal processing it is more common to characterize rise times in terms of the time to go from 10% to 90% of the final value, the analysis here uses the actual time at which the peak value is obtained.

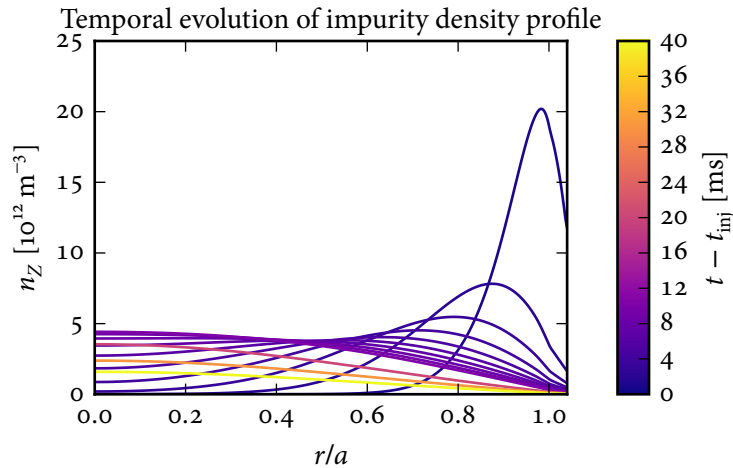
After the peak there is an exponential decay of the density at all locations, the time constant of which is the impurity confinement time  $\tau_{\text{imp}}$ . This is best defined in terms of the temporal decay of the total number of impurity atoms inside the LCFS,

$$N_Z = \int \sum_{i=0}^Z n_{Z,i} dV, \quad (3.15)$$

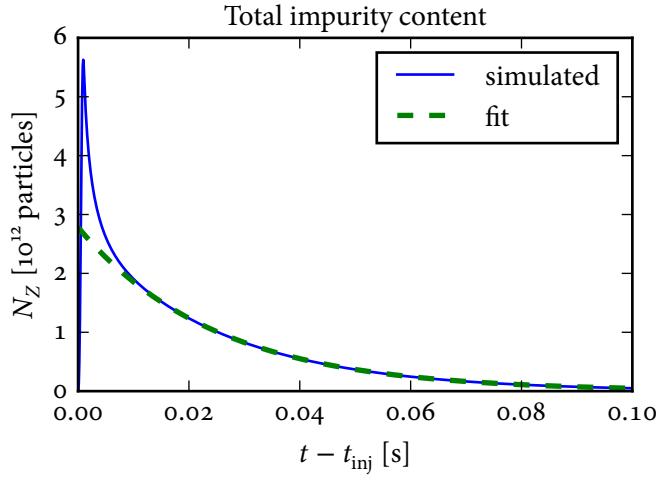
where the volume integral is estimated using the trapezoid rule. This is shown in



**Figure 3.2:** Temporal evolution of the impurity density  $n_Z$  at several radii. The vertical scale was set to cut off the curve for  $r/a = 1$  in order to show the details at the other radii. Note that all three signals peak at different times because of the finite time for the impurities to penetrate to the core of the plasma. This figure was produced using `make_DV_plot_matrix.py`.



**Figure 3.3:** Temporal evolution of the impurity density profile. Profiles are drawn every 1 ms for the first 10 ms, then every 10 ms after that. During the first 10 ms the profile shape is evolving as the impurities move from the edge in. After 10 ms the decay of  $N_Z$  enters the purely exponential decay phase and the profile shape is constant as it decays. This figure was produced using `make_DV_plot_matrix.py`.



**Figure 3.4:** Temporal evolution of the total impurity content  $N_Z$ . The exponential fit used to obtain the impurity confinement time  $\tau_{\text{imp}}$  is shown as the **green dashed curve**. Note that the initial 10 ms of the decay exhibits a different time constant because the impurity density profile shape is still evolving. This figure was produced using `make_DV_plot_matrix.py`.

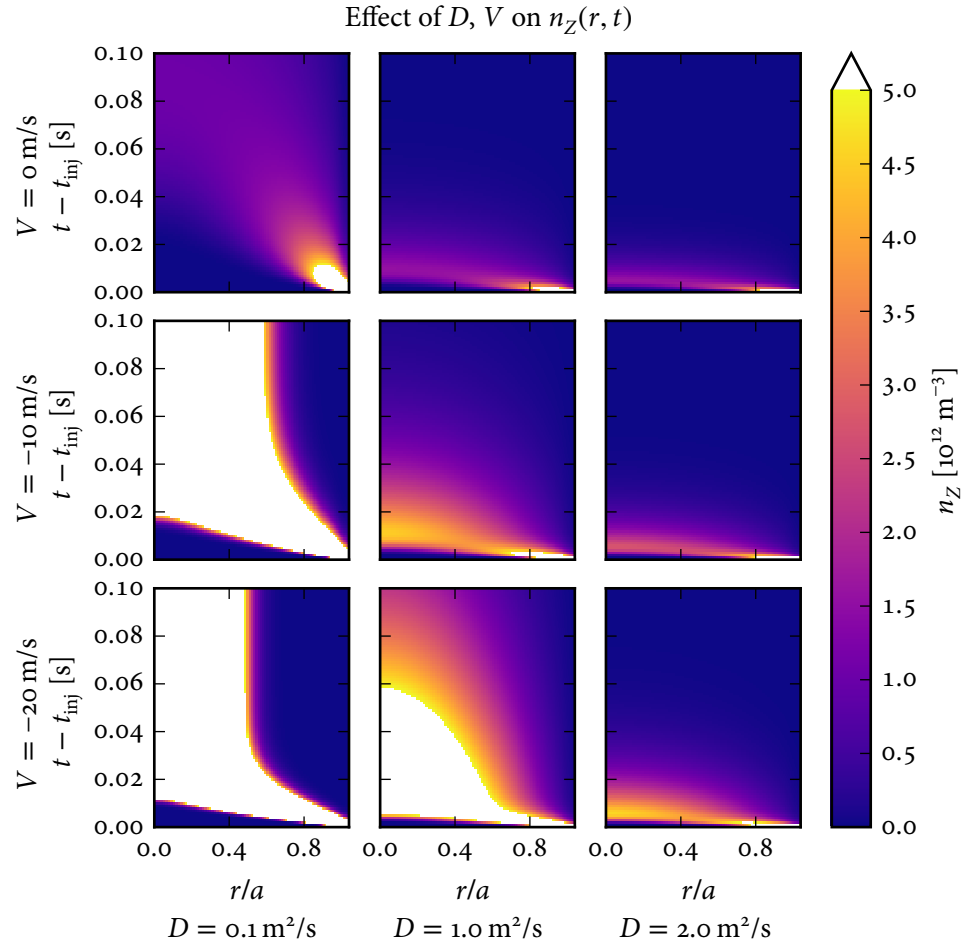
figure 3.4. The impurity confinement time is then determined by fitting<sup>4</sup>  $N_Z \sim N_{Z,0} \exp(-t/\tau_{\text{imp}})$ . Note, however, that the initial period of the decay is not purely exponential/exhibits a different time constant. This is a result of the fact that the impurity density profile shape is still evolving up until about 10 ms after the injection, as noted in figure 3.3. Therefore, when post-processing the results of STRAHL runs, the impurity confinement time is determined by fitting  $N_Z(t)$  from 10 ms after it peaks and on.<sup>5</sup> Note that this has a key implication for the analysis of experimental data: if confinement times are fit using the initial part of the decay of a signal (when the signal-to-noise ratio is highest), this will likely lead to an incorrect value because the profiles may still be evolving instead exhibiting pure exponential decay with the desired time constant.

Figure 3.5 illustrates the effect of changing  $D$  and  $V$  on  $n_Z(r, t)$ . It is clear that a wide variety of behaviors arises even from this simple model. The figure implies that four specific figures of merit are particularly relevant:

- the core rise time  $t_r$ ,

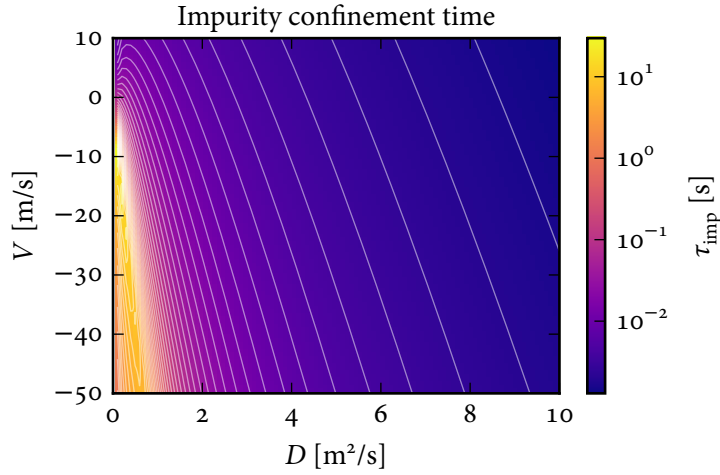
4. In the present analysis this is accomplished by fitting a straight line to  $\ln N_Z$ .

5. The actual time for the profile shape to settle depends on  $D$ ,  $V$ ; 10 ms was found to be sufficient for the cases considered here.



**Figure 3.5:** Effect of  $D$ ,  $V$  on  $n_Z(r, t)$ . All plots share the same coloring so that absolute magnitudes can be compared. To make the details visible, the coloring saturates at  $5 \times 10^{12} \text{ m}^{-3}$  (white regions). When  $V$  is large relative to  $D$  the impurities remain in the core for a very long time, though a gradual loss to the SOL is still observed. It is clear that both  $D$  and  $V$  affect the rise time  $t_r$ , the impurity confinement time  $\tau_{\text{imp}}$  and the peak impurity density  $n_{Z,p}(o)$ , as well as the fine details of the profile shape and time evolution. This figure was produced using `make_DV_plot_matrix.py`.





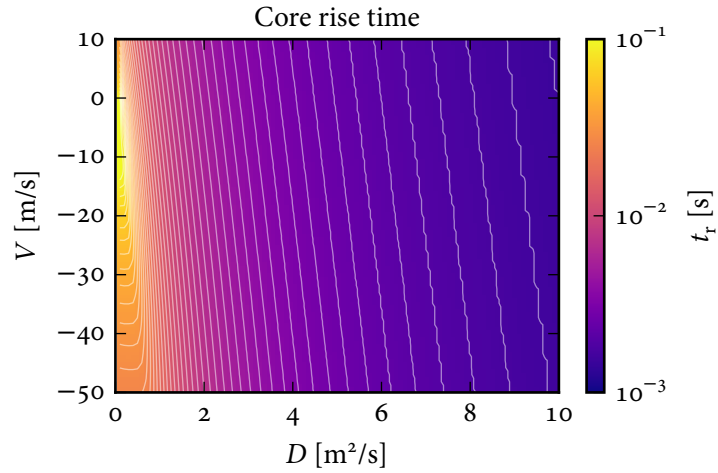
**Figure 3.6:** Impurity confinement time  $\tau_{\text{imp}}$  as a function of  $D$  and  $V$ . The coloring and the contours are logarithmically spaced. Outside of the low  $D$ , strongly negative  $V$  region where accumulation occurs,  $\tau_{\text{imp}}$  is a function of both  $D$  and  $V$ . Higher diffusion and lower inwards convection leads to shorter confinement of impurities, as expected. This figure was produced using `make_time_res_plots.py`.

- the impurity confinement time  $\tau_{\text{imp}}$ ,
- how broad versus peaked the impurity density profile is during the exponential decay phase, and
- the peak core impurity density  $n_{Z,p}(0)$ .

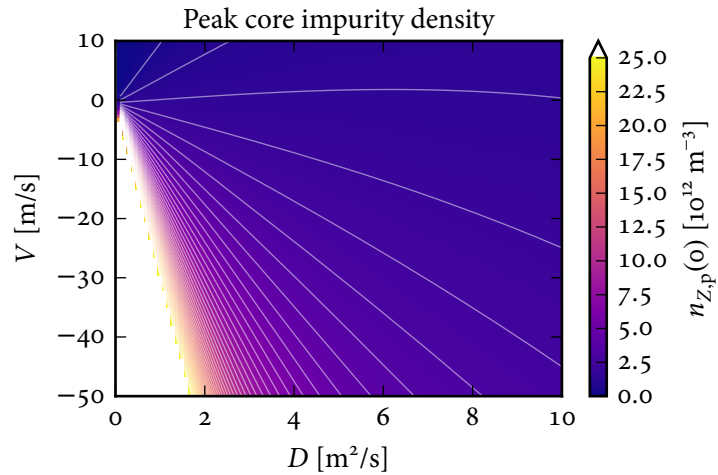
For the analysis in this section the broadness of the profile is represented by the ratio  $b_{0.75} = n_Z(r/a = 0.75)/n_Z(0)$ . This was chosen over more sophisticated estimates of profile peaking like  $n_Z(0)/\langle n_Z \rangle_{\text{vol}}$  because it can be estimated from two diagnostic channels and hence enables the measurement to be made with simpler diagnostics. Plots of these quantities as functions of  $D$  and  $V$  are given in [figure 3.6](#) through [figure 3.9](#).

The impurity confinement time  $\tau_{\text{imp}}$  is a function of both  $D$  and  $V$ , and was found to be consistent with the expression given in [432]:

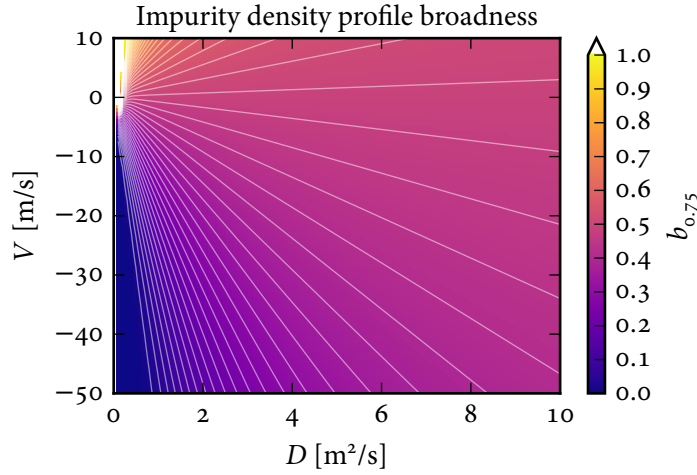
$$\tau_{\text{imp}} \approx \frac{77 + S^2}{56 + S^2} \frac{e^S - S - 1}{4S^2} \frac{a^2}{D}, \quad (3.16)$$



**Figure 3.7:** Core rise time  $t_r$  as a function of  $D$  and  $V$ . The coloring and the contours are logarithmically spaced. The “zig-zag” shape of the low  $t_r$  contours arises from the  $10\ \mu\text{s}$  time resolution. Outside of the low  $D$ , strongly negative  $V$  region where accumulation occurs,  $t_r$  is a function primarily of  $D$ . Higher  $D$  sweeps the impurities into the core of the plasma faster, as is expected. This figure was produced using `make_time_res_plots.py`.



**Figure 3.8:** Peak core impurity density as a function of  $D$  and  $V$ . The coloring and the contours are linearly spaced and the coloring saturates at  $n_{Z,p}(0) = 25 \times 10^{12}\ \text{m}^{-3}$ . The contours are approximately lines from the origin, which means they are also contours of  $V/D$ . This figure was produced using `make_time_res_plots.py`.



**Figure 3.9:** Impurity density profile broadness  $b_{0.75} = n_Z(0.75)/n_Z(0)$  as a function of  $D$  and  $V$ . The coloring and the contours are linearly spaced and the coloring saturates at  $b_{0.75} = 1$ . The contours are lines from the origin, which means they are also contours of  $V/D$ . This figure was produced using `make_time_res_plots.py`.

where

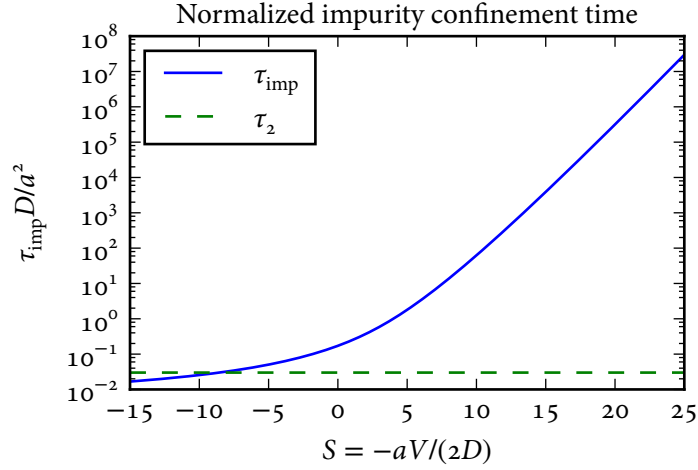
$$S = -\frac{aV}{2D}. \quad (3.17)$$

Note that

$$\tau_{\text{imp}} \frac{D}{a^2} \sim \begin{cases} e^S/(4S^2), & S \gg 0 \\ -1/(4S), & S \ll 0 \end{cases}. \quad (3.18)$$

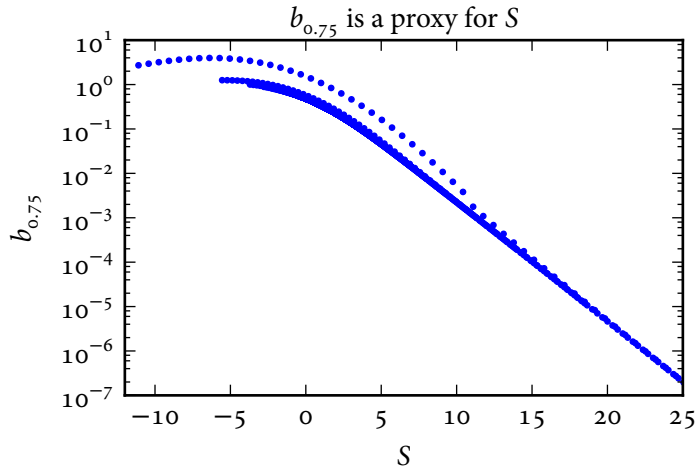
This is shown as a function of  $S$  in [figure 3.10](#).

The rise time  $t_r$  is primarily a function of  $D$ . This is consistent with [\[433\]](#), where an eigenvalue expansion for the Green's function (i.e., impulse response) of the convection-diffusion equation in circular geometry is given. The first eigenvalue corresponds to the impurity confinement time from [\[432\]](#) while the second eigenvalue determines the rise time and corresponds to a time scale  $\tau_2 \approx 0.03a^2/D$  which is a function of  $D$  only. This is also consistent with the results given in [section C.1](#) for typical values of  $D$ ,  $V$  which shows that the diffusive flux dominates the early portion of the injection: when all of the injected impurities are localized to the edge of the plasma, there is a very high density gradient to drive them inwards. This time scale is shown as the horizontal line in [figure 3.10](#).



**Figure 3.10:** Normalized impurity confinement time  $\tau_{\text{imp}} D/a^2$  as a function of  $S = -aV/(2D)$  (blue solid curve), as predicted by equation (3.16). Positive  $S$  corresponds to inwards convection. Increasing  $S$  dramatically increases the impurity confinement time. Also shown is the (normalized) time scale of the second eigenvalue  $\tau_2 D/a^2 \sim 0.03$  (green dashed line) which determines the rise time  $t_r$ . For all cases with inwards convection,  $\tau_{\text{imp}} > \tau_2$ . This figure was produced using `make_seguin_plot.py`.

The profile broadness  $b_{0.75}$  has contours which are lines originating from the origin, which means  $b_{0.75}$  is primarily a function of the peaking factor  $V/D$ . This is consistent with figure 4 of [432], where it is found that the profile shape is a function of  $S \propto V/D$ . The profile broadness is shown as a function of  $S$  in figure 3.11, which confirms that this interpretation holds for  $S \gtrsim 10$ . Note that, because the profile shape is dictated by  $V/D$ , having extra spatial resolution provides a better constraint on the profile shape and therefore is equivalent to knowing  $b_{0.75} \propto V/D$  to a higher precision in the context of the following analysis. If  $D$  and  $V$  have a more complicated shape than what is assumed here the spatial resolution must be higher in order to determine the spatial structure of the profile. The peak core impurity density  $n_{Z,p}(o)$  is also primarily a function of  $V/D$ . In the present analysis,  $b_{0.75}$  is preferred as the proxy for  $V/D$  because measuring  $n_{Z,p}(o)$  requires an absolute calibration of the diagnostics whereas measuring  $b_{0.75}$  only requires a relative calibration between the spatial channels.



**Figure 3.11:** Impurity density profile broadness  $b_{0.75}$  as a function of normalized peaking factor  $S$ . This figure was produced by making a scatterplot of the results from the STRAHL runs on the dense grid in  $(D, V)$  space. Over most of the domain,  $b_{0.75}$  is a function of  $S$  only, indicating that it is a good proxy for  $S$ . At low  $D$  and low/outwards  $V$ , however, the relationship breaks down. This figure was produced using `make_time_res_plots.py`.

### 3.4 Bayesian analysis of required tolerances

This section derives a model describing the inference of the transport coefficients  $D$  and  $V$  from measurements of  $\tau_{\text{imp}}$ ,  $t_r$  and  $b_{r/a} = n_Z(r/a)/n_Z(0)$ . As illustrated in the previous section, these three observables represent nearly<sup>6</sup> all of the information which can be readily extracted from spatially- and temporally-resolved measurements of the impurity density profile, and hence this analysis should be representative of the more complete analysis which combines STRAHL with a synthetic diagnostic in order to find the best fit to the observed diagnostic signals. This analysis is useful because it allows the diagnostic capabilities necessary to reconstruct  $D$  and  $V$  to a desired precision to be estimated without requiring many computationally-expensive STRAHL runs. Furthermore, the following analysis is applicable to any experimental situation which can be cast as an inverse problem where the observables  $y$  are a func-

6. There is additional information on  $D$  carried by the rise times of the other spatial channels and hence the evolution of the profile shape, but this is not included in the present analysis for two reasons: first, the fitting of the spline to determine the rise time is one of the more computationally-expensive steps in the analysis. Second, the analysis in section 3.4.3.1 indicates that the profile shape measurements  $b_{r/a}$  will be more important than the rise time in the inference.

tion of the quantities to be inferred  $\mathbf{T}$ :  $\mathbf{y} = f(\mathbf{T})$ , provided that  $f(\mathbf{T})$  can be linearized in a region around the true values bounded by the diagnostic uncertainties.

### 3.4.1 Derivation of the model

As shown in figure 3.6 through figure 3.9,  $\tau_{\text{imp}}(D, V)$ ,  $t_r(D, V)$  and  $b_{0.75}(D, V)$  have contours that are not parallel to each other. Therefore, provided that the assumed profile shapes for  $D$  and  $V$  are correct, two or more of these measurements can reconstruct  $D$  and  $V$ . This is shown in figure 3.12 where the contours corresponding to  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$  are shown. This construction allows a simple estimate of the spatial resolution, temporal resolution and noise level requirements to determine  $D$  and  $V$ . Note that the following analysis does not account for the uncertainties in the atomic physics data and the background  $n_e$ ,  $T_e$  profiles, nor does it account for the issues inherent in the tomographic inversion of line-averaged measurements.

Consider a measurement of some quantity  $y_i$  which is a function of both  $D$  and  $V$ :

$$y_i = y_i(D, V). \quad (3.19)$$

Linearize at  $D = D_o$ ,  $V = V_o$ :

$$y_i \approx y_{i,o} + \left. \frac{\partial y_i}{\partial D} \right|_{D=D_o, V=V_o} (D - D_o) + \left. \frac{\partial y_i}{\partial V} \right|_{D=D_o, V=V_o} (V - V_o) \quad (3.20)$$

$$= \left( y_{i,o} - \left. \frac{\partial y_i}{\partial D} \right|_{D=D_o, V=V_o} D_o - \left. \frac{\partial y_i}{\partial V} \right|_{D=D_o, V=V_o} V_o \right) + \left. \frac{\partial y_i}{\partial D} \right|_{D=D_o, V=V_o} D + \left. \frac{\partial y_i}{\partial V} \right|_{D=D_o, V=V_o} V \quad (3.21)$$

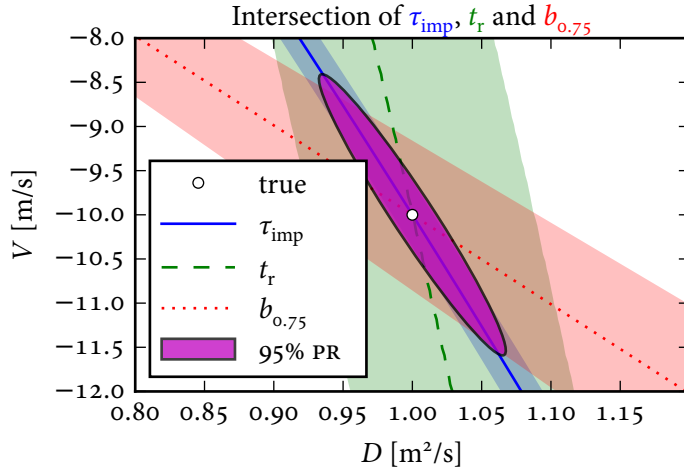
$$= a_i + b_i D + c_i V, \quad (3.22)$$

where  $y_{i,o} = y_i(D_o, V_o)$ . To prepare for the next step, write this in vector form:

$$y_i \approx a_i + [b_i \quad c_i] \begin{bmatrix} D \\ V \end{bmatrix} = a_i + \mathbf{C}_i \mathbf{T}, \quad (3.23)$$

where  $\mathbf{C}_i$  is the row vector  $[b_i, c_i]$  and  $\mathbf{T} = [D, V]^T$  is the column vector of transport coefficients. Consider each measurement to have Gaussian noise:

$$f_{Y_i|\mathbf{T}}(y_i|\mathbf{T}) = \mathcal{N}(y_i; a_i + \mathbf{C}_i \mathbf{T}, \sigma_i^2). \quad (3.24)$$



**Figure 3.12:** Contours of  $\tau_{\text{imp}}$  (blue solid),  $t_r$  (green dashed) and  $b_{0.75}$  (red dotted) for  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$  (white point). Each measurement is shown with its  $\pm 2\sigma$  (i.e., 95% confidence) interval for  $\sigma_{\tau_{\text{imp}}} = 0.3 \text{ ms}$ ,  $\sigma_{t_r} = 0.5 \text{ ms}$  and  $\sigma_{b_{0.75}} = 6 \times 10^{-3}$ . Any given measurement only constrains  $D$  and  $V$  to lie in a neighborhood around a contour that extends through all of  $D, V$  space. The unique intersection of the contours corresponding to two or more measurements determines  $D$  and  $V$ . If multiple, noisy measurements are used then the  $D$  and  $V$  values which fit their intersection in the least-squares sense are found. The angles of the intersections determine how much information a measurement adds to the inference: the closer to perpendicular two measurements are, the better they can constrain  $D$  and  $V$ . The magenta ellipse gives the 95% posterior region for the combination of these three measurements according to equation (3.34) through equation (3.38). The  $\pm 2\sigma$  bands on the contours are slightly asymmetric because they were generated using the actual STRAHL outputs and not the linearization. This figure was produced using `make_bayes_time_res_plot.py`.

The joint distribution for  $n$  such independent<sup>7</sup> measurements  $\mathbf{y} \in \mathbb{R}^n$  is then

$$f_{\mathbf{Y}|T}(\mathbf{y}|T) = \mathcal{N}(\mathbf{y}; \mathbf{a} + \mathbf{C}T, \mathbf{\Sigma}_{\mathbf{y}}), \quad (3.25)$$

where  $\mathbf{a} \in \mathbb{R}^n$  is the vector containing  $a_i$  for each measurement,  $\mathbf{C} \in \mathbb{R}^{n \times 2}$  is the matrix where each row contains  $C_i$ , and  $\mathbf{\Sigma}_{\mathbf{y}}$  has each  $\sigma_i^2$  on the diagonal and zeros

<sup>7</sup> In general,  $\mathbf{\Sigma}_{\mathbf{y}}$  could include correlations between the measurements, but this level of sophistication is not adopted for the present analysis. A non-diagonal covariance matrix would be necessary to represent the results of a tomographic inversion, for instance.

everywhere else. Assume a Gaussian prior distribution for  $T$ :

$$f_T(T) = \mathcal{N}(T; \boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T). \quad (3.26)$$

The posterior distribution for  $T$  is then

$$f_{T|Y}(T|y) \propto f_{y|T}(y|T)f_T(T) = \mathcal{N}(y; \mathbf{a} + \mathbf{C}T, \boldsymbol{\Sigma}_y) \mathcal{N}(T; \boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T). \quad (3.27)$$

The Gaussian distribution is self-conjugate, meaning that a Gaussian likelihood with Gaussian prior distribution results in a Gaussian posterior distribution [286, 304]. The posterior distribution for  $T$  is then given by the standard result for linear Gaussian systems [304]:

$$f_{T|Y}(T|y) = \mathcal{N}(T; \boldsymbol{\mu}_{T|y}, \boldsymbol{\Sigma}_{T|y}) \quad (3.28)$$

$$\boldsymbol{\Sigma}_{T|y}^{-1} = \boldsymbol{\Sigma}_T^{-1} + \mathbf{C}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{C} \quad (3.29)$$

$$\boldsymbol{\mu}_{T|y} = \boldsymbol{\Sigma}_{T|y} (\mathbf{C}^T \boldsymbol{\Sigma}_y^{-1} (y - \mathbf{a}) + \boldsymbol{\Sigma}_T^{-1} \boldsymbol{\mu}_T). \quad (3.30)$$

Now make  $f_T(T)$  non-informative by making it infinitely wide, such that  $\boldsymbol{\Sigma}_T^{-1} \rightarrow \mathbf{0}$ :

$$\boldsymbol{\Sigma}_{T|y} = (\mathbf{C}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{C})^{-1} \quad (3.31)$$

$$\boldsymbol{\mu}_{T|y} = (\mathbf{C}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{C})^{-1} (\mathbf{C}^T \boldsymbol{\Sigma}_y^{-1} (y - \mathbf{a})). \quad (3.32)$$

Note that this is exactly the result of heteroscedastic linear least squares regression for the model  $y - \mathbf{a} = \mathbf{C}T$ . The use of a Gaussian likelihood and non-informative prior distribution means that this analysis combines the noisy, potentially conflicting observations  $y$  to reconstruct the transport coefficients  $T$  which are most consistent with the observations in the weighted least squares sense.

Expanding the covariance matrix given in equation (3.31) yields

$$\boldsymbol{\Sigma}_{T|y} = \frac{1}{\left(\sum_{i=1}^n b_i^2/\sigma_i^2\right) \left(\sum_{i=1}^n c_i^2/\sigma_i^2\right) - \left(\sum_{i=1}^n b_i c_i/\sigma_i^2\right)^2} \cdot \begin{bmatrix} \sum_{i=1}^n c_i^2/\sigma_i^2 & -\sum_{i=1}^n b_i c_i/\sigma_i^2 \\ -\sum_{i=1}^n b_i c_i/\sigma_i^2 & \sum_{i=1}^n b_i^2/\sigma_i^2 \end{bmatrix}. \quad (3.33)$$

Therefore, the (marginal) uncertainties in the individual parameters are

$$\sigma_D^2 = \frac{\sum_{i=1}^n c_i^2/\sigma_i^2}{\left(\sum_{i=1}^n b_i^2/\sigma_i^2\right) \left(\sum_{i=1}^n c_i^2/\sigma_i^2\right) - \left(\sum_{i=1}^n b_i c_i/\sigma_i^2\right)^2} \quad (3.34)$$

$$\sigma_V^2 = \frac{\sum_{i=1}^n b_i^2/\sigma_i^2}{\left(\sum_{i=1}^n b_i^2/\sigma_i^2\right) \left(\sum_{i=1}^n c_i^2/\sigma_i^2\right) - \left(\sum_{i=1}^n b_i c_i/\sigma_i^2\right)^2} \quad (3.35)$$



and the covariance is

$$\sigma_{DV} = \frac{-\sum_{i=1}^n b_i c_i / \sigma_i^2}{\left(\sum_{i=1}^n b_i^2 / \sigma_i^2\right) \left(\sum_{i=1}^n c_i^2 / \sigma_i^2\right) - \left(\sum_{i=1}^n b_i c_i / \sigma_i^2\right)^2}. \quad (3.36)$$

The predictions are then

$$\mu_D = \sigma_D^2 \sum_{i=1}^n \frac{b_i (y_i - a_i)}{\sigma_i^2} + \sigma_{DV} \sum_{i=1}^n \frac{c_i (y_i - a_i)}{\sigma_i^2} \quad (3.37)$$

$$\mu_V = \sigma_V^2 \sum_{i=1}^n \frac{c_i (y_i - a_i)}{\sigma_i^2} + \sigma_{DV} \sum_{i=1}^n \frac{b_i (y_i - a_i)}{\sigma_i^2}. \quad (3.38)$$

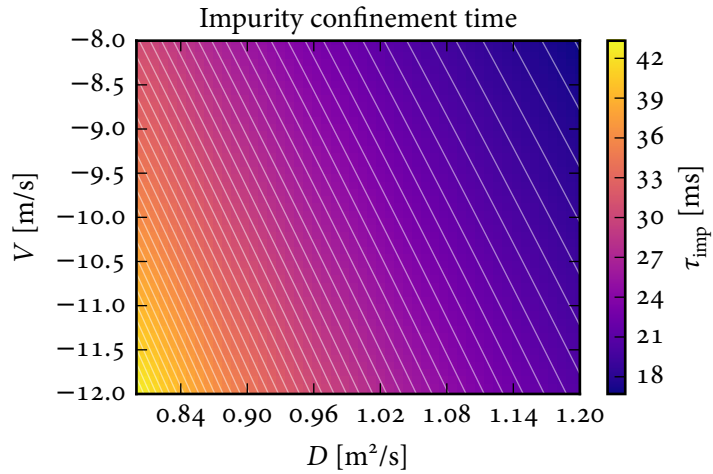
This framework allows the use of measurements of any number of features (including repeated measurements of the same feature) and the estimation of any number of parameters. It can be applied to analyzing the inferences to be obtained from any diagnostic provided that the features to be observed are roughly linear in the parameters to be inferred and the noise on the features to be observed is roughly Gaussian.

### 3.4.2 Determining the required tolerances on $\tau_{\text{imp}}$ , $t_r$ and $b_{0.75}$

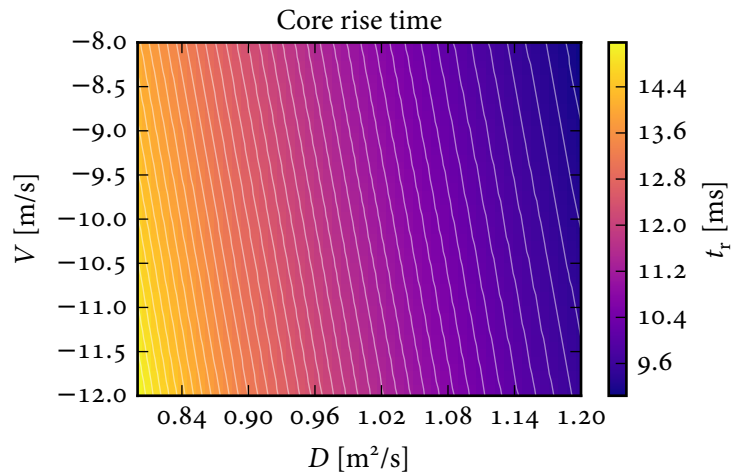
Now consider the reconstruction of the true parameters  $D_0 = 1 \text{ m}^2/\text{s}$ ,  $V_0 = -10 \text{ m/s}$  using measurements of the three parameters discussed previously:  $\tau_{\text{imp}}$ ,  $t_r$  and  $b_{0.75}$ . The absolutely-calibrated measurement of the peak impurity density on axis  $n_{z,p}(0)$  is not included because  $b_{0.75}$  contains the same information but does not require an absolute calibration of the measurement.

These quantities were evaluated over a dense grid covering a  $\pm 20\%$  window around the true parameters and contour plots are given in [figure 3.13](#) through [figure 3.15](#). The slightly uneven spacing of the contours indicates that the linearization is not exact, but this analysis will still be useful to establish the basic specifications required. In particular,  $\tau_{\text{imp}}$  seems to exhibit the most nonlinearity in the sense that the spacing of the contours is visibly nonuniform. This is not an issue as  $\tau_{\text{imp}}$  is typically the quantity that can be measured to the highest precision because, even at moderate time resolution, there are many samples in the decay of the impurity density.

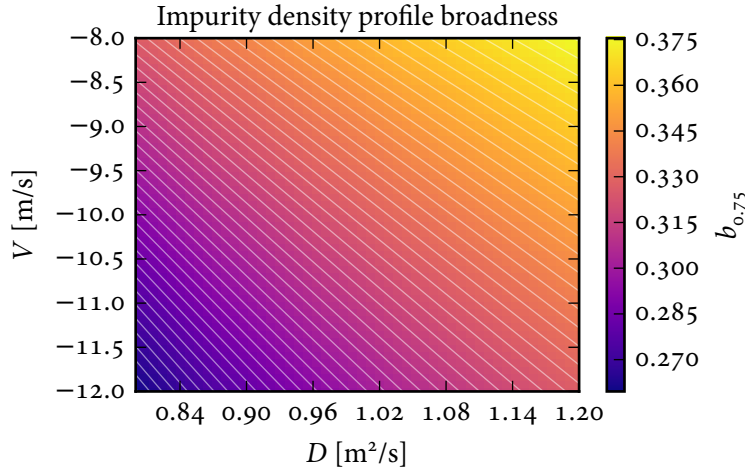
The coefficients  $a_i$ ,  $b_i$  and  $c_i$  were evaluated for each of the three signals by fitting the results with a bivariate cubic interpolating spline. Because of the discretization error in  $t_r$  noted in [figure 3.7](#), the spline for  $t_r$  had a small smoothing parameter to allow an accurate estimate of the derivatives to be obtained. Plots showing the discrepancy between the linearizations and the actual STRAHL results are given in [figure c.3](#)



**Figure 3.13:** Impurity confinement time  $\tau_{\text{imp}}$  in the vicinity of  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$ . The contours and coloring are linearly spaced. This figure was produced using `make_bayes_time_res_plot.py`.



**Figure 3.14:** Rise time  $t_r$  in the vicinity of  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$ . The contours and coloring are linearly spaced. This figure was produced using `make_bayes_time_res_plot.py`.

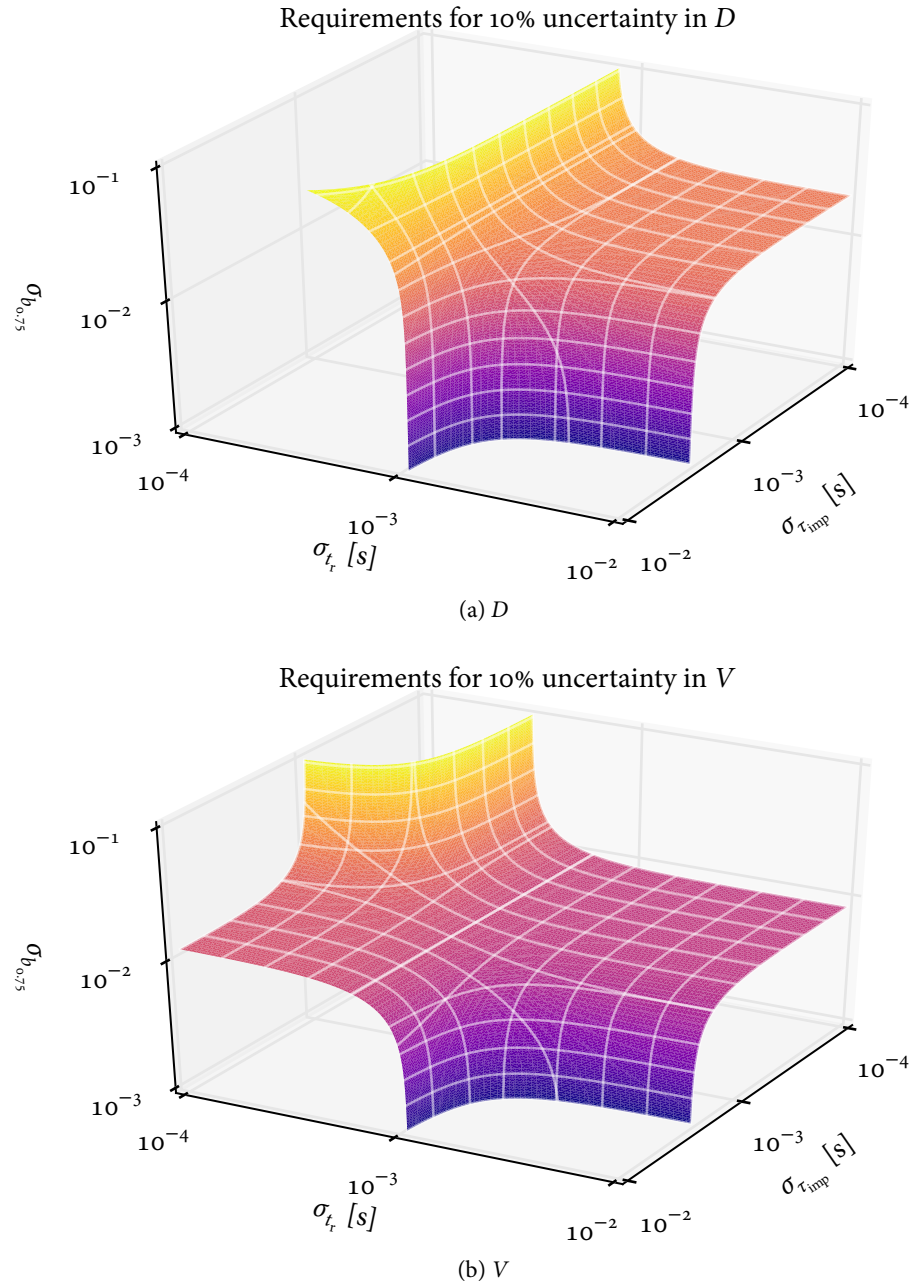


**Figure 3.15:** Impurity density profile broadness  $b_{0.75}$  in the vicinity of  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$ . The contours and coloring are linearly spaced. This figure was produced using `make_bayes_time_res_plot.py`.

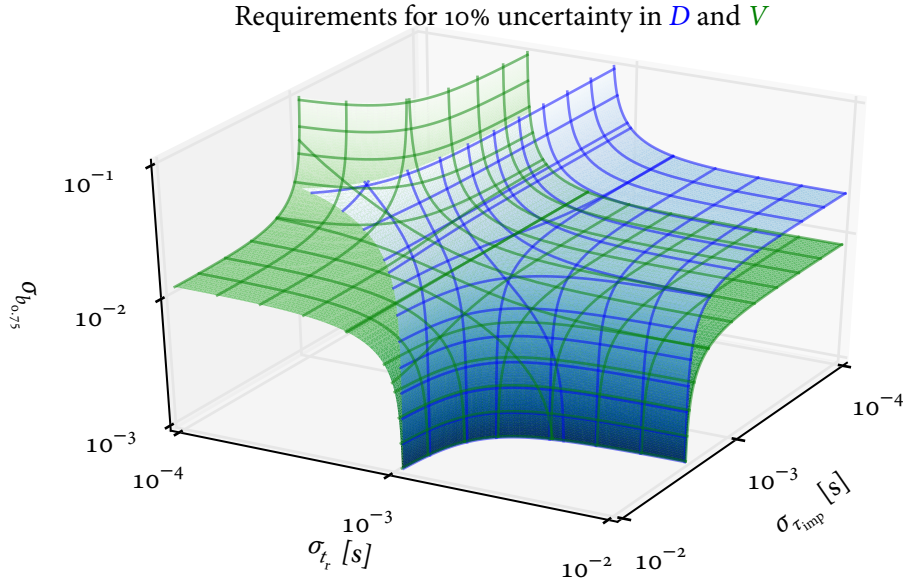
through [figure c.5](#). The expected result of combining these three measurements with reasonable uncertainties is shown in [figure 3.12](#).

The  $1\sigma$  uncertainties in  $D$  and  $V$  were evaluated over a dense, logarithmically-spaced grid of  $\sigma_{\tau_{\text{imp}}}$ ,  $\sigma_{t_r}$ ,  $\sigma_{b_{0.75}}$  using [equation \(3.34\)](#) and [equation \(3.35\)](#). The isosurfaces in  $(\sigma_{\tau_{\text{imp}}}, \sigma_{t_r}, \sigma_{b_{0.75}})$  space corresponding to  $\sigma_D = 0.1D_0$  and  $\sigma_V = 0.1|V_0|$  are given in [figure 3.16](#). These isosurfaces were computed using the marching cubes algorithm in `scikit-image` [[434](#), [435](#)]. These figures illustrate what is intuitively expected: when any two quantities are known to high precision, the third quantity becomes redundant and has very low required precision. Also of interest is the fact that the surface for  $\sigma_V = 0.1|V_0|$  lies entirely below the surface for  $\sigma_D = 0.1D_0$ , which is also illustrated in [figure 3.17](#) where the isosurfaces are overlaid. This means that  $D$  will always be known better than  $V$  unless  $b_{0.75}$  is known very well, outside of the domain of the plots shown. That reconstructing  $V$  requires  $b_{0.75}$  to be measured to a higher precision than reconstructing  $D$  does makes sense, as  $b_{0.75}$  is the observable which is most sensitive to  $V$ .

Note that  $\tau_{\text{imp}}$  is expected to be known to a much higher precision than  $b_{0.75}$  or  $t_r$ , so plots of  $\sigma_D$  and  $\sigma_V$  as functions of  $\sigma_{t_r}$  and  $\sigma_{b_{0.75}}$  for  $\sigma_{\tau_{\text{imp}}} = 1 \mu\text{s}$  are given in [figure 3.18](#). Note from [figure 3.17](#) that these plots are expected to be the same for  $\sigma_{\tau_{\text{imp}}} \lesssim 1 \text{ ms}$ ;  $\sigma_{\tau_{\text{imp}}} = 1 \mu\text{s}$  was chosen for convenience. [Figure 3.18](#) confirms the intuition from the isosurfaces in  $(\sigma_{\tau_{\text{imp}}}, \sigma_{t_r}, \sigma_{b_{0.75}})$  space: when  $\tau_{\text{imp}}$  is known with



**Figure 3.16:** Isosurfaces of (a)  $\sigma_D = 0.1D_0$  and (b)  $\sigma_V = 0.1|V_0|$  in  $(\sigma_{\tau_{\text{imp}}}, \sigma_{\tau_t}, \sigma_{b_{0.75}})$  space. The coloring is proportional to  $\sigma_{b_{0.75}}$  in order to make the vertical changes more visible. The white lines are logarithmically-spaced isolines of  $\sigma_{\tau_{\text{imp}}}$ ,  $\sigma_{\tau_t}$  and  $\sigma_{b_{0.75}}$ . Any point on the surface will enable the quantity to be reconstructed with  $\pm 10\% 1\sigma$  uncertainty. Points below the surface enable a higher precision. The “right angle” features correspond to the fact that, given sufficient precision for any two quantities, the third quantity is redundant. For instance, the yellow region in the back corner corresponds to both  $\tau_{\text{imp}}$  and  $\tau_t$  being known to very high precision. In this case  $b_{0.75}$  is redundant, and does not need to be estimated to better than  $\pm 100\%$  uncertainty. This figure was produced using `make_bayes_time_res_plot.py`.



**Figure 3.17:** Isosurfaces for  $\sigma_D = 0.1D_0$  (blue) and  $\sigma_V = 0.1|V_0|$  (green). The surface for  $V$  lies below that for  $D$  at all points, indicating that  $D$  will always be known better than  $V$  for reasonably attainable uncertainties. This figure was produced using `make_bayes_time_res_plot.py`.

small uncertainty, either  $b_{0.75}$  or  $t_r$  must be known with small uncertainty, but not both.

### 3.4.3 Determining the number of spatial channels, sampling rate and noise level

#### 3.4.3.1 Estimating the uncertainty in the profile broadness $b_{0.75}$

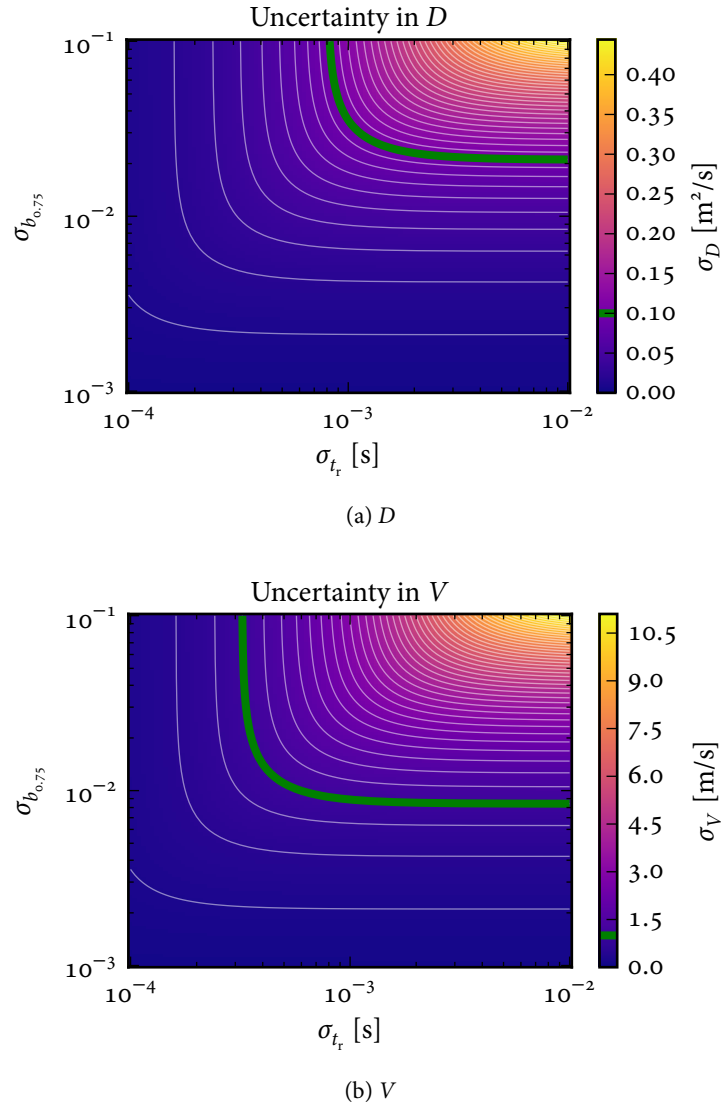
The profile broadness  $b_{0.75}$  is defined as

$$b_{0.75} = \frac{n_Z(0.75)}{n_Z(0)}. \quad (3.39)$$

Applying the uncertainty propagation equation yields an estimate of the variance:

$$\sigma_{b_{0.75}}^2 = \left( \frac{1}{n_Z(0)} \right)^2 \sigma_{n_Z(0.75)}^2 + \left( -\frac{n_Z(0.75)}{n_Z(0)^2} \right)^2 \sigma_{n_Z(0)}^2 \quad (3.40)$$

$$= \frac{\sigma_{n_Z(0.75)}^2}{n_Z(0)^2} + \frac{n_Z(0.75)^2 \sigma_{n_Z(0)}^2}{n_Z(0)^4}. \quad (3.41)$$



**Figure 3.18:** Uncertainty in (a)  $D$  and (b)  $V$  as a function of  $\sigma_{t_r}$  and  $\sigma_{b_{0.75}}$  for  $\sigma_{r_{\text{imp}}} = 1 \mu\text{s}$ . The parameters corresponding to  $\sigma_D = 0.1 D_0$  and  $\sigma_V = 0.1 |V_0|$  are indicated by the green contours. This figure was produced using `make_bayes_time_res_plot.py`.

Assume each density measurement is made with relative noise level  $u = \sigma_{n_z}/n_z$ :

$$\sigma_{b_{0.75}}^2 = \frac{u^2 n_z(0.75)^2}{n_z(0)^2} + \frac{n_z(0.75)^2 u^2 n_z(0)^2}{n_z(0)^4} = 2u^2 b_{0.75}^2. \quad (3.42)$$

Note, however, that  $b_{0.75}$  can be estimated from every sample in the exponential decay period. Therefore, the higher the time resolution is (assuming fixed noise level) the better  $b_{0.75}$  can be estimated. Given a sampling interval  $\Delta t$ , there will be of order  $\tau_{\text{imp}}/\Delta t$  points in the decay. Therefore, the uncertainty in the mean of the samples of  $b_{0.75}$  is

$$\sigma_{b_{0.75}} = \sqrt{\frac{2\Delta t}{\tau_{\text{imp}}}} u b_{0.75}. \quad (3.43)$$

Note that this analysis does not take into account the increased precision which comes from having multiple spatial channels; this effect is handled in the next step but is neglected here in order to demonstrate the general characteristics. The expected  $\sigma_{b_{0.75}}$  is shown as a function of  $\Delta t$  and  $u$  in [figure 3.19](#). As shown by the red contour,  $\sigma_{b_{0.75}} < 10^{-2}$  (the level at which  $b_{0.75}$  begins to dominate the estimation of  $D$  and  $V$ ) can be obtained for a wide range of high noise levels and modest sampling rates, indicating that given even a single additional point of spatial resolution, the spatial information in  $b_{0.75}$  combined with the temporal information in  $\tau_{\text{imp}}$  makes the temporal information in  $t_r$  redundant.

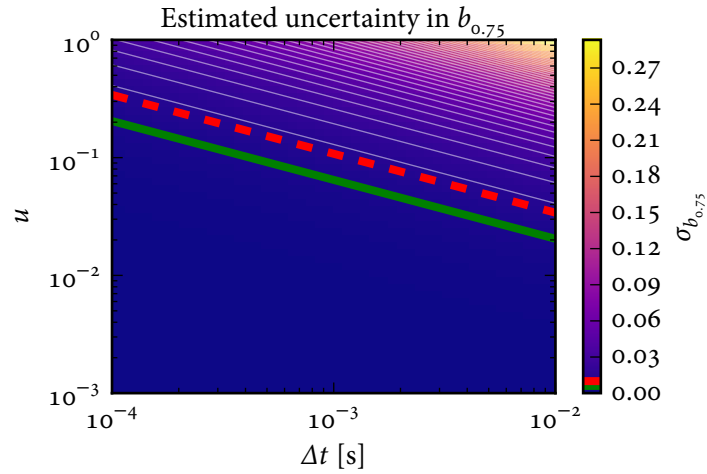
Note, however, that the assumption of Gaussian noise produces several pathologies which have an impact on the interpretation of this and the following analysis. The uncertainty propagation equation as used to obtain [equation \(3.41\)](#) assumes that the distribution of the ratio is approximately Gaussian, when in fact the PDF of  $z = x/y$  where  $x$  and  $y$  are uncorrelated Gaussian random variables is [[436](#), [437](#)]:

$$f_Z(z) = \frac{b(z)d(z)}{a^3(z)} \frac{1}{\sqrt{2\pi}\sigma_x\sigma_y} \left( \Phi\left(\frac{b(z)}{a(z)}\right) - \Phi\left(-\frac{b(z)}{a(z)}\right) \right) + \frac{\exp(-c/2)}{\pi a^2(z)\sigma_x\sigma_y} \quad (3.44)$$

$$a(z) = \sqrt{\frac{z^2}{\sigma_x^2} + \frac{1}{\sigma_y^2}}, \quad b(z) = \frac{\mu_x}{\sigma_x^2} z + \frac{\mu_y}{\sigma_y^2} \quad (3.45)$$

$$c = \frac{\mu_x^2}{\sigma_x^2} + \frac{\mu_y^2}{\sigma_y^2}, \quad d(z) = \exp\left(\frac{b^2(z) - ca^2(z)}{2a^2(z)}\right). \quad (3.46)$$

This is shown along with the assumed Gaussian distribution and the results of a Monte Carlo simulation of the distribution in [figure 3.20](#). The exact distribution is somewhat skewed toward higher values and has a much longer right-hand tail

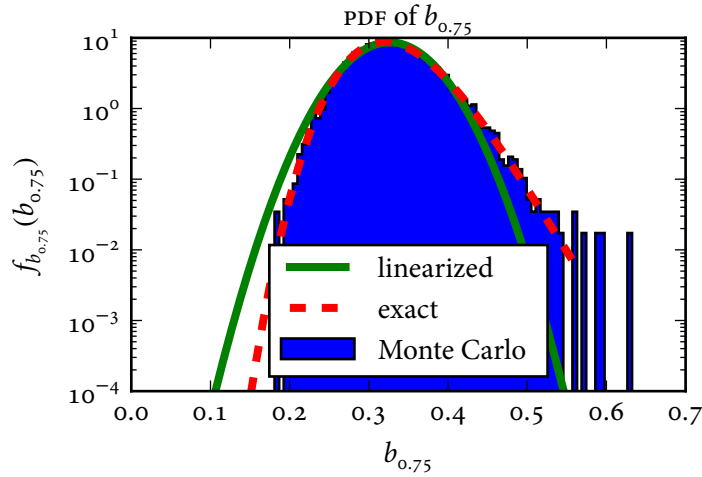


**Figure 3.19:** Uncertainty in the profile broadness  $b_{0.75}$  as a function of the sampling interval  $\Delta t$  and the relative noise level  $u = \sigma_y/y$ . The **red dashed contour** is  $\sigma_{b_{0.75}} = 10^{-2}$ , the point at which  $b_{0.75}$  and  $\tau_{\text{imp}}$  are sufficient to reconstruct  $D$  and  $V$  independent of  $t_r$ . The **green contour** is  $\sigma_{b_{0.75}} = 6 \times 10^{-3}$ , the uncertainty used in figure 3.12. This figure was produced using `make_bayes_time_res_plot.py`.

than the assumed Gaussian distribution. In fact, it can be shown that the right-hand tail is sufficiently long that the mean and variance of this distribution are undefined and the central limit theorem does not hold. This implies two things: first, when interpreting the synthetic data sets used in the next section, standard estimators for the mean and variance will not be reliable because they will be thrown off by tail events. Second, the intuition which led to equation (3.43) is suspect.

In order to verify the intuition behind equation (3.43) a series of Monte Carlo simulations was run. Consider the distribution of the mean and median of collections of  $N \in \{10^1, 10^2, 10^3, 10^4\}$  samples. For each value of  $N$ ,  $10^4$  data sets were generated, and the histograms of the sample means and medians are given in figure 3.21. For small numbers of points the distributions of the mean and median coincide, but for larger numbers of points they begin to diverge because the exact distribution of  $b_{0.75}$  is skewed to the right and has undefined mean and variance. The assumed behavior from equation (3.43) approximately describes the location and width of the distribution of the median, but the actual distribution has longer tails on both sides. The width of the distribution for the mean does appear to go roughly like  $1/\sqrt{N}$ , but the distribution does not converge to  $b_{0.75,0}$ . Therefore, when describing the distribution of  $b_{0.75}$  arising in a synthetic data set, it is necessary to use robust



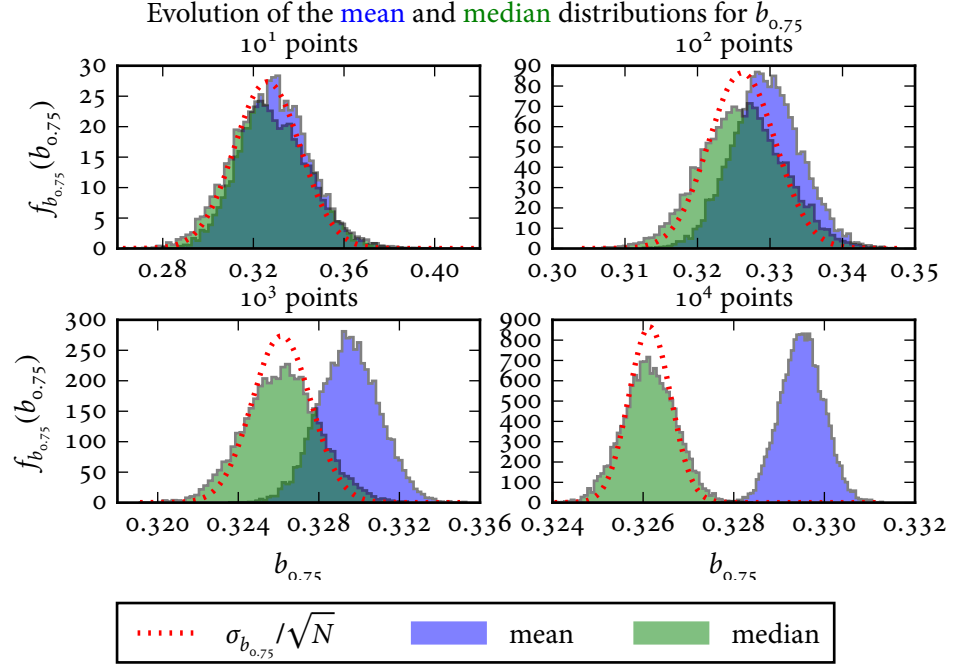


**Figure 3.20:** Assumed Gaussian distribution  $\mathcal{N}(b_{0.75,0}, 2u^2b_{0.75,0}^2)$  from equation (3.41) (green solid curve), exact distribution from equation (3.44) (red dashed curve) and normalized histogram of a Monte Carlo simulation of  $b_{0.75} = n_Z(0.75)/n_Z(0)$ . Note the logarithmic scale, which was used to make the difference in the tails of the distributions visible. The curves were constructed with  $u = 0.1$  and  $b_{0.75,0} = 0.326$  (the value for  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$ ). The Monte Carlo simulation used  $10^4$  random draws for  $n_Z(0.75) \sim \mathcal{N}(b_{0.75,0}, u^2b_{0.75,0}^2)$  and  $n_Z(0) \sim \mathcal{N}(1, u^2)$ . The exact distribution has a longer right-hand tail than the assumed Gaussian distribution and is therefore skewed to larger values. This figure was produced using `make_bayes_time_res_plot.py`.

estimators like the median and interquartile range.

### 3.4.3.2 Estimating the uncertainty in $D$ and $V$ as a function of the system parameters

In order to determine the diagnostic parameters which enable  $D$  and  $V$  to be reconstructed, synthetic data sets were constructed with a variety of sampling rates, noise levels and spatial resolutions. The sampling interval (i.e., the reciprocal of the sampling rate) was varied over the range  $10^{-4} \text{ s} \leq \Delta t \leq 10^{-2} \text{ s}$ , the relative noise level was varied over the range  $10^{-3} \leq u \leq 1$  and the calculation was performed for  $N_s \in \{1, 3, 5, 32\}$  spatial points. The 32 point case corresponds roughly to the spatial resolution of the HIREX-SR XICS instrument on Alcator C-Mod, though note that this analysis does not treat the details of the tomographic inversion necessary to convert



**Figure 3.21:** Normalized histograms of the mean (blue) and median (green) of  $N$  samples of  $b_{0.75}$  along with  $\mathcal{N}(b_{0.75,0}, 2u^2 b_{0.75,0}^2/N)$ , the approximate result from equation (3.43) (red dotted). Each panel represents the results from  $10^4$  synthetic data sets, with each set containing the indicated number of points,  $N$ . This figure was produced using `make_bayes_time_res_plot.py`.

the line-integrated measurements to local measurements.<sup>8</sup> For the single point case the value  $n_Z(o)$  at the core of the plasma was used. For the other cases the points were equally spaced over  $0 \leq r/a \leq 1$ . For each point in  $(\Delta t, u, N_s)$  space,  $10^3$  realizations were constructed. The phase of each realization's synthetic data with respect to the injection was varied by taking the sampling to start at time  $t_o \sim \mathcal{U}(t_{inj} - \Delta t, t_{inj})$ . For  $N_s > 1$ , the ratios  $b_{r/a} = n_Z(r/a)/n_Z(o)$  were included in the inference, and the coefficients of the linearization of  $b_{r/a}(D, V)$  were computed as described above. A lower bound on the noise of  $\sigma_{n_Z} = 10^{-3} \cdot \max(n_Z)$  was used to avoid issues with the calculation of  $t_r$ .

For each realization,  $t_r$  is determined by fitting a weighted cubic smoothing

8. The details of the uniqueness of tomographic inversions are beyond the scope of this thesis, but note that section 3.6.2 demonstrates a successful reconstruction of  $D$  and  $V$  using line-integrated measurements similar to those on Alcator C-Mod.

spline to the noisy synthetic  $n_Z(o, t)$ . This spline is then evaluated over a dense grid of time values to determine the time at which the impurity density reaches its maximum value. This is done instead of using an optimizer both because it runs faster and because the optimizer sometimes fails to find the correct solution.

The impurity confinement time  $\tau_{\text{imp}}$  is found by adding up the data from all of the spatial points to obtain a rough estimate<sup>9</sup> of  $N_Z(t)$ , then fitting an exponential to the data for  $t > t_{\text{inj}} + t_r + 10$  ms. This approach takes into account the increase in precision of the estimate of  $\tau_{\text{imp}}$  which comes from having multiple chords and higher time resolution.

For  $N_s > 1$ , the broadness factors  $b_{r/a}$  are computed by taking the median of  $b_{r/a}(t) = n_Z(r/a, t)/n_Z(o, t)$  over the same set of time points used to fit  $\tau_{\text{imp}}$ . This has the effect of accounting for the improvement in precision on  $b_{r/a}$  that comes from having higher time resolution while avoiding the pathologies discussed in the previous section.

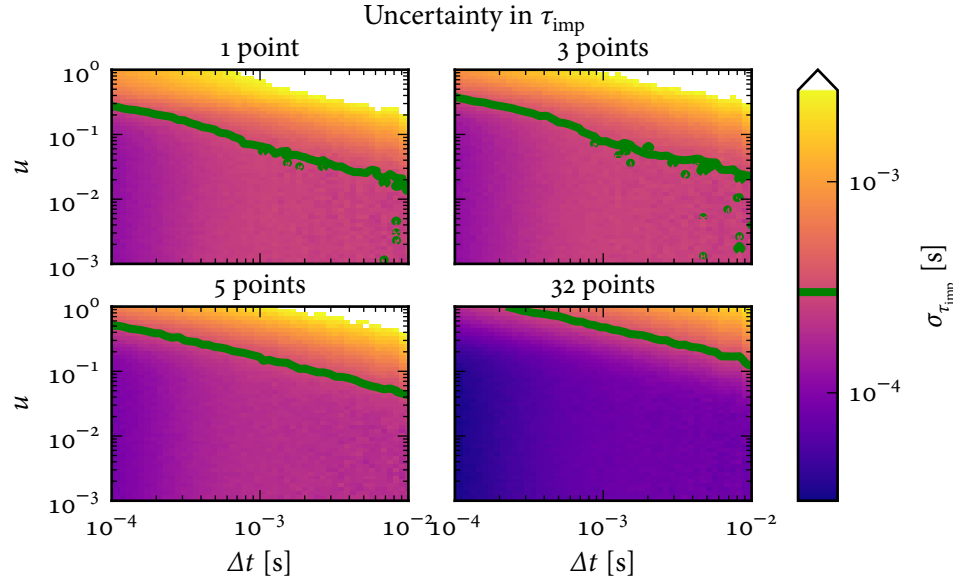
For each point in  $(\Delta t, u, N_s)$  space, the estimate of  $t_r$  is computed as the median of the results from the realizations. The uncertainty in this estimate is given by  $\sigma_{t_r} = 1.349 \cdot \text{IQR}$ , where IQR is the interquartile range (see [section 1.4.2](#)). Robust estimators are used for this case because there were occasionally poor spline fits which produced outliers which threw off the fit. For  $N_s > 1$ , robust estimators are also used to find  $b_{r/a}$  and  $\sigma_{b_{r/a}}$  in order to avoid the pathologies discussed in the previous section. Conventional estimators (mean, standard deviation) are used to obtain  $\tau_{\text{imp}}$  and  $\sigma_{\tau_{\text{imp}}}$ . The uncertainties in  $D$  and  $V$  are then obtained using [equation \(3.34\)](#) and [equation \(3.35\)](#).

Note that all of the following discussion of expected uncertainties in the various observables concerns only the uncertainties arising from random errors in the diagnostic signals. Systematic effects arising from diagnostic calibration errors and variation of the background plasma conditions are likely to dominate the uncertainties in quantities such as  $\tau_{\text{imp}}$  and  $b_{0.75}$  which are determined by averages over many points during the exponential decay period. Unfortunately this framework is unable to handle this class of uncertainties.

The uncertainties in  $\tau_{\text{imp}}$  are given in [figure 3.22](#). As expected, decreasing  $\Delta t$  or  $u$  or increasing  $N_s$  allows  $\tau_{\text{imp}}$  to be determined with a smaller uncertainty.

The uncertainties in  $t_r$  for the single chord case are given in [figure 3.23](#). While the results are somewhat ragged as a result of the distribution being somewhat non-Gaussian because of poor spline fits when determining  $t_r$ , the behavior is as

9. This calculation does not include the quadrature weights needed to be a proper estimate of  $N_Z(t)$ . But, the density at each point goes like  $A_i e^{-t/\tau_{\text{imp}}}$ , so fitting a line to  $\ln \sum_i A_i e^{-t/\tau_{\text{imp}}}$  still yields the correct impurity confinement time.

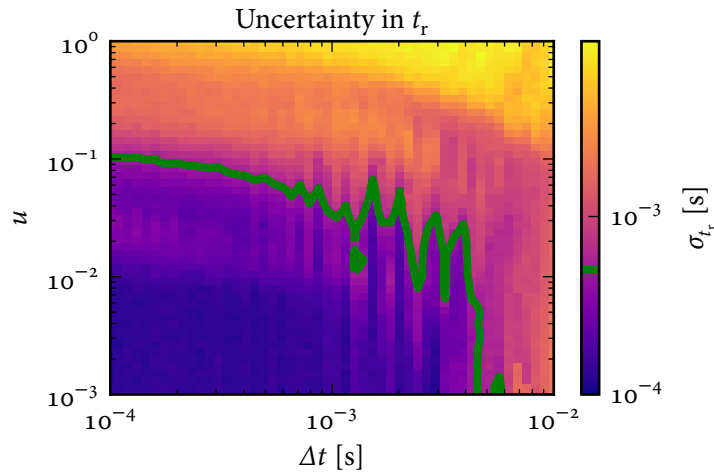


**Figure 3.22:** Uncertainty in  $\tau_{\text{imp}}$  as a function of  $\Delta t$ ,  $u$  and  $N_s$ . The color scale for all of the figures is the same and is set to saturate at the maximum value in the  $N_s = 32$  plot in order to make the changes visible. The green contour corresponds to  $\sigma_{\tau_{\text{imp}}} = 0.3$  ms, the value used in figure 3.12. This figure was produced using `make_bayes_time_res_plot.py`.

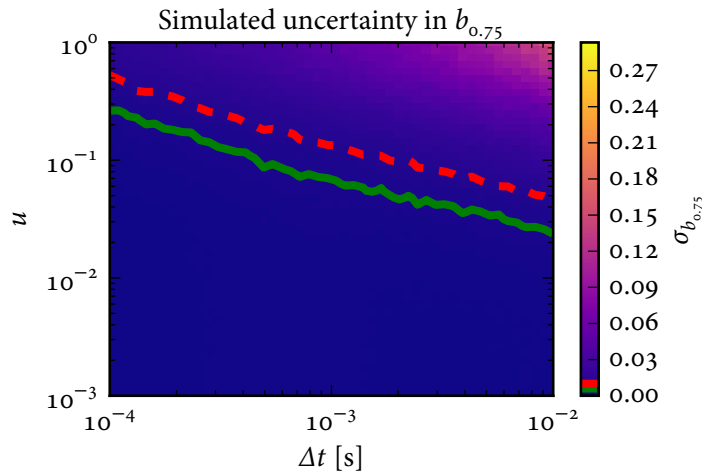
expected: sampling faster or with lower noise allows  $t_r$  to be reconstructed with a smaller uncertainty. But for  $\Delta t \gtrsim 5$  ms it is impossible to adequately reconstruct  $t_r$  accurate to the 0.5 ms level even with very low-noise measurements. The required time resolution is approximately half of the rise time, which makes sense in light of the Nyquist sampling theorem.

The observed uncertainties in  $b_{0.75}$  are given in figure 3.24. The results are in very good agreement with figure 3.19 and equation (3.43). The observed uncertainties are slightly lower than shown in figure 3.19 because data from more than one impurity confinement time were included in the estimate. When the code was run without robust estimators the results were very far off, illustrating the importance of accounting for the pathologies noted in the previous section.

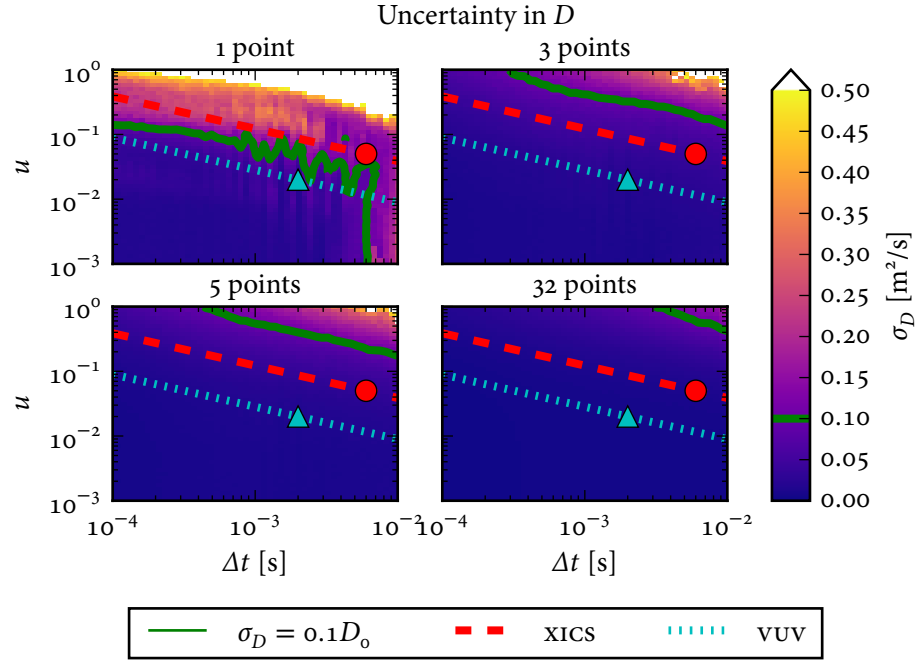
The expected uncertainties in  $D$  and  $V$  are given in figure 3.25 and figure 3.26, respectively. For the  $N_s = 1$  case, the uncertainties are dominated by  $\sigma_{t_r}$ , and hence a very fine time resolution and low noise level is required. For the  $N_s = 3$  case the reliance on  $t_r$  has almost been completely removed, and it is possible to use a much slower sampling rate: the conventional intuition that a high time resolution during



**Figure 3.23:** Uncertainty in  $t_r$  as a function of  $\Delta t$  and  $u$  for  $N_s = 1$ . (Additional chords do not help to constrain  $t_r$ .) The **green contour** corresponds to  $\sigma_{t_r} = 0.5$  ms, the value used in [figure 3.12](#). The ragged shape of the contour is the result of poor fits for  $t_r$  when few points are available in the rise. This figure was produced using `make_bayes_time_res_plot.py`.



**Figure 3.24:** Uncertainty in  $b_{0.75}$  as a function of  $\Delta t$  and  $u$ . The color scale is set to be the same as [figure 3.19](#). The **red dashed contour** is  $\sigma_{b_{0.75}} = 10^{-2}$ , the point at which  $b_{0.75}$  and  $\tau_{\text{imp}}$  are sufficient to reconstruct  $D$  and  $V$  independent of  $t_r$ . The **green contour** is  $\sigma_{b_{0.75}} = 6 \times 10^{-3}$ , the precision used in [figure 3.12](#). This figure was produced using `make_bayes_time_res_plot.py`.



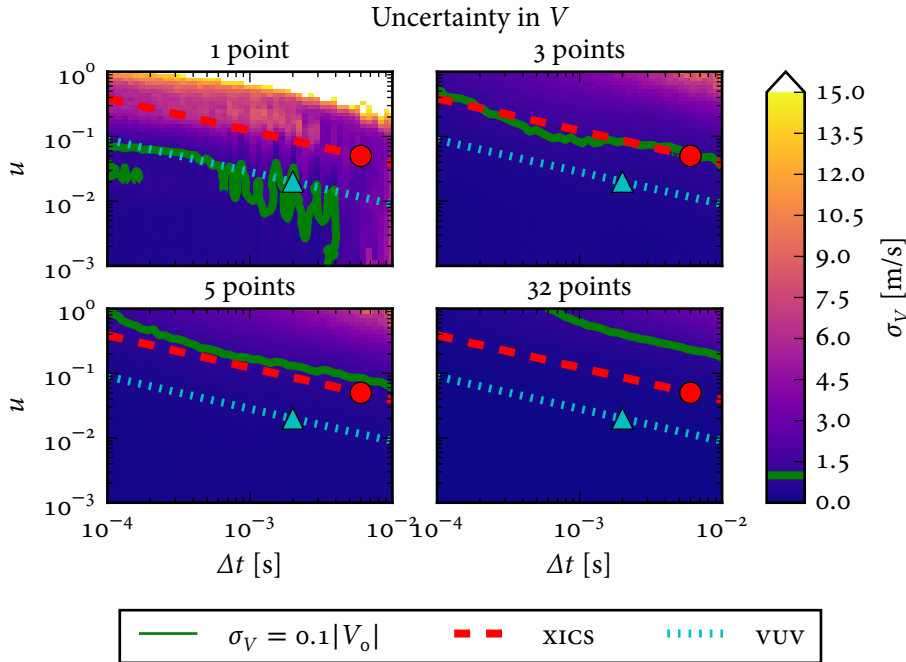
**Figure 3.25:** Uncertainty in  $D$  as a function of  $\Delta t$ ,  $u$  and  $N_s$ . The color scale is the same for all of the subplots and was set to saturate at  $0.5 \text{ m}^2/\text{s}$  in order to make the changes visible. The green contour corresponds to  $\sigma_D = 0.1D_0$ . The red dashed line represents a typical HIREX-SR signal level, and the red circle is the  $\Delta t = 6 \text{ ms}$  time resolution the instrument is normally operated at for impurity transport studies. Similarly, the cyan dotted line and triangle represent a typical VUV spectrometer signal level. This figure was produced using `make_bayes_time_res_plot.py`.

the rise is necessary to separate  $D$  from  $V$  appears to be incorrect, as mediocre time resolution (simply sufficient to estimate  $\tau_{\text{imp}}$ ) combined with modest spatial resolution should be able to reconstruct  $D$  and  $V$  accurate to better than 10%.

In order to gain some intuition for the parameter ranges indicated, consider a diagnostic which is dominated by Poisson noise (i.e., photon statistics): the number of photons reaching the detector is Poisson-distributed, with an average rate of  $\lambda$  photons per second. Therefore, the relative noise level corresponding to data collected during a sampling interval  $\Delta t$  is

$$u = \frac{1}{\sqrt{\lambda \Delta t}}. \quad (3.47)$$

The HIREX-SR diagnostic on Alcator C-Mod typically has a relative uncertainty of



**Figure 3.26:** Uncertainty in  $V$  as a function of  $\Delta t$ ,  $u$  and  $N_s$ . The color scale is the same for all of the subplots and was set to saturate at 15 m/s in order to make the changes visible. The green contour corresponds to  $\sigma_V = 0.1|V_0|$ . The red dashed line represents a typical HIREX-SR signal level, and the red circle is the  $\Delta t = 6$  ms time resolution the instrument is normally operated at for impurity transport studies. Similarly, the cyan dotted line and triangle represent a typical VUV spectrometer signal level. This figure was produced using `make_bayes_time_res_plot.py`.

about 1% to 5% at the 6 ms sampling interval used for impurity transport studies; the relative noise as a function of time resolution extrapolated from this is shown as the dashed red curve in figure 3.25 and figure 3.26. The XEUS VUV spectrometer typically obtains a relative uncertainty of about 1% to 2% with a 2 ms sampling interval; the relative noise for this instrument is shown as the dotted cyan curve. The SXR instruments on Alcator C-Mod sample at 4  $\mu$ s and have enough signal strength to lie off the scale of the figures. The XICS and VUV diagnostics are only just inside the envelope of being able to reconstruct  $D$  and  $V$  to the  $\pm 10\%$  level, but the SXR instruments should be more than capable of delivering this level of precision, provided that the atomic physics model used to infer  $n_Z$  from the SXR emissivities is sufficiently accurate. Therefore, this analysis implies that SXR instruments with

high throughput and high temporal and spatial resolution have the best chance of being able to reconstruct  $D$  and  $V$ . But because of the complexity of the atomic physics model necessary to interpret sXR measurements, spectroscopic diagnostics are invaluable both as an extra signal to constrain the inference and as a cross-check of the results from the sXR measurements using simpler atomic physics.

Another interesting point is that the contours of constant  $\sigma_D$  and  $\sigma_V$  for the cases where  $t_r$  is redundant are also roughly contours of  $1/\sqrt{\lambda\Delta t}$ : changing the averaging of the data makes plots look nicer, but does not add or remove data from the inference. To understand this result, note that both  $b_{r/a}$  and  $\tau_{\text{imp}}$  represent averages over the samples,<sup>10</sup> and hence have uncertainties which are expected to go like  $1/\sqrt{N}$ , where the number of samples  $N$  will go like  $\Delta t$ . Therefore, binning a Poisson-distributed signal over a larger window will increase the precision in the individual measurements by a factor of  $\sqrt{\Delta t}$  while decreasing the precision in the average of those quantities by the same factor.

Additional spatial and temporal resolution will, of course, be necessary to reconstruct more complicated profile shapes for  $D$  and  $V$ . Note that this framework can be used to estimate the requirements to infer profiles with many more parameters than what was shown here, but the curse of dimensionality makes the use of a dense grid in parameter space to evaluate  $y_i(D, V)$  very computationally expensive. A better strategy for such cases would be to compute the derivatives needed for the linearization with just a few STRAHL runs using finite differences, then use a few more runs to check the accuracy of the linearization.

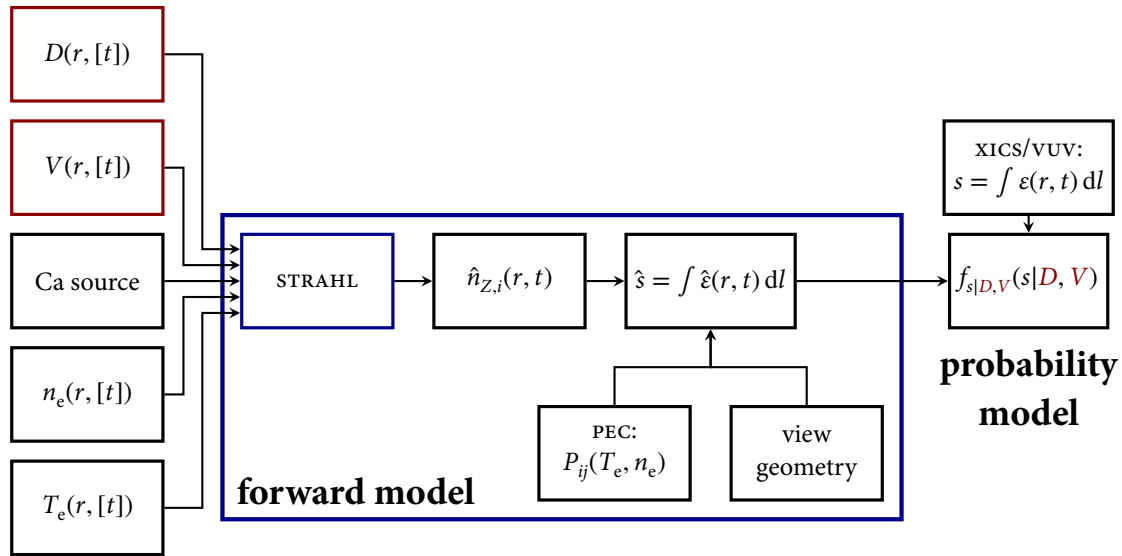
## 3.5 Inferring impurity transport coefficient profiles is an inverse problem

### 3.5.1 The structure of the forward model

The previous sections used the very simple profiles given in equation (3.14) to illustrate the basic features of the impurity transport equation given in equation (3.7). This section expands the problem to inferring the parameters<sup>11</sup> of some arbitrary pair of functions  $D(r)$  and  $V(r)$ . The structure of the problem is shown in figure 3.27:

10. The impurity confinement time represents an average over samples in the sense that the slope of  $\ln N_z$  is estimated from all of the samples as  $1/\tau_{\text{imp}} = -\text{cov}[t, \ln N_z]/\text{var}[t]$ . For homoscedastic uncertainties (which are assumed in the present analysis) the standard deviation of the estimate for  $1/\tau_{\text{imp}}$  goes down like  $1/\sqrt{N}$ .
11. In the rest of this chapter, the notations  $D$  and  $V$  are taken to refer to the complete sets of parameters for the diffusion and convection profiles, respectively.





**Figure 3.27:** Diagram of the forward model and probability model used to infer the impurity transport coefficients  $D$  and  $V$  from spectroscopic measurements  $s = \int \epsilon dl$ . Note that  $s$  may be measured along many chords. The *forward model* converts guesses for  $D$  and  $V$  into predictions of the diagnostic signals using the STRAHL impurity transport code and a synthetic diagnostic. The *probability model* uses the output from the forward model  $\hat{s}$  together with the experimental measurements  $s$  to compute the likelihood function: the probability of the data given the guess for  $D$  and  $V$ .

the plasma is assumed to exhibit behavior described by equation (3.7) for some true profiles  $D(r)$ ,  $V(r)$ . What is measured directly is the line-integrated spatiotemporal evolution of the line radiation emitted by the injected impurity, the impurity source function, and the background  $n_e$  and  $T_e$  profiles. The diagnostic signals  $s$  can be thought of as being functions of  $D$  and  $V$ :  $s(t) = F(D, V)$ . This is therefore an *inverse problem* [438–441]: given measurements of  $s(t)$ , the objective is to find the set of inputs  $D$ ,  $V$  which gave rise to it. Or, in the presence of noise and/or a mismatch between the assumed forms for  $D$  and  $V$  and reality, the objective is to reconstruct the  $D$  and  $V$  which best match the observations.

Note that the function  $F(D, V)$  is referred to as the *forward model*. In order to relate the predicted diagnostic signals  $\hat{s}$  to the observations  $s$  it is also necessary to construct a *probability model* which computes the likelihood of observing  $s$  as a function of  $D$  and  $V$ . Bayesian inference was in fact developed to solve problems like this: it lets us convert the likelihood  $f_{s|D,V}(s|D, V)$  into the posterior distribution

$f_{D,V|s}(D, V|s)$ . A key point of solving inverse problems is that, for a complicated forward model like the one used here, we do not know a priori whether or not  $F$  is uniquely invertible: there may be multiple solutions which describe the data equally well. It is therefore important to use the probability model with an inference procedure which accounts for this possibility.

In practice the diagnostic signals are usually line integrals along various lines of sight:

$$s_i(t) = \int_{l_i} \varepsilon_i(r, t) dl, \quad (3.48)$$

where  $s_i$  is the  $i^{\text{th}}$  diagnostic signal,  $l_i$  is the corresponding diagnostic line of sight and  $\varepsilon_i(r, t)$  is the local emissivity profile integrated against the spectral response of the diagnostic. With measurements like the HIREX-SR XICS and XTOMO SXR diode arrays which have many lines of sight, two approaches can be taken to handle the line-integrated data: the data can be included in the probability model as-is or it can be tomographically inverted before comparing to the forward model. Including the data as-is essentially solves the tomography problem at the same time it solves the inverse problem to get  $D$  and  $V$ . This has the advantage of requiring fewer assumptions and allowing the solution procedure to fully capture the uncertainty that comes from the potential for non-uniqueness in the tomographic inversion, and was therefore used in the previous work on Alcator C-Mod. The disadvantage of this approach is that non-uniqueness makes it much more difficult to solve the inverse problem.

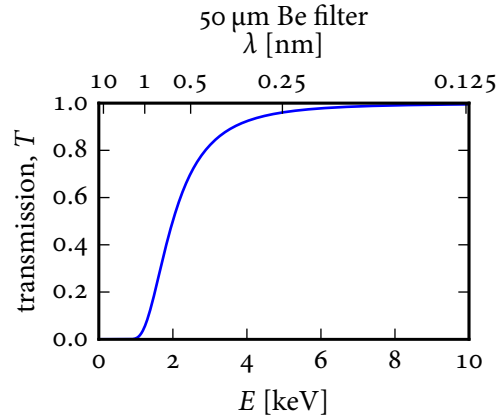
### 3.5.2 The spectral model

The spectral model is a key part of the synthetic diagnostic used to convert the charge state density profiles predicted by STRAHL,  $\hat{n}_{Z,i}(r, t)$ , into local emissivities  $\hat{\varepsilon}(r, t)$  which are then integrated along the relevant line of sight to yield predicted diagnostic signals  $\hat{s}(t)$  according to equation (3.48). There are two parts of the spectral model: one to handle line radiation as seen by spectroscopic diagnostics and one to handle emission seen by broadband soft x-ray instruments.

#### 3.5.2.1 Line radiation

The line radiation model handles collisional excitation from the ground state and simply uses photon emission coefficients (PEC) from the ADAS [442] database to compute the emissivity according to

$$\varepsilon = P_{ij}(T_e, n_e)n_e n_{Z,i}, \quad (3.49)$$



**Figure 3.28:** Transmission of 50  $\mu\text{m}$  beryllium filter used in XTOMO 1, 3 and 5. The lower scale is linearly calibrated in energy units and the upper scale gives the corresponding wavelength units. The data are from [443, 444]. This figure was produced using `plot_xtomo_filter.py`.

where  $P_{ij}$  is the PEC of the  $j^{\text{th}}$  transition of charge state  $i$ . This yields the volumetric rate of photon emission (photons/(s m<sup>3</sup>)), and can be converted to W/m<sup>3</sup> by multiplying the result by  $hc/\lambda_j$ . This is then converted to the signal received at the detector using the line integration capabilities in TRIPPy [362].

### 3.5.2.2 Soft x-ray emission

The model for SXR emission is necessarily more complicated: ensuring one has *all* of the lines from a given impurity in the diagnostic's spectral range is a much more difficult task than analyzing a single transition. For the purposes of generating synthetic data for this thesis, a very basic spectral model was constructed. This will need to be expanded in the ways noted below to adequately handle real experimental data, but the present state of development is sufficient for the construction of synthetic datasets. In particular, bremsstrahlung and recombination radiation were not computed as part of the simplified model: only line radiation is included. The transmission of the 50  $\mu\text{m}$  beryllium filter used in the XTOMO 1, 3 and 5 diagnostics on Alcator C-Mod is shown in figure 3.28.<sup>12</sup> As shown in the figure, only lines with  $E \gtrsim 1$  keV need to be included in the model. This excludes all but the lines from He- and H-like calcium. The edge XTOMO diagnostics (XTOMO 2 and 4) have thinner filters to allow lower energy lines through, but these diagnostics were not included in the analysis

<sup>12</sup>. Note that the response curve of the detector itself is not accounted for in this basic model.

because the radiation from fluorine (recall that  $\text{CaF}_2$  is what is injected in practice) is expected to dominate those signals.

The He-like calcium emission is computed using the extensive collection of collisionally-excited lines obtained from ADAS. Note that this underestimates the forbidden line, whose upper level can be populated by radiative recombination, which is often quite strong in the outer portion of the plasma [420]. This will need to be implemented in order to properly handle experimental data, but was not included in the synthetic data in order to simplify the analysis.

Data from ADAS for the H-like emission were not available and these lines are instead handled using a simple atomic physics model adapted from the IDL program `lines.pro` used for many years at the MIT PSFC [45]. The energy levels of an H-like atom are [445]:

$$E_n = -hcR_M \frac{Z^2}{n^2}, \quad (3.50)$$

where  $n$  is the principal quantum number,  $hcR_M$  is the Rydberg energy for nuclear mass  $M$ :

$$hcR_M = hcR_\infty \frac{M}{M + m_e}, \quad (3.51)$$

and  $hcR_\infty = 13.6 \text{ eV}$  is the Rydberg energy. This means that the transitions from upper level  $n_2$  to lower level  $n_1$  will emit a photon with energy

$$E_{21} = hcR_M Z^2 \left( \frac{1}{n_1^2} - \frac{1}{n_2^2} \right). \quad (3.52)$$

Transitions with the same lower level  $n_1$  are said to form a *series*, with the Lyman series corresponding to  $n_1 = 1$ . To denote a specific member of a series, lowercase Greek letters are used to denote the upper level  $n_2$ . So, the so-called Lyman alpha (“Ly  $\alpha$ ”) line corresponds to  $n_2 = 2$ ,  $n_1 = 1$ . Note that the highest-energy line in a spectral series (the *series limit*) is

$$E_{\infty 1} = hcR_M \frac{Z^2}{n_1^2}. \quad (3.53)$$

The series limits for calcium are given in table 3.1. From this table and figure 3.28, it is clear that only the Lyman and Balmer series need to be considered to model the emission for XTOMO 1, 3 and 5. For simplicity, the spectral model used to produce the synthetic data only includes the Lyman series. Furthermore, the model ignores the fine structure and satellite lines: only the  $1s - np$  transitions are included.

**Table 3.1:** Series limits for calcium.

Name	$n_1$	$E_{\infty 1}$ [keV]
Lyman	1	5.4
Balmer	2	1.4
Paschen	3	0.60
Brackett	4	0.34
Pfund	5	0.22
Humphreys	6	0.15

It is assumed that all upper states are populated by electron impact excitation and then promptly decay through spontaneous emission of a photon. Therefore, the photon emission rate is equal to the excitation rate. Mewe [446] provides the expression

$$S_{ij} \propto \frac{f_{ij} \bar{g}(y) e^{-y}}{T_e^{1/2} E_{ij}}, \quad y = \frac{E_{ij}}{T_e} \quad (3.54)$$

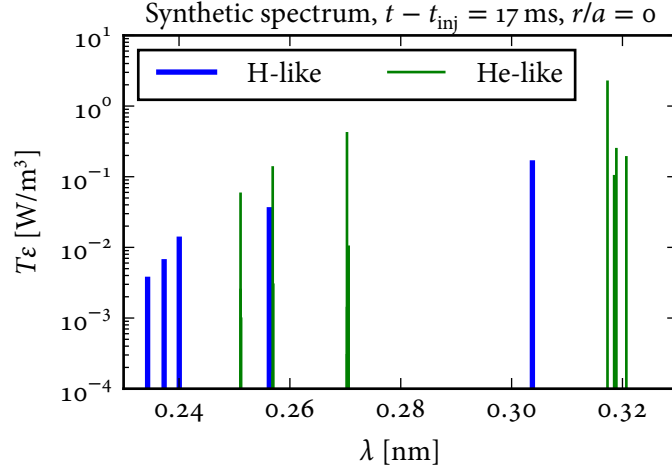
$$\bar{g}(y) = A + (By - Cy^2 + D)e^y E_1(y) + Cy, \quad E_1(y) = \int_1^\infty t^{-1} e^{-yt} dt, \quad (3.55)$$

where  $S_{ij}$  is the excitation rate coefficient,  $f_{ij}$  is the oscillator strength,  $\bar{g}(y)$  is the integrated Gaunt factor,  $E_{ij}$  is the transition energy, and  $A$ ,  $B$ ,  $C$  and  $D$  are coefficients provided in [446]. The oscillator strength is the same along the H-like isoelectronic sequence [447], so the values given for hydrogen in table 3.1(b) of [448] are all that is needed to evaluate equation (3.54). Because  $f_{ij}$  drops off so rapidly at higher energy levels, it is only necessary to include Ly  $\alpha$  through Ly  $\epsilon$ . A simulated spectrum (weighted by the filter transmission) is given in figure 3.29. The synthetic XTOMO diagnostic then adds up the contributions throughout the entire spectral range to yield the total SXR emissivity, which is then integrated along the diagnostic lines of sight using TRIPPy.

### 3.5.3 The probability model

#### 3.5.3.1 The likelihood and the posterior distribution

The noise on the measurements is taken to be Gaussian and independently distributed with heteroscedastic uncertainties  $\sigma_i$ , such that the likelihood of observing any given signal  $s_i$  (in this section  $i$  is taken to be a multi-index which spans both



**Figure 3.29:** Synthetic spectrum weighted by the 50  $\mu\text{m}$  beryllium filter transmission. The spectrum is for  $t - t_{\text{inj}} = 17$  ms,  $r/a = 0$  and was generated using the transport coefficient profiles given in equation (3.14). This figure was produced using `make_synth_spectrum_plot.py`.

sight lines and time points) is given by

$$f_{s_i|D,V,n_e,T_e}(s_i|D, V, n_e, T_e) = \mathcal{N}(\hat{s}_i(D, V, n_e, T_e), \sigma_i^2). \quad (3.56)$$

The likelihood of the entire data set  $\mathbf{s} \in \mathbb{R}^n$  is then

$$f_{\mathbf{s}|D,V,n_e,T_e}(\mathbf{s}|D, V, n_e, T_e) = \frac{1}{(2\pi)^{n/2} \prod_{i=1}^n \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^n \frac{(s_i - \hat{s}_i(D, V, n_e, T_e))^2}{\sigma_i^2}\right). \quad (3.57)$$

In order to obtain the posterior distribution for  $D$  and  $V$  it is necessary to select a prior distribution,  $f_{D,V,n_e,T_e}(D, V, n_e, T_e)$ . Assume independent prior distributions for  $n_e$  and  $T_e$ :

$$f_{D,V,n_e,T_e}(D, V, n_e, T_e) = f_{D,V}(D, V) f_{n_e}(n_e) f_{T_e}(T_e). \quad (3.58)$$

The prior distribution for  $D$  and  $V$  used in practice is designed to restrict  $D$  and  $V$  to physically reasonable values and to impose the conditions  $dD/dr|_{r=0} = 0$  and  $V(0) = 0$ . All of the information about the inference is contained in the posterior

distribution, computed from Bayes' rule:

$$f_{D,V,n_e,T_e|\mathbf{s}}(D, V, n_e, T_e|\mathbf{s}) = \frac{f_{\mathbf{s}|D,V,n_e,T_e}(\mathbf{s}|D, V, n_e, T_e)f_{D,V}(D, V)f_{n_e}(n_e)f_{T_e}(T_e)}{f_{\mathbf{s}}(\mathbf{s})}. \quad (3.59)$$

Assume that  $n_e$  and  $T_e$  are represented as Gaussian processes so their prior distributions are multivariate normal:

$$f_{n_e}(n_e) = f_{n_e|\mathcal{P}}(n_e|\mathcal{P}) = \mathcal{N}(\boldsymbol{\mu}_{n_e|\mathcal{P}}, \boldsymbol{\Sigma}_{n_e|\mathcal{P}}) \quad (3.60)$$

$$f_{T_e}(T_e) = f_{T_e|\mathcal{P}}(T_e|\mathcal{P}) = \mathcal{N}(\boldsymbol{\mu}_{T_e|\mathcal{P}}, \boldsymbol{\Sigma}_{T_e|\mathcal{P}}), \quad (3.61)$$

where  $\mathcal{P}$  represents the observed profile data from Thomson scattering, etc. These prior distributions can be found in a separate step using the procedures described in [chapter 2](#): the mean vectors and covariance matrices are given by [equation \(2.17\)](#).

### 3.5.3.2 Efficient handling of the uncertainties in the $n_e$ and $T_e$ profiles

The machinery given in [section 2.3.8](#) enables these profiles to be represented in a low-dimensional space: [equation \(2.53\)](#) can be used to represent the  $n_e$  and  $T_e$  profiles with a truncated eigendecomposition. Specifically, the vector of  $n_e$  values evaluated on some dense grid is given by

$$\mathbf{n}_e = \mathbf{Q}\boldsymbol{\Lambda}^{1/2}\mathbf{u} + \boldsymbol{\mu}_{n_e|\mathcal{P}}, \quad (3.62)$$

where

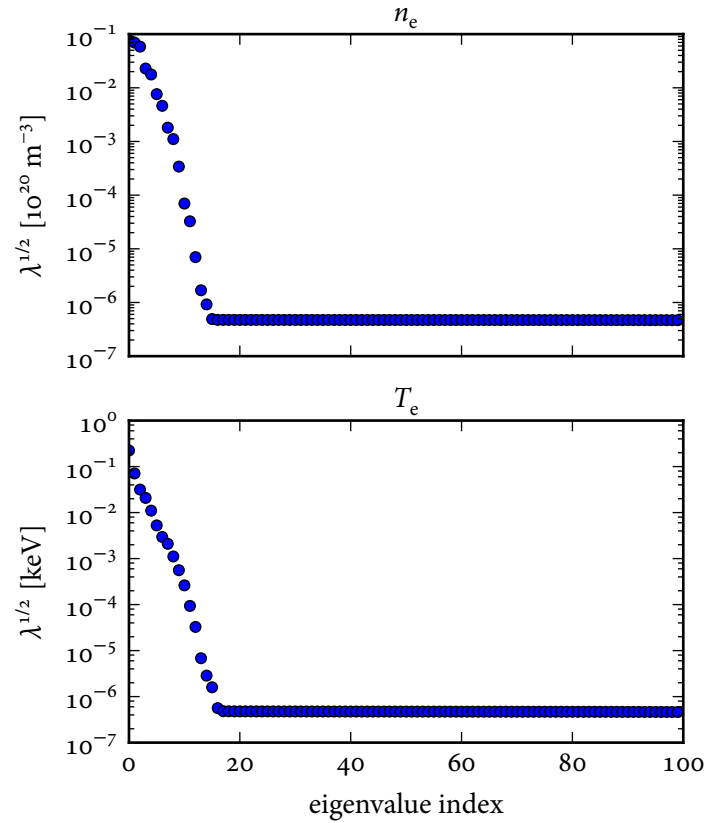
$$\boldsymbol{\Sigma}_{n_e|\mathcal{P}} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1} \quad (3.63)$$

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (3.64)$$

The eigenvalue spectra of the  $n_e$  and  $T_e$  profiles are given in [figure 3.30](#). The spectra decay quite rapidly, indicating that only the first few elements of  $\mathbf{u}$  are needed to represent these profiles. Therefore, instead of needing to use 100 variables with a correlated prior distribution, the  $n_e$  profile can be represented with about four variables having uncorrelated, standard normal prior distributions. Finally, the desired result is the posterior distribution for  $D$  and  $V$  marginalized over  $n_e$  and  $T_e$ :

$$f_{D,V|\mathbf{s}}(D, V|\mathbf{s}) = \int f_{D,V,n_e,T_e|\mathbf{s}}(D, V, n_e, T_e|\mathbf{s}) dn_e dT_e. \quad (3.65)$$

This integral can be performed efficiently using Markov chain Monte Carlo (MCMC) techniques, as described in the next section.



**Figure 3.30:** Eigenvalue spectra for the  $n_e$  and  $T_e$  profiles. The square root of the eigenvalue is shown as this is what determines the extent to which the associated eigenvector can pull the value away from the mean. (Recall that the eigenvectors are normalized.) The spectra correspond to the evaluation of the fitted profile at 100 equally-spaced points. Note that a factor of  $10^3$  times machine epsilon (i.e., the floating point precision) was added to the diagonal of the covariance matrix to stabilize the calculation, which is what causes the spectrum to become a flat line for the higher eigenvalues. This figure was produced using `plot_eigenvalues.py`.



### 3.5.3.3 Handling multiple injections

As was noted in [section 3.1](#), binning multiple injections together to produce “sawtooth-averaged” estimates of transport coefficients is a questionable practice. That being said, there is much utility in being able to combine the data from multiple injections to better constrain the inference. Assuming that an appropriate sawtooth model is included in the forward model, data from multiple injections can be included in [equation \(3.57\)](#) simply by performing multiple STRAHL runs with the appropriate sawtooth timing to properly compute  $\hat{s}(t)$  for each injection. For the sawtooth model, one could either use the model which is already included in STRAHL which simply flattens the impurity density profile inside of the sawtooth inversion radius during each crash [[67](#), [162](#), [185](#), [223](#)], or one could adopt the approach used in [[72](#), [103](#), [117](#), [119](#), [172](#), [173](#), [215](#), [449](#)] and assume that the transport coefficients are transiently increased during the sawtooth crash. This framework would then be capable of inferring both the quiescent transport coefficients as well as the perturbed transport coefficients which occur during the sawtooth crash.

### 3.5.4 Performing the inference

The posterior distribution [equation \(3.65\)](#) contains all of the information that is known about  $D$  and  $V$  given the observations  $\mathbf{s}$  and the prior information contained in  $f_{D,V,n_e,T_e}(D, V, n_e, T_e)$ . In principle, a brute-force evaluation of  $f_{D,V|\mathbf{s}}(D, V|\mathbf{s})$  over the entire parameter space would yield a clear picture of the uncertainty in  $D$  and  $V$ , but this is not a computationally tractable approach.

There are two steps to performing the inference:

**Parameter estimation** is the process whereby the parameters and their uncertainties are estimated. Just like in [section 2.3.6](#), it is possible to either use a point estimate such as the maximum a posteriori (MAP) estimate,

$$\hat{D}^{\text{MAP}}, \hat{V}^{\text{MAP}} = \arg \max_{D,V} f_{D,V|\mathbf{s}}(D, V|\mathbf{s}), \quad (3.66)$$

or an MCMC technique which produces samples  $\{D^{(i)}, V^{(i)}\}$  from which summary statistics can be computed. When using a point estimate, the uncertainties are often taken to be given by the inverse Hessian matrix which is estimated by derivative-based optimizers such as Levenberg-Marquardt and BFGS [[302](#)].

**Model selection** is the process whereby multiple models are compared to select the one with the appropriate level of complexity to describe the observed data. Too

simple of a model will be too inflexible to describe the real data and can lead to an underestimation of the posterior uncertainty. Too complex of a model will infer more structure in the solution than is physically justified and can make the parameter estimation step computationally intractable. While model selection was neglected for the profile fitting task in [section 2.3.6](#), [section 3.6.7](#) shows that it is very important when inferring impurity transport coefficient profiles.

### 3.5.4.1 The previous attempt at obtaining a point estimate and its uncertainty

As described in [section 2.6](#), the previous work [[18](#), [19](#), [55](#), [305](#)] used a point estimate which attempted to account for the uncertainty in  $n_e$  and  $T_e$ . First, an ensemble of  $N$  realizations of  $n_e$  and  $T_e$  was produced using one of the methods described in [section 2.3.8](#). In order to obtain a smooth curve from a piecewise linear spline, the knots for each realization were randomly moved. For each realization, the maximum of the likelihood  $f_{s|D,V,n_e,T_e}(\mathbf{s}|D, V, n_e, T_e)$  was determined by minimizing

$$\chi^2 \propto -2 \ln f_{s|D,V,n_e,T_e}(\mathbf{s}|D, V, n_e, T_e) = \sum_{i=1}^n \frac{(s_i - \hat{s}_i(D, V, n_e, T_e))^2}{\sigma_i^2} \quad (3.67)$$

with  $n_e$  and  $T_e$  held fixed. The mean and variance of the estimates for  $D$  and  $V$  were then found using weighted estimators (shown here just for  $D$ ):

$$\mathbb{E}[D] = \frac{\sum_{i=1}^N w_i \hat{D}_i^{\text{MAP}}}{\sum_{i=1}^N w_i} \quad (3.68)$$

$$\text{var}[D] = \frac{\sum_{i=1}^N w_i (\hat{D}_i^{\text{MAP}} - \mathbb{E}[D])^2}{\sum_{i=1}^N w_i}, \quad (3.69)$$

where the weights are

$$w_i = \frac{\min \chi^2}{\chi_i^2}. \quad (3.70)$$

This procedure suffers from a number of issues:

1. The width of the posterior distribution for any given realization of  $n_e$  and  $T_e$  is not accounted for: this ignores the uncertainties in the spectroscopic measurements as well as the possibility that there are multiple values of  $D$  and  $V$  which describe the data equally well.

2. The knot positions are not inferred in a direct sense: they are only inferred in so far as realizations with worse knot positions will be weighted less in the weighted mean.
3. The use of  $\min \chi^2 / \chi_i^2$  for the weights is not, in general, a valid way of computing the marginalized values of  $D$  and  $V$ . That being said, the solutions generally all had  $\chi^2$  of about the same order of magnitude, so the old approach had the effect of approximating the correct approach given below.

### 3.5.4.2 An improved framework for obtaining a point estimate and its uncertainty

A better approach would be to find the maximum of the conditional posterior density (i.e., the MAP estimate) for each realization of  $n_e$ ,  $T_e$ :

$$f_{D,V|s,n_e,T_e}(D, V|s, n_e, T_e) \propto f_{s|D,V,n_e,T_e}(s|D, V, n_e, T_e) f_{D,V}(D, V) \quad (3.71)$$

$$\ln f_{D,V|s,n_e,T_e}(D, V|s, n_e, T_e) \propto -\frac{1}{2} \sum_{i=1}^n \frac{(s_i - \hat{s}_i(D, V, n_e, T_e))^2}{\sigma_i^2} + \ln f_{D,V}(D, V), \quad (3.72)$$

where the knot positions are included in the parameters of  $D$  and  $V$  (and the best locations are therefore found during the optimization) and it has been assumed that the prior distribution for  $D$  and  $V$  is independent of  $n_e$  and  $T_e$ . Assuming that the distribution is symmetric, the MAP estimate will be equal to the (conditional) posterior mean:

$$\hat{D}^{\text{MAP}}, \hat{V}^{\text{MAP}} = \arg \max_{D,V} f_{D,V|s,n_e,T_e}(D, V|s, n_e, T_e) = \mathbb{E}[D, V|s, n_e, T_e]. \quad (3.73)$$

Given the assumption of symmetry and furthermore assuming there is only one mode, the width of the posterior distribution can be estimated from the inverse of the Hessian matrix estimate by most derivative-based optimizers. This means that we are equipped to estimate  $\text{var}[D, V|s, n_e, T_e]$ , which includes the contributions to the posterior uncertainty from the uncertainties in the diagnostic measurements. The marginalized estimate for  $D$  and  $V$  can then be found from the law of iterated expectations:

$$\mathbb{E}[D, V|s] = \mathbb{E}_{n_e, T_e} [\mathbb{E}[D, V|s, n_e, T_e]] = \frac{1}{N} \sum_{i=1}^N \mathbb{E}[D, V|s, n_{e,i}, T_{e,i}], \quad (3.74)$$

where all samples are weighted equally because  $n_e$  and  $T_e$  were sampled according to  $f_{n_e, T_e | \mathcal{P}}(n_e, T_e | \mathcal{P})$ . The uncertainty is then given by the law of total variance:

$$\begin{aligned} \text{var}[D, V | \mathbf{s}] &= \mathbb{E}_{n_e, T_e} [\text{var}[D, V | \mathbf{s}, n_e, T_e]] + \text{var}_{n_e, T_e} [\mathbb{E}[D, V | \mathbf{s}, n_e, T_e]] \quad (3.75) \\ &= \frac{1}{N} \sum_{i=1}^N \text{var}[D, V | \mathbf{s}, n_{e,i}, T_{e,i}] \\ &\quad + \frac{1}{N-1} \sum_{i=1}^N (\mathbb{E}[D, V | \mathbf{s}, n_{e,i}, T_{e,i}] - \mathbb{E}[D, V | \mathbf{s}])^2. \quad (3.76) \end{aligned}$$

While this approach rectifies the shortcomings identified in the previous section, it still suffers from the fact that free-knot splines lead to non-identifiable posterior distributions (see [section 3.6.4](#)) and fails to account for the possibility that the posterior distribution is multimodal. Furthermore, it is shown in [section 3.6.5](#) that the contribution of  $n_e$  and  $T_e$  to the posterior uncertainty is minimal: the previous results can be explained purely in terms of the different choices of knot locations.

### 3.5.4.3 Fully characterizing the posterior distribution using multimodal nested sampling

The approach outlined in the previous section is only valid when the posterior distribution is unimodal and approximately Gaussian. In practice, application of the MAP estimate fails to yield unique solutions and does not rectify the problems discussed in [section 3.1](#). Therefore, it is necessary to fully characterize the posterior distribution using Markov chain Monte Carlo (MCMC) sampling. As described in [section 2.3.6](#), MCMC techniques can be used to draw samples from unnormalized probability distributions such as the posterior distribution [equation \(3.59\)](#). These samples can then be post-processed to yield marginalized estimates such as [equation \(3.65\)](#).

The affine-invariant ensemble sampler [[353](#), [354](#)] which was used for profile fitting with Gaussian process regression proved to be very inefficient for the high-dimensional, potentially multi-modal posterior distributions encountered when inferring  $D$  and  $V$ . Parallel tempering [[450](#)] and adaptive parallel tempering [[451](#)] capabilities built around the affine-invariant ensemble sampler were also tested, but these approaches were far too computationally expensive to yield useful results. The MULTINEST algorithm [[452–455](#)] was found to have better performance than any of the ensemble sampler-based alternatives, but still struggled when confronted with some of the high-dimensional, non-identifiable cases.<sup>13</sup> The MULTINEST algorithm

13. Refer to [[456](#)] for a comparison of the Metropolis-Hastings, affine-invariant ensemble sampling and MULTINEST algorithms applied to various problems.

has the advantage that it is built specifically to handle non-identifiable and/or multimodal posterior distributions. Furthermore, the algorithm computes the evidence  $f_{\mathbf{s}|\mathcal{M}}(\mathbf{s}|\mathcal{M})$  of the model  $\mathcal{M}$  (i.e., the specific choice of basis functions, knot locations, etc.), which can then be used to directly select the best model (see [section 3.6.6](#)).

Just like in [chapter 2](#), using MCMC<sup>14</sup> to marginalize over the parameters of the model is the most rigorous way to get an accurate estimate of the uncertainty. Furthermore, it offers an easy path forward to marginalize over other nuisance parameters such as uncertainties in the atomic physics parameters [457] and calibration constants for absolutely-calibrated detectors.

## 3.6 Application to synthetic data

The following sections test the MULTINEST-based approach to various synthetic datasets. [Section 3.6.1](#) proves the concept using local measurements of the total impurity density and compares the results of the full model to the linearized model obtained in [section 3.4](#). [Section 3.6.2](#) proves the concept using line-integrated measurements equivalent to the diagnostic situation on Alcator C-Mod. [Section 3.6.3](#) then uses the line-integrated synthetic data to confirm the hypothesis from [section 3.4.3.2](#) that temporal resolution is unimportant given sufficient spatial resolution. [Section 3.6.4](#) demonstrates the pathologies which can arise from the use of free knot splines. [Section 3.6.5](#) demonstrates the surprising result that the background  $n_e$  and  $T_e$  profiles have little effect on the inferred  $D$  and  $V$  profiles and explains the previous results in terms of the sensitivity of the solution to the knot locations. [Section 3.6.6](#) demonstrates the use of the model evidence to select the model complexity in a very simple case, and [section 3.6.7](#) applies the whole framework to synthetic data which matches the complexity inferred by the previous work.

### 3.6.1 Basic case with local impurity density measurements

A synthetic data set was constructed using the basic transport coefficient profiles given in [equation \(3.14\)](#). The total impurity density  $n_Z$  was measured at five points linearly spaced over  $0 \leq r/a \leq 1$ . The sampling interval was  $\Delta t = 6$  ms and the noise level was 5%, consistent with the HIREX-SR XICS. As shown in [figure 3.26](#), this should be capable of reconstructing  $D$  and  $V$  accurate to better than 10% uncertainty. The prior distribution used was

$$f_{D,V}(D, V) = \mathcal{U}(D; 0 \text{ m}^2/\text{s}, 10 \text{ m}^2/\text{s})\mathcal{U}(V; -100 \text{ m/s}, 10 \text{ m/s}). \quad (3.77)$$

14. Technically MULTINEST is not a pure MCMC sampler, but the term is used loosely in this context to refer to techniques which produce samples from the posterior distribution.

**Table 3.2:** Summary statistics for  $D$  and  $V$  under various conditions.

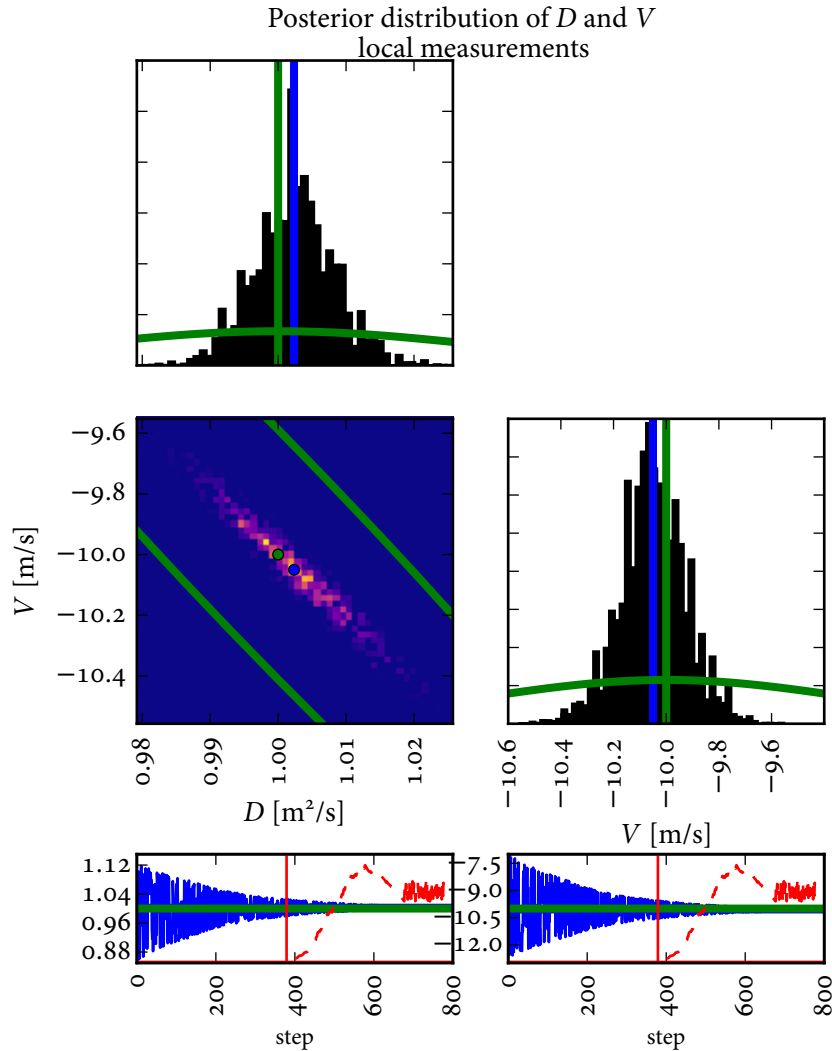
Case	Parameter [units]	Mean	95% interval
Basic	$D$ [ $\text{m}^2/\text{s}$ ]	1.002	[ 0.9911, 1.014]
	$V$ [ $\text{m}/\text{s}$ ]	-10.05	[ -10.31 , -9.797]
Line-integrated	$D$ [ $\text{m}^2/\text{s}$ ]	1.000	[ 0.9957, 1.006]
	$V$ [ $\text{m}/\text{s}$ ]	-10.01	[ -10.12 , -9.916]
No rise	$D$ [ $\text{m}^2/\text{s}$ ]	1.011	[ 0.9977, 1.025]
	$V$ [ $\text{m}/\text{s}$ ]	-10.23	[ -10.53 , -9.951]
Free $n_e, T_e$	$D$ [ $\text{m}^2/\text{s}$ ]	0.9981	[ 0.9906, 1.005]
	$V$ [ $\text{m}/\text{s}$ ]	-9.966	[ -10.14 , -9.777]

**Table 3.3:** Comparison of uncertainty estimates from the linearized Bayesian analysis and MULTINEST. The standard deviations are five times smaller for the full MULTINEST-based analysis, but the correlation coefficient is the same.

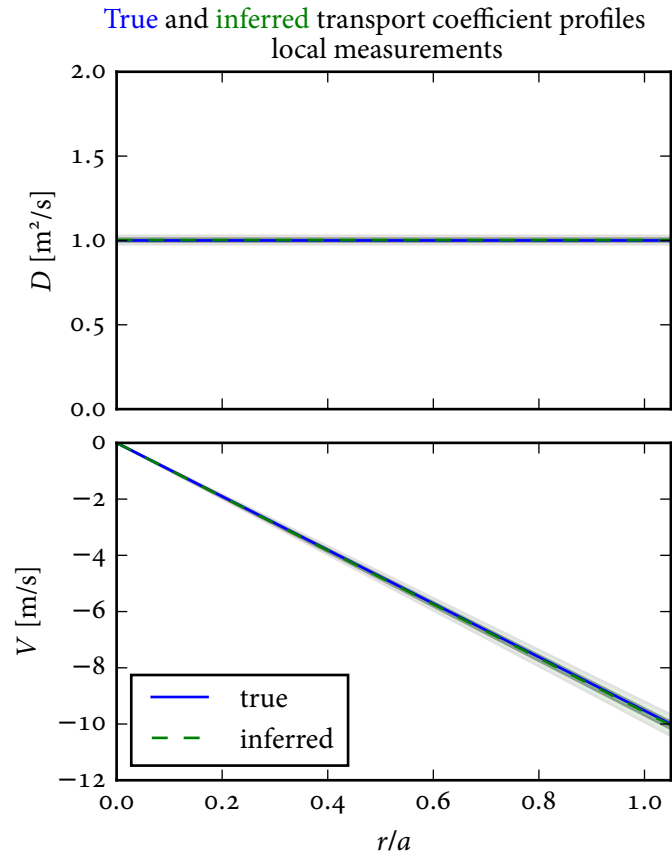
Method	$\sigma_D$ [ $\text{m}^2/\text{s}$ ]	$\sigma_V$ [ $\text{m}/\text{s}$ ]	$\sigma_{DV}$ [ $\text{m}^3/\text{s}^2$ ]	$\rho_{DV}$
Linearized	0.030	0.69	-0.020	-0.97
MULTINEST	0.0059	0.13	-0.00072	-0.97

The MULTINEST sampler was used with 100 live points, the default sampling efficiency of 0.8 and the default tolerance of 0.5 on the log-evidence. The posterior distribution is shown in [figure 3.31](#), the marginalized posterior  $D$  and  $V$  profiles are given in [figure 3.32](#) and summary statistics for the posterior distribution are given in the first block of [table 3.2](#) (labelled “Basic”).

The uncertainty estimates from the linearized analysis and the full analysis are given in [table 3.3](#). As shown in the tables and [figure 3.32](#), the linearized analysis is pessimistic, and the transport coefficients are known to about five times higher precision than was expected from the linearized analysis. It makes sense that the full analysis would have smaller uncertainty as it includes *all* of the data. The correlation between  $D$  and  $V$  is the same, however: the result is a tilted ellipse similar to what was predicted in [figure 3.12](#). This confirms that the linearized model gets the uncertainties correct to within an order of magnitude, and that it captures the correct intuition about which terms have the most influence on the result.

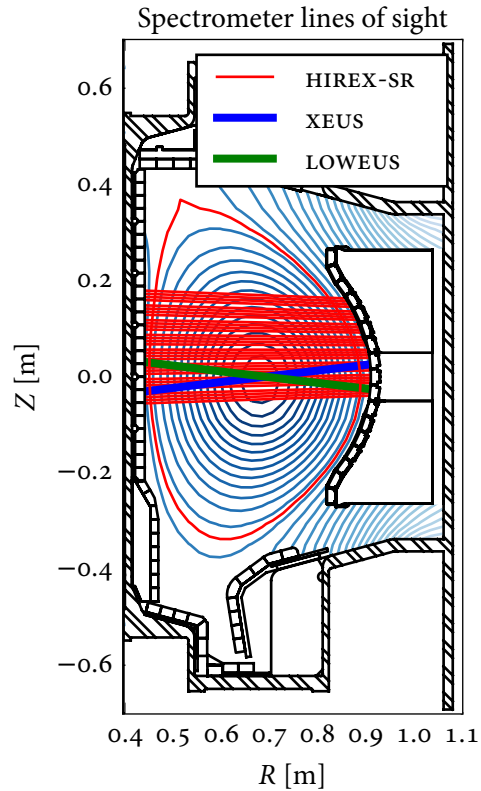


**Figure 3.31:** Posterior distribution for  $D$  and  $V$  inferred from five local measurements. MULTINEST's outputs are slightly different from those of the affine-invariant ensemble sampler used in chapter 2. The posterior samples are given as samples combined with weights representing the posterior density of that sample. The samples are shown as the blue solid curve at the bottom and the weights are shown as the red dashed curve at the bottom. Samples with weights less than  $10^{-2}$  times the maximum weight were not included in the histogram so as to focus on the center of the distribution. This is indicated by the vertical red bar in the bottom plots. The posterior mean is shown as the blue point and vertical lines and the true value is shown as the green point and vertical lines. The green curves in each of the plots correspond to the linearized estimate (equation (3.34) through equation (3.36)): in the univariate plots the green curve is the predicted PDF and in the bivariate plot the two green lines are the edges of the 95% confidence ellipse, similar to what was shown in figure 3.12. The uncertainties estimated from the full model are about five times smaller than those estimated from the linearized analysis, but the correlation between  $D$  and  $V$  (i.e., the angle of the ellipse) is identical. This figure was produced using `settings_1101014006_3_basic.py` and `post_process_basic_MCMC.py`.



**Figure 3.32:** Marginalized posterior  $D$  and  $V$  profiles inferred from five local measurements. The true (blue solid line) and inferred (green dashed line with  $\pm 1\sigma$  and  $\pm 3\sigma$  envelopes) profiles overlay very well and the uncertainty envelopes are rather narrow: there are only a few combinations of  $D$  and  $V$  which fit the data. This figure was produced using `settings_1101014006_3_basic.py` and `post_process_basic_MCMC.py`.





**Figure 3.33:** Spectrometer lines of sight for Alcator C-Mod shot 1101014006, the shot used as the basis for the synthetic impurity transport data. HIREX-SR has 32 lines of sight organized in eight bundles. XEUS and LOWEUS only have a single chord each. This figure was produced using `plot_hirex_chords.py`.

### 3.6.2 Basic case with line-integrated spectral measurements

In order to assess the effect of using line-integrated, spectrally-resolved measurements, the synthetic diagnostic described in section 3.5.2 was used to produce synthetic spectroscopic data. All 32 chords of the HIREX-SR XICS were used, viewing the 0.32 nm He-like calcium line with 6 ms time resolution. The XEUS VUV spectrometer was represented using a single chord viewing seven Li-like calcium lines covering  $1.8 \text{ nm} \leq \lambda \leq 2.0 \text{ nm}$ . The LOWEUS VUV spectrometer was represented using a single chord viewing the 19 nm Be-like calcium line. Both of the VUV instruments were represented as having 2 ms time resolution. The spectrometer lines of sight are shown in figure 3.33. Synthetic XTOMO SXR data were *not* included. The results

are given in [figure 3.34](#) and the second block (labelled “Line-integrated”) of [table 3.2](#). With the extra spatial resolution provided by the 32 HIREX-SR chords and the two VUV chords, the uncertainties on  $D$  and  $V$  are very small. The uncertainty envelopes on  $D$  and  $V$  are too small to be visible, so a plot of the marginalized profiles is not given.

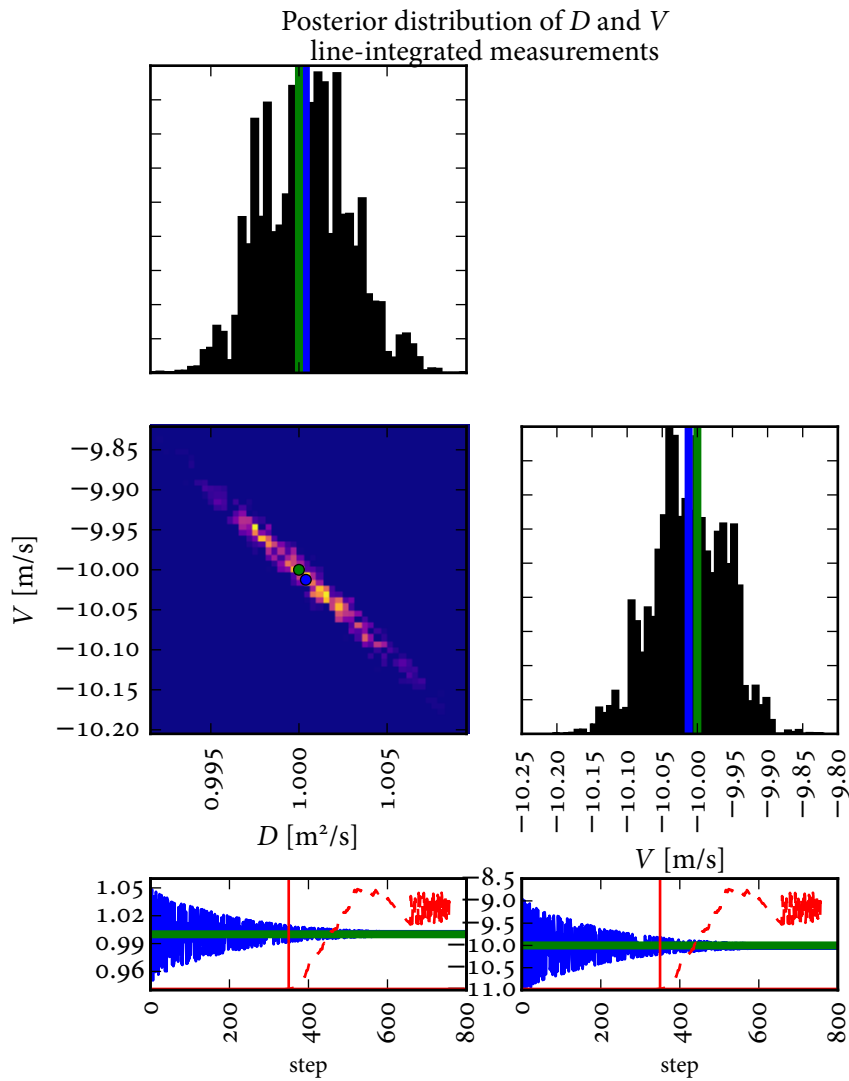
### 3.6.3 Resolving the rise does not matter

In order to verify the hypothesis that the time resolution (and therefore the ability to resolve the details of the rise) does not matter in the presence of sufficient spatial resolution, a synthetic data set including only the data from 20 ms after the injection and on was constructed. The diagnostic configuration was otherwise the same as the previous section in order to represent the situation on Alcator C-Mod. The results are given in [figure 3.35](#) and the third block (labelled “No rise”) of [table 3.2](#). The uncertainties are higher than they were in the case where the rise is resolved and the posterior mean does not quite match the true value, but the 95% posterior interval encompasses the true value and the parameters  $D$  and  $V$  are uniquely constrained to lie in reasonable posterior intervals. This has a critical implication for experiment design: having spatial resolution replaces the information which would be conveyed by having enough temporal resolution to resolve the rise. Therefore, instead of binning injections in the manner used for previous work on Alcator C-Mod, it should be possible to use a single injection combined with a proper sawtooth model to perform the inference.

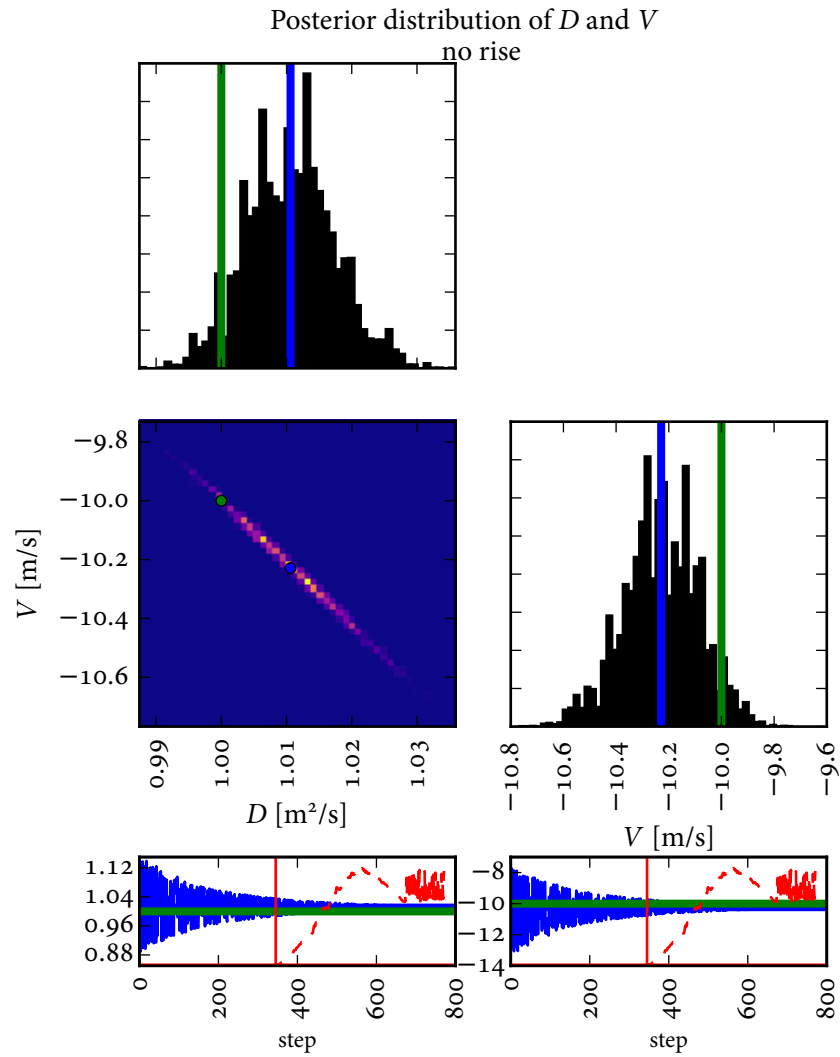
### 3.6.4 Free knots produce non-identifiable posterior distributions

In order to obtain a flexible way of parameterizing the  $D$  and  $V$  profiles while still being able to exercise control over the allowable amount of spatial structure, it is tempting to use a free-knot spline. Here, the (internal) knot locations are treated as free parameters to be inferred along with the spline coefficients.<sup>15</sup> This can, however, lead to pathologies which make the inference much more difficult. To see this, assume that the basic  $D$  and  $V$  profiles as shown in [figure 3.1](#) are the true profiles. But at the start of the inference this is not known. Consider the use of a free knot spline with one internal knot at location  $r/a = t$  with prior distribution  $f_t(t) = \mathcal{U}(t; 0, a_c/a)$ . The value of each transport coefficient profile is then to be inferred at two points  $r/a = t$  and  $r/a = a_c/a$ . It is clear that the exact same profile for  $D$  results for any  $0 < t_D < a_c/a$  as long as  $D(t_D) = D(a_c/a)$ . Likewise, the exact same profile for  $V$

15. For the piecewise linear profiles used here and in the previous work, specifying the spline coefficients is accomplished by specifying the value at each knot.



**Figure 3.34:** Posterior distribution for  $D$  and  $V$  inferred from line-integrated measurements representing the diagnostic situation on Alcator C-Mod. The figure is drawn as in figure 3.31. The posterior mean is shown as the blue line and the true value is shown as the green line. This figure was produced using `settings_1101014006_14_basic_line_integ.py` and `post_process_basic_line_integral_MCMC.py`.



**Figure 3.35:** Posterior distribution for  $D$  and  $V$  inferred from line-integrated measurements which only have data from the decay portion. The posterior mean is shown as the blue line and the true value is shown as the green line. This figure was produced using `settings_1101014006_15_basic_line_integ_no_rise.py` and `post_process_basic_line_integral_no_rise_MCMC.py`.

results for any  $0 < t_V < a_c/a$  as long as  $V(t_V)$  moves up and down so as to form a line with no break in slope. Therefore, a free knot spline such as this is non-identifiable: no set of observations can uniquely determine values for  $t_D$ ,  $D(t_D)$ ,  $t_V$  and  $V(t_V)$ .

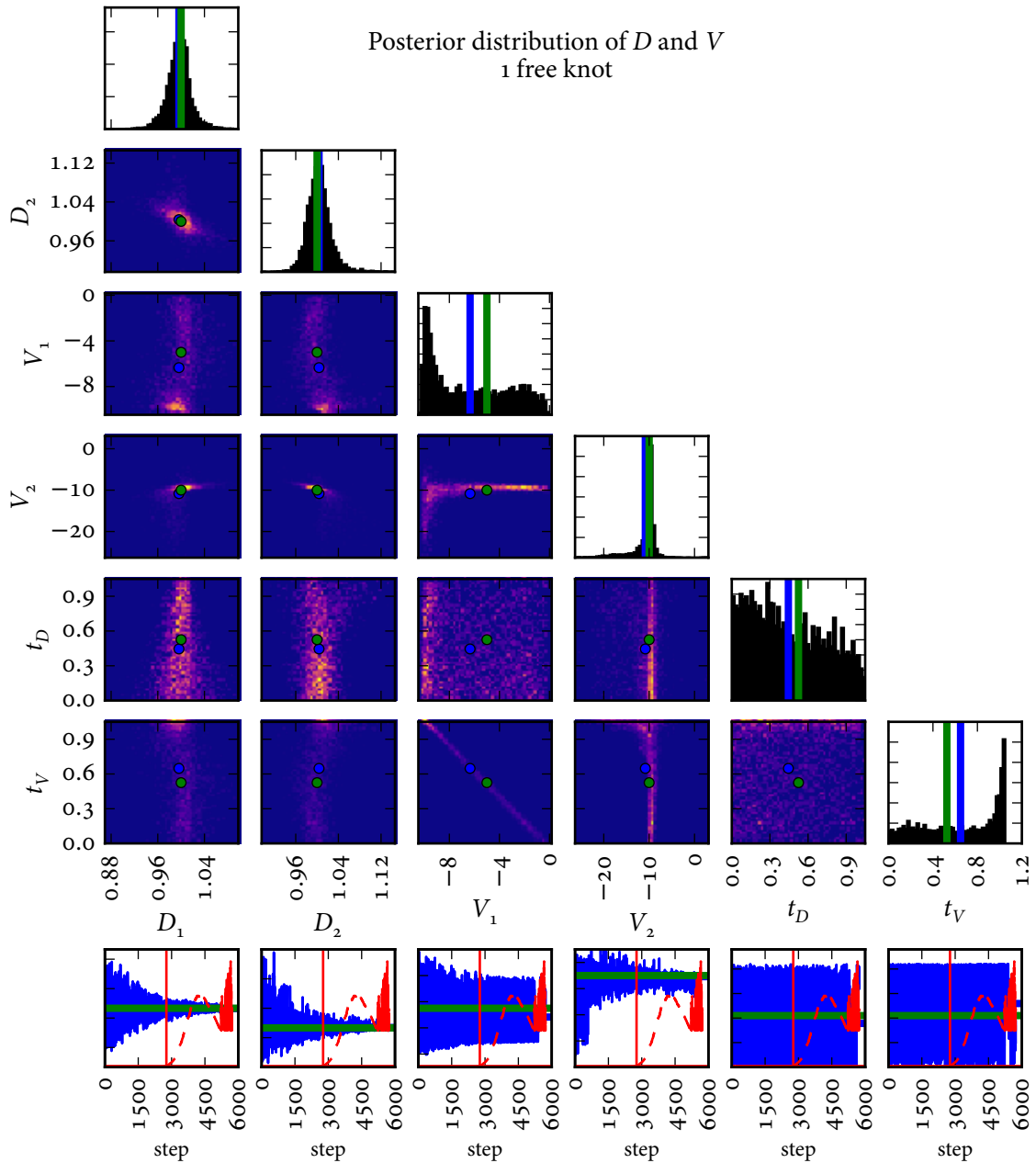
To illustrate this, the inference was performed using the profiles shown in [figure 3.1](#) and five local measurements. Performing the inference required 500 active points and was rather computationally expensive. The results are shown in [figure 3.36](#). As expected, there are large regions of the posterior distribution with constant posterior density indicating that this combination of parameters is not identifiable. It would be expected that the marginal posterior distributions for  $D_1$ ,  $V_1$ ,  $t_D$  and  $t_V$  should be uniform. But there is a large number of points with  $t_V \approx a_c/a$ ,  $V_1 \approx V_2$ . This is either an artifact of MULTINEST failing to adequately sample such a non-identifiable posterior distribution or a result of the simulated diagnostic noise causing the fit to infer some extra structure/uncertainty in the edge. This is illustrated by [figure 3.37](#) which shows the marginalized  $D$  and  $V$  profiles for this case: the uncertainty envelope on  $V$  spreads out considerably at the edge. In either case, the implication of this analysis is clear: avoiding free-knot splines (or putting a strong prior distribution on the knot locations) is necessary for the inference problem to be tractable.

### 3.6.5 The uncertainty attributable to $n_e$ and $T_e$ is minimal

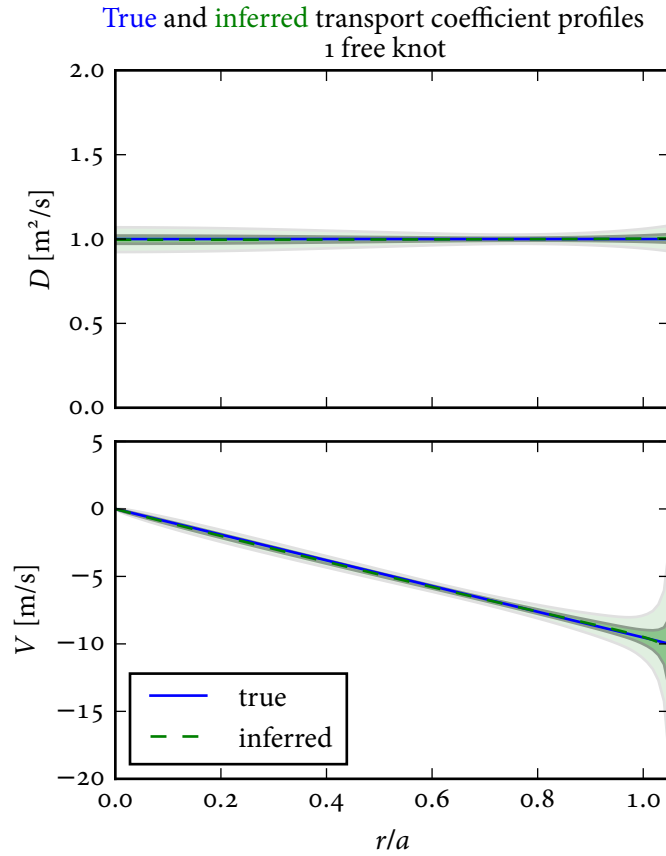
#### 3.6.5.1 Demonstration with synthetic data

In order to demonstrate the ability to marginalize over the  $n_e$  and  $T_e$  profiles using MCMC, the inference for the line-integrated data (including the rise) was repeated with the temperature and density profiles allowed to vary. The truncated eigendecomposition described in [section 3.5.3.2](#) was used. Informed by [figure 3.30](#), the first three eigenvalues for each profile were retained. The number of active points was increased to 200 to get a better picture of the posterior distribution. The posterior distribution is shown in [figure 3.38](#), the marginalized  $D$  and  $V$  profiles are shown in [figure 3.39](#) and summary statistics are given in the fourth block (labelled “Free  $n_e$ ,  $T_e$ ”) in [table 3.2](#). The posterior  $n_e$  and  $T_e$  profiles are shown in [figure 3.40](#).

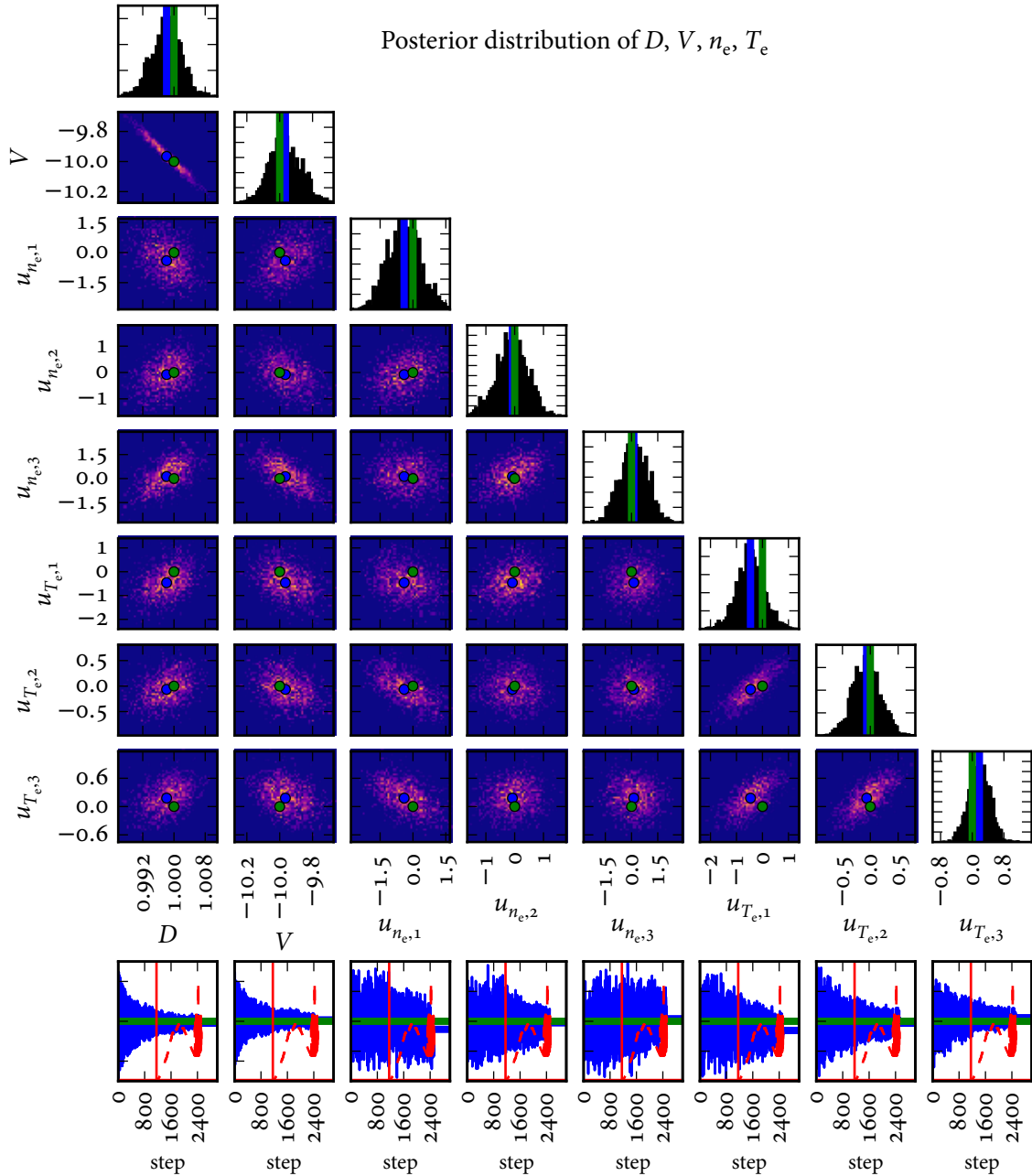
As shown in the figures and the table, the effect of  $n_e$  and  $T_e$  on the posterior uncertainty in  $D$  and  $V$  is small. To further bolster this claim, the posterior correlation matrix of the parameters is shown in [figure 3.41](#). There is, in general, somewhat weak correlation between the weights for the  $n_e$  and  $T_e$  profiles and  $D$  and  $V$ , with the strongest correlation ( $\rho = -0.69$ ) being between  $V$  and  $u_{n_e,3}$  and the second highest ( $\rho = 0.58$ ) between  $D$  and  $u_{n_e,3}$ . Note, however, that the correlation between  $D$  and  $V$  is very high compared to this ( $\rho = -0.98$ ).



**Figure 3.36:** Example of non-identifiable posterior distribution resulting from the use of a free-knot spline. The parameters are  $D_1 = D(t_D)$ ,  $D_2 = D(a_c/a)$ ,  $V_1 = V(t_V)$ ,  $V_2 = V(a_c/a)$  plus the knot locations  $t_D$  and  $t_V$ . The posterior mean is shown as the **blue points** and the true value is shown as the **green points**. The posterior distribution exhibits the classic symptoms of non-identifiable parameters: there are large areas of the posterior distribution with constant posterior density. This figure was produced using `settings_1101014006_18_free_knots_local.py` and `post_process_free_knots_local_MCMC.py`.

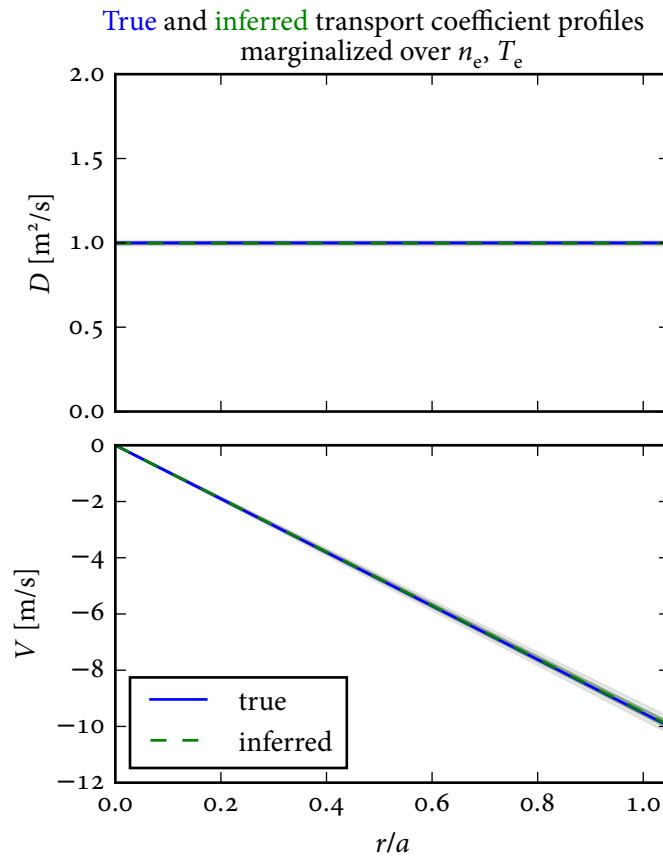


**Figure 3.37:** Marginalized  $D$  and  $V$  profiles for the case with one free internal knot and local measurements. As shown in figure 3.36, the posterior distribution for the convective velocity profile's knot  $t_V$  is peaked around  $r/a = 1$  which allowed extra flexibility and hence uncertainty there, consistent with the presumed lack of knowledge about the outer region of the plasma. This figure was produced using `settings_1101014006_18_free_knots_local.py` and `post_process_free_knots_local_MCMC.py`.

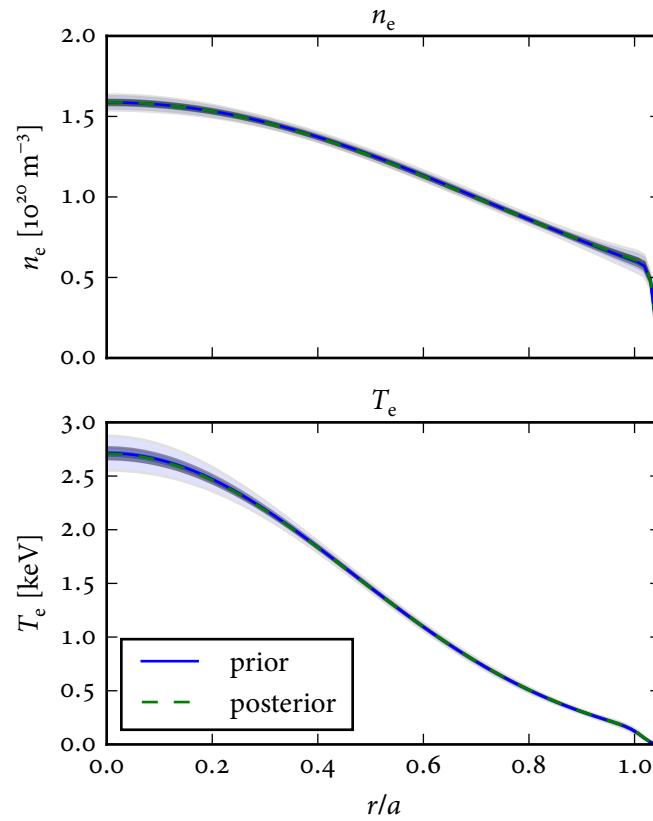


**Figure 3.38:** Posterior distribution for  $D, V, n_e, T_e$  inferred from line-integrated spectroscopic measurements. The posterior mean is shown as the [blue points](#) and the true value is shown as the [green points](#). The effect of  $n_e$  and  $T_e$  on the posterior uncertainty in  $D$  and  $V$  is minimal. This figure was produced using `settings_1101014006_16_ne3_Te3_line_integ.py` and `post_process_ne3_Te3_line_integ_MCMC.py`.

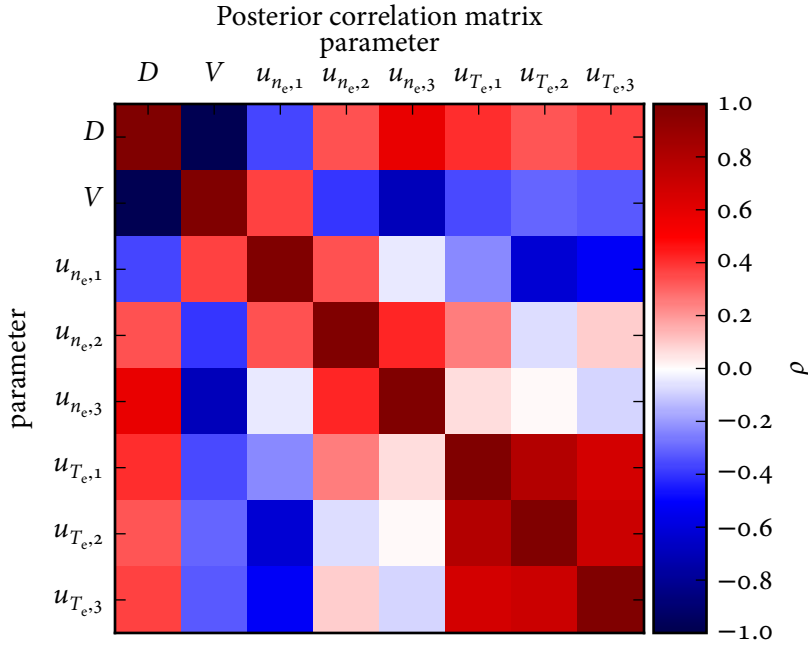




**Figure 3.39:** Posterior  $D$  and  $V$  profiles inferred from line-integrated spectroscopic measurements and marginalized over  $n_e$  and  $T_e$ . The effect of  $n_e$  and  $T_e$  on the marginalized profiles is minimal. This figure was produced using `settings_1101014006_16_ne3_Te3_line_integ.py` and `post_process_ne3_Te3_line_integ_MCMC.py`.

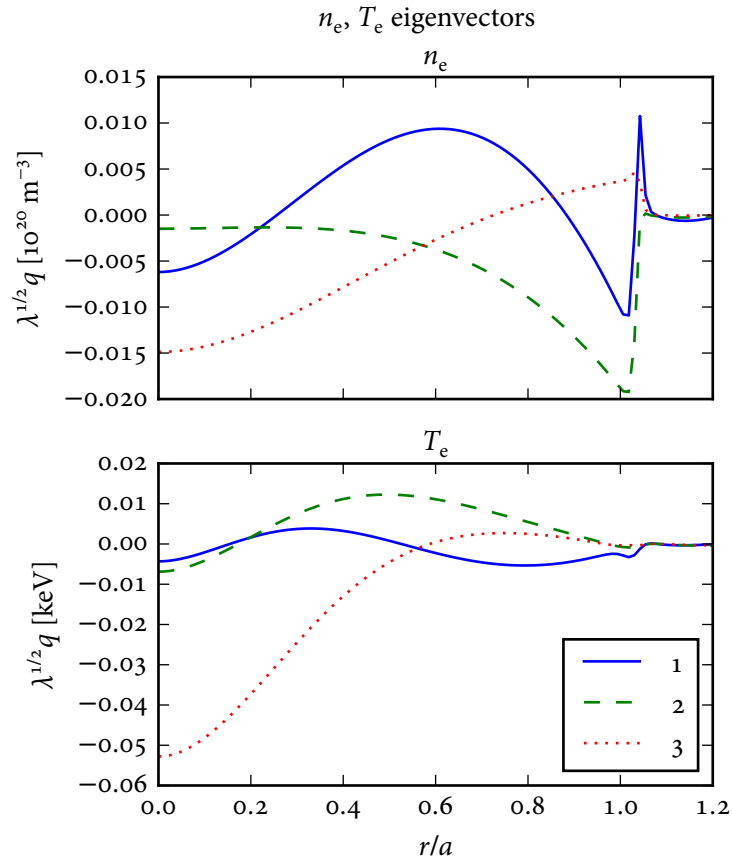


**Figure 3.40:** Prior (blue solid) and posterior (green dashed)  $n_e$  and  $T_e$  profiles. The posterior profiles are *very* slightly different from the prior profiles: the mean values are essentially the same, but the uncertainty envelopes are much smaller. This figure was produced using `settings_1101014006_16_ne3_Te3_line_integ.py` and `post_process_ne3_Te3_line_integ_MCMC.py`.



**Figure 3.41:** Posterior correlation matrix for the inference with line-integrated spectroscopic measurements and  $n_e$  and  $T_e$  free to vary. The correlation between  $D$  and  $V$  is very strongly negative, but the correlation of  $D$  and  $V$  with any of the weights for the  $n_e$  and  $T_e$  profiles is comparatively low. This figure was produced using `settings_1101014006_16_ne3_Te3_line_integ.py` and `post_process_ne3_Te3_line_integ_MCMC.py`.

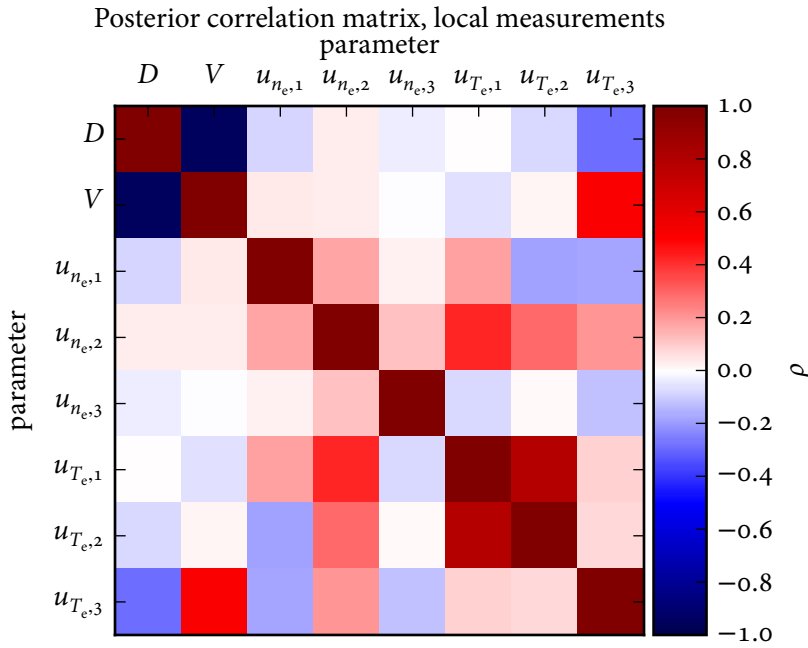
The weighted eigenvectors (i.e., the basis functions that the weights  $u$  multiply) are shown in figure 3.42. The red dashed curve in the top plot corresponds to  $u_{n_e,3}$  and hence is the one which was observed to have the biggest effect on  $D$  and  $V$ . The effect of increasing  $u_{n_e,3}$  is primarily to lower the core electron density. The ionization and recombination rate coefficients are very weak functions of  $n_e$  so any change in  $n_e$  will preserve the ratio of ionization to recombination, with the total source/sink term  $Q_{Z,i}$  going linearly with  $n_e$ . Likewise, the photon emissivity  $\epsilon_{\text{ph}} = n_e n_{Z,i} P_{ij}$  is linear in  $n_e$ . As shown in figure C.2, the source/sink term  $Q_{Z,i}$  is negative in the core for these parameters, so decreasing the core electron density will make this source/sink term less negative. Therefore, the effect of increasing  $u_{n_e,3}$  is to create two competing effects on the diagnostic signals: the photon emissivity for a given impurity density goes down, but the impurity density can reach a slightly higher level because of the reduced particle sink. The loss of photon emissivity appears to



**Figure 3.42:** First three weighted eigenvectors for  $n_e$  and  $T_e$ . This figure was produced using `settings_1101014006_16_ne3_Te3_line_integ.py` and `post_process_ne3_Te3_line_integ_MCMC.py`.

be the dominant effect given that, as  $u_{n_{e,3}}$  increases, the diffusion increases and the convection gets more strongly inwards to restore the impurity density and hence the core photon emissivity to the levels given in the synthetic data.

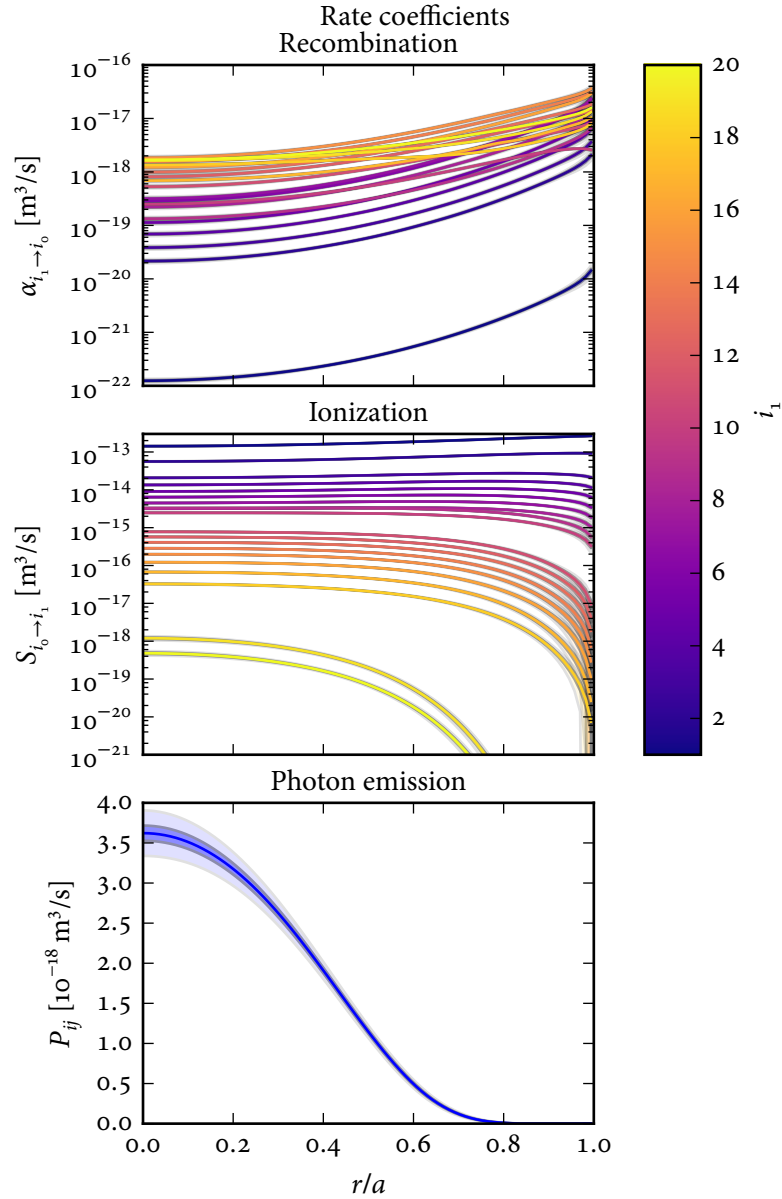
The hypothesis that the correlations between  $n_e$ ,  $T_e$  and  $D$ ,  $V$  are primarily the result of changes in the photon emissivity was verified by running the same test with synthetic data consisting of 32 local measurements of the He-like density. Because the charge state density is used directly the photon emission rate does not enter this calculation. The posterior correlation matrix from this case is shown in [figure 3.43](#). The strong negative correlation between  $D$  and  $V$  is preserved, but now



**Figure 3.43:** Posterior correlation matrix for the inference with  $n_e$  and  $T_e$  free to vary and local charge state density measurements. The correlation between  $D$  and  $V$  remains very strongly negative, but the correlation of  $D$  and  $V$  with any of the weights for the  $n_e$  and  $T_e$  profiles is negligible. This figure was produced using `settings_1101014006_25_ne3_Te3_local.py` and `post_process_ne3_Te3_MCMC.py`.

only  $u_{T_e,3}$  has a non-negligible correlation with  $D$  and  $V$ , but with much lower values of  $\text{corr}[D, u_{T_e,3}] = -0.28$  and  $\text{corr}[V, u_{T_e,3}] = 0.51$ . This appears to be a result of the sensitivity of the ionization and recombination coefficients to  $T_e$ : decreasing the core  $T_e$  changes the charge state distribution, thus requiring slightly less diffusion and slightly less inwards convection to match the observed signals.

The lack of sensitivity of  $D$  and  $V$  to  $n_e$  and  $T_e$  is surprising, especially given that the previous work on Alcator C-Mod ascribed *all* of the uncertainty in  $D$  and  $V$  to the effects of  $n_e$  and  $T_e$ . The charge state distributions and emissivity profiles for STRAHL runs with  $n_e$  and  $T_e$  sent to the extremes of their error bars, however, showed very little change, thus confirming that this is not simply an artifact of the profiles for  $D$  and  $V$  being too stiff to accommodate the changes in  $n_e$  and  $T_e$ . This conclusion is further bolstered by [figure 3.44](#) which shows the recombination, ionization and photon emission rate coefficients averaged over the uncertainties in  $n_e$  and  $T_e$ . These



**Figure 3.44:** Recombination, ionization and photon emission (for the 0.32 nm He-like calcium line) rate coefficients with uncertainty envelopes consistent with the prior uncertainty in  $n_e$  and  $T_e$ . The  $\pm 1\sigma$  and  $\pm 3\sigma$  uncertainty envelopes shown were computed by drawing samples from the prior distribution  $f_{n_e, T_e} | \mathcal{P}(n_e, T_e | \mathcal{P})$  and evaluating the rate coefficients at these values. For the recombination and ionization rates the uncertainty envelopes are comparable to the width of the line. Note that these are *not* proper uncertainty estimates for the rate coefficients: no attempt has been made to include the uncertainties in the atomic physics modeling used to obtain these quantities. This figure was produced using `make_rate_plots.py`.

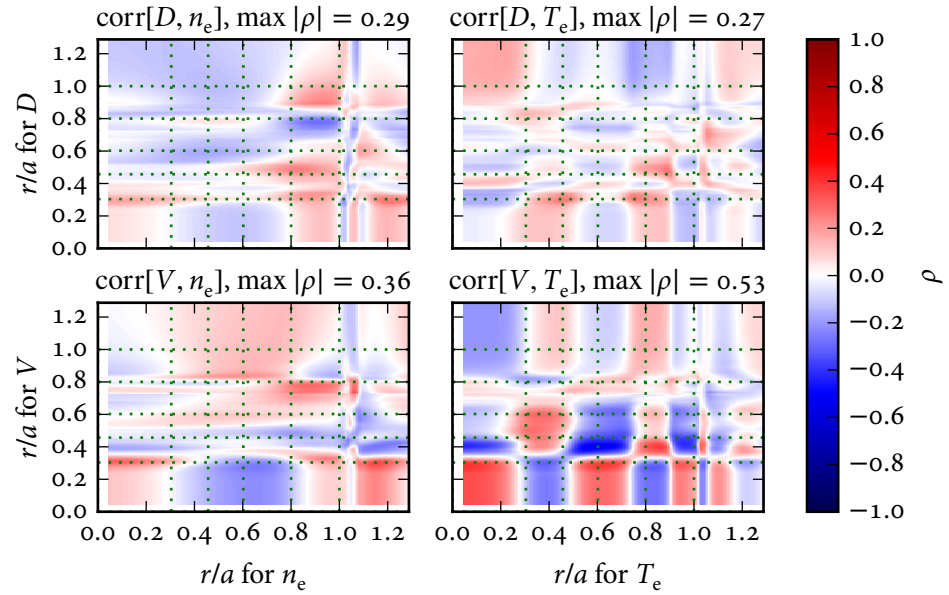
rate coefficients are the only way that  $n_e$  and  $T_e$  enter the impurity transport equation: the background profiles only act to determine the ionization balance (and hence the source/sink terms) and the spectral emission seen by the diagnostics. The spread in the rate coefficients (in particular the ionization and recombination rates) over the prior uncertainty in  $n_e$  and  $T_e$  is very small: therefore, it is entirely expected that the  $n_e$  and  $T_e$  uncertainties should have little impact on the posterior uncertainty in  $D$  and  $V$ .

### 3.6.5.2 Explaining the previous result

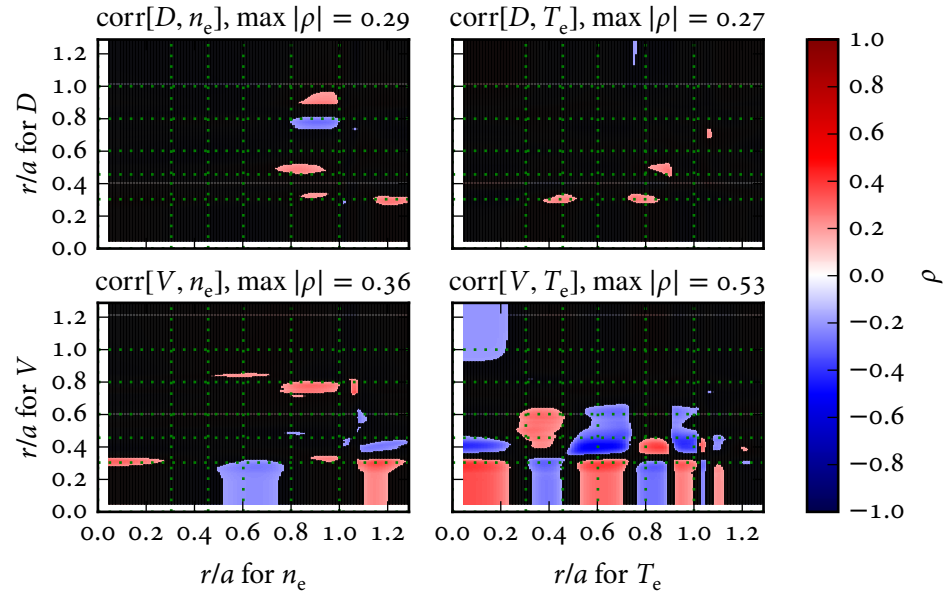
In order to explain the presumed sensitivity of the previous result to  $n_e$  and  $T_e$ , the results from [section 2.6](#) were re-examined. The dataset used for the following analysis consists of all of the samples from the MAP and MCMC cases. The spline samples were not used because that case had a substantially different  $T_e$  profile. The correlations of the local  $D$  and  $V$  values with the local  $n_e$  and  $T_e$  values are shown in [figure 3.45](#). The correlations are, in general, very low, with the peak correlation being a small region of the  $V$  profile having  $\rho = -0.53$ . Specifically, to quantify this, the significance test for moderate numbers of samples which are assumed to be approximately bivariate normal given in [\[302\]](#) was used. Plots of the correlations masked to  $p \leq 0.05$  (i.e., less than 5% probability of seeing a correlation this big by chance from an uncorrelated distribution) are given in [figure 3.45\(b\)](#). Scatterplots of the data with the highest correlations from each pair of variables are given in [figure 3.46](#). The highest correlation observed is between  $V$  and  $T_e$ , and can potentially be ascribed to outliers given that most of the data occupy an uncorrelated cluster. The scatterplots confirm the lack of correlation between the transport coefficients and the background profiles during the Monte Carlo sampling.

The only other thing which changed during the Monte Carlo sampling was the knot locations. The correlations of the local  $D$  and  $V$  values with the knot locations are given in [figure 3.47](#). Both of these quantities show very clear, statistically significant correlations with the knot location, in particular with the outermost knot. This is confirmed by the scatterplots of the data with the highest correlations given in [figure 3.48](#). This indicates that the uncertainty in  $D$  and  $V$  seen before can be almost entirely attributed to the variation in knot location and not to the sensitivity to  $n_e$  and  $T_e$ . This has two key implications:

1. Because there is so much sensitivity to the knot locations, the existing basis functions when used with five coefficients per profile and fixed knots appear to be too inflexible to adequately describe the experimental data. Instead, it is necessary either to use free knots (and hence have to deal with the potential

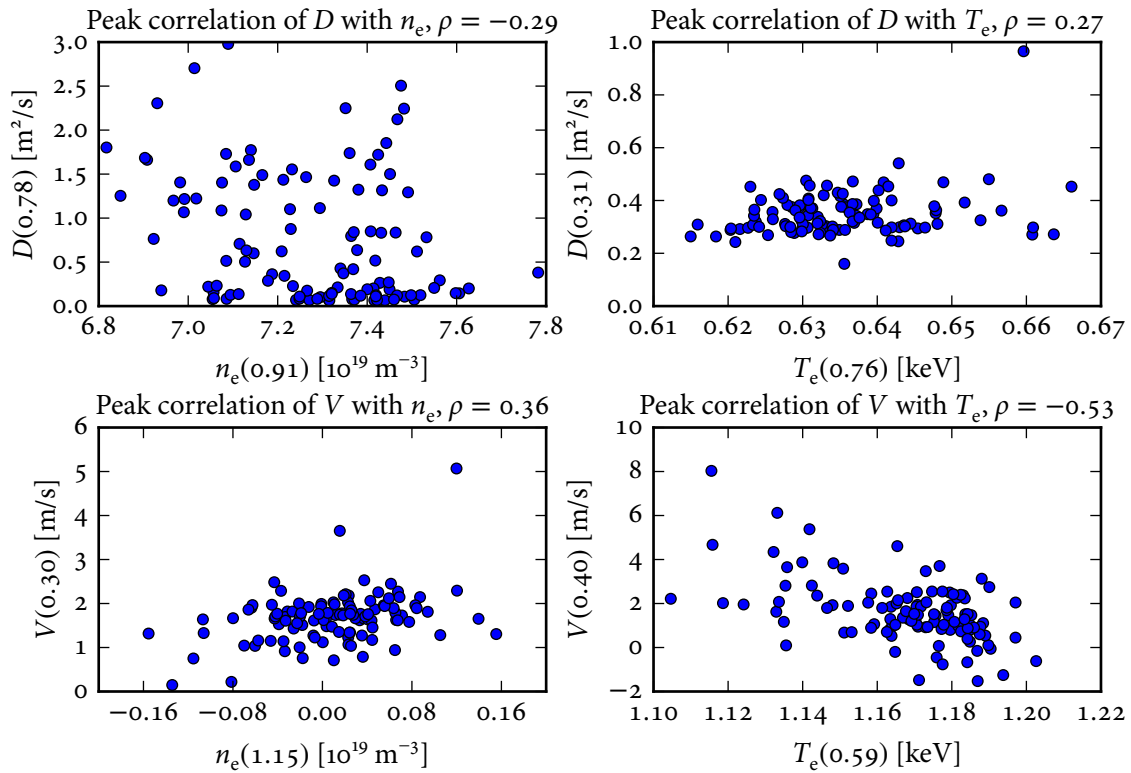


(a) full

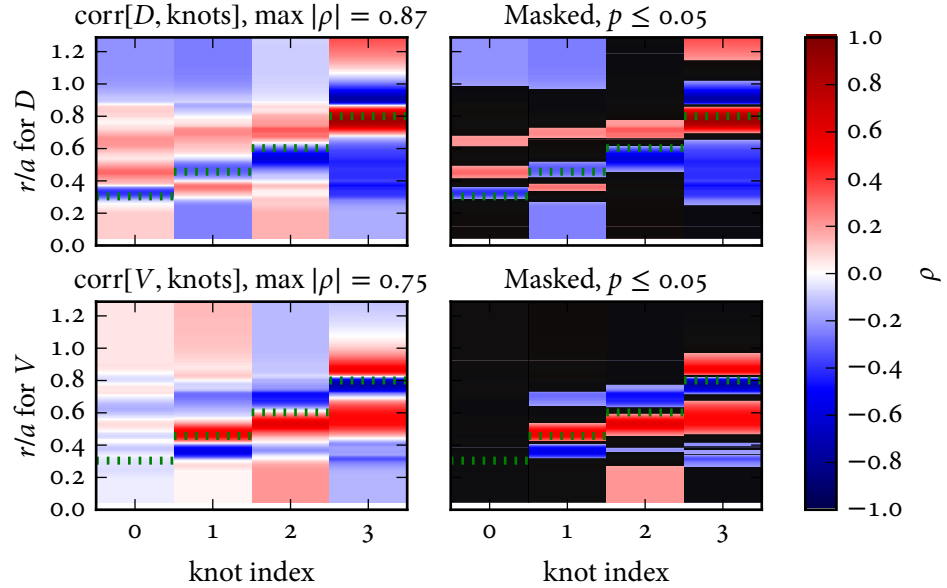
(b) masked to  $p \leq 0.05$ 

**Figure 3.45:** Correlations between the local  $D(r/a)$  and  $V(r/a)$  values and the local  $n_e(r/a)$  and  $T_e(r/a)$  values. (b) shows only the statistically significant ( $p \leq 0.05$ ) correlations. The mean knot positions are shown as the green dotted lines. These figures were produced using `process_old_result.py`.





**Figure 3.46:** Scatterplots of  $D$ ,  $V$  versus  $n_e$ ,  $T_e$  for the pairs of points with the highest correlations. The slightly negative  $n_e$  values in the plot on the lower left are a numerical effect relating to the very small density at  $r/a = 1.15$ , well outside of the LCFS. This figure was produced using `process_old_result.py`.



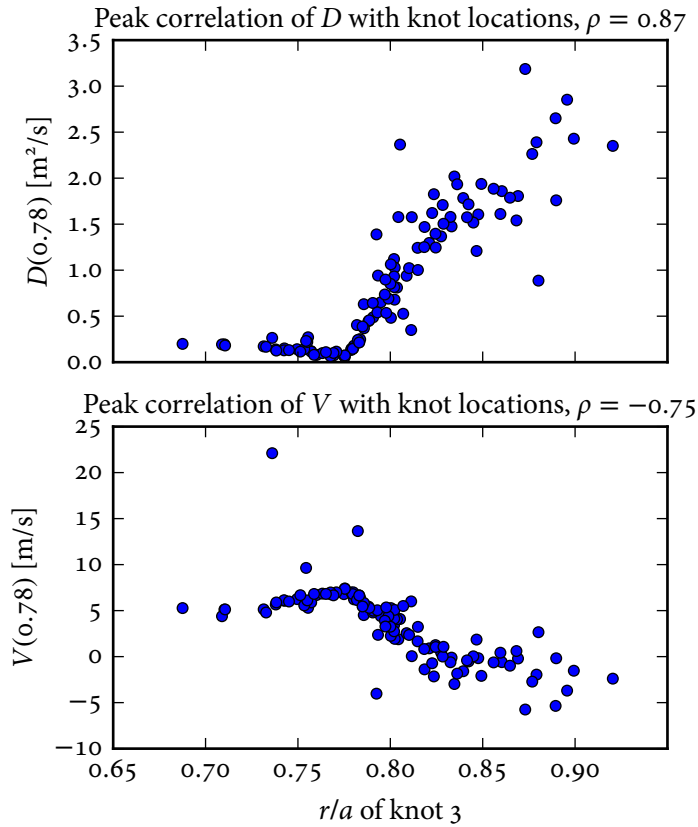
**Figure 3.47:** Correlation of  $D$  (top) and  $V$  (bottom) with the knot positions for the four internal knots. The horizontal green dotted lines give the mean knot locations. The plots on the right are masked to show only the statistically significant ( $p \leq 0.05$ ) correlations. This figure was produced using `process_old_result.py`.

issues illustrated in section 3.6.4) or use many more coefficients to obtain a comparable level of flexibility without such pathologies.

2. Considerable computational effort can be saved by neglecting the effects of  $n_e$  and  $T_e$  on the inferred  $D$  and  $V$  profiles with minimal effect on the results.

### 3.6.6 The model evidence allows rigorous selection of the model complexity

The previous section points to the need to consider more complicated models: either including free knots as free parameters, or including many more coefficients to obtain a comparable level of flexibility in the basis functions describing the transport coefficient profiles. This then raises the question of what level of complexity is justified given the data on hand. As was noted in section 3.5.4.3, the use of MULTINEST to characterize the posterior distribution has the key advantage of providing an estimate of the model evidence  $f_{s|\mathcal{M}}(\mathbf{s}|\mathcal{M})$ . This quantity is the probability of observing data  $\mathbf{s}$  given model  $\mathcal{M}$ , averaged over all values of the parameters  $\theta$ . Here,  $\mathcal{M}$  refers to



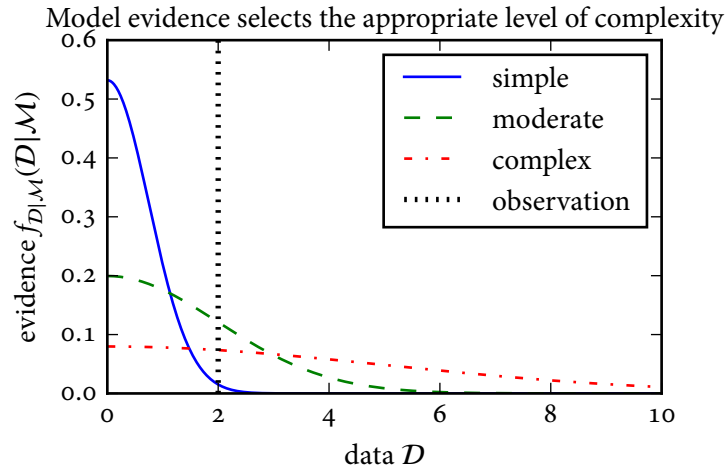
**Figure 3.48:** Scatterplot of  $D$  (top) and  $V$  (bottom) versus knot location for the pairs of points with the highest correlations. The break in slope occurs because  $D$  must be greater than 0. This figure was produced using `process_old_result.py`.

the choice a specific number of coefficients and configuration of knots (fixed or free) to describe the transport coefficient profiles. Mathematically, the evidence is the likelihood  $f_{s|\theta, \mathcal{M}}(s|\theta, \mathcal{M})$  marginalized over all values of the parameters permitted by the prior distribution  $f_{\theta|\mathcal{M}}(\theta|\mathcal{M})$ :

$$f_{s|\mathcal{M}}(s|\mathcal{M}) = \int f_{s|\theta, \mathcal{M}}(s|\theta, \mathcal{M}) f_{\theta|\mathcal{M}}(\theta|\mathcal{M}) d\theta. \quad (3.78)$$

For this reason, the evidence is sometimes also referred to as the *marginal likelihood*. This quantity gives an objective way of comparing models, and allows the model with the appropriate level of complexity given the data to be selected.

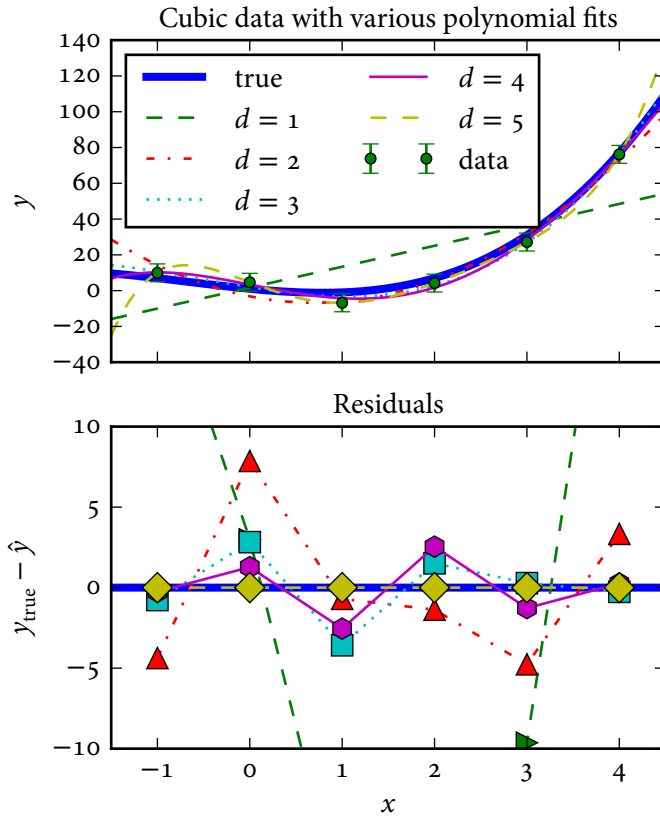
To see how this works, [figure 3.49](#) gives a basic demonstration, based on [figure 5.6](#)



**Figure 3.49:** Demonstration of model selection using the model evidence. The plot shows the evidence  $f_{D|M}(D|M)$  for three models of varying complexity as a function of a single observation  $D$ , where the value actually observed is shown by the black dotted line. The simple model (blue solid curve) can only explain data in a small range, so it has very high evidence for that narrow range, but small evidence for other observations. The complex model (red dot-dash curve) can explain many possible observations, so it necessarily has low evidence over the whole domain. The model with moderate complexity (green dashed curve) can explain a moderate number of possible observations in a region encompassing the observed data, and so has the highest evidence for the observation shown. This figure is based on figure 5.6 of [304]. The figure was produced using `make_ml_demo.py`.

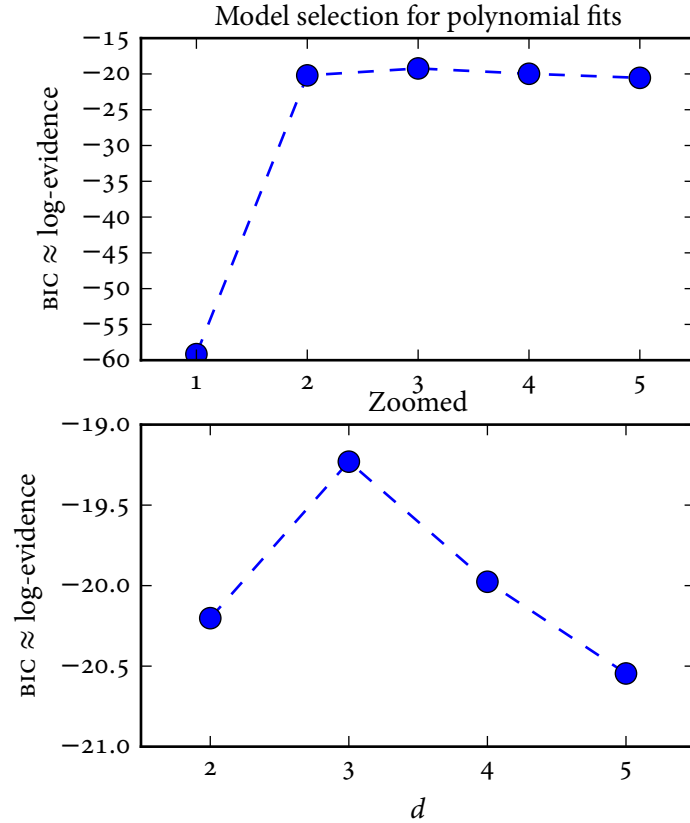
of [304]. For a more concrete example, consider selecting what order of polynomial to fit some data with. Six synthetic data points were generated from the cubic polynomial  $y = x^3 + 2x^2 - 5x + 1$ , with homoscedastic noise  $\sigma_y = 5$ . The results of fitting polynomials of order  $d = 1$  through  $d = 5$  are shown in figure 3.50. It is obvious that the linear fit (green dashed line) is not a good description of the data, and one might also guess that the quintic fit (yellow dashed line) is overfitting because of its zero residual and strange behavior at the edges of the fit. But, it is not obvious from inspecting the fits and residuals where between these two extremes the “best” model is. Figure 3.51 shows an estimate of the log-evidence as a function of polynomial degree. This correctly shows that the cubic polynomial has the right level of complexity given the data – it is neither too simple (and hence misses real structure) nor is it too complex (and hence infers structure which is not there).

In order to demonstrate the applicability of the evidence to model selection for



**Figure 3.50:** Synthetic data (green points), true curve (thick solid blue line) and polynomial fits of various degrees. The upper plot shows the data and the fits, the lower plots shows the residuals. As expected, the quintic ( $d = 5$ ) polynomial can fit the six data points with zero residual. In order to show the structure for the more reasonable fits, the plot range was set to cut off the very large residuals from the linear fit. This figure was produced using `model_selection_demo.py`.

the impurity transport problem, a synthetic dataset consisting of 32 local measurements of the He-like calcium density with a time resolution of 6 ms and a noise level of 5% was used. This is roughly equivalent to the situation on Alcator C-Mod, but using the local measurements is a little more computationally efficient. The basic  $D$  and  $V$  profiles from equation (3.14) were used to generate the data. MULTINEST was run for cases with one, two, three and four free coefficients for each of the transport coefficient profiles. In all cases the knots were fixed at positions linearly spaced between  $r/a = 0$  and  $r/a = a_c/a$ . In order to obtain accurate estimates of the evidence,

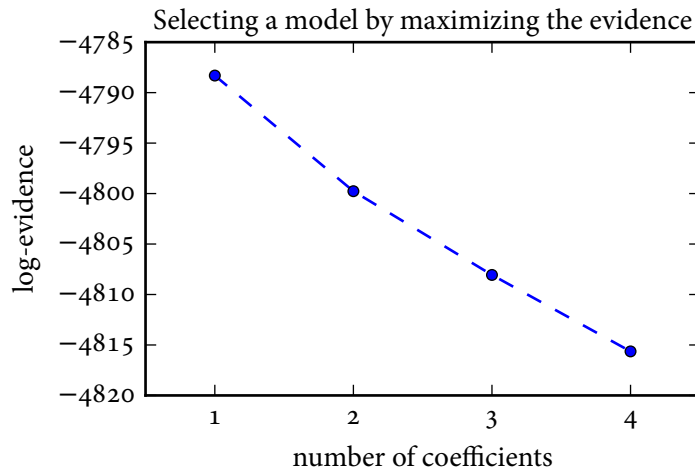


**Figure 3.51:** Estimated log-evidence as a function of polynomial degree  $d$ . The log-evidence was estimated using [equation \(3.79\)](#). This procedure correctly picks  $d = 3$  as the best model. This figure was produced using `model_selection_demo.py`.

400 active points were used and the sampling efficiency was set to 0.3. The evidence is given as a function of the number of coefficients in [figure 3.52](#) and the various posterior profiles for  $D$  and  $V$  are given in [figure 3.53](#). The evidence is highest for the simplest case, which is the model which generated the data. Furthermore, past the optimal case the log-evidence follows a linear trend in the number of free parameters. This is consistent with behavior of the Bayesian information criterion (BIC) [[304](#), [458](#)]:

$$\text{BIC}(\mathcal{M}) = \ln f_{s|\hat{\theta}^{\text{ML}}, \mathcal{M}}(s|\hat{\theta}^{\text{ML}}, \mathcal{M}) - \frac{d}{2} \ln N \approx \ln f_{s|\mathcal{M}}(s|\mathcal{M}), \quad (3.79)$$

where  $d$  is the number of parameters and  $N$  is the number of observations. Once



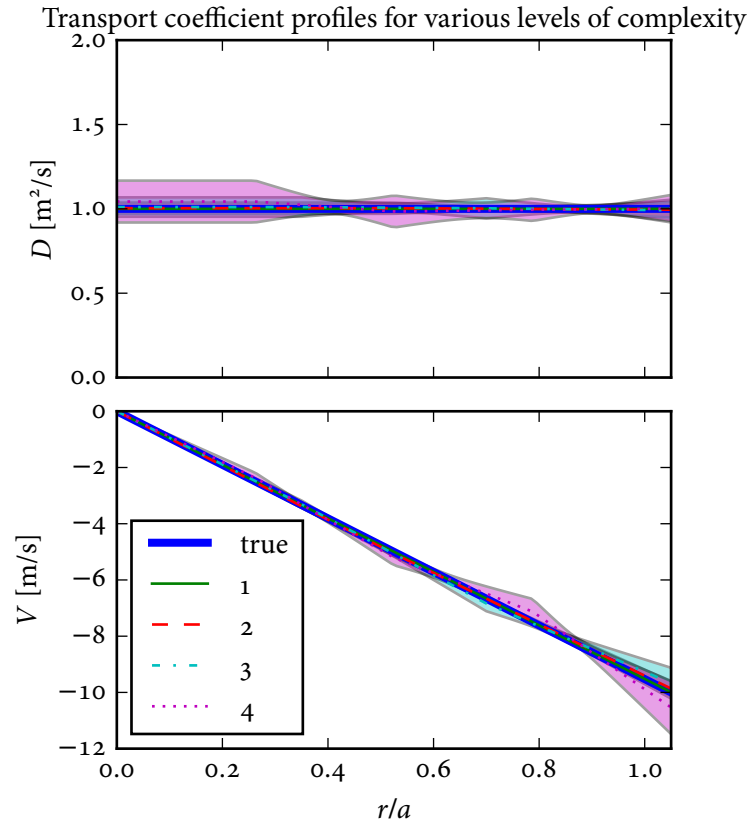
**Figure 3.52:** Evidence as a function of model complexity for the basic synthetic data. The horizontal axis gives the number of free coefficients for each transport coefficient profile (both  $D$  and  $V$  were taken to have the same number of coefficients). The synthetic data were generated by a model with only one coefficient per profile, which is correctly identified as the best model. This figure was produced using `settings_1101014006_24_model_selection_local.py` and `plot_model_selection_evidence_trend_local.py`.

a model is complex enough to describe the data well, the improvement to the log-likelihood term is negligible and the term which is linear in the number of parameters takes over. This both verifies that the log-evidence computed by MULTINEST has the correct behavior and confirms that the BIC is an appropriate approximation to the log-evidence for comparing models when a reliable maximum likelihood estimate is available.

Note that as an alternative to model selection using the evidence, techniques such as reversible-jump MCMC can be used to marginalize over discrete parameters like the number of coefficients, as was done for smoothing splines in [459]. This process is very computationally expensive, however, so it was not attempted here.

### 3.6.7 Testing parameter estimation and model selection on more complicated synthetic data

The previous sections have demonstrated the ability of the MULTINEST-based approach to reconstruct the very simple  $D$  and  $V$  profiles given by equation (3.14). There is no reason to expect that real experimental transport coefficient profiles will



**Figure 3.53:** Posterior  $D$  and  $V$  profiles for the four different levels of complexity. Only the  $\pm 1\sigma$  uncertainty envelopes are shown. In all cases the posterior mean is similar to the true profile, but the higher complexity cases permit more flexibility and hence have higher uncertainties. This figure was produced using `settings_1101014006_24_model_selection_local.py` and `post_process_model_selection_MCMC_local.py`.

be so well-behaved, however. In order to verify the ability of this approach (and the diagnostic arrangement on Alcator C-Mod) to reconstruct more complicated profiles, synthetic data were constructed using the mean profile from the previous work as presented in section 2.6. To represent the diagnostic situation on Alcator C-Mod, the same line-integrated synthetic diagnostics as in section 3.6.2 were used. To help speed up convergence, a prior distribution which was designed to keep  $D$



and  $V$  on the correct order of magnitude was used:

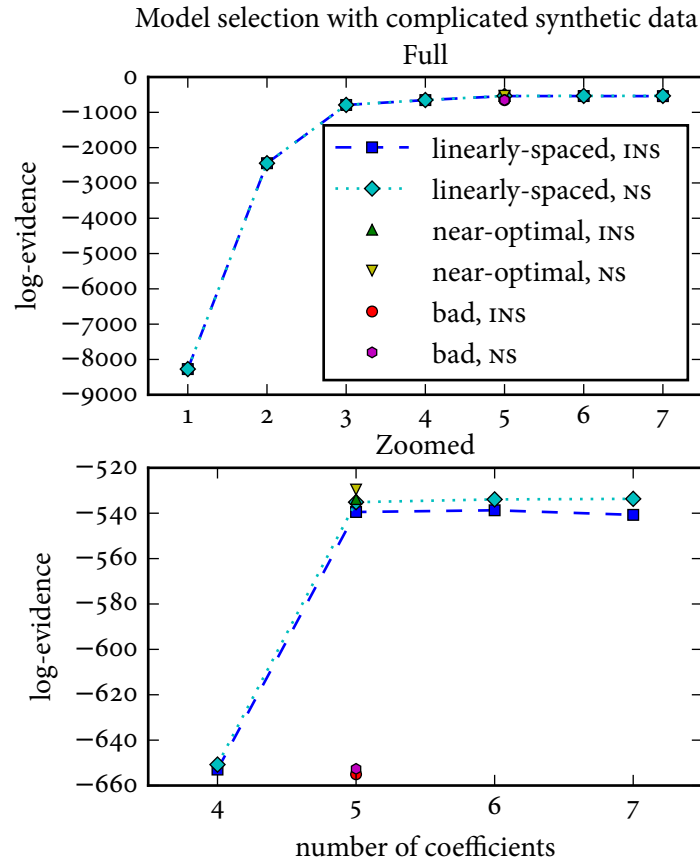
$$f_{D,V}(D, V) = \mathcal{U}(D; 0 \text{ m}^2/\text{s}, 15 \text{ m}^2/\text{s})\mathcal{U}(V; -60 \text{ m/s}, 20 \text{ m/s}), \quad (3.80)$$

where in this notation  $D$  refers to all of the coefficients of the diffusion coefficient profile and  $V$  refers to all of the coefficients of the convective velocity profile. As in the previous section, 400 active points were used and the sampling efficiency was set to 0.3. Importance nested sampling was used to attempt to get a more accurate estimate of the evidence.

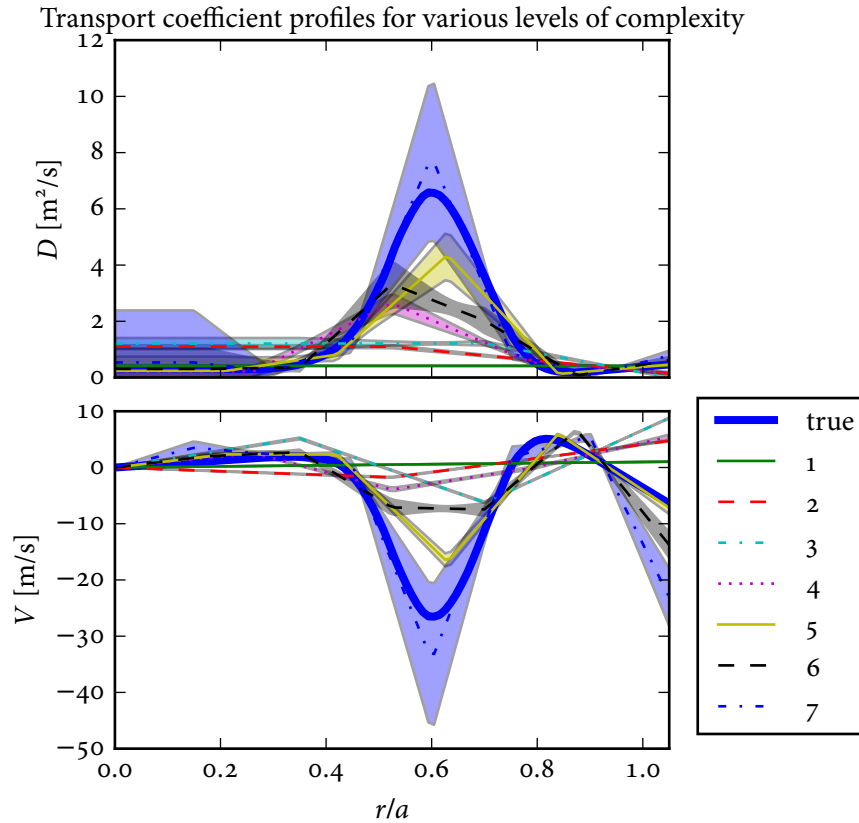
The evidence as a function of model complexity is given in [figure 3.54](#) and the corresponding profiles are given in [figure 3.55](#). With up to seven coefficients per profile the evidence has still not shown a clear rollover. Curiously, the importance nested sampling estimate shows a slight decrease in the evidence for the seven coefficient case, but the conventional nested sampling estimate shows the curve as continuing to increase. Unfortunately it was not possible to evaluate the evidence for more complicated models: as [figure 3.56](#) shows, this problem is very much subject to the curse of dimensionality. The number of forward model evaluations needed for MULTINEST to converge increases exponentially with the number of free parameters. The forward model takes a little more than one CPU-second per evaluation, so the seven coefficient case took approximately 7000 CPU-hours to converge. Extrapolating this trend, the eight coefficient case is expected to take about 15 000 CPU-hours and the nine coefficient case 53 000 CPU-hours. The total time to test all cases from one to ten coefficients is expected to be about 260 000 CPU-hours. The actual time may be worse, as at some point the number of active points will need to be increased to ensure adequate coverage of the posterior distribution is obtained. For comparison, a single nonlinear, ion-scale GYRO run takes about 10 000 CPU-hours, but is run on many processors.<sup>16</sup> Therefore, to make it practical to apply the MULTINEST-based approach to real data, the next step is to do extensive development work to eliminate bottlenecks between STRAHL and the rest of the forward model and to deploy the analysis on more powerful, massively-parallel computers than what was used for the present work. For reference, to analyze the seven coefficient model took a total of about 15 wall-clock days on a workstation which supports 24 threads. Optimistically assuming that the problem scales linearly with the number of processors, this would be reduced to two wall-clock days on 200 processors and four wall-clock hours on 2000 processors.

That being said, it is still possible to get some insight from the runs which were obtained. From [figure 3.55](#), it is clear that the simple  $D$  and  $V$  profiles of

16. It often takes a total of about  $2 \times 10^6$  to  $3 \times 10^6$  CPU-hours to simulate a given shot with GYRO, including setting up the baseline run, doing resolution checks and performing sensitivity scans [460].

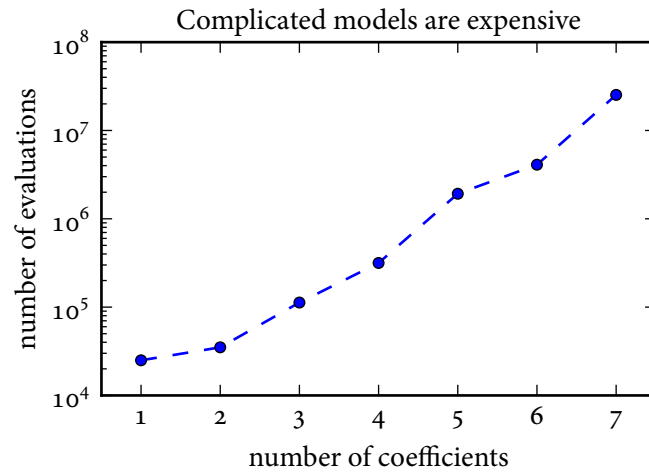


**Figure 3.54:** Evidence as a function of model complexity for the synthetic data using complicated  $D$  and  $V$  profiles. The upper plot shows all of the cases which were run while the lower plot has been zoomed in to show the more reasonable cases with four or more coefficients per profile. Also shown are the evidence values for the five coefficient case with good (green triangle) and bad (red circle) knot locations. In the legend, “INS” refers to importance nested sampling and “NS” refers to nested sampling. The INS estimates tend to lie below the NS estimates, and in fact reach opposite conclusions about the relative suitability of the six and seven coefficient cases. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `plot_NTH_as_true_evidence_trend.py`.



**Figure 3.55:** Posterior  $D$  and  $V$  profiles for various levels of complexity. Only the  $\pm 1\sigma$  uncertainty envelopes are shown. The dashed blue curve corresponding to the seven coefficient per profile case is the only case shown which fully encompasses the true solution in its uncertainty envelope. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.

equation (3.14) do a terrible job of matching the true profiles and have very small uncertainties as a result of how inflexible they are. This case is relevant because simple profiles like this were widely used in the older impurity transport literature. On their own, the parameter estimates for  $D$  and  $V$  in this case provide no direct indication that the model is a poor description of reality: the qualitative features of a fast rise followed by a slow decay are present in the simulated diagnostic signals, and the large residuals could naively be attributed to a mis-estimation of the diagnostic uncertainties. In fact, as shown in figure 3.57, the diagnostic signals are reproduced

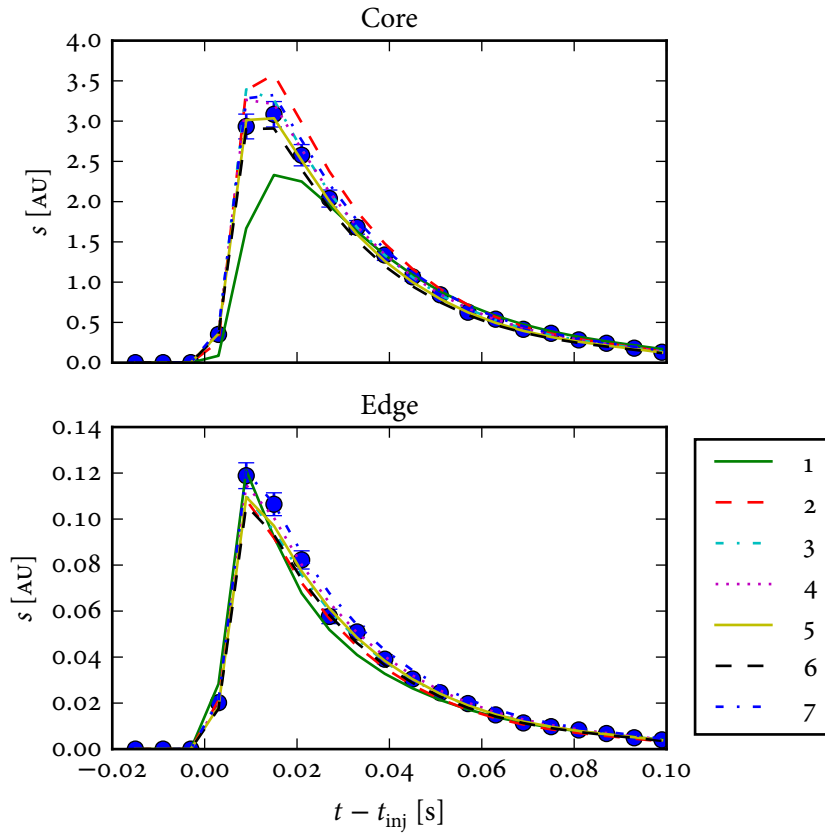


**Figure 3.56:** Number of forward model evaluations for MULTINEST to converge as a function of model complexity. Note the logarithmic scale: this shows the exponential growth characteristic of the curse of dimensionality. Note that on the computer used for this study a single forward model evaluation takes approximately one CPU-second: doing model selection properly requires tens to hundreds of thousands of CPU-hours. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `make_calls_plot.py`.

reasonably well even for the two coefficient per profile case. This highlights the critical importance of model selection: even simple profiles which bear absolutely no resemblance to the true profiles can reproduce the experimental observations to an extent that “looks good.” But, as [figure 3.54](#) makes clear, the model evidence soundly rejects the models with fewer than five coefficients per profile.

Note that, until there are at least seven coefficients per profile, the uncertainty envelopes still do not overlap with the major features of the true profiles. One intriguing aspect is that the five coefficient model qualitatively resembles the true profiles more strongly than the six coefficient model. This is particularly relevant as the previous work which generated the truth data used five coefficients per profile. This appears to be a result of the knot positions: when using linearly-spaced knots, models with an odd number of knots have a knot right near the large peak in the true profiles, whereas models with an even number of knots have a knot to either side of the peak. This means that odd-knot cases like the five and seven coefficient models are better able to match the features of the true profiles.

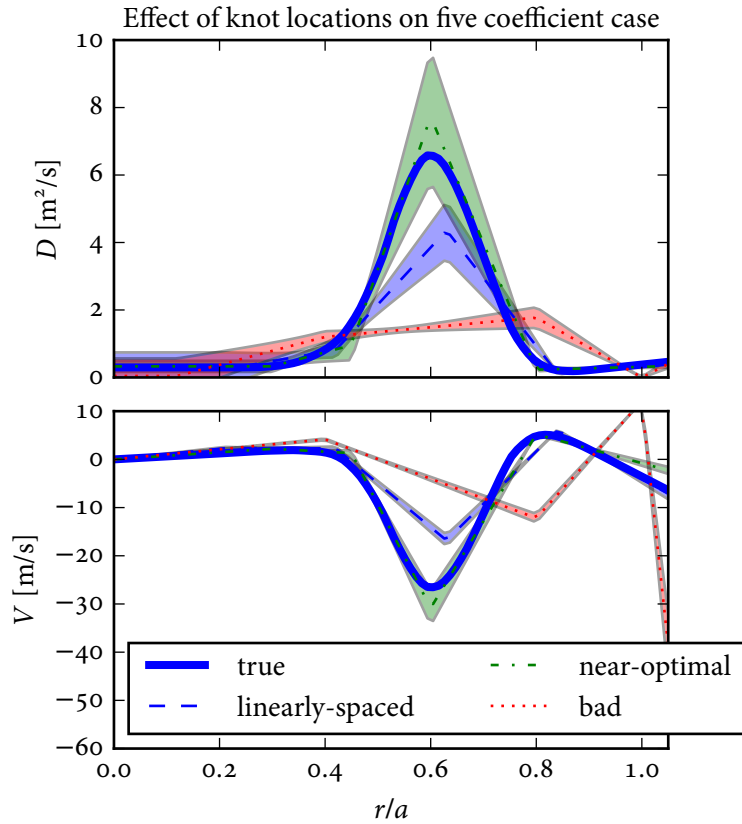
To explore the effect of the knot positions further, two additional five coefficient



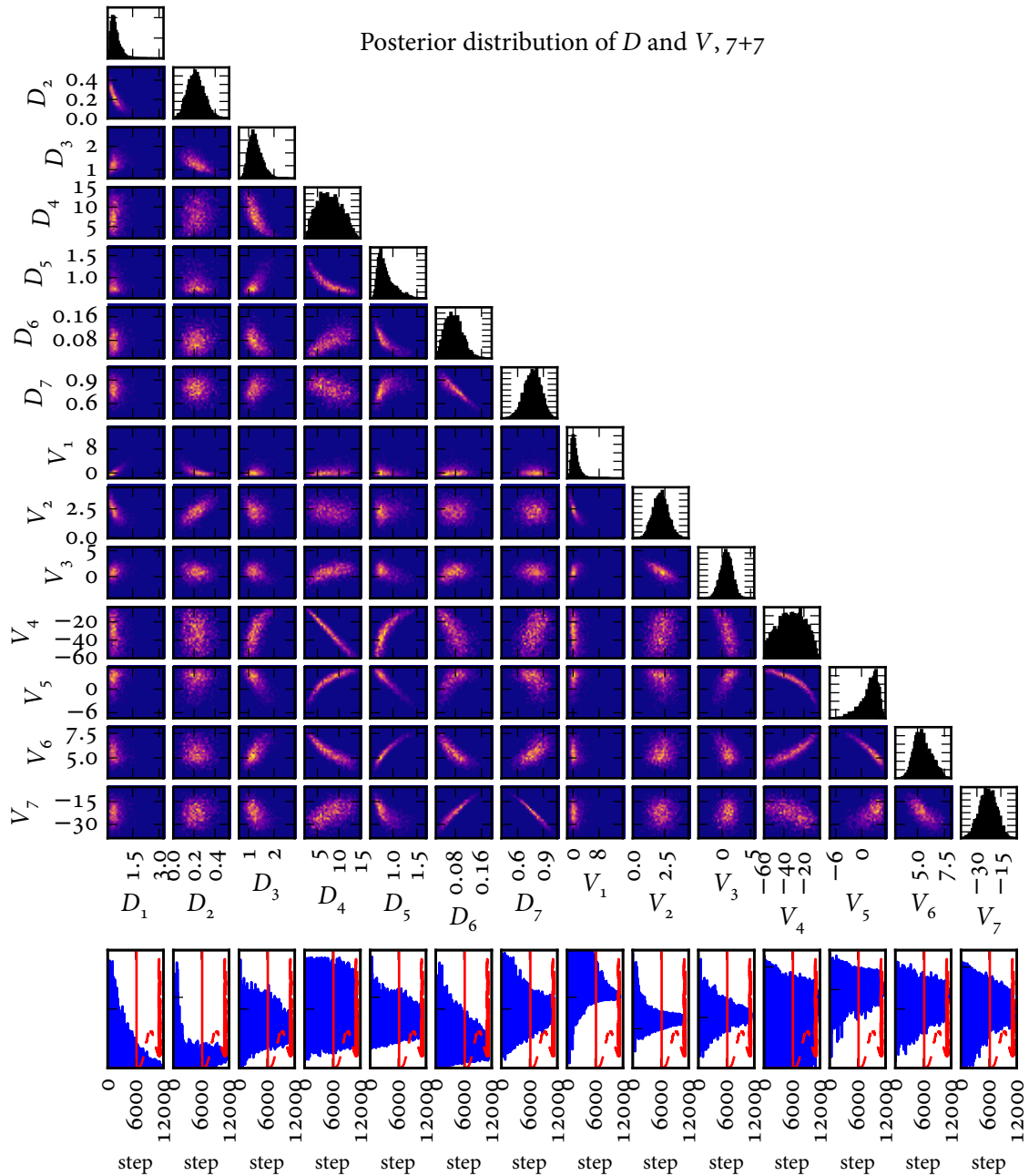
**Figure 3.57:** Synthetic data and predicted diagnostic signals for varying levels of complexity. The upper plot is from a HIREX-SR chord viewing the core of the plasma, the lower plot is from the outermost chord. These signals were evaluated using the posterior mean  $D$ ,  $V$  profiles for each case. When there are at least two coefficients per profile, the data appear to be reproduced quite well. But [figure 3.54](#) clearly shows that the data are more likely under the more complicated models, and [figure 3.55](#) confirms that the more complicated models do in fact match the true profiles better. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.

models were evaluated: one with knots at the mean knot locations from the previous work, and one with the knots purposefully placed in bad locations. The profiles are shown in [figure 3.58](#) and the corresponding evidence values are shown as the green triangle and red square in [figure 3.54](#). The near-optimal knots yield profiles which match the true profiles well and the evidence value is correspondingly higher. The purposefully bad knots, however, yield a very bad match to the true profiles and a dramatically lower evidence value. Note that the bad knots case is somewhat questionable as the outermost coefficient of the  $V$  profile ended up being very close to the edge of the prior distribution, which may have caused the sampling to be unreliable. Independent of that detail, this test would seem to imply that some benefits may be claimed by using free knots. Unfortunately, when this was tried, the number of forward model evaluations needed to converge was found to be much worse than for the fixed-knot case. The effect of the knot positions should be much smaller in the more flexible models with more coefficients per profile, so the appropriate course of action seems to be to use many coefficients with fixed, linearly-spaced knots in order to avoid the issues (and associated convergence slowdown) discussed in [section 3.6.4](#).

A final interesting result from the seven coefficient case is shown in [figure 3.59](#), which shows the posterior distribution for the coefficients of the  $D$  and  $V$  profiles. The mode has a curved shape in a number of the bivariate marginal distributions and many of the univariate marginal distributions are asymmetric: the posterior distribution is very much non-Gaussian. This is a very important result, as the common approach of finding the MAP or ML estimate using a local optimizer then estimating the parameter uncertainties using the inverse Hessian matrix assumes that the posterior distribution is Gaussian. That approach would yield an inaccurate uncertainty estimate when applied to this case. Curiously, the simpler cases do not exhibit such strongly non-Gaussian behavior except where  $D$  is close to zero. (Posterior distributions for the other cases are given in [figure C.6](#) through [figure C.13](#).) It is believed that this is a result of the seven coefficient per profile case being the first one to have sufficient flexibility for the sampling procedure to be able to explore the nonlinear interplay between convection and diffusion when determining the line-integrated diagnostic signals. This behavior may also be partly an effect of the distribution for  $V_4$  (the negative peak of the  $V$  profile) being close to the lower bound of the prior distribution, which may have created issues with MULTINEST's sampling.



**Figure 3.58:** Posterior  $D$  and  $V$  profiles with various fixed knot locations for the five coefficient per profile case. Only the  $\pm 1\sigma$  uncertainty envelopes are shown. The colors of the thin curves were chosen to match figure 3.54. Note that the original curves are a mixture of five coefficient profiles with perturbed knot locations. Linearly-spaced knots (thin blue dashed line) do a reasonable job of matching the general features of the true profiles (thick blue solid line). Using the mean knot locations from the previous analysis (dot-dash green line) gives a much better fit, with the true curves lying almost entirely inside the uncertainty envelopes. Using knots which were selected to lie far from the optimal locations (red dotted line) produces a very poor fit to the data. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.



**Figure 3.59:** Posterior distribution for  $D$  and  $V$  for the seven coefficient per profile case. Note the curved shape of some of the bivariate marginal distributions such as between  $V_5$  and  $V_4$  as well as the asymmetric shape of the univariate marginal distributions such as the one for  $D_5$ : the posterior distribution is very much non-Gaussian. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.



### 3.7 Summary, conclusions and next steps

This chapter has illustrated the difficulties endemic to the inference of experimental impurity transport coefficient profiles. The general features of the impurity transport equation were explored with the help of the STRAHL code and a linearized Bayesian model was constructed to help estimate diagnostic requirements for impurity transport measurements. This model led to the interesting conclusion that spatial resolution appears to be more important than temporal resolution for typical x-ray spectrometer sampling rates and noise levels: from the impurity transport perspective, it is a better investment to add more lines of sight than to develop faster detectors.

A fully Bayesian framework for inferring transport coefficient profiles was then developed in order to provide rigorous accounting of the uncertainty in the solution. Use of the MULTINEST algorithm allows a complete accounting of uncertainty, even in the presence of potentially multimodal or otherwise non-Gaussian posterior distributions. Furthermore, MULTINEST computes the model evidence, an objective metric for selecting the appropriate level of complexity in the inferred  $D$  and  $V$  profiles.

Several synthetic datasets were used to illustrate the utility of the fully Bayesian approach and verify the predictions of the linearized Bayesian analysis. This analysis led to the very surprising result that the uncertainties in the background  $n_e$  and  $T_e$  profiles do not affect the result very much. Instead, the uncertainties which were attributed to  $n_e$  and  $T_e$  in the previous work on Alcator C-Mod can be explained by the shifting of knot positions between code runs. This indicates that the basis functions used in the previous work were not flexible enough, and underscores the need for proper model selection. Application of the new impurity transport analysis methodology to synthetic data based on the previous Alcator C-Mod results shows the promise of the new methodology, but completing the analysis and revisiting existing experimental data was not computationally tractable.

In light of the difficulties encountered when measuring impurity transport coefficients for this thesis, and the fact that many of the pathologies only became clear upon very close examination of the situation, any study which attempts to measure impurity transport coefficients using the forward model approach must be considered suspect unless it addresses all of the following questions:

1. How is parameter estimation performed? Specifically:
  - (a) If using a derivative-based optimizer such as Levenberg-Marquardt or BFGS, how were the derivatives computed? If finite differences are used

to compute the derivatives for a derivative-based optimizer, was the step size sufficient to avoid roundoff error? It was found in the present work that finite differences require a prohibitively large step size. This affects both the search direction of the optimizer and the quality of the inverse Hessian matrix used to estimate parameter uncertainties in this approach.

- (b) How was the possibility of multiple modes and/or plateaus in the parameter space handled? It was found in the present work that local optimizers started at various points distributed throughout the parameter space would often either get stuck in regions of the parameter space with poor fit or would converge to various local optima.
2. How is model selection performed? It was found in the present work that simple models can give seemingly reasonable agreement with the data while having very little relationship to the true  $D$  and  $V$  profiles. Furthermore, the inflexibility of overly simple models was found to lead to an underestimation of the uncertainty in the answer.
  3. How was the analysis procedure verified? It was found in the present work that some of the issues observed do not occur in synthetic data produced using very simple  $D$  and  $V$  profiles.

Given the issues associated with the measurement of impurity transport coefficients, it would appear that they are not, in fact, the appropriate level of the primacy hierarchy [34] to use for validation of gyrokinetic simulations of impurity transport. A more trustworthy comparison can be obtained by using quantities which require far less complicated post-processing of both the experimental data and the simulation, as was done to compare measured and simulated boron density profiles in [21, 461, 462].

While the present work has demonstrated a framework which can deliver reliable estimates of impurity transport coefficient profiles and their associated uncertainties, there are still many avenues to build on the present work and improve it for use in situations where a summary of impurity transport in terms of transport coefficient profiles is desirable:

- Handling the full complexity of real data is not computationally tractable with the present approach. The basic step to improve this is to eliminate the bottlenecks between STRAHL and the rest of the forward model and then deploy the code on a massively-parallel computer. Beyond that, it may be possible to replace MULTINEST with some form of faster, approximate methodology such

as a global optimizer for parameter estimation and BIC for model selection. This will require considerable work to select an appropriate optimizer given the issues observed during the present work.

- The present approach did not consider the effects of sawteeth. To handle multiple injections at different sawtooth phases will require multiple STRAHL runs per forward model evaluation, so it is essential to get the analysis to be computationally tractable before handling this problem.
- While the present framework is in principle capable of handling uncertainties in the ionization, recombination and photon emission rate coefficients, uncertainty estimates are not presently available. Some preliminary work on this topic was published in [457], but more work is needed to fully characterize the uncertainties in measurements of impurity transport.
- The present analysis ignored poloidal asymmetries, which are known to be important for high- $Z$  impurities. This will require development of a new impurity transport code, as STRAHL is a 1D code.
- The present analysis assumed that the diagnostics were absolutely-calibrated, including the measurement of the source function. Observations during the present work as well as published results in [159, 160] suggest that the lack of an absolute calibration can lead to a multimodal and/or non-identifiable posterior distribution.
- The spectral model developed for the present work needs to be extended to include the H-like calcium Balmer series, proper handling of the He-like calcium forbidden line, satellite lines, recombination radiation and bremsstrahlung in order to handle experimental data from the XTOMO systems.

# 4

## Conclusions and future directions

### 4.1 Contributions and conclusions

This thesis has made contributions to experimental plasma physics under two main topics:

1. Analysis of profile data using nonstationary Gaussian process regression.
2. Inference of impurity transport coefficient profiles.

In both cases, the goal has been to improve the quality of experimental data analysis in order to improve the ability to validate simulations of turbulent transport.

#### 4.1.1 Profile data analysis

As shown in [chapter 2](#), the use of Gaussian process regression helps to overcome the major issues endemic to previous approaches for profile data smoothing by providing a framework for statistically rigorous, automated profile fitting. The major contributions of this thesis to profile fitting and Gaussian process regression are:

1. Development of general-purpose (`gptools`) and domain-specific (`eqtools`, `profiletools`) open source software tools to fit profile data. `gptools` is presently in use around the world in fields including plasma physics, astrophysics, and biophysics and `eqtools` has been deployed at several American and European tokamaks. (See [appendix D](#), [appendix E](#) and [appendix F](#).)
2. Testing of a wide variety of approaches to nonstationary Gaussian process regression, including the novel I-spline input warping function and the tanh covariance length scale function. (See [section 2.4.3.2](#).)

3. Demonstration of the importance of using a fully Bayesian (i.e., MCMC-based) approach when evaluating gradient uncertainties. (See [section 2.5.1.](#))
4. Demonstration of nonstationary Gaussian process regression combining local and line-integrated measurements, including the ability to reconstruct the core electron density without local measurements. (See [section 2.3.9](#) and [section 2.5.2.](#))
5. Demonstration of the use of Gaussian process regression to compute second derivative profiles with rigorous uncertainty estimates in order to test theories of momentum transport. Within the experimental uncertainties there is little evidence that there is a strong change in second derivatives across the density threshold where a toroidal rotation reversal is seen in L-mode. (See [section 2.5.3.](#))
6. Demonstration of Gaussian process regression to fit the space and time dependence of sawtooth-free profile data. (See [section 2.5.4.](#))
7. Demonstration of the combination of Gaussian process regression with a parametric (mtanh) mean function to fit H-mode pedestal data. This application revealed that the parameters obtained from a Gaussian process with a parametric mean function can end up differing substantially from those obtained from a purely parametric fit. (See [section 2.5.5.](#))
8. Demonstration of the use of a truncated eigendecomposition of the Gaussian process posterior distribution to efficiently propagate profile uncertainty through an analysis code. This approach made it computationally tractable to show that the uncertainties in  $n_e$  and  $T_e$  have little effect on the uncertainties in  $D$  and  $V$ . (See [section 3.6.5.](#))
9. Extensive calculations of the derivatives of stationary and nonstationary covariance kernels, including expressions for derivatives of arbitrary order of multidimensional squared exponential, rational quadratic and Matérn covariance kernels. (See [appendix B.](#))

Widespread adoption of these techniques will increase the level of rigor with which plasma data analysis and vvuq of simulation codes is conducted. These techniques have already been applied to a basic analysis of experimental impurity transport [305] and to the modeling of lower hybrid current drive [463]. The truncated eigendecomposition demonstrated in contribution 8 in particular has the potential to dramatically speed up the propagation of profile uncertainties through both analysis and simulation codes when conducting vvuq of turbulent transport simulations.

### 4.1.2 Inference of impurity transport coefficient profiles

This thesis has led to a major reevaluation of the procedures used to infer impurity transport coefficient profiles, calling much of the existing literature into question. The major contributions of this thesis to the measurement of impurity transport are:

1. Demonstration of major issues with existing approaches for inferring impurity transport coefficients. (See [chapter 3](#).)
2. Development of guidelines for assessing attempts at inferring impurity transport coefficients. (See [section 3.7](#).)
3. Development of a linearized model to efficiently estimate diagnostic requirements to infer impurity transport coefficients. Application of this model revealed that spatial resolution is more important than time resolution, contrary to the conventional intuition. (See [section 3.4](#).)
4. Development of a fully Bayesian approach to inferring the posterior distribution of the impurity transport coefficient profiles. Application of the MULTINEST algorithm enables a complete accounting of uncertainty as well as rigorous selection of the model complexity. (See [section 3.5](#) and [section 3.6](#).)
5. Demonstration of the critical importance of model selection in obtaining correct estimates of impurity transport coefficient profiles, including:
  - (a) Demonstration that the inferred transport coefficient profiles are relatively insensitive to the  $n_e$  and  $T_e$  profiles. In particular, the large amount of uncertainty previously attributed to the uncertainties in the  $n_e$  and  $T_e$  profiles is actually a result of the use of overly simple basis functions combined with incomplete handling of the knot locations. (See [section 3.6.5](#).)
  - (b) Demonstration that the inferred transport coefficient profiles do not in any way resemble the true values until a sufficiently complex model is used. This calls into question much of the early work which used a constant value for  $D$  and a linear function for  $V$ , as well as more recent work using rather simple basis functions. (See [section 3.6.7](#).)
6. Development of `bayesimp`, an open source software tool to perform Bayesian analysis of impurity transport data. (See [appendix G](#).)

This work has shown that, despite having been investigated since the 1970s [74], the measurement of impurity transport coefficients is still an open topic (though this thesis has made considerable progress on solving it) and the validation of turbulent

transport simulations against the impurity channel is still an open question. Future validation work informed by the present work is sure to yield further insight into the behavior of impurities in magnetically confined plasmas and the ability of turbulent transport simulations to describe reality.

## 4.2 Future directions

### 4.2.1 Profile data analysis

The software for Gaussian process regression developed as part of this thesis work has reached a high state of development and is being used for routine data analysis. That being said, further improvements to speed up the analysis could be made, such as a generalization of the fast approximate methods described in [297] to handle derivatives and nonstationary Gaussian processes. This combined with more efficient selection of quadrature points would make routine inclusion of TCI data in profile fits more practical. Given that the majority of the time to fit a profile with GPR is spent optimizing or sampling the hyperparameters, there is considerable room for improvement through selection of more efficient optimizers and samplers and/or the development of fast techniques to approximate the hyperparameters. The logical next step to improve the quality of profile data analysis is to incorporate the new techniques for nonstationary GPR developed for this thesis into an integrated data analysis such as was described in [section 2.3.10](#).

### 4.2.2 Impurity transport

As was discussed in [section 3.7](#), this thesis has demonstrated that the new system for inferring impurity transport coefficients works on synthetic data, but more work is needed before it is practical to apply it to real experimental data. To make the new analysis practical requires considerable development to eliminate the bottlenecks between STRAHL and the rest of the forward model and to deploy the analysis code on a massively-parallel computer. Equipped with a more efficient analysis code, it would then be possible to properly handle sawteeth, account for atomic physics uncertainties and incorporate data from broadband SXR diagnostics. A key shortcoming of the present analysis is that the STRAHL code is 1D so it is necessary to ignore poloidal asymmetries. Fixing this will require the use of a 2D impurity transport code. With these improvements in place it would then be possible to revisit the validation of transport simulations using the impurity channel. That being said, given the issues inherent to inferring impurity transport coefficients, it is probably more useful to develop validation metrics which use data from lower on the primacy hierarchy.

# Bibliography

- [1] Jeffrey Freidberg. *Plasma Physics and Fusion Energy*. The Edinburgh Building, Cambridge CB2 8RU, UK: Cambridge University Press, 2007. ISBN: 978-0521733175. URL: <http://ebooks.cambridge.org/ebook.jsf?bid=CB09780511755705>.
- [2] John Wesson. *Tokamaks*. 4<sup>th</sup> edition. International Series of Monographs on Physics 149. Great Clarendon Street, Oxford OX2 6DP, United Kingdom: Oxford University Press, 2011. ISBN: 978-0199592234. URL: <https://global.oup.com/academic/product/tokamaks-9780199592234>.
- [3] ITER Physics Expert Group on Confinement and Transport, ITER Physics Expert Group on Confinement Modelling and Database, and ITER Physics Basis Editors. “Chapter 2: Plasma confinement and transport”. In: *Nuclear Fusion* 39.12 (1999), pages 2175–2249. DOI: [10.1088/0029-5515/39/12/302](https://doi.org/10.1088/0029-5515/39/12/302). URL: <http://iopscience.iop.org/0029-5515/39/12/302>.
- [4] Peter C. Stangeby. *The Plasma Boundary of Magnetic Fusion Devices*. Plasma Physics Series. IOP Publishing, Ltd., 2000. ISBN: 0-7503-0559-2.
- [5] A. Loarte, B. Lipschultz, A. S. Kukushkin, G. F. Matthews, P. C. Stangeby, N. Asakura, G. F. Counsell, G. Federici, A. Kallenbach, K. Krieger, A. Mahdavi, V. Philipps, D. Reiter, J. Roth, J. Strachan, D. Whyte, R. Doerner, T. Eich, W. Fundamenski, A. Herrmann, M. Fenstermacher, P. Ghendrih, M. Groth, A. Kirschner, S. Konoshima, B. LaBombard, P. Lang, A. W. Leonard, P. Monier-Garbet, R. Neu, H. Pacher, B. Pegourie, R. A. Pitts, S. Takamura, J. Terry, E. Tsitrone, and the ITPA Scrape-off Layer and Divertor Physics Topical Group. “Chapter 4: Power and particle control”. In: *Nuclear Fusion* 47.6 (2007), s203–s263. DOI: [10.1088/0029-5515/47/6/S04](https://doi.org/10.1088/0029-5515/47/6/S04). URL: <http://iopscience.iop.org/0029-5515/47/6/S04>.
- [6] M. Z. Tokar, J. Rapp, G. Bertschinger, L. Könen, H. R. Koslowski, A. Krämer-Flecken, V. Philipps, U. Samm, and B. Unterberg. “Nature of high-Z impurity accumulation in tokamaks”. In: *Nuclear Fusion* 37.12 (1997), pages 1691–1708. DOI: [10.1088/0029-5515/37/12/I03](https://doi.org/10.1088/0029-5515/37/12/I03). URL: <http://iopscience.iop.org/0029-5515/37/12/I03>.



- [7] S. P. Hirshman and D. J. Sigmar. “Neoclassical transport of impurities in tokamak plasmas”. In: *Nuclear Fusion* 21.9 (1981), pages 1079–1201. DOI: [10.1088/0029-5515/21/9/003](https://doi.org/10.1088/0029-5515/21/9/003). URL: <http://iopscience.iop.org/0029-5515/21/9/003>.
- [8] Per Helander and Dieter J. Sigmar. *Collisional Transport in Magnetized Plasmas*. Cambridge Monographs on Plasma Physics 4. Cambridge University Press, 2002. ISBN: 978-0-521-02098-5. URL: <http://www.cambridge.org/us/academic/subjects/physics/plasma-physics-and-fusion-physics/collisional-transport-magnetized-plasmas>.
- [9] J. Candy and R. E. Waltz. “An Eulerian gyrokinetic-Maxwell solver”. In: *Journal of Computational Physics* 186.2 (Apr. 2003), pages 545–581. DOI: [10.1016/S0021-9991\(03\)00079-2](https://doi.org/10.1016/S0021-9991(03)00079-2). URL: <http://www.sciencedirect.com/science/article/pii/S0021999103000792>.
- [10] J. Candy and E. Belli. *GYRO Technical Guide*. Technical report. General Atomics, 2015. URL: [https://github.com/gafusion/doc/raw/master/gyro/gyro\\_technical\\_manual.pdf](https://github.com/gafusion/doc/raw/master/gyro/gyro_technical_manual.pdf).
- [11] C. Holland, J. Candy, R. E. Waltz, A. E. White, G. R. McKee, M. W. Shafer, L. Schmitz, and G. R. Tynan. “Validating simulations of core tokamak turbulence: current status and future directions”. In: *Journal of Physics: Conference Series* 125.1 (2008), 012043. DOI: [10.1088/1742-6596/125/1/012043](https://doi.org/10.1088/1742-6596/125/1/012043). URL: <http://iopscience.iop.org/1742-6596/125/1/012043>.
- [12] A. E. White, L. Schmitz, G. R. McKee, C. Holland, W. A. Peebles, T. A. Carter, M. W. Shafer, M. E. Austin, K. H. Burrell, J. Candy, J. C. DeBoo, E. J. Doyle, M. A. Makowski, R. Prater, T. L. Rhodes, G. M. Staebler, G. R. Tynan, R. E. Waltz, and G. Wang. “Measurements of core electron temperature and density fluctuations in DIII-D and comparison to nonlinear gyrokinetic simulations”. In: *Physics of Plasmas* 15.5 (2008), 056116. DOI: [10.1063/1.2895408](https://doi.org/10.1063/1.2895408). URL: <http://scitation.aip.org/content/aip/journal/pop/15/5/10.1063/1.2895408>.
- [13] A. Casati, T. Gerbaud, P. Hennequin, C. Bourdelle, J. Candy, F. Clairet, X. Garbet, V. Grandgirand, Ö. D. Gürçan, S. Heuraux, G. T. Hoang, C. Honoré, F. Imbeaux, R. Sabot, Y. Sarazin, L. Vermare, and R. E. Waltz. “Turbulence in the TORE SUPRA Tokamak: Measurements and Validation of Nonlinear Simulations”. In: *Physical Review Letters* 102.16 (Apr. 2009), 165005. DOI: [10.1103/PhysRevLett.102.165005](https://doi.org/10.1103/PhysRevLett.102.165005). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.102.165005>.

- [14] C. Holland, A. E. White, G. R. McKee, M. W. Shafer, J. Candy, R. E. Waltz, L. Schmitz, and G. R. Tynan. "Implementation and application of two synthetic diagnostics for validating simulations of core tokamak turbulence". In: *Physics of Plasmas* 16.5 (2009), 052301. DOI: [10.1063/1.3085792](https://doi.org/10.1063/1.3085792). URL: <http://scitation.aip.org/content/aip/journal/pop/16/5/10.1063/1.3085792>.
- [15] C. Holland, L. Schmitz, T. L. Rhodes, W. A. Peebles, J. C. Hillesheim, G. Wang, L. Zeng, E. J. Doyle, S. P. Smith, R. Prater, K. H. Burrell, J. Candy, R. E. Waltz, J. E. Kinsey, G. M. Staebler, J. C. DeBoo, C. C. Petty, G. R. McKee, Z. Yan, and A. E. White. "Advances in validating gyrokinetic turbulence models against L- and H-mode plasmas". In: *Physics of Plasmas* 18.5 (2011), 056113. DOI: [10.1063/1.3574518](https://doi.org/10.1063/1.3574518). URL: <http://scitation.aip.org/content/aip/journal/pop/18/5/10.1063/1.3574518>.
- [16] T. L. Rhodes, C. Holland, S. P. Smith, A. E. White, K. H. Burrell, J. Candy, J. C. DeBoo, E. J. Doyle, J. C. Hillesheim, J. E. Kinsey, G. R. McKee, D. Mikkelsen, W. A. Peebles, C. C. Petty, R. Prater, S. Parker, Y. Chen, L. Schmitz, G. M. Staebler, R. E. Waltz, G. Wang, Z. Yan, and L. Zeng. "L-mode validation studies of gyrokinetic turbulence simulations via multiscale and multifield turbulence measurements on the DIII-D tokamak". In: *Nuclear Fusion* 51.6 (2011), 063022. DOI: [10.1088/0029-5515/51/6/063022](https://doi.org/10.1088/0029-5515/51/6/063022). URL: <http://iopscience.iop.org/0029-5515/51/6/063022>.
- [17] C. Holland, C. C. Petty, L. Schmitz, K. H. Burrell, G. R. McKee, T. L. Rhodes, and J. Candy. "Progress in GYRO validation studies of DIII-D H-mode plasmas". In: *Nuclear Fusion* 52.11 (2012), 114007. DOI: [10.1088/0029-5515/52/11/114007](https://doi.org/10.1088/0029-5515/52/11/114007). URL: <http://iopscience.iop.org/0029-5515/52/11/114007>.
- [18] N. T. Howard, M. Greenwald, D. R. Mikkelsen, M. L. Reinke, A. E. White, D. Ernst, Y. Podpaly, and J. Candy. "Quantitative comparison of experimental impurity transport with nonlinear gyrokinetic simulation in an Alcator C-Mod L-mode plasma". In: *Nuclear Fusion* 52.6 (2012), 063002. DOI: [10.1088/0029-5515/52/6/063002](https://doi.org/10.1088/0029-5515/52/6/063002). URL: <http://iopscience.iop.org/0029-5515/52/6/063002>.
- [19] N. T. Howard, M. Greenwald, D. R. Mikkelsen, A. E. White, M. L. Reinke, D. Ernst, Y. Podpaly, and J. Candy. "Measurement of plasma current dependent changes in impurity transport and comparison with nonlinear gyrokinetic simulation". In: *Physics of Plasmas* 19.5 (2012), 056110. DOI: [10.1063/1.3694113](https://doi.org/10.1063/1.3694113). URL: <http://scitation.aip.org/content/aip/journal/pop/19/5/10.1063/1.3694113>.

- [20] D. A. Spong, E. M. Bass, W. Deng, W. W. Heidbrink, Z. Lin, B. Tobias, M. A. Van Zeeland, M. E. Austin, C. W. Domier, and N. C. Luhmann Jr. “Verification and validation of linear gyrokinetic simulation of Alfvén eigenmodes in the DIII-D tokamak”. In: *Physics of Plasmas* 19.8 (2012), 082511. DOI: [10.1063/1.4747505](https://doi.org/10.1063/1.4747505). URL: <http://scitation.aip.org/content/aip/journal/pop/19/8/10.1063/1.4747505>.
- [21] F. J. Casson, R. M. McDermott, C. Angioni, Y. Camenen, R. Dux, E. Fable, R. Fischer, B. Geiger, P. Manas, L. Menchero, G. Tardini, and the ASDEX Upgrade Team. “Validation of gyrokinetic modelling of light impurity transport including rotation in ASDEX Upgrade”. In: *Nuclear Fusion* 53.6 (2013), 063026. DOI: [10.1088/0029-5515/53/6/063026](https://doi.org/10.1088/0029-5515/53/6/063026). URL: <http://iopscience.iop.org/0029-5515/53/6/063026>.
- [22] M. Greenwald, A. Bader, S. Baek, H. Barnard, W. Beck, W. Bergerson, I. Bespamyatnov, M. Bitter, P. Bonoli, M. Brookman, D. Brower, D. Brunner, W. Burke, J. Candy, M. Chilenski, M. Chung, M. Churchill, I. Cziegler, E. Davis, G. Dekow, L. Delgado-Aparicio, A. Diallo, W. Ding, A. Dominguez, R. Ellis, P. Ennever, D. Ernst, I. Faust, C. Fiore, E. Fitzgerald, T. Fredian, O. E. Garcia, C. Gao, M. Garrett, T. Golfopoulos, R. Granetz, R. Goebner, S. Harrison, R. Harvey, Z. Hartwig, K. Hill, J. Hillairet, N. Howard, A. E. Hubbard, J. W. Hughes, I. Hutchinson, J. Irby, A. N. James, A. Kanojia, C. Kasten, J. Kesner, C. Kessel, R. Kube, B. LaBombard, C. Lau, J. Lee, K. Liao, Y. Lin, B. Lipschultz, Y. Ma, E. Marmor, P. McGibbon, O. Meneghini, D. Mikkelsen, D. Miller, R. Mumgaard, R. Murray, R. Ochoukov, G. Olynyk, D. Pace, S. Park, R. Parker, Y. Podpaly, M. Porkolab, M. Preynas, I. Pusztai, M. Reinke, J. Rice, W. Rowan, S. Scott, S. Shiraiwa, J. Sierchio, P. Snyder, B. Sorbom, V. Soukhanovskii, J. Stillerman, L. Sugiyama, C. Sung, D. Terry, J. Terry, C. Theiler, N. Tsuji, R. Vieira, J. Walk, G. Wallace, A. White, D. Whyte, J. Wilson, S. Wolfe, K. Woller, G. Wright, J. Wright, S. Wukitch, G. Wurden, P. Xu, C. Yang, and S. Zweben. “Overview of experimental results and code validation activities at Alcator C-Mod”. In: *Nuclear Fusion* 53.10 (2013), 104004. DOI: [10.1088/0029-5515/53/10/104004](https://doi.org/10.1088/0029-5515/53/10/104004). URL: <http://iopscience.iop.org/0029-5515/53/10/104004>.
- [23] C. Holland, J. E. Kinsey, J. C. DeBoo, K. H. Burrell, T. C. Luce, S. P. Smith, C. C. Petty, A. E. White, T. L. Rhodes, L. Schmitz, E. J. Doyle, J. C. Hillesheim, G. R. McKee, Z. Yan, G. Wang, L. Zeng, B. A. Grierson, A. Marinori, P. Mantica, P. B. Snyder, R. E. Waltz, G. M. Staebler, and J. Candy. “Validation studies of gyrofluid and gyrokinetic predictions of transport and turbulence stiffness using the DIII-D tokamak”. In: *Nuclear Fusion* 53.8 (2013), 083027.

- DOI: [10.1088/0029-5515/53/8/083027](https://doi.org/10.1088/0029-5515/53/8/083027). URL: <http://iopscience.iop.org/0029-5515/53/8/083027>.
- [24] N. T. Howard, A. E. White, M. L. Reinke, M. Greenwald, C. Holland, J. Candy, and J. R. Walk. “Validation of the gyrokinetic model in ITG and TEM dominated L-mode plasmas”. In: *Nuclear Fusion* 53.12 (2013), 123011. DOI: [10.1088/0029-5515/53/12/123011](https://doi.org/10.1088/0029-5515/53/12/123011). URL: <http://iopscience.iop.org/0029-5515/53/12/123011>.
- [25] N. T. Howard, A. E. White, M. Greenwald, M. L. Reinke, J. Walk, C. Holland, J. Candy, and T. Görler. “Investigation of the transport shortfall in Alcator C-Mod L-mode plasmas”. In: *Physics of Plasmas* 20.3 (2013), 032510. DOI: [10.1063/1.4795301](https://doi.org/10.1063/1.4795301). URL: <http://scitation.aip.org/content/aip/journal/pop/20/3/10.1063/1.4795301>.
- [26] A. E. White, N. T. Howard, M. Greenwald, M. L. Reinke, C. Sung, S. Baek, M. Barnes, J. Candy, A. Dominguez, D. Ernst, C. Gao, A. E. Hubbard, J. W. Hughes, Y. Lin, D. Mikkelsen, F. Parra, M. Porkolab, J. E. Rice, J. Walk, S. J. Wukitch, and Alcator C-Mod Team. “Multi-channel transport experiments at Alcator C-Mod and comparison with gyrokinetic simulations”. In: *Physics of Plasmas* 20.5 (2013), 056106. DOI: [10.1063/1.4803089](https://doi.org/10.1063/1.4803089). URL: <http://scitation.aip.org/content/aip/journal/pop/20/5/10.1063/1.4803089>.
- [27] N. T. Howard, A. E. White, M. Greenwald, C. Holland, and J. Candy. “Multi-scale gyrokinetic simulation of Alcator C-Mod tokamak discharges”. In: *Physics of Plasmas* 21.3 (2014), 032308. DOI: [10.1063/1.4869078](https://doi.org/10.1063/1.4869078). URL: <http://scitation.aip.org/content/aip/journal/pop/21/3/10.1063/1.4869078>.
- [28] N. Bonanomi, P. Mantica, G. Szepesi, N. Hawkes, E. Lerche, P. Migliano, A. Peeters, C. Sozzi, M. Tsalas, D. Van Eester, and JET Contributors. “Trapped electron mode driven electron heat transport in JET: experimental investigation and gyro-kinetic theory validation”. In: *Nuclear Fusion* 55.11 (2015), 113016. DOI: [10.1088/0029-5515/55/11/113016](https://doi.org/10.1088/0029-5515/55/11/113016). URL: <http://iopscience.iop.org/0029-5515/55/11/113016>.
- [29] A. E. White, N. T. Howard, A. J. Creely, M. A. Chilenski, M. Greenwald, A. E. Hubbard, J. W. Hughes, E. Marmor, J. E. Rice, J. M. Sierchio, C. Sung, J. R. Walk, D. G. Whyte, D. R. Mikkelsen, E. M. Edlund, C. Kung, C. Holland, J. Candy, C. C. Petty, M. L. Reinke, and C. Theiler. “Nonlinear gyrokinetic simulations of the I-mode high confinement regime and comparisons with experiment”. In: *Physics of Plasmas* 22.5 (2015), 056109. DOI: [10.1063/1.4869078](https://doi.org/10.1063/1.4869078).

4921150. URL: <http://scitation.aip.org/content/aip/journal/pop/22/5/10.1063/1.4921150>.
- [30] T. Görler, A. E. White, D. Told, F. Jenko, C. Holland, and T. L. Rhodes. “On the Validation of Gyrokinetic L-Mode Simulations”. In: *Fusion Science and Technology* 69.2 (Apr. 2016), pages 537–545. DOI: [10.13182/FST15-182](https://doi.org/10.13182/FST15-182). URL: [http://www.ans.org/pubs/journals/fst/a\\_38303](http://www.ans.org/pubs/journals/fst/a_38303).
- [31] William L. Oberkampf, Timothy G. Trucano, and Charles Hirsch. “Verification, validation, and predictive capability in computational engineering and physics”. In: *Applied Mechanics Reviews* 57.5 (2004), pages 345–384. DOI: [10.1115/1.1767847](https://doi.org/10.1115/1.1767847). URL: <http://appliedmechanicsreviews.asmedigitalcollection.asme.org/article.aspx?articleid=1398084>.
- [32] William L. Oberkampf and Matthew F. Barone. “Measures of agreement between computation and experiment: Validation metrics”. In: *Journal of Computational Physics* 217.1 (2006), pages 5–36. DOI: [10.1016/j.jcp.2006.03.037](https://doi.org/10.1016/j.jcp.2006.03.037). URL: <http://www.sciencedirect.com/science/article/pii/S0021999106001860>.
- [33] William L. Oberkampf and Timothy G. Trucano. “Verification and validation benchmarks”. In: *Nuclear Engineering and Design* 238.3 (Mar. 2008), pages 716–743. DOI: [10.1016/j.nucengdes.2007.02.032](https://doi.org/10.1016/j.nucengdes.2007.02.032). URL: <http://www.sciencedirect.com/science/article/pii/S0029549307003548>.
- [34] P. W. Terry, M. Greenwald, J.-N. Leboeuf, G. R. McKee, D. R. Mikkelsen, W. M. Nevins, D. E. Newman, D. P. Stotler, Task Group on Verification and Validation, us Burning Plasma Organization, and us Transport Task Force. “Validation in fusion research: Towards guidelines and best practices”. In: *Physics of Plasmas* 15.6 (2008), 062503. DOI: [10.1063/1.2928909](https://doi.org/10.1063/1.2928909). URL: <http://scitation.aip.org/content/aip/journal/pop/15/6/10.1063/1.2928909>.
- [35] Patrick J. Roache. *Fundamentals of Verification and Validation*. Hermosa Publishers, 2009. ISBN: 978-0913478127.
- [36] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010. ISBN: 978-0521113601. URL: <http://www.cambridge.org/us/academic/subjects/computer-science/scientific-computing-scientific-software/verification-and-validation-scientific-computing>.

- [37] Christopher J. Roy and William L. Oberkampf. “A Complete Framework for Verification, Validation, and Uncertainty Quantification in Scientific Computing (Invited)”. In: *48<sup>th</sup> AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. (Orlando, Florida, Jan. 4–7, 2010). American Institute of Aeronautics and Astronautics. 2010, 2010–124. DOI: [10.2514/6.2010-124](https://doi.org/10.2514/6.2010-124). URL: <http://arc.aiaa.org/doi/10.2514/6.2010-124>.
- [38] Ralph C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*. Computational Science and Engineering. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014. ISBN: 978-1611973211. URL: <http://bookstore.siam.org/cs12/>.
- [39] M. B. Chowdhuri, J. Ghosh, S. Banerjee, Ritu Dey, R. Manchanda, Vinay Kumar, P. Vasu, K. M. Patel, P. K. Atrey, Y. Shankara Joisa, C. V. S. Rao, R. L. Tanna, D. Raju, P. K. Chattopadhyay, R. Jha, C. N. Gupta, S. B. Bhatt, Y. C. Saxena, and the Aditya Team. “Investigation of oxygen impurity transport using the O<sup>4+</sup> visible spectral line in the Aditya tokamak”. In: *Nuclear Fusion* 53.2 (2013), 023006. DOI: [10.1088/0029-5515/53/2/023006](https://doi.org/10.1088/0029-5515/53/2/023006). URL: <http://iopscience.iop.org/0029-5515/53/2/023006>.
- [40] E. S. Marmor, J. E. Rice, and S. L. Allen. “Confinement of Injected Silicon in the Alcator-A Tokamak”. In: *Physical Review Letters* 45.25 (Dec. 1980), pages 2025–2028. DOI: [10.1103/PhysRevLett.45.2025](https://doi.org/10.1103/PhysRevLett.45.2025). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.45.2025>.
- [41] S. L. Allen, H. W. Moos, E. S. Marmor, and J. E. Rice. “Radial dependence of decay times of emissions from injected impurities in Alcator A tokamak”. In: *Nuclear Fusion* 23.3 (1983), pages 303–309. DOI: [10.1088/0029-5515/23/3/004](https://doi.org/10.1088/0029-5515/23/3/004). URL: <http://iopscience.iop.org/0029-5515/23/3/004>.
- [42] E. S. Marmor, J. E. Rice, J. L. Terry, and F. H. Seguin. “Impurity injection experiments on the Alcator C tokamak”. In: *Nuclear Fusion* 22.12 (1982), pages 1567–1575. DOI: [10.1088/0029-5515/22/12/003](https://doi.org/10.1088/0029-5515/22/12/003). URL: <http://iopscience.iop.org/0029-5515/22/12/003>.
- [43] J. E. Rice, E. S. Marmor, B. Lipschultz, and J. L. Terry. “Influx and sources of medium- and high-Z intrinsic impurities in the Alcator C tokamak”. In: *Nuclear Fusion* 24.3 (1984), pages 329–336. DOI: [10.1088/0029-5515/24/3/007](https://doi.org/10.1088/0029-5515/24/3/007). URL: <http://iopscience.iop.org/0029-5515/24/3/007>.

- [44] M. A. Graf, J. E. Rice, J. L. Terry, E. S. Marmor, J. A. Goetz, G. M. McCracken, F. Bombarda, and M. J. May. "Spectroscopic measurement of impurity transport coefficients and penetration efficiencies in Alcator C-Mod plasmas". In: *Review of Scientific Instruments* 66.1 (Jan. 1995), pages 636–638. DOI: [10.1063/1.1146311](https://doi.org/10.1063/1.1146311). URL: <http://scitation.aip.org/content/aip/journal/rsi/66/1/10.1063/1.1146311>.
- [45] Michael A. Graf. "Impurity Injection Experiments on the Alcator C-Mod Tokamak". PhD thesis. Massachusetts Institute of Technology, June 1995. URL: <http://hdl.handle.net/1721.1/11888>.
- [46] J. E. Rice, M. A. Graf, J. L. Terry, E. S. Marmor, K. Giesing, and F. Bombarda. "X-ray observations of helium-like scandium from the Alcator C-Mod tokamak". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 28.5 (1995), pages 893–905. DOI: [10.1088/0953-4075/28/5/021](https://doi.org/10.1088/0953-4075/28/5/021). URL: <http://iopscience.iop.org/0953-4075/28/5/021>.
- [47] 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". In: *Review of Scientific Instruments* 66.1 (Jan. 1995), pages 752–754. DOI: [10.1063/1.1146279](https://doi.org/10.1063/1.1146279). URL: <http://scitation.aip.org/content/aip/journal/rsi/66/1/10.1063/1.1146279>.
- [48] J. E. Rice, J. L. Terry, J. A. Goetz, Y. Wang, E. S. Marmor, M. Greenwald, I. Hutchinson, Y. Takase, S. Wolfe, H. Ohkawa, and A. Hubbard. "Impurity transport in Alcator C-Mod plasmas". In: *Physics of Plasmas* 4.5 (May 1997), pages 1605–1609. DOI: [10.1063/1.872291](https://doi.org/10.1063/1.872291). URL: <http://scitation.aip.org/content/aip/journal/pop/4/5/10.1063/1.872291>.
- [49] J. E. Rice, J. A. Goetz, R. S. Granetz, M. J. Greenwald, A. E. Hubbard, I. H. Hutchinson, E. S. Marmor, D. Mossessian, T. Sunn Pedersen, J. A. Snipes, J. L. Terry, and S. M. Wolfe. "Impurity toroidal rotation and transport in Alcator C-Mod ohmic high confinement mode plasmas". In: *Physics of Plasmas* 7.5 (May 2000), pages 1825–1830. DOI: [10.1063/1.874004](https://doi.org/10.1063/1.874004). URL: <http://scitation.aip.org/content/aip/journal/pop/7/5/10.1063/1.874004>.
- [50] T. Sunn Pedersen, R. S. Granetz, A. E. Hubbard, I. H. Hutchinson, E. S. Marmor, J. E. Rice, and J. Terry. "Radial impurity transport in the H mode transport barrier region in Alcator C-Mod". In: *Nuclear Fusion* 40.10 (2000), pages 1795–1804. DOI: [10.1088/0029-5515/40/10/310](https://doi.org/10.1088/0029-5515/40/10/310). URL: <http://iopscience.iop.org/0029-5515/40/10/310>.



- [51] Thomas Sunn Pedersen. “Edge Plasma Phenomena in the Alcator C-Mod Tokamak measured by High Resolution X-Ray Imaging Diagnostics”. PhD thesis. Massachusetts Institute of Technology, June 2000. URL: <http://hdl.handle.net/1721.1/9025>.
- [52] M. Greenwald, N. Basse, P. Bonoli, R. Bravenec, E. Edlund, D. Ernst, C. Fiore, R. Granetz, A. Hubbard, J. Hughes, I. Hutchinson, J. Irby, B. LaBombard, L. Lin, Y. Lin, B. Lipschultz, E. Marmor, D. Mikkelsen, D. Mossessian, P. Phillips, M. Porkolab, J. Rice, W. Rowan, S. Scott, J. Snipes, J. Terry, S. Wolfe, S. Wukitch, and K. Zhurovich. “Confinement and Transport Research in Alcator C-Mod”. In: *Fusion Science and Technology* 51.3 (Apr. 2007), pages 266–287. URL: [http://www.ans.org/pubs/journals/fst/a\\_1422](http://www.ans.org/pubs/journals/fst/a_1422).
- [53] J. E. Rice, J. L. Terry, E. S. Marmor, R. S. Granetz, M. J. Greenwald, A. E. Hubbard, J. H. Irby, S. M. Wolfe, and T. Sunn Pedersen. “Impurity transport in Alcator C-Mod plasmas”. In: *Fusion Science and Technology* 51.3 (2007), pages 357–368. URL: [http://www.ans.org/pubs/journals/fst/a\\_1427](http://www.ans.org/pubs/journals/fst/a_1427).
- [54] N. T. Howard, M. Greenwald, and J. E. Rice. “Characterization of impurity confinement on Alcator C-Mod using a multi-pulse laser blow-off system”. In: *Review of Scientific Instruments* 82.3 (2011), 033512. DOI: [10.1063/1.3565448](https://doi.org/10.1063/1.3565448). URL: <http://scitation.aip.org/content/aip/journal/rsi/82/3/10.1063/1.3565448>.
- [55] Nathaniel Thomas Howard. “Experimental and Gyrokinetic Studies of Impurity Transport in the Core of Alcator C-Mod Plasmas”. PhD thesis. Massachusetts Institute of Technology, June 2012. URL: <http://hdl.handle.net/1721.1/76920>.
- [56] N. T. Howard, M. Greenwald, D. R. Mikkelsen, A. E. White, M. L. Reinke, D. Ernst, Y. Podpaly, and J. Candy. “Quantitative Comparison of Experimental and Gyrokinetic Simulated ICRH and  $I_p$  Dependent Impurity Transport”. In: *Proceedings of the 24<sup>th</sup> IAEA Fusion Energy Conference*. (San Diego, CA, USA, Oct. 8–13, 2012). Volume IAEA-CN-197. International Atomic Energy Agency. 2012, EX/P3–23. URL: <https://conferences.iaea.org/indico/event/10/session/6/contribution/296>.
- [57] N. T. Howard, A. E. White, M. Greenwald, C. Holland, J. Candy, and J. E. Rice. “Impurity transport, turbulence transitions and intrinsic rotation in Alcator C-Mod plasmas”. In: *Plasma Physics and Controlled Fusion* 56.12 (2014), 124004. DOI: [10.1088/0741-3335/56/12/124004](https://doi.org/10.1088/0741-3335/56/12/124004). URL: <http://iopscience.iop.org/0741-3335/56/12/124004>.



- [58] A. Loarte, M. L. Reinke, A. R. Polevoi, M. Hosokawa, M. Chilenski, N. Howard, A. Hubbard, J. W. Hughes, J. E. Rice, J. Walk, Alcator C-Mod Team, F. Köchl, T. Pütterich, R. Dux, and V. E. Zhogolev. “Tungsten impurity transport experiments in Alcator C-Mod to address high priority research and development for ITER”. In: *Physics of Plasmas* 22.5 (2015), 056117. DOI: [10.1063/1.4921253](https://doi.org/10.1063/1.4921253). URL: <http://scitation.aip.org/content/aip/journal/pop/22/5/10.1063/1.4921253>.
- [59] J. E. Rice, M. L. Reinke, C. Gao, N. T. Howard, M. A. Chilenski, L. Delgado-Aparicio, R. S. Granetz, M. J. Greenwald, A. E. Hubbard, J. W. Hughes, J. H. Irby, Y. Lin, E. S. Marmor, R. T. Mumgaard, S. D. Scott, J. L. Terry, J. R. Walk, A. E. White, D. G. Whyte, S. M. Wolfe, and S. J. Wukitch. “Core impurity transport in Alcator C-Mod L-, I- and H-mode plasmas”. In: *Nuclear Fusion* 55.3 (2015), 033014. DOI: [10.1088/0029-5515/55/3/033014](https://doi.org/10.1088/0029-5515/55/3/033014). URL: <http://iopscience.iop.org/0029-5515/55/3/033014>.
- [60] K. H. Behringer and K. Büchl. “Impurity transport studies in ASDEX by means of neon-seeded pellets”. In: *Nuclear Fusion* 29.3 (1989), pages 415–422. DOI: [10.1088/0029-5515/29/3/005](https://doi.org/10.1088/0029-5515/29/3/005). URL: <http://iopscience.iop.org/0029-5515/29/3/005>.
- [61] K. Krieger, G. Fussmann, and ASDEX Team. “Determination of impurity transport coefficients by harmonic analysis”. In: *Nuclear Fusion* 30.11 (1990), pages 2392–2396. DOI: [10.1088/0029-5515/30/11/015](https://doi.org/10.1088/0029-5515/30/11/015). URL: <http://iopscience.iop.org/0029-5515/30/11/015>.
- [62] Sascha de Peña Hempel. *Untersuchungen zum Transport leichter Verunreinigungen an ASDEX-Upgrade*. Technical report 10/8. Garching Bei München: Max-Planck-Institut für Plasmaphysik, Aug. 1997. URL: <http://hdl.handle.net/11858/00-001M-0000-0027-610D-A>.
- [63] R. Dux, A. G. Peeters, A. Gude, A. Kallenbach, R. Neu, and ASDEX Upgrade Team. “Z dependence of the core impurity transport in ASDEX Upgrade H mode discharges”. In: *Nuclear Fusion* 39.11 (1999), pages 1509–1522. DOI: [10.1088/0029-5515/39/11/302](https://doi.org/10.1088/0029-5515/39/11/302). URL: <http://iopscience.iop.org/0029-5515/39/11/302>.
- [64] R. Dux, A. G. Peeters, A. Gude, A. Kallenbach, R. Neu, and ASDEX Upgrade Team. “Measurement of Impurity Transport Coefficients in the Confined Plasma of ASDEX Upgrade”. In: *18<sup>th</sup> International Conference on Fusion Energy*. (Sorrento, Italy, Oct. 4–10, 2000). Volume IAEA-CN-77. International Atomic Energy Agency. 2000, EXP5/32. URL: [http://www-pub.iaea.org/MTCD/publications/PDF/csp\\_008c/html/node231.htm](http://www-pub.iaea.org/MTCD/publications/PDF/csp_008c/html/node231.htm).

- [65] Ralph Dux. “Impurity transport in ASDEX Upgrade”. In: *Fusion Science and Technology* 44.3 (Nov. 2003), pages 708–715. URL: [http://www.ans.org/pubs/journals/fst/a\\_409](http://www.ans.org/pubs/journals/fst/a_409).
- [66] R. Dux, R. Neu, A. G. Peeters, G. Pereverzev, A. Mück, F. Ryter, J. Stober, and ASDEX Upgrade Team. “Influence of the heating profile on impurity transport in ASDEX Upgrade”. In: *Plasma Physics and Controlled Fusion* 45.9 (2003), pages 1815–1825. DOI: [10.1088/0741-3335/45/9/317](https://doi.org/10.1088/0741-3335/45/9/317). URL: <http://iopscience.iop.org/0741-3335/45/9/317>.
- [67] R. Dux, R. Neu, C. F. Maggi, A. G. Peeters, T. Pütterich, G. Pereverzev, A. Mück, F. Ryter, J. Stober, B. Zaniol, and ASDEX Upgrade team. “Impurity Transport and Control in ASDEX Upgrade”. In: *20<sup>th</sup> IAEA Fusion Energy Conference*. (Vilamoura, Portugal, Nov. 1–6, 2004). Volume IAEA-CN-116. International Atomic Energy Agency. 2004, EX/P6-14. URL: [http://www-pub.iaea.org/mtcd/meetings/PDFplus/fusion-20-preprints/EX\\_P6-14.pdf](http://www-pub.iaea.org/mtcd/meetings/PDFplus/fusion-20-preprints/EX_P6-14.pdf).
- [68] C. Angioni, L. Carraro, T. Dannert, N. Dubuit, R. Dux, C. Fuchs, X. Garbet, L. Garzotti, C. Giroud, R. Guirlet, F. Jenko, O. J. W. F. Kardaun, L. Lauro-Taroni, P. Mantica, M. Maslov, V. Naulin, R. Neu, A. G. Peeters, G. Pereverzev, M. E. Puiatti, T. Pütterich, J. Stober, M. Valovič, M. Valisa, H. Weisen, A. Zabolotsky, ASDEX Upgrade Team, and JET EFDA Contributors. “Particle and impurity transport in the Axial Symmetric Divertor Experiment Upgrade and the Joint European Torus, experimental observations and theoretical understanding”. In: *Physics of Plasmas* 14.5 (2007), 055905. DOI: [10.1063/1.2515300](https://doi.org/10.1063/1.2515300). URL: <http://scitation.aip.org/content/aip/journal/pop/14/5/10.1063/1.2515300>.
- [69] T. Pütterich, R. Dux, M. A. Janzer, R. M. McDermott, and ASDEX Upgrade Team. “ELM flushing and impurity transport in the H-mode edge barrier in ASDEX Upgrade”. In: *Journal of Nuclear Materials* 415.1 (Aug. 2010), s334–s339. DOI: [10.1016/j.jnucmat.2010.09.052](https://doi.org/10.1016/j.jnucmat.2010.09.052). URL: <http://www.sciencedirect.com/science/article/pii/S0022311510005623>.
- [70] M. Sertoli, C. Angioni, R. Dux, R. Neu, T. Pütterich, V. Igochine, and the ASDEX Upgrade Team. “Local effects of ECRH on argon transport in L-mode discharges at ASDEX Upgrade”. In: *Plasma Physics and Controlled Fusion* 53.3 (2011), 035024. DOI: [10.1088/0741-3335/53/3/035024](https://doi.org/10.1088/0741-3335/53/3/035024). URL: <http://iopscience.iop.org/0741-3335/53/3/035024>.

- [71] T. Pütterich, R. Dux, R. Neu, M. Bernert, M. N. A. Beurskens, V. Bobkov, S. Brezinsek, C. Challis, J. W. Coenen, I. Coffey, A. Czarnecka, C. Giroud, P. Jacquet, E. Joffrin, A. Kallenbach, M. Lehnen, E. Lerche, E. de la Luna, S. Marsen, G. Matthews, M.-L. Mayoral, R. M. McDermott, A. Meigs, J. Mlynar, M. Sertoli, G. van Rooij, the ASDEX Upgrade Team, and JET EFDA Contributors. “Observations on the W-transport in the core plasma of JET and ASDEX Upgrade”. In: *Plasma Physics and Controlled Fusion* 55.12 (2013), 124036. DOI: [10.1088/0741-3335/55/12/124036](https://doi.org/10.1088/0741-3335/55/12/124036). URL: <http://iopscience.iop.org/0741-3335/55/12/124036>.
- [72] M. Sertoli, R. Dux, T. Pütterich, and the ASDEX Upgrade Team. “Modification of impurity transport in the presence of saturated  $(m, n) = (1, 1)$  MHD activity at ASDEX Upgrade”. In: *Plasma Physics and Controlled Fusion* 57.7 (2015), 075004. DOI: [10.1088/0741-3335/57/7/075004](https://doi.org/10.1088/0741-3335/57/7/075004). URL: <http://iopscience.iop.org/0741-3335/57/7/075004>.
- [73] S. A. Cohen, J. L. Cecchi, and E. S. Marmor. “Impurity Transport in a Quiescent Tokamak Plasma”. In: *Physical Review Letters* 35.22 (Dec. 1975), pages 1507–1510. DOI: [10.1103/PhysRevLett.35.1507](https://doi.org/10.1103/PhysRevLett.35.1507). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.35.1507>.
- [74] E. S. Marmor, J. L. Cecchi, and S. A. Cohen. “System for rapid injection of metal atoms into plasmas”. In: *Review of Scientific Instruments* 46.9 (Sept. 1975), pages 1149–1154. DOI: [10.1063/1.1134432](https://doi.org/10.1063/1.1134432). URL: <http://scitation.aip.org/content/aip/journal/rsi/46/9/10.1063/1.1134432>.
- [75] R. C. Isler, L. D. Horton, E. C. Crume, M. Murakami, G. L. Bell, T. S. Bigelow, A. C. England, J. C. Glowienka, S. Hiroe, T. C. Jernigan, R. A. Langley, J. F. Lyon, P. K. Mioduszewski, S. Morita, D. A. Rasmussen, J. E. Simpkins, J. B. Wilgen, and W. R. Wing. “Impurity effects and impurity transport in ATF”. In: *Proceedings of the 13<sup>th</sup> International Conference on Fusion Energy*. (Washington, DC, USA, Oct. 1–6, 1990). Volume 2. IAEA-CN-53. International Atomic Energy Agency. Vienna, 1991, pages 685–692. URL: [https://www.iaea.org/inis/collection/NCLCollectionStore/\\_Public/22/082/22082081.pdf#page=701](https://www.iaea.org/inis/collection/NCLCollectionStore/_Public/22/082/22082081.pdf#page=701).
- [76] L. D. Horton, R. C. Isler, H. C. Howe, E. C. Crume, and D. C. Giles. “Impurity transport in the Advanced Toroidal Facility”. In: *Nuclear Fusion* 32.3 (1992), pages 481–487. DOI: [10.1088/0029-5515/32/3/I10](https://doi.org/10.1088/0029-5515/32/3/I10). URL: <http://iopscience.iop.org/0029-5515/32/3/I10>.

- [77] V. A. Soukhanovskii, M. Finkenthal, H. W. Moos, D. Stutman, T. Munsat, B. Jones, D. Hoffman, R. Kaita, and R. Majeski. "Observation of neoclassical impurity transport in Ohmically heated plasmas of CDX-U low aspect ratio tokamak". In: *Plasma Physics and Controlled Fusion* 44.11 (2002), pages 2339–2355. DOI: [10.1088/0741-3335/44/11/304](https://doi.org/10.1088/0741-3335/44/11/304). URL: <http://iopscience.iop.org/0741-3335/44/11/304>.
- [78] T. Hirayama, T. Takizuka, M. Shimada, M. Nagami, S. Konoshima, M. Abe, A. Kameari, A. Kitsunezaki, K. Kodama, S. Sengoku, M. Washizu, T. Yamamoto, K. H. Burrell, G. Zawadzki, N. Brooks, and R. Groebner. "Trace impurity transport in Doublet III beam heated plasma: Comparison of numerical simulations with decay time of titanium line emission in L- and H-mode discharges". In: *Journal of Nuclear Materials* 128–129 (Dec. 1984), pages 271–274. DOI: [10.1016/0022-3115\(84\)90365-9](https://doi.org/10.1016/0022-3115(84)90365-9). URL: <http://www.sciencedirect.com/science/article/pii/0022311584903659>.
- [79] M. E. Perry, N. H. Brooks, D. A. Content, R. A. Hulse, M. Ali Mahdavi, and H. W. Moos. "Impurity transport during the H-mode in DIII-D". In: *Nuclear Fusion* 31.10 (1991), pages 1859–1875. DOI: [10.1088/0029-5515/31/10/005](https://doi.org/10.1088/0029-5515/31/10/005). URL: <http://iopscience.iop.org/0029-5515/31/10/005>.
- [80] Andrew P. Zwicker, Michael Finkenthal, and H. Warren Moos. "Analysis of impurity content and transport in tokamak plasmas using low-resolution xuv spectra". In: *Journal of X-Ray Science and Technology* 4.1 (Dec. 1993), pages 57–66. DOI: [10.1016/S0895-3996\(05\)80033-3](https://doi.org/10.1016/S0895-3996(05)80033-3). URL: <http://www.sciencedirect.com/science/article/pii/S0895399605800333>.
- [81] M. R. Wade, D. G. Whyte, R. D. Wood, and W. P. West. *Characterization of Core Impurity Transport and Accumulation in Various Operating Regimes in DIII-D*. Technical report GA-A22404. General Atomics, July 1996. URL: [https://www.iaea.org/inis/collection/NCLCollectionStore/\\_Public/28/066/28066047.pdf](https://www.iaea.org/inis/collection/NCLCollectionStore/_Public/28/066/28066047.pdf).
- [82] M. R. Wade, T. C. Luce, and C. C. Petty. "Gyroradius Scaling of Helium Transport". In: *Physical Review Letters* 79.3 (July 1997), pages 419–422. DOI: [10.1103/PhysRevLett.79.419](https://doi.org/10.1103/PhysRevLett.79.419). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.79.419>.
- [83] M. R. Wade, W. A. Houlberg, and L. R. Baylor. "Experimental Confirmation of Impurity Convection Driven by the Ion-Temperature Gradient in Toroidal Plasmas". In: *Physical Review Letters* 84.2 (Jan. 2000), pages 282–285. DOI: [10.1103/PhysRevLett.84.282](https://doi.org/10.1103/PhysRevLett.84.282). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.84.282>.

- [84] W. P. West, M. R. Wade, C. M. Greenfield, E. J. Doyle, K. H. Burrell, N. H. Brooks, P. Gohil, R. J. Groebner, G. L. Jackson, J. E. Kinsey, C. J. Lasnier, J. Mandrekas, G. R. McKee, T. L. Rhodes, G. M. Staebler, G. Wang, J. G. Watkins, and L. Zeng. “Energy, particle and impurity transport in quiescent double barrier discharges in DIII-D”. In: *Physics of Plasmas* 9.5 (May 2002), pages 1970–1980. DOI: [10.1063/1.1456064](https://doi.org/10.1063/1.1456064). URL: <http://scitation.aip.org/content/aip/journal/pop/9/5/10.1063/1.1456064>.
- [85] B. A. Grierson, K. H. Burrell, R. M. Nazikian, W. M. Solomon, A. M. Garofalo, E. A. Belli, G. M. Staebler, M. E. Fenstermacher, G. R. McKee, T. E. Evans, D. M. Orlov, S. P. Smith, C. Chrobak, C. Chrystal, and DIII-D Team. “Impurity confinement and transport in high confinement regimes without edge localized modes on DIII-D”. In: *Physics of Plasmas* 22.5 (2015), 055901. DOI: [10.1063/1.4918359](https://doi.org/10.1063/1.4918359). URL: <http://scitation.aip.org/content/aip/journal/pop/22/5/10.1063/1.4918359>.
- [86] F. De Marco, R. Giannella, and G. Mazzitelli. “Behaviour of oxygen impurities in the Frascati Tokamak”. In: *Plasma Physics* 24.3 (1982), pages 257–264. DOI: [10.1088/0032-1028/24/3/004](https://doi.org/10.1088/0032-1028/24/3/004). URL: <http://iopscience.iop.org/0032-1028/24/3/004>.
- [87] D. Pacella, L. Gabellieri, G. Mazzitelli, K. B. Fournier, and M. Finkenthal. “Direct measurement of the impurity radial flux in the FTU plasma core”. In: *Plasma Physics and Controlled Fusion* 39.10 (1997), pages 1501–1508. DOI: [10.1088/0741-3335/39/10/001](https://doi.org/10.1088/0741-3335/39/10/001). URL: <http://iopscience.iop.org/0741-3335/39/10/001>.
- [88] D. Pacella, M. Leigheb, and M. Mattioli. “Space Resolved Measurements of Titanium Soft X-Ray Emissions from FTU Tokamak Plasmas”. In: *Physica Scripta* 57.2 (1998), pages 265–271. DOI: [10.1088/0031-8949/57/2/021](https://doi.org/10.1088/0031-8949/57/2/021). URL: <http://iopscience.iop.org/1402-4896/57/2/021>.
- [89] D. Pacella, F. Romanelli, B. C. Gregory, M. Leigheb, G. Mazzitelli, L. Gabellieri, R. de Angelis, M. Finkenthal, M. May, K. Fournier, and W. H. Goldstein. “Impurity Transport Studies on the FTU Tokamak”. In: *17<sup>th</sup> International Conference on Fusion Energy*. (Yokohama, Japan, Oct. 19–24, 1998). Volume IAEA-CN-69. International Atomic Energy Agency. 1998, EXP1/01. URL: [http://www-pub.iaea.org/MTCD/publications/PDF/csp\\_001c/html/node109.htm](http://www-pub.iaea.org/MTCD/publications/PDF/csp_001c/html/node109.htm).
- [90] M. J. May, M. Finkenthal, H. W. Moos, K. B. Fournier, W. H. Goldstein, M. Mattioli, D. Pacella, G. Mazzitelli, M. Leigheb, and L. Gabellieri. “Observations of the vacuum ultraviolet and x-ray brightness profiles of Fe, Ni,

- and Ge in magnetically confined fusion plasmas”. In: *Physical Review E* 64.3 (Sept. 2001), 036406. DOI: [10.1103/PhysRevE.64.036406](https://doi.org/10.1103/PhysRevE.64.036406). URL: <http://journals.aps.org/pre/abstract/10.1103/PhysRevE.64.036406>.
- [91] M. J. May, M. Finkenthal, H. W. Moos, K. B. Fournier, W. H. Goldstein, M. Mattioli, D. Pacella, G. Mazzitelli, M. Leigheb, and L. Gabellieri. “Improved confinement regime in the current ramp down phase of Frascati Tokamak Upgrade plasmas”. In: *Nuclear Fusion* 42.11 (2002), pages 1299–1303. DOI: [10.1088/0029-5515/42/11/302](https://doi.org/10.1088/0029-5515/42/11/302). URL: <http://iopscience.iop.org/0029-5515/42/11/302>.
- [92] L. Carraro, L. Gabellieri, M. Mattioli, M. Finkenthal, K. Fournier, M. Leigheb, M. E. Puiatti, P. Scarin, M. Valisa, and D. Pacella. “Study of impurity transport in FTU ITB plasmas”. In: *Plasma Physics and Controlled Fusion* 46.2 (2004), pages 389–407. DOI: [10.1088/0741-3335/46/2/005](https://doi.org/10.1088/0741-3335/46/2/005). URL: <http://iopscience.iop.org/0741-3335/46/2/005>.
- [93] M. Leigheb, M. Romanelli, L. Gabellieri, L. Carraro, M. Mattioli, C. Mazzotta, M. E. Puiatti, L. Lauro-Taroni, M. Marinucci, S. Nowak, L. Panaccione, V. Pericoli, P. Smeulders, O. Tudisco, C. Sozzi, M. Valisa, and the FTU team. “Molybdenum transport in high density FTU plasmas with strong radio frequency electron heating”. In: *Plasma Physics and Controlled Fusion* 49.11 (2007), pages 1897–1912. DOI: [10.1088/0741-3335/49/11/010](https://doi.org/10.1088/0741-3335/49/11/010). URL: <http://iopscience.iop.org/0741-3335/49/11/010>.
- [94] Yin-jia Zheng, Rui-rong Wang, Zhen Feng, and Shang-jie Qian. “Investigation on impurity transport using LBO technique”. In: *Plasma Science and Technology* 3.3 (2001), pages 797–802. DOI: [10.1088/1009-0630/3/3/007](https://doi.org/10.1088/1009-0630/3/3/007). URL: <http://iopscience.iop.org/1009-0630/3/3/007>.
- [95] Zheng-Ying Cui, Yuan Huang, Ping Sun, Yong-Zhen Zheng, Pei-Lan Si, Jie Lu, Bing-Zhong Fu, Peng Zhang, Yu-Dong Pan, Yun-Bo Dong, Wei Deng, Qing-Wei Yang, and Xuan-Tong Ding. “Investigation of Impurity Ion Transport with Laser Blow-off in HL-2A Tokamak”. In: *Chinese Physics Letters* 23.8 (2006), pages 2143–2146. DOI: [10.1088/0256-307X/23/8/050](https://doi.org/10.1088/0256-307X/23/8/050). URL: <http://iopscience.iop.org/0256-307X/23/8/050>.
- [96] Wei Chen, Zheng-Ying Cui, Ping Sun, Yuan Huang, Peng Zhang, Wei Deng, Yu-Dong Pan, Zhong-Bing Shi, Yan Zhou, Yong-Zhen Zheng, and Qing-Wei Yang. “Preliminary analysis of impurity transport in HL-2A ohmic discharges”. In: *Chinese Physics* 16.11 (2007), pages 3451–3457. DOI: [10.1088/1009-1963/16/11/050](https://doi.org/10.1088/1009-1963/16/11/050). URL: <http://iopscience.iop.org/1009-1963/16/11/050>.

- [97] Xue-Wu Cui, Zheng-Ying Cui, Bei-Bin Feng, Yu-Dong Pan, Hang-Yu Zhou, Ping Sun, Bing-Zhong Fu, Ping Lu, Yun-Bo Dong, Jin-Ming Gao, Shao-Dong Song, and Qing-Wei Yang. "Investigation of impurity transport using supersonic molecular beam injected neon in HL-2A ECRH plasma". In: *Chinese Physics B* 22.12 (2013), 125201. DOI: [10.1088/1674-1056/22/12/125201](https://doi.org/10.1088/1674-1056/22/12/125201). URL: <http://iopscience.iop.org/1674-1056/22/12/125201>.
- [98] Z. Y. Cui, S. Morita, H. Y. Zhou, X. T. Ding, P. Sun, M. Kobayashi, X. W. Cui, Y. Xu, X. L. Huang, Z. B. Shi, J. Cheng, Y. G. Li, B. B. Feng, S. D. Song, L. W. Yan, Q. W. Yang, and X. R. Duan. "Enhancement of edge impurity transport with ECRH in the HL-2A tokamak". In: *Nuclear Fusion* 53.9 (2013), 093001. DOI: [10.1088/0029-5515/53/9/093001](https://doi.org/10.1088/0029-5515/53/9/093001). URL: <http://iopscience.iop.org/0029-5515/53/9/093001>.
- [99] Rong Huang, Ji-kang Xie, Lin-zhong Li, Ye-xi He, Shu-ya Wang, Chuan-bao Deng, Guo-xiang Li, Le-han Wei, and Li-jian Qiu. "The impurity transport in HT-6B tokamak". In: *Acta Physica Sinica (Overseas Edition)* 2.1 (Jan. 1993), pages 22-34. DOI: [10.1088/1004-423X/2/1/004](https://doi.org/10.1088/1004-423X/2/1/004). URL: <http://iopscience.iop.org/1004-423X/2/1/004>.
- [100] Qian Zhou, Bao-Nian Wan, Zhen-Wei Wu, and Juan Huang. "The carbon impurity particle transport in ohmic discharges on the HT-7 tokamak". In: *Chinese Physics* 14.12 (Dec. 2005), pages 2539-2545. DOI: [10.1088/1009-1963/14/12/027](https://doi.org/10.1088/1009-1963/14/12/027). URL: <http://iopscience.iop.org/1009-1963/14/12/027>.
- [101] K. H. Burrell, S. K. Wong, C. H. Muller III, M. P. Hacker, H. E. Ketterer, R. C. Isler, and E. A. Lazarus. "Observation of long impurity confinement times in the ISX tokamak". In: *Nuclear Fusion* 21.8 (1981), pages 1009-1014. DOI: [10.1088/0029-5515/21/8/010](https://doi.org/10.1088/0029-5515/21/8/010). URL: <http://iopscience.iop.org/0029-5515/21/8/010>.
- [102] K. Lackner, K. Behringer, W. Engelhardt, and R. Wunderlich. "An Algorithm for the Solution of Impurity Diffusion under Finite Reaction Rates". In: *Zeitschrift für Naturforschung A* 37.8 (1982), pages 931-938. DOI: [10.1515/zna-1982-0827](https://doi.org/10.1515/zna-1982-0827). URL: <http://www.degruyter.com/view/j/zna.1982.37.issue-8/zna-1982-0827/zna-1982-0827.xml>.
- [103] D. Pasini, M. Mattioli, A. W. Edwards, R. Giannella, R. D. Gill, N. C. Hawkes, G. Magyar, B. Saoutic, Z. Wang, and D. Zasche. "Impurity transport in JET using laser injected impurities in Ohmic and radiofrequency heated plasmas". In: *Nuclear Fusion* 30.10 (1990), pages 2049-2062. DOI: [10.1088/0029-5515/30/10/007](https://doi.org/10.1088/0029-5515/30/10/007). URL: <http://iopscience.iop.org/0029-5515/30/10/007>.



- [104] R. Giannella, N. C. Hawkes, L. Lauro Taroni, M. Mattioli, J. O'Rourke, and D. Pasini. "Comparison of impurity and electron particle transport in JET discharges". In: *Plasma Physics and Controlled Fusion* 34.5 (1992), pages 687–697. DOI: [10.1088/0741-3335/34/5/003](https://doi.org/10.1088/0741-3335/34/5/003). URL: <http://iopscience.iop.org/0741-3335/34/5/003>.
- [105] D. Pasini, R. Giannella, L. Lauro Taroni, M. Mattioli, B. Denne-Hinnov, N. Hawkes, G. Magyar, and H. Weisen. "Measurements of impurity transport in JET". In: *Plasma Physics and Controlled Fusion* 34.5 (1992), pages 677–685. DOI: [10.1088/0741-3335/34/5/002](https://doi.org/10.1088/0741-3335/34/5/002). URL: <http://iopscience.iop.org/0741-3335/34/5/002>.
- [106] R. Giannella, L. Lauro-Taroni, M. Mattioli, B. Alper, B. Denne-Hinnov, G. Magyar, J. O'Rourke, and D. Pasini. "Role of current profile in impurity transport in JET L mode discharges". In: *Nuclear Fusion* 34.9 (1994), pages 1185–1202. DOI: [10.1088/0029-5515/34/9/I01](https://doi.org/10.1088/0029-5515/34/9/I01). URL: <http://iopscience.iop.org/0029-5515/34/9/I01>.
- [107] L. Lauro-Taroni, B. Alper, R. Giannella, K. Lawson, F. Marcus, M. Mattioli, P. Smeulders, and M. von Hellermann. "Impurity Transport of high performance discharges in JET". In: *21<sup>st</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (Montpellier, France, June 27–July 1, 1994). Volume 18B. European Conference Abstracts. European Physical Society, 1994, page 102.
- [108] H. Chen, I. Coffey, R. Giannella, N. C. Hawkes, L. C. Ingesson, L. Lauro-Taroni, K. Lawson, M. G. O'Mullane, M. Romanelli, N. J. Peacock, M. von Hellermann, and K.-D. Zastrow. "Impurity particle transport with radial electric field and plasma rotation in JET optimised shear plasmas". In: *25<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (Praha, Czech Republic, June 29–July 3, 1998). Edited by P. Pavlo. Volume 22C. European Conference Abstracts. European Physical Society, 1998, p3.175. URL: [http://epsppd.epfl.ch/Praha/WEB/98ICPP\\_W/G077PR.PDF](http://epsppd.epfl.ch/Praha/WEB/98ICPP_W/G077PR.PDF).
- [109] L. C. Ingesson, B. Alper, H. Chen, A. W. Edwards, G. C. Fehmers, J. C. Fuchs, R. Giannella, R. D. Gill, L. Lauro-Taroni, and M. Romanelli. "Soft X ray tomography during ELMS and impurity injection in JET". In: *Nuclear Fusion* 38.11 (1998), pages 1675–1694. DOI: [10.1088/0029-5515/38/11/307](https://doi.org/10.1088/0029-5515/38/11/307). URL: <http://iopscience.iop.org/0029-5515/38/11/307>.
- [110] K. D. Lawson, B. Alper, I. H. Coffey, S. K. Erents, M. von Hellermann, L. D. Horton, L. C. Ingesson, H. J. F. Jäckel, L. Lauro-Taroni, G. McCracken, M. G.



- O'Mullane, M. Romanelli, P. R. Thomas, and K.-D. Zastrow. "Impurity accumulation in JET ELMY H-mode discharges". In: *25<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (Praha, Czech Republic, June 29–July 3, 1998). Edited by P. Pavlo. Volume 22c. European Conference Abstracts. European Physical Society, 1998, P1.018. URL: [http://epsppd.epfl.ch/Praha/WEB/98ICPP\\_W/B164PR.PDF](http://epsppd.epfl.ch/Praha/WEB/98ICPP_W/B164PR.PDF).
- [111] K. D. Lawson, L. Lauro-Taroni, R. Giannella, N. A. C. Gottardi, N. C. Hawkes, P. D. Morgan, M. G. O'Mullane, N. J. Peacock, and P. Smeulders. *Impurity Transport in JET L-mode Discharges*. Technical report JET-P(99)03. Abingdon, Oxfordshire, OX14 3EA, UK: JET Joint Undertaking, Mar. 1999. URL: <http://www.euro-fusionscipub.org/archives/jet-archive/impurity-transport-in-jet-l-mode-discharges>.
- [112] JET Team (prepared by K.-D. Zastrow). "Trace tritium transport in JET". In: *Nuclear Fusion* 39.11Y (1999), pages 1891–1896. DOI: [10.1088/0029-5515/39/11Y/331](https://doi.org/10.1088/0029-5515/39/11Y/331). URL: <http://iopscience.iop.org/0029-5515/39/11Y/331>.
- [113] H. Chen, R. Giannella, N. C. Hawkes, L. Lauro-Taroni, N. J. Peacock, and M. von Hellermann. "Study of impurity behaviour during JET radiative boundary experiments". In: *Plasma Physics and Controlled Fusion* 43.1 (2001), pages 1–12. DOI: [10.1088/0741-3335/43/1/301](https://doi.org/10.1088/0741-3335/43/1/301). URL: <http://iopscience.iop.org/0741-3335/43/1/301>.
- [114] H. Chen, N. C. Hawkes, L. C. Ingesson, N. J. Peacock, and M. G. Haines. "Impurity transport with strong and weak internal thermal barriers in JET optimized shear plasmas". In: *Nuclear Fusion* 41.1 (2001), pages 31–46. DOI: [10.1088/0029-5515/41/1/303](https://doi.org/10.1088/0029-5515/41/1/303). URL: <http://iopscience.iop.org/0029-5515/41/1/303>.
- [115] M. Mattioli, K. B. Fournier, L. Carraro, I. Coffey, C. Giroud, K. Lawson, P. Monier-Garbet, M. O'Mullane, J. Ongena, M. E. Puiatti, F. Sattin, P. Scarin, and M. Valisa. "Experimental and simulated argon spectra in the 2.3–3.4 nm region from tokamak plasmas". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 34.2 (2001), pages 127–142. DOI: [10.1088/0953-4075/34/2/301](https://doi.org/10.1088/0953-4075/34/2/301). URL: <http://iopscience.iop.org/0953-4075/34/2/301>.
- [116] M. Mattioli, K. B. Fournier, M. E. Puiatti, M. Valisa, L. Carraro, I. Coffey, M. O'Mullane, F. Sattin, and P. Scarin. "Experimental and simulated vuv spectra from the JET tokamak and the reversed field pinch RFX". In: *Plasma Physics and Controlled Fusion* 44.1 (2002), pages 33–50. DOI: [10.1088/0741-3335/44/1/304](https://doi.org/10.1088/0741-3335/44/1/304). URL: <http://iopscience.iop.org/0741-3335/44/1/304>.

- [117] M. Mattioli, M. E. Puiatti, M. Valisa, I. Coffey, R. Dux, P. Monier-Garbet, M. F. F. Nave, J. Ongena, M. Stamp, J. Strachan, M. von Hellermann, and contributors to the EFDA-JET workprogramme. “Simulation of the time behavior of impurities in Jet Ar-seeded discharges and its relation with sawteething”. In: *29<sup>th</sup> European Physical Society Conference on Plasma Physics and Controlled Fusion*. (Montreux, Switzerland, June 17–21, 2002). Edited by R. Behn and C. Varandas. Volume 26B. European Conference Abstracts. European Physical Society, 2002, P–2.042. URL: [http://crpppc42.epfl.ch/Montreux/pdf/P2\\_042.pdf](http://crpppc42.epfl.ch/Montreux/pdf/P2_042.pdf).
- [118] M. E. Puiatti, M. Mattioli, G. Telesca, M. Valisa, I. Coffey, P. Dumortier, C. Giroud, L. C. Ingesson, K. D. Lawson, G. Maddison, A. M. Messiaen, P. Monier-Garbet, A. Murari, M. F. F. Nave, J. Ongena, J. Rapp, J. Strachan, B. Unterberg, M. von Hellermann, and contributors to the EFDA-JET Workprogramme. “Radiation pattern and impurity transport in argon seeded ELMY H-mode discharges in JET”. In: *Plasma Physics and Controlled Fusion* 44.9 (2002), pages 1863–1878. DOI: [10.1088/0741-3335/44/9/305](https://doi.org/10.1088/0741-3335/44/9/305). URL: <http://iopscience.iop.org/0741-3335/44/9/305>.
- [119] M. E. Puiatti, M. Valisa, M. Mattioli, T. Bolzonella, A. Bortolon, I. Coffey, R. Dux, M. von Hellermann, P. Monier-Garbet, M. F. F. Nave, J. Ongena, and contributors to the EFDA-JET Workprogramme. “Simulation of the time behaviour of impurities in JET Ar-seeded discharges and its relation with sawtoothing and RF heating”. In: *Plasma Physics and Controlled Fusion* 45.12 (2003), pages 2011–2024. DOI: [10.1088/0741-3335/45/12/003](https://doi.org/10.1088/0741-3335/45/12/003). URL: <http://iopscience.iop.org/0741-3335/45/12/003>.
- [120] R. Dux, C. Giroud, K.-D. Zastrow, and JET EFDA contributors. “Impurity transport in internal transport barrier discharges on JET”. In: *Nuclear Fusion* 44.2 (2004), pages 260–264. DOI: [10.1088/0029-5515/44/2/006](https://doi.org/10.1088/0029-5515/44/2/006). URL: <http://iopscience.iop.org/0029-5515/44/2/006>.
- [121] C. Giroud, P. Belo, R. Barnsley, I. Coffey, R. Dux, M. von Hellermann, E. Joffrin, C. Jupen, A. Meigs, M. O’Mullane, V. Pericoli Ridolfini, A. C. C. Sips, A. Whiteford, K.-D. Zastrow, and JET EFDA contributors. “Z-dependence of impurity transport in the Hybrid scenario at JET”. In: *31<sup>st</sup> European Physical Society Conference on Plasma Physics*. (London, UK, June 28–July 2, 2004). Edited by P. Norreys and H. Hutchinson. Volume 28G. European Conference Abstracts. European Physical Society, 2004, P–1.144. URL: [http://epsppd.epfl.ch/London/pdf/P1\\_144.pdf](http://epsppd.epfl.ch/London/pdf/P1_144.pdf).

- [122] A. D. Whiteford, K.-D. Zastrow, M. Adams, L. Bertalot, S. Conroy, M. G. O'Mullane, S. Popovichev, H. P. Summers, A. Zabolotsky, and JET EFDA contributors. "Quantitative forward modelling of neutron emission to derive transport coefficients of tritium in JET, including error propagation through to transport parameters". In: *31<sup>st</sup> European Physical Society Conference on Plasma Physics*. (London, UK, June 28–July 2, 2004). Edited by P. Norreys and H. Hutchinson. Volume 28G. European Conference Abstracts. European Physical Society, 2004, P–1.159. URL: [http://epsppd.epfl.ch/London/pdf/P1\\_159.pdf](http://epsppd.epfl.ch/London/pdf/P1_159.pdf).
- [123] Allan D. Whiteford. "On the spectral emission of impurity species for diagnostic application to magnetically confined fusion plasmas". PhD thesis. University of Strathclyde, Sept. 2004. URL: [http://www.adas.ac.uk/theses/whiteford\\_thesis.pdf](http://www.adas.ac.uk/theses/whiteford_thesis.pdf).
- [124] K.-D. Zastrow, J. M. Adams, Yu. Baranov, P. Belo, L. Bertalot, J. H. Brzozowski, C. D. Challis, S. Conroy, M. de Baar, P. de Vries, P. Dumortier, J. Ferreira, L. Garzotti, T. C. Hender, E. Joffrin, V. Kiptily, J. Mailloux, D. C. McDonald, R. Neu, M. O'Mullane, M. F. F. Nave, J. Ongena, S. Popovichev, M. Stamp, J. Stober, D. Stork, I. Voitsekhovitch, M. Valovič, H. Weisen, A. D. Whiteford, A. Zabolotsky, and JET EFDA Contributors. "Tritium transport experiments on the JET tokamak". In: *Plasma Physics and Controlled Fusion* 46.12B (2004), B255–B265. DOI: [10.1088/0741-3335/46/12B/022](https://doi.org/10.1088/0741-3335/46/12B/022). URL: <http://iopscience.iop.org/0741-3335/46/12B/022>.
- [125] D. Stork, Yu. Baranov, P. Belo, L. Bertalot, D. Borba, J. H. Brzozowski, C. D. Challis, D. Ciric, S. Conroy, M. de Baar, P. de Vries, P. Dumortier, L. Garzotti, N. C. Hawkes, T. C. Hender, E. Joffrin, T. T. C. Jones, V. Kiptily, P. Lamalle, J. Mailloux, M. Mantsinen, D. C. McDonald, M. F. F. Nave, R. Neu, M. O'Mullane, J. Ongena, R. J. Pearce, S. Popovichev, S. E. Sharapov, M. Stamp, J. Stober, E. Surrey, M. Valovic, I. Voitsekhovitch, H. Weisen, A. D. Whiteford, L. Worth, V. Yavorskij, K.-D. Zastrow, and JET EFDA contributors. "Overview of transport, fast particle and heating and current drive physics using tritium in JET plasmas". In: *Nuclear Fusion* 45.10 (2005), S181–S194. DOI: [10.1088/0029-5515/45/10/S15](https://doi.org/10.1088/0029-5515/45/10/S15). URL: <http://iopscience.iop.org/0029-5515/45/10/S15>.
- [126] C. Giroud, C. Angioni, G. Bonheure, I. Coffey, N. Dubuit, X. Garbet, R. Guirlet, P. Mantica, V. Naulin, M. E. Puiatti, M. Valisa, A. D. Whiteford, K.-D. Zastrow, M. N. A. Beurskens, M. Brix, E. de la Luna, K. Lawson, L. Lauro-Taroni, A. Meigs, M. O'Mullane, T. Parisot, C. Perez Von Thun, and

- O. Zimmermann. “Progress in understanding anomalous impurity transport at JET”. In: *21<sup>st</sup> International Conference on Fusion Energy*. (Chengdu, China, Oct. 16–21, 2006). Volume IAEA-CN-149. International Atomic Energy Agency. 2006, EX/8–3. URL: <http://www-naweb.iaea.org/napc/physics/FEC/FEC2006/html/node407.htm#81216>.
- [127] M. E. Puiatti, M. Valisa, C. Angioni, L. Garzotti, P. Mantica, M. Mattioli, L. Carraro, I. Coffey, C. Sozzi, and JET-EFDA contributors. “Analysis of metallic impurity density profiles in low collisionality Joint European Torus H-mode and L-mode plasmas”. In: *Physics of Plasmas* 13.4 (2006), 042501. DOI: [10.1063/1.2187424](https://doi.org/10.1063/1.2187424). URL: <http://scitation.aip.org/content/aip/journal/pop/13/4/10.1063/1.2187424>.
- [128] L. Carraro, C. Angioni, C. Giroud, M. E. Puiatti, M. Valisa, P. Buratti, R. Buttery, I. Coffey, L. Garzotti, D. Van Eester, L. Lauro Taroni, K. Lawson, E. Lerche, P. Mantica, M. Mattioli, V. Naulin, and JET-EFDA contributors. “Impurity profile control in JET plasmas with radio-frequency power injection”. In: *34<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Warsaw, Poland, July 2–6, 2007). Edited by Paweł Gašior and Jerzy Wołowski. Volume 31F. European Conference Abstracts. European Physical Society, 2008, 0–4.028. URL: [http://epsppd.epfl.ch/Warsaw/pdf/04\\_028.pdf](http://epsppd.epfl.ch/Warsaw/pdf/04_028.pdf).
- [129] C. Giroud, C. Angioni, L. Carraro, I. H. Coffey, J. Hobirk, M. E. Puiatti, M. Valisa, A. D. Whiteford, P. Belo, T. M. Biewer, M. Brix, R. Buttery, E. Joffrin, L. Lauro Taroni, K. Lawson, P. Mantica, A. Meigs, V. Naulin, M. G. O’Mullane, K.-D. Zastrow, and the JET-EFDA Contributors. “Study of Z-dependence of impurity transport at JET”. In: *34<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Warsaw, Poland, July 2–6, 2007). Edited by Paweł Gašior and Jerzy Wołowski. Volume 31F. European Conference Abstracts. European Physical Society, 2007, P–2.049. URL: [http://epsppd.epfl.ch/Warsaw/pdf/P2\\_049.pdf](http://epsppd.epfl.ch/Warsaw/pdf/P2_049.pdf).
- [130] C. Giroud, R. Barnsley, P. Buratti, I. H. Coffey, M. von Hellermann, C. Jupén, K. D. Lawson, A. Meigs, M. O’Mullane, A. D. Whiteford, K.-D. Zastrow, and the JET EFDA contributors. “Method for experimental determination of Z dependence of impurity transport on JET”. In: *Nuclear Fusion* 47.4 (2007), pages 313–330. DOI: [10.1088/0029-5515/47/4/010](https://doi.org/10.1088/0029-5515/47/4/010). URL: <http://iopscience.iop.org/0029-5515/47/4/010>.
- [131] P. Mantica, G. Corrigan, X. Garbet, F. Imbeaux, J. Lonroth, V. Parail, T. Tala, A. Taroni, M. Valisa, and H. Weisen. “Chapter 10: Core Transport Studies

- in JET”. In: *Fusion Science and Technology* 53.4 (May 2008), pages 1152–1216. URL: [http://www.ans.org/pubs/journals/fst/a\\_1750](http://www.ans.org/pubs/journals/fst/a_1750).
- [132] H. Nordman, A. Skyman, P. Strand, C. Giroud, F. Jenko, F. Merz, V. Naulin, T. Tala, and the JET-EFDA Contributors. “Fluid and gyrokinetic simulations of impurity transport at JET”. In: *Plasma Physics and Controlled Fusion* 53.10 (2011), 105005. DOI: [10.1088/0741-3335/53/10/105005](https://doi.org/10.1088/0741-3335/53/10/105005). URL: <http://iopscience.iop.org/0741-3335/53/10/105005>.
- [133] M. Valisa, L. Carraro, I. Predebon, M. E. Puiatti, C. Angioni, I. Coffey, C. Giroud, L. Lauro Taroni, B. Alper, M. Baruzzo, P. Belo daSilva, P. Buratti, L. Garzotti, D. Van Eester, E. Lerche, P. Mantica, V. Naulin, T. Tala, M. Tsalas, and JET EFDA contributors. “Metal impurity transport control in JET H-mode plasmas with central ion cyclotron radiofrequency power injection”. In: *Nuclear Fusion* 51.3 (2011), 033002. DOI: [10.1088/0029-5515/51/3/033002](https://doi.org/10.1088/0029-5515/51/3/033002). URL: <http://iopscience.iop.org/0029-5515/51/3/033002>.
- [134] C. Angioni, P. Mantica, T. Pütterich, M. Valisa, M. Baruzzo, E. A. Belli, P. Belo, F. J. Casson, C. Challis, P. Drewelow, C. Giroud, N. Hawkes, T. C. Hender, J. Hobirk, T. Koskela, L. Lauro Taroni, C. F. Maggi, J. Mlynar, T. Odstrcil, M. L. Reinke, M. Romanelli, and JET EFDA Contributors. “Tungsten transport in JET H-mode plasmas in hybrid scenario, experimental observations and modelling”. In: *Nuclear Fusion* 54.8 (2014), 083028. DOI: [10.1088/0029-5515/54/8/083028](https://doi.org/10.1088/0029-5515/54/8/083028). URL: <http://iopscience.iop.org/0029-5515/54/8/083028>.
- [135] Masayasu Sato, Tsuneo Amano, Kuninori Sato, and Kenro Miyamoto. “Impurity Behavior in JIPP T-II Tokamak Plasma”. In: *Journal of the Physical Society of Japan* 50.6 (June 1981), pages 2114–2121. DOI: [10.1143/JPSJ.50.2114](https://doi.org/10.1143/JPSJ.50.2114). URL: <http://journals.jps.jp/doi/abs/10.1143/JPSJ.50.2114>.
- [136] K. Ida, T. Amano, K. Kawahata, O. Kaneko, and H. Tawara. “ $Z_{\text{eff}}$  measurements and low- $Z$  impurity transport for NBI and ICRF heated plasmas in JIPP T-IIU”. In: *Nuclear Fusion* 30.4 (1990), pages 665–674. DOI: [10.1088/0029-5515/30/4/008](https://doi.org/10.1088/0029-5515/30/4/008). URL: <http://iopscience.iop.org/0029-5515/30/4/008>.
- [137] Y. Koide, T. Hirayama, T. Sugie, A. Sakasai, H. Kubo, N. Akaoka, T. Nishitani, K. Nagashima, H. Shirai, H. Takeuchi, and JT-60 Team. “Study of impurity transport in ohmically and neutral beam heated divertor discharges in JT-60”. In: *Nuclear Fusion* 28.10 (1988), pages 1835–1844. DOI: [10.1088/0029-5515/28/10/011](https://doi.org/10.1088/0029-5515/28/10/011). URL: <http://iopscience.iop.org/0029-5515/28/10/011>.

- [138] A. Sakasai, Y. Koide, H. Kubo, T. Sugie, M. Shimada, T. Hirayama, N. Asakura, Y. Kawano, N. Hosogane, and H. Nakamura. “Impurity profiles and transport characteristics in JT-60U”. In: *Journal of Nuclear Materials* 196–198 (Dec. 1992), pages 472–475. DOI: [10.1016/S0022-3115\(06\)80081-4](https://doi.org/10.1016/S0022-3115(06)80081-4). URL: <http://www.sciencedirect.com/science/article/pii/S0022311506800814>.
- [139] H. Takenaga, K. Nagashima, A. Sakasai, T. Oikawa, and T. Fujita. “Determination of particle transport coefficients in reversed shear plasma of JT-60U”. In: *Plasma Physics and Controlled Fusion* 40.2 (1998), pages 183–190. DOI: [10.1088/0741-3335/40/2/002](https://doi.org/10.1088/0741-3335/40/2/002). URL: <http://iopscience.iop.org/0741-3335/40/2/002>.
- [140] H. Takenaga, K. Nagashima, A. Sakasai, N. Asakura, K. Shimizu, H. Kubo, S. Higashijima, T. Oikawa, T. Fujita, Y. Kamada, N. Hosogane, M. Shimada, and the JT-60 Team. “Particle confinement and transport in JT-60U”. In: *17<sup>th</sup> International Conference on Fusion Energy*. (Yokohama, Japan, Oct. 19–24, 1998). Volume IAEA-CN-69. International Atomic Energy Agency. 1998, EXP1/17. URL: [http://www-pub.iaea.org/mtcd/publications/pdf/csp\\_008c/fec1998/html/node139.htm](http://www-pub.iaea.org/mtcd/publications/pdf/csp_008c/fec1998/html/node139.htm).
- [141] Hidenobu Takenaga, Akira Sakasai, Yoshihiko Koide, Yoshiteru Sakamoto, Hirotaka Kubo, Satoru Higashijima, Toshihiro Oikawa, Hiroshi Shirai, Takaaki Fujita, and Yutaka Kamada. “Impurity Transport in Reversed Shear and ELMY H-Mode Plasmas of JT-60U”. In: *Journal of Plasma and Fusion Research* 75.8 (Aug. 1999), pages 952–966. DOI: [10.1585/jspf.75.952](https://doi.org/10.1585/jspf.75.952). URL: [https://www.jstage.jst.go.jp/article/jspf/75/8/75\\_8\\_952/\\_article](https://www.jstage.jst.go.jp/article/jspf/75/8/75_8_952/_article).
- [142] H. Takenaga, K. Nagashima, A. Sakasai, N. Asakura, K. Shimizu, H. Kubo, S. Higashijima, T. Oikawa, H. Shirai, T. Fujita, Y. Kamada, N. Hosogane, M. Shimada, and JT-60 Team. “Particle confinement and transport in JT-60U”. In: *Nuclear Fusion* 39.11Y (1999), pages 1917–1928. DOI: [10.1088/0029-5515/39/11Y/335](https://doi.org/10.1088/0029-5515/39/11Y/335). URL: <http://iopscience.iop.org/0029-5515/39/11Y/335>.
- [143] S. Sakurai, H. Kubo, A. Takenaga, N. Asakura, H. Tamai, T. Ishijima, S. Konoshima, K. Itami, A. Sakasai, S. Higashijima, T. Sugie, and JT-60 Team. “Impurity behavior in high performance radiative discharges of JT-60U”. In: *Journal of Nuclear Materials* 290–293 (Mar. 2001), pages 1002–1008. DOI: [10.1016/S0022-3115\(00\)00475-X](https://doi.org/10.1016/S0022-3115(00)00475-X). URL: <http://www.sciencedirect.com/science/article/pii/S002231150000475X>.

- [144] H. Takenaga and JT-60 Team. “Improved particle control for high integrated plasma performance in Japan Atomic Energy Research Institute Tokamak-60 Upgrade”. In: *Physics of Plasmas* 8.5 (May 2001), pages 2217–2223. DOI: [10.1063/1.1357827](https://doi.org/10.1063/1.1357827). URL: <http://scitation.aip.org/content/aip/journal/pop/8/5/10.1063/1.1357827>.
- [145] Joohwan Hong, Seung Hun Lee, Juhung Kim, C. R. Seon, S. G. Lee, G. Y. Park, K. D. Lee, S. S. Henderson, H. Y. Lee, Jae Sun Park, Juhyeok Jang, Siwon Jang, Taemin Jeon, M. O’Mullane, and Wonho Choe. “Control of core argon impurity profile by ECH in KSTAR L-mode plasmas”. In: *Nuclear Fusion* 55.6 (2015), 063016. DOI: [10.1088/0029-5515/55/6/063016](https://doi.org/10.1088/0029-5515/55/6/063016). URL: <http://iopscience.iop.org/0029-5515/55/6/063016>.
- [146] I. Lehane, G. Turri, R. J. Akers, R. J. Buttery, P. G. Carolan, S. Cowley, M. P. Gryaznevich, H. Meyer, M. G. O’Mullane, A. Patel, and the MAST Team. “Core Instabilities and Impurity Transport in MAST”. In: *30<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (St. Petersburg, Russia, July 7–11, 2003). Edited by R. Koch and S. Lebedev. Volume 27A. European Conference Abstracts. European Physical Society, 2003, P–3.92. URL: [http://epsppd.epfl.ch/StPetersburg/PDF/P3\\_092.PDF](http://epsppd.epfl.ch/StPetersburg/PDF/P3_092.PDF).
- [147] S. S. Henderson, L. Garzotti, F. J. Casson, D. Dickinson, M. O’Mullane, A. Patel, C. M. Roach, H. P. Summers, M. Valovič, and the MAST Team. “Low-Z perturbative impurity transport and microstability analysis on MAST”. In: *40<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Espoo, Finland, July 1–5, 2013). Edited by V. Naulin, C. Angioni, M. Borghesi, S. Ratynskaia, S. Poedts, T. Donné, T. Kurki-Suonio, S. Äkäslompolo, A. Hakola, and M. Airila. Volume 37D. European Conference Abstracts. European Physical Society, 2013, P4.146. URL: <http://ocs.ciemat.es/EPS2013PAP/pdf/P4.146.pdf>.
- [148] S. S. Henderson, L. Garzotti, F. J. Casson, D. Dickinson, M. F. J. Fox, M. O’Mullane, A. Patel, C. M. Roach, H. P. Summers, M. Valovič, and The MAST Team. “Neoclassical and gyrokinetic analysis of time-dependent helium transport experiments on MAST”. In: *Nuclear Fusion* 54.9 (2014), 093013. DOI: [10.1088/0029-5515/54/9/093013](https://doi.org/10.1088/0029-5515/54/9/093013). URL: <http://iopscience.iop.org/0029-5515/54/9/093013>.
- [149] Stuart Henderson. “Impurity Transport Studies on MAST”. PhD thesis. University of Strathclyde, July 2014. URL: <http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.635519>.



- [150] S. S. Henderson, L. Garzotti, F. J. Casson, D. Dickinson, M. O'Mullane, A. Patel, C. M. Roach, H. P. Summers, H. Tanabe, M. Valovič, and the MAST team. "Charge dependence of neoclassical and turbulent transport of light impurities on MAST". In: *Plasma Physics and Controlled Fusion* 57.9 (2015), 095001. DOI: [10.1088/0741-3335/57/9/095001](https://doi.org/10.1088/0741-3335/57/9/095001). URL: <http://iopscience.iop.org/0741-3335/57/9/095001>.
- [151] D. Stutman, M. Finkenthal, C. Bourdelle, E. Synakowski, J. Menard, D. Darrow, B. Leblanc, R. Bell, V. Soukhanovskii, M. Gilmore, and S. Kaye. "Experimental Evidence of Low Impurity Transport in NSTX". In: *29<sup>th</sup> European Physical Society Conference on Plasma Physics and Controlled Fusion*. (Montreux, Switzerland, June 17–21, 2002). Edited by R. Behn and C. Varandas. Volume 26B. European Conference Abstracts. European Physical Society, 2002, P-4.067. URL: [http://epsppd.epfl.ch/Montreux/pdf/P4\\_067.pdf](http://epsppd.epfl.ch/Montreux/pdf/P4_067.pdf).
- [152] D. Stutman, M. Finkenthal, M. G. Bell, R. E. Bell, C. Bourdelle, S. M. Kaye, B. P. LeBlanc, J. E. Menard, V. Soukhanovskii, E. J. Synakowski, and the NSTX Team. "Correlation between impurity transport and toroidal rotation in NSTX". In: *30<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (St. Petersburg, Russia, July 7–11, 2003). Edited by R. Koch and S. Lebedev. Volume 27A. European Conference Abstracts. European Physical Society, 2003, P-3.100. URL: [http://epsppd.epfl.ch/StPetersburg/PDF/P3\\_100.PDF](http://epsppd.epfl.ch/StPetersburg/PDF/P3_100.PDF).
- [153] D. Stutman, M. Finkenthal, R. E. Bell, S. M. Kaye, B. P. LeBlanc, J. E. Menard, E. J. Synakowski, D. S. Darrow, V. Soukhanovskii, C. Bourdelle, and The NSTX Team. "Impurity transport measurements in beam heated low-confinement mode discharges in the National Spherical Torus Experiment". In: *Physics of Plasmas* 10.11 (Nov. 2003), pages 4387–4395. DOI: [10.1063/1.1610473](https://doi.org/10.1063/1.1610473). URL: <http://scitation.aip.org/content/aip/journal/pop/10/11/10.1063/1.1610473>.
- [154] D. Stutman, M. Finkenthal, H. W. Moos, K. B. Fournier, R. Kaita, D. Johnson, and L. Roquemore. "Integrated impurity diagnostic package for magnetic fusion experiments". In: *Review of Scientific Instruments* 74.3 (Mar. 2003), pages 1982–1987. DOI: [10.1063/1.1538327](https://doi.org/10.1063/1.1538327). URL: <http://scitation.aip.org/content/aip/journal/rsi/74/3/10.1063/1.1538327>.
- [155] Luis F. Delgado-Aparicio. "Novel soft x-ray techniques for the study of particle transport phenomena in magnetically confined fusion plasmas". PhD thesis. Baltimore, MD: Johns Hopkins University, Oct. 2007. URL: <http://search.proquest.com/docview/304615267>.



- [156] L. Delgado-Aparicio, D. Stutman, K. Tritz, M. Finkenthal, S. Kaye, R. Bell, R. Kaita, B. LeBlanc, F. Levinton, J. Menard, S. Paul, D. Smith, and H. Yuh. “Impurity transport studies in NSTX neutral beam heated H-mode plasmas”. In: *Nuclear Fusion* 49.8 (2009), 085028. DOI: [10.1088/0029-5515/49/8/085028](https://doi.org/10.1088/0029-5515/49/8/085028). URL: <http://iopscience.iop.org/0029-5515/49/8/085028>.
- [157] L. Delgado-Aparicio, D. Stutman, K. Tritz, F. Volpe, K. L. Wong, S. Gerhardt, E. Fredrickson, R. Bell, B. LeBlanc, S. Kaye, J. Menard, M. Finkenthal, S. Paul, and L. Roquemore. “Dependence of impurity transport on  $\rho^*$ ,  $v^*$  and MHD in NSTX”. In: *Proceedings of the 23<sup>rd</sup> IAEA Fusion Energy Conference*. (Daejeon, South Korea, Oct. 11–16, 2010). Volume IAEA-CN-180. International Atomic Energy Agency. 2010, EXC/P4-04. URL: <http://www-naweb.iaea.org/napc/physics/FEC/FEC2010/html/node306.htm>.
- [158] L. Delgado-Aparicio, D. Stutman, K. Tritz, F. Volpe, K. L. Wong, R. Bell, M. Finkenthal, E. Fredrickson, S. P. Gerhardt, S. Kaye, B. LeBlanc, J. Menard, S. Paul, and L. Roquemore. “Impurity transport experiments and effects on MHD in the National Spherical Torus Experiment (NSTX)”. In: *Nuclear Fusion* 51.8 (2011), 083047. DOI: [10.1088/0029-5515/51/8/083047](https://doi.org/10.1088/0029-5515/51/8/083047). URL: <http://iopscience.iop.org/0029-5515/51/8/083047>.
- [159] D. J. Clayton, K. Tritz, M. Finkenthal, D. Kumar, D. Stutman, R. E. Bell, and B. P. LeBlanc. “Edge transport measurements with the new multi-energy soft-x-ray diagnostic on NSTX”. In: *38<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Strasbourg, France, June 27–July 1, 2011). Edited by A. Becoulet, T. Hoang, and U. Stroth. Volume 35G. European Conference Abstracts. European Physical Society, 2011, P1.134. URL: <http://ocs.ciemat.es/EPS2011PAP/pdf/P1.134.pdf>.
- [160] D. J. Clayton, K. Tritz, D. Stutman, M. Finkenthal, S. M. Kaye, D. Kumar, B. P. LeBlanc, S. Paul, and S. A. Sabbagh. “Multi-energy soft-x-ray technique for impurity transport measurements in the fusion plasma edge”. In: *Plasma Physics and Controlled Fusion* 54.10 (2012), 105022. DOI: [10.1088/0741-3335/54/10/105022](https://doi.org/10.1088/0741-3335/54/10/105022). URL: <http://iopscience.iop.org/0741-3335/54/10/105022>.
- [161] F. Scotti, V. A. Soukhanovskii, R. E. Bell, S. Gerhardt, W. Guttenfelder, S. Kaye, R. Andre, A. Diallo, R. Kaita, B. P. LeBlanc, M. Podestá, and the NSTX Team. “Core transport of lithium and carbon in ELM-free discharges with lithium wall conditioning in NSTX”. In: *Nuclear Fusion* 53.8 (2013), 083001. DOI: [10.1088/0029-5515/53/8/083001](https://doi.org/10.1088/0029-5515/53/8/083001). URL: <http://iopscience.iop.org/0029-5515/53/8/083001>.

- [162] K. Ida, R. J. Fonck, R. A. Hulse, and B. Leblanc. “Some effects of MHD activity on impurity transport in the PBX Tokamak”. In: *Plasma Physics and Controlled Fusion* 28.6 (1986), pages 879–895. DOI: [10.1088/0741-3335/28/6/004](https://doi.org/10.1088/0741-3335/28/6/004). URL: <http://iopscience.iop.org/0741-3335/28/6/004>.
- [163] K. Ida, R. J. Fonck, S. Sesnic, R. A. Hulse, and B. LeBlanc. “Observation of Z-dependent impurity accumulation in the PBX tokamak”. In: *Physical Review Letters* 58.2 (Jan. 1987), pages 116–119. DOI: [10.1103/PhysRevLett.58.116](https://doi.org/10.1103/PhysRevLett.58.116). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.58.116>.
- [164] S. S. Sesnic, R. J. Fonck, K. Ida, K. Bol, P. Couture, G. Gammel, R. Kaita, S. Kaye, H. Kugel, B. LeBlanc, W. Morris, M. Okabayashi, S. Paul, E. T. Powell, M. Reusch, and H. Takahashi. “Observation of impurity accumulation and concurrent impurity influx in PBX”. In: *Journal of Nuclear Materials* 145–147 (Feb. 1987), pages 580–586. DOI: [10.1016/0022-3115\(87\)90403-X](https://doi.org/10.1016/0022-3115(87)90403-X). URL: <http://www.sciencedirect.com/science/article/pii/002231158790403X>.
- [165] R. J. Fonck, M. Finkenthal, R. J. Goldston, D. L. Herndon, R. A. Hulse, R. Kaita, and D. D. Meyerhofer. “Spatially Resolved Measurements of Fully Ionized Low-Z Impurities in the PDX Tokamak”. In: *Physical Review Letters* 49.10 (Sept. 1982), pages 737–740. DOI: [10.1103/PhysRevLett.49.737](https://doi.org/10.1103/PhysRevLett.49.737). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.49.737>.
- [166] R. J. Fonck and R. A. Hulse. “He<sup>++</sup> Transport in the PDX Tokamak”. In: *Physical Review Letters* 52.7 (Feb. 1984), pages 530–533. DOI: [10.1103/PhysRevLett.52.530](https://doi.org/10.1103/PhysRevLett.52.530). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.52.530>.
- [167] S. P. Regan, M. J. May, V. Soukhanovskii, M. Finkenthal, H. W. Moos, D. A. Diebold, M. Doczy, N. Hershkowitz, E. Y. Wang, and X. Wang. “Impurity transport measurements of biased H mode in hydrogen and deuterium plasmas on the Phaedrus-T tokamak”. In: *Nuclear Fusion* 37.5 (1997), pages 657–671. DOI: [10.1088/0029-5515/37/5/I08](https://doi.org/10.1088/0029-5515/37/5/I08). URL: <http://iopscience.iop.org/0029-5515/37/5/I08>.
- [168] K. Brau, S. von Goeler, M. Bitter, R. D. Cowan, D. Eames, K. Hill, N. Sauthoff, E. Silver, and W. Stodiek. “Observations of giant recombination edges on the Princeton Large Torus tokamak induced by particle transport”. In: *Physical Review A* 22.6 (Dec. 1980), pages 2769–2775. DOI: [10.1103/PhysRevA.22.2769](https://doi.org/10.1103/PhysRevA.22.2769).

- 22.2769. URL: <http://journals.aps.org/pr/abstract/10.1103/PhysRevA.22.2769>.
- [169] O. Demokan, F. Waelbroeck, and N. Demokan. "Iron transport in a confined high-temperature plasma". In: *Nuclear Fusion* 22.7 (1982), pages 921–934. DOI: [10.1088/0029-5515/22/7/005](https://doi.org/10.1088/0029-5515/22/7/005). URL: <http://iopscience.iop.org/0029-5515/22/7/005>.
- [170] B. C. Stratton, H. W. Moos, W. L. Hodge, S. Suckewer, J. C. Hosea, R. A. Hulse, D. Q. Hwang, and J. R. Wilson. "Changes in impurity radiation during ICRF heating of PLT tokamak plasmas". In: *Nuclear Fusion* 24.6 (1984), pages 767–777. DOI: [10.1088/0029-5515/24/6/009](https://doi.org/10.1088/0029-5515/24/6/009). URL: <http://iopscience.iop.org/0029-5515/24/6/009>.
- [171] S. Suckewer, A. Cavallo, S. Cohen, C. Daughney, B. Denne, E. Hinnov, J. Hosea, R. Hulse, D. Hwang, G. Schilling, B. Stratton, and R. Wilson. "Impurity ion transport studies on the PLT tokamak during neutral-beam injection". In: *Nuclear Fusion* 24.7 (1984), pages 815–826. DOI: [10.1088/0029-5515/24/7/001](https://doi.org/10.1088/0029-5515/24/7/001). URL: <http://iopscience.iop.org/0029-5515/24/7/001>.
- [172] A. Compant La Fontaine, M. A. Dubois, A. L. Pecquet, D. Boyd, A. Cavallo, S. Cohen, S. Von Goeler, R. Hulse, D. Manos, R. Smith, and R. Wilson. " $q = 1$  magnetohydrodynamic activity in PLT studied with aluminum impurity injection as a diagnostic tool". In: *Plasma Physics and Controlled Fusion* 27.3 (1985), pages 229–243. DOI: [10.1088/0741-3335/27/3/001](https://doi.org/10.1088/0741-3335/27/3/001). URL: <http://iopscience.iop.org/0741-3335/27/3/001>.
- [173] Alexei Zabolotski and Henri Weisen. *Analysis of impurity transport in TCV ohmic discharges*. Technical report INT 199/00. École Polytechnique Fédérale de Lausanne, July 2000. URL: <http://infoscience.epfl.ch/record/213218>.
- [174] E. Scavino, J. S. Bakos, H. Weisen, and A. Zabolotsky. "Impurity transport in shaped TCV plasmas". In: *28<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (Madeira, Portugal, June 18–22, 2001). Volume 25A. European Conference Abstracts. European Physical Society, 2001, pages 1425–1428. URL: <http://www.cfn.ist.utl.pt/EPS2001/CD/html/pdf/P4.025.pdf>.
- [175] E. Scavino, J. S. Bakos, R. Dux, H. Weisen, and TCV Team. "Effects of plasma shape on laser blow-off injected impurity transport in TCV". In: *Plasma Physics and Controlled Fusion* 45.11 (2003), pages 1961–1974. DOI: [10.1088/0741-3335/45/11/002](https://doi.org/10.1088/0741-3335/45/11/002). URL: <http://iopscience.iop.org/0741-3335/45/11/002>.

- [176] Edgar Scavino. “Transport of Laser-Ablated Impurities in TCV”. PhD thesis. Ecole Polytechnique Fédérale de Lausanne, Apr. 2003. URL: [http://infoscience.epfl.ch/record/121317/files/lrp\\_758\\_03\\_hq.pdf](http://infoscience.epfl.ch/record/121317/files/lrp_758_03_hq.pdf).
- [177] V. Piffl, H. Weisen, A. Zabolotsky, and the TCV Team. “Ultrasoft x-ray spectroscopy using multilayer mirrors on TCV”. In: *Plasma Physics and Controlled Fusion* 46.11 (2004), pages 1659–1674. DOI: [10.1088/0741-3335/46/11/001](https://doi.org/10.1088/0741-3335/46/11/001). URL: <http://iopscience.iop.org/0741-3335/46/11/001>.
- [178] E. Scavino, J. Bakos, H. Weisen, and TCV Team. “Effects of ECRH power and safety factor on laser blow-off injected impurity confinement in TCV”. In: *Plasma Physics and Controlled Fusion* 46.5 (2004), pages 857–868. DOI: [10.1088/0741-3335/46/5/008](https://doi.org/10.1088/0741-3335/46/5/008). URL: <http://iopscience.iop.org/0741-3335/46/5/008>.
- [179] Emiliano Fable. “Experimental and Theoretical Study of Particle Transport in the TCV Tokamak”. PhD thesis. École Polytechnique Fédérale de Lausanne, June 2009. DOI: [10.5075/epfl-thesis-4334](https://doi.org/10.5075/epfl-thesis-4334). URL: <http://infoscience.epfl.ch/record/130845>.
- [180] D. Lafrance, A. Boileau, B. L. Stansfield, and W. Zuzak. “Visible spectroscopy on the Tokamak de Varennes with high spatial resolution”. In: *Review of Scientific Instruments* 61.12 (Dec. 1990), pages 3793–3796. DOI: [10.1063/1.1141555](https://doi.org/10.1063/1.1141555). URL: <http://scitation.aip.org/content/aip/journal/rsi/61/12/10.1063/1.1141555>.
- [181] E. Haddad, C. Belanger, B. C. Gregory, and G. Abel. “Impurity transport and density in TdeV”. In: *Plasma Physics and Controlled Fusion* 34.4 (1992), pages 579–591. DOI: [10.1088/0741-3335/34/4/016](https://doi.org/10.1088/0741-3335/34/4/016). URL: <http://iopscience.iop.org/0741-3335/34/4/016>.
- [182] E. Haddad, H. H. Mai, B. C. Gregory, D. G. Whyte, C. Janicki, G. Ratel, D. Lafrance, A. Cote, D. Michaud, and A. H. Sarkissian. “The effect of divertor plate biasing on impurity transport and levels in the central plasma of TdeV”. In: *Nuclear Fusion* 36.5 (1996), pages 613–628. DOI: [10.1088/0029-5515/36/5/I08](https://doi.org/10.1088/0029-5515/36/5/I08). URL: <http://iopscience.iop.org/0029-5515/36/5/I08>.
- [183] K. W. Gentle, R. D. Bengtson, R. Bravenec, D. L. Brower, W. L. Hodge, T. P. Kochanski, N. C. Luhmann Jr., D. Paterson, W. A. Peebles, P. E. Phillips, E. J. Powers, T. P. Price, B. Richards, C. P. Ritz, D. W. Ross, W. L. Rowan, R. Savage, J. C. Wiley, and Y. S. Wan. “Transport studies in the TEXT Tokamak”. In: *Plasma Physics and Controlled Fusion* 26.12A (1984), pages 1407–1418. DOI: [10.1088/0741-3335/26/12A/006](https://doi.org/10.1088/0741-3335/26/12A/006). URL: <http://iopscience.iop.org/0741-3335/26/12A/006>.

- [184] E. J. Synakowski, R. D. Bengtson, A. Ouroua, A. J. Wootton, and S. K. Kim. “Spectroscopic measurements of light impurity peaking after pellet injection on TEXT”. In: *Nuclear Fusion* 29.2 (1989), pages 311–315. DOI: [10.1088/0029-5515/29/2/013](https://doi.org/10.1088/0029-5515/29/2/013). URL: <http://iopscience.iop.org/0029-5515/29/2/013>.
- [185] Kevin Wayne Wenzel. “Measurements of Injected Impurity Transport in TEXT Using Multiply Filtered Soft X-ray Detectors”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology, Feb. 1990. URL: <http://hdl.handle.net/1721.1/93625>.
- [186] S. M. Mahajan, P. M. Valanju, and William L. Rowan. “Closed, form invariant solutions of convective-diffusive systems with applications to impurity transport”. In: *Physics of Fluids B: Plasma Physics* 4.8 (Aug. 1992), pages 2495–2498. DOI: [10.1063/1.860216](https://doi.org/10.1063/1.860216). URL: <http://scitation.aip.org/content/aip/journal/pofb/4/8/10.1063/1.860216>.
- [187] W. Horton and W. Rowan. “Impurity transport studies in the Texas Experimental Tokamak (TEXT)”. In: *Physics of Plasmas* 1.4 (Apr. 1994), pages 901–908. DOI: [10.1063/1.870749](https://doi.org/10.1063/1.870749). URL: <http://scitation.aip.org/content/aip/journal/pop/1/4/10.1063/1.870749>.
- [188] B. Zurro, K. J. McCarthy, J. Vega, and A. Baciero. “On the interpretation of laser ablation data in fusion plasmas”. In: *25<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (Praha, Czech Republic, June 29–July 3, 1998). Edited by P. Pavlo. Volume 22C. European Conference Abstracts. European Physical Society, 1998, p4.083. URL: [http://epsppd.epfl.ch/Praha/WEB/98ICPP\\_W/F092PR.PDF](http://epsppd.epfl.ch/Praha/WEB/98ICPP_W/F092PR.PDF).
- [189] J. Castracane, Y. Demers, and H. H. Mai. “EUV impurity measurements on the TEXTOR tokamak”. In: *Plasma Physics and Controlled Fusion* 29.6 (1987), pages 759–768. DOI: [10.1088/0741-3335/29/6/006](https://doi.org/10.1088/0741-3335/29/6/006). URL: <http://iopscience.iop.org/0741-3335/29/6/006>.
- [190] J. Rapp, M. Z. Tokar, L. Könen, H. R. Koslowski, G. Bertschinger, M. Brix, H. Claassen, R. Jaspers, A. Krämer-Flecken, K. Ohya, V. Philipps, A. Pospieszczyk, U. Samm, T. Tanabe, G. Telesca, B. Unterberg, and G. Van Oost. “Transport studies of high-Z elements in neon edge radiation cooled discharges in TEXTOR-94”. In: *Plasma Physics and Controlled Fusion* 39.10 (1997), pages 1615–1634. DOI: [10.1088/0741-3335/39/10/009](https://doi.org/10.1088/0741-3335/39/10/009). URL: <http://iopscience.iop.org/0741-3335/39/10/009>.

- [191] W. Biel, R. Dux, G. Bertschinger, M. Brix, E. Graffmann, G. Fuchs, H. R. Koslowski, A. Krämer-Flecken, J. Rapp, and the TEXTOR Team. “Impurity transport studies at TEXTOR-94”. In: *25<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (Praha, Czech Republic, June 29–July 3, 1998). Edited by P. Pavlo. Volume 22c. European Conference Abstracts. European Physical Society, 1998, P2.027. URL: [http://epsppd.epfl.ch/Praha/WEB/98ICPP\\_W/B182PR.PDF](http://epsppd.epfl.ch/Praha/WEB/98ICPP_W/B182PR.PDF).
- [192] W. Biel, I. Ahmad, C. J. Barth, G. Bertschinger, R. Dux, G. Fuchs, M. von Hellermann, H. R. Koslowski, A. Krämer-Flecken, M. Lehnen, A. Marchuk, H. J. van der Meiden, J. Rapp, B. Unterberg, and the TEXTOR Team. “Progress in impurity transport studies on TEXTOR using new VUV spectrometers with high time resolution”. In: *28<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (Madeira, Portugal, June 18–22, 2001). Volume 25A. European Conference Abstracts. European Physical Society, 2001, pages 1389–1392. URL: <http://www.cfn.ist.utl.pt/EPS2001/fin/pdf/P4.015.pdf>.
- [193] O. Marchuk, M. Z. Tokar, G. Bertschinger, A. Urnov, H.-J. Kunze, D. Pilipenko, X. Loozen, D. Kalupin, D. Reiter, A. Pospieszczyk, W. Biel, M. Goto, and F. Goryaev. “Comparison of impurity transport model with measurements of He-like spectra of argon at the tokamak TEXTOR”. In: *Plasma Physics and Controlled Fusion* 48.11 (2006), pages 1633–1646. DOI: [10.1088/0741-3335/48/11/005](https://doi.org/10.1088/0741-3335/48/11/005). URL: <http://iopscience.iop.org/0741-3335/48/11/005>.
- [194] A. Greiche, Y. Liang, O. Marchuk, G. Bertschinger, W. Biel, R. Burhenn, R. Dux, H. R. Koslowski, A. Krämer-Flecken, K. Löwenbrück, O. Schmitz, G. W. Spakman, R. Wolf, and the TEXTOR-team. “Transport of argon and iron during a resonant magnetic perturbation at TEXTOR-DED”. In: *Plasma Physics and Controlled Fusion* 51.3 (2009), 032001. DOI: [10.1088/0741-3335/51/3/032001](https://doi.org/10.1088/0741-3335/51/3/032001). URL: <http://iopscience.iop.org/0741-3335/51/3/032001>.
- [195] TFR Group. “Light impurity transport in the TFR tokamak: Comparison of oxygen and carbon line emission with numerical simulations”. In: *Nuclear Fusion* 22.9 (1982), pages 1173–1189. DOI: [10.1088/0029-5515/22/9/004](https://doi.org/10.1088/0029-5515/22/9/004). URL: <http://iopscience.iop.org/0029-5515/22/9/004>.
- [196] TFR Group. “Heavy-impurity transport in the TFR tokamak – comparison of line emission with numerical simulations”. In: *Nuclear Fusion* 23.5 (1983), pages 559–569. DOI: [10.1088/0029-5515/23/5/001](https://doi.org/10.1088/0029-5515/23/5/001). URL: <http://iopscience.iop.org/0029-5515/23/5/001>.

- [197] C. Breton, C. De Michelis, W. Hecq, M. Mattioli, J. Ramette, and B. Saoutic. “TFR discharges with low metal impurity radiation level”. In: *Plasma Physics and Controlled Fusion* 27.3 (1985), pages 355–359. DOI: [10.1088/0741-3335/27/3/012](https://doi.org/10.1088/0741-3335/27/3/012). URL: <http://iopscience.iop.org/0741-3335/27/3/012>.
- [198] TFR Group. “Laser-injected impurity transport in ohmically and ICRF-heated TFR plasmas”. In: *Nuclear Fusion* 25.12 (1985), pages 1767–1774. DOI: [10.1088/0029-5515/25/12/007](https://doi.org/10.1088/0029-5515/25/12/007). URL: <http://iopscience.iop.org/0029-5515/25/12/007>.
- [199] TFR Group. “Charge-exchange measurements of fully stripped oxygen and carbon ion radial density profiles in TFR”. In: *Physics Letters A* 112.1–2 (Oct. 1985), pages 29–32. DOI: [10.1016/0375-9601\(85\)90455-4](https://doi.org/10.1016/0375-9601(85)90455-4). URL: <http://www.sciencedirect.com/science/article/pii/0375960185904554>.
- [200] B. C. Stratton, S. A. Cohen, F. P. Boody, C. E. Bush, R. Ellis III, R. J. Fonck, E. Fredrickson, R. J. Groebner, K. W. Hill, R. A. Hulse, E. S. Marmor, A. T. Ramsey, J. F. Schivell, S. S. Sesnic, and J. Timberlake. “Impurity transport in ohmic and neutral-beam-heated TFTR discharges”. In: *Journal of Nuclear Materials* 145–147 (Feb. 1987), pages 587–591. DOI: [10.1016/0022-3115\(87\)90404-1](https://doi.org/10.1016/0022-3115(87)90404-1). URL: <http://www.sciencedirect.com/science/article/pii/0022311587904041>.
- [201] B. C. Stratton, A. T. Ramsey, F. P. Boody, C. E. Bush, R. J. Fonck, R. J. Groebner, R. A. Hulse, R. K. Richards, and J. Schivell. “Spectroscopic study of impurity behaviour in neutral beam heated and ohmically heated TFTR discharges”. In: *Nuclear Fusion* 27.7 (1987), pages 1147–1164. DOI: [10.1088/0029-5515/27/7/008](https://doi.org/10.1088/0029-5515/27/7/008). URL: <http://iopscience.iop.org/0029-5515/27/7/008>.
- [202] P. C. Efthimion, M. Bitter, E. D. Fredrickson, R. J. Goldston, G. W. Hammett, K. W. Hill, H. Hsuan, R. A. Hulse, R. Kaita, D. K. Mansfield, D. C. McCune, K. M. McGuire, S. S. Medley, D. Mueller, A. T. Ramsey, S. D. Scott, B. C. Stratton, K.-L. Wong, TFTR Group, H. Biglari, P. H. Diamond, Y. Takase, and V. A. Vershkov. “Transport studies on TFTR utilizing perturbation techniques”. In: *Proceedings of the 12<sup>th</sup> International Conference on Fusion Energy*. (Nice, France, Oct. 12–19, 1988). Volume 1. IAEA-CN-50/A-V-4. International Atomic Energy Agency. Vienna, 1989, pages 307–321. URL: [https://www.iaea.org/inis/collection/NCLCollectionStore/\\_Public/20/077/20077811.pdf#page=333](https://www.iaea.org/inis/collection/NCLCollectionStore/_Public/20/077/20077811.pdf#page=333).
- [203] R. J. Knize, R. J. Fonck, R. B. Howell, R. A. Hulse, and K. P. Jaehnig. “Utilization of charge exchange recombination spectroscopy for the study of metallic ion transport in TFTR”. In: *Review of Scientific Instruments* 59.8 (Aug.



- 1988), pages 1518–1520. DOI: [10.1063/1.1140182](https://doi.org/10.1063/1.1140182). URL: <http://scitation.aip.org/content/aip/journal/rsi/59/8/10.1063/1.1140182>.
- [204] B. C. Stratton, R. J. Fonck, R. A. Hulse, A. T. Ramsey, J. Timberlake, P. C. Efthimion, E. D. Fredrickson, B. Grek, K. W. Hill, D. W. Johnson, D. K. Mansfield, H. Park, F. J. Stauffer, and G. Taylor. “Impurity transport in ohmically heated TFTR plasmas”. In: *Nuclear Fusion* 29.3 (1989), pages 437–447. DOI: [10.1088/0029-5515/29/3/007](https://doi.org/10.1088/0029-5515/29/3/007). URL: <http://iopscience.iop.org/0029-5515/29/3/007>.
- [205] E. J. Synakowski, B. C. Stratton, P. C. Efthimion, R. J. Fonck, R. A. Hulse, D. W. Johnson, D. K. Mansfield, H. Park, S. D. Scott, and G. Taylor. “Measurements of radial profiles of  $\text{He}^{2+}$  transport coefficients on the TFTR tokamak”. In: *Physical Review Letters* 65.18 (Oct. 1990), pages 2255–2258. DOI: [10.1103/PhysRevLett.65.2255](https://doi.org/10.1103/PhysRevLett.65.2255). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.65.2255>.
- [206] B. C. Stratton, E. J. Synakowski, P. C. Efthimion, R. J. Fonck, K. W. Hill, R. A. Hulse, D. W. Johnson, H. Park, G. Taylor, and J. Timberlake. “Measurement of iron transport in the TFTR tokamak by charge exchange recombination spectroscopy”. In: *Nuclear Fusion* 31.1 (1991), pages 171–175. DOI: [10.1088/0029-5515/31/1/015](https://doi.org/10.1088/0029-5515/31/1/015). URL: <http://iopscience.iop.org/0029-5515/31/1/015>.
- [207] E. J. Synakowski, P. C. Efthimion, G. Rewoldt, B. C. Stratton, W. M. Tang, B. Grek, K. W. Hill, R. A. Hulse, D. W. Johnson, M. W. Kissick, D. K. Mansfield, D. McCune, D. R. Mikkelsen, H. K. Park, A. T. Ramsey, M. H. Redi, S. D. Scott, G. Taylor, J. Timberlake, and M. C. Zarnstorff. “Helium, iron, and electron particle transport and energy transport studies on the Tokamak Fusion Test Reactor”. In: *Physics of Fluids B: Plasma Physics* 5.7 (July 1993), pages 2215–2228. DOI: [10.1063/1.860755](https://doi.org/10.1063/1.860755). URL: <http://scitation.aip.org/content/aip/journal/pofb/5/7/10.1063/1.860755>.
- [208] P. C. Efthimion, L. C. Johnson, J. D. Strachan, E. J. Synakowski, M. Zarnstorff, H. Adler, C. Barnes, R. V. Budny, F. C. Jobes, M. Louglin, D. McCune, D. Mueller, A. T. Ramsey, G. Rewoldt, A. L. Roquemore, W. M. Tang, and G. Taylor. “Tritium Particle Transport Experiments on TFTR during D-T Operation”. In: *Physical Review Letters* 75.1 (July 1995), pages 85–88. DOI: [10.1103/PhysRevLett.75.85](https://doi.org/10.1103/PhysRevLett.75.85). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.75.85>.



- [209] P. C. Efthimion, S. von Goeler, W. A. Houlberg, E. J. Synakowski, M. C. Zarnstorff, S. H. Batha, R. E. Bell, M. Bitter, C. E. Bush, F. M. Levinton, E. Mazzucato, D. McCune, D. Mueller, H. Park, A. T. Ramsey, A. L. Roquemore, and G. Taylor. "Observation of particle transport barriers in reverse shear plasmas on the Tokamak Fusion Test Reactor". In: *Physics of Plasmas* 5.5 (May 1998), pages 1832–1838. DOI: [10.1063/1.872853](https://doi.org/10.1063/1.872853). URL: <http://scitation.aip.org/content/aip/journal/pop/5/5/10.1063/1.872853>.
- [210] P. C. Efthimion, S. von Goeler, W. A. Houlberg, E. J. Synakowski, M. C. Zarnstorff, S. Batha, R. E. Bell, M. Bitter, C. E. Bush, F. M. Levinton, E. Mazzucato, D. McCune, D. Mueller, H. Park, A. T. Ramsey, A. L. Roquemore, and G. Taylor. "Observation of Neoclassical Transport in Reverse Shear Plasmas on the Tokamak Fusion Test Reactor". In: *17<sup>th</sup> International Conference on Fusion Energy*. (Yokohama, Japan, Oct. 19–24, 1998). Volume IAEA-CN-69. International Atomic Energy Agency. 1998, EXP1/15. URL: [http://www-pub.iaea.org/mtcd/publications/pdf/csp\\_008c/fec1998/html/node137.htm](http://www-pub.iaea.org/mtcd/publications/pdf/csp_008c/fec1998/html/node137.htm).
- [211] K. W. Hill, M. G. Bell, R. E. Bell, R. Budny, C. E. Bush, D. R. Ernst, G. W. Hammett, D. R. Mikkelsen, H. K. Park, A. T. Ramsey, S. A. Sabbagh, S. D. Scott, E. J. Synakowski, G. Taylor, and M. C. Zarnstorff. "Highly radiative plasmas for local transport studies and power and particle handling in reactor regimes". In: *Nuclear Fusion* 39.11Y (1999), pages 1949–1954. DOI: [10.1088/0029-5515/39/11Y/339](https://doi.org/10.1088/0029-5515/39/11Y/339). URL: <http://iopscience.iop.org/0029-5515/39/11Y/339>.
- [212] M. Mattioli, C. de Michelis, P. Monier-Garbet, A. L. Pecquet, and W. Hecq. *Laser blow-off injection of metal impurities in Tore Supra*. Technical report EUR-CEA-FC-1491. Association Euratom-CEA sur la Fusion, June 1993. URL: [https://inis.iaea.org/search/search.aspx?orig\\_q=RN:25013137](https://inis.iaea.org/search/search.aspx?orig_q=RN:25013137).
- [213] M. Mattioli, C. De Michelis, and P. Monier-Garbet. "Behaviour of intrinsic carbon and laser blow-off injected nickel in Tore Supra during ergodic divertor activation". In: *Nuclear Fusion* 35.7 (1995), pages 807–825. DOI: [10.1088/0029-5515/35/7/I05](https://doi.org/10.1088/0029-5515/35/7/I05). URL: <http://iopscience.iop.org/0029-5515/35/7/I05>.
- [214] L. Carraro, M. Mattioli, F. Sattin, C. Demichelis, J. T. Hogan, W. Mandl, M. E. Puiatti, P. Scarin, and M. Valisa. "Collisional-Radiative models for hydrogen-like and helium-like carbon and oxygen ions and applications to experimental data from the TS tokamak and the reversed field pinch RFX". In: *Physica Scripta* 55.5 (1997), pages 565–578. DOI: [10.1088/0031-8949/55/5/009](https://doi.org/10.1088/0031-8949/55/5/009). URL: <http://iopscience.iop.org/1402-4896/55/5/009>.

- [215] M. Mattioli, M. Erba, T. Dudok de Wit, A.-L. Pecquet, J.-L. Segui, and J. C. Vallet. “Enhanced transport of impurity ions and electron heat in Tore Supra ohmic discharges following nickel laser blow-off injection”. In: *Nuclear Fusion* 38.2 (1998), pages 189–206. DOI: [10.1088/0029-5515/38/2/304](https://doi.org/10.1088/0029-5515/38/2/304). URL: <http://iopscience.iop.org/0029-5515/38/2/304>.
- [216] M. Mattioli, C. De Michelis, and A. L. Pecquet. “Laser blow-off injected impurity transport in L mode Tore Supra plasmas”. In: *Nuclear Fusion* 38.11 (1998), pages 1629–1635. DOI: [10.1088/0029-5515/38/11/303](https://doi.org/10.1088/0029-5515/38/11/303). URL: <http://iopscience.iop.org/0029-5515/38/11/303>.
- [217] O. Marchuk, G. Bertschinger, C. Fenzi-Bonizec, B. Schunke, M. Mattioli, A. Whiteford, and N. R. Badnell. “Impurity transport studies in TORE SUPRA with He-like spectroscopy”. In: *32<sup>nd</sup> European Physical Society Conference on Plasma Physics and Controlled Fusion combined with the 8<sup>th</sup> International Workshop on Fast Ignition of Fusion Targets*. (Tarragona, Spain, June 26–July 1, 2005). Edited by C Hidalgo and B. Ph. van Milligen. Volume 29c. European Conference Abstracts. European Physical Society, 2005, P–2.025. URL: [http://epsppd.epfl.ch/Tarragona/pdf/P2\\_025.pdf](http://epsppd.epfl.ch/Tarragona/pdf/P2_025.pdf).
- [218] T. Parisot, R. Guirlet, N. Dubuit, A. Sirinelli, X. Garbet, C. Bourdelle, P. R. Thomas, and L. Godbert-Mouret. “Charge and electron density dependences of anomalous impurity transport in Tore Supra”. In: *33<sup>rd</sup> European Physical Society Conference on Plasma Physics*. (Rome, Italy, June 19–23, 2006). Edited by F. De Marco and G. Vlad. Volume 301. European Conference Abstracts. European Physical Society, 2006, P–5.102. URL: [http://epsppd.epfl.ch/Roma/pdf/P5\\_102.pdf](http://epsppd.epfl.ch/Roma/pdf/P5_102.pdf).
- [219] T. Parisot, R. Guirlet, C. Bourdelle, X. Garbet, N. Dubuit, F. Imbeaux, and P. R. Thomas. “Experimental impurity transport and theoretical interpretation in a Tore Supra lower-hybrid heated plasma”. In: *Plasma Physics and Controlled Fusion* 50.5 (2008), 055010. DOI: [10.1088/0741-3335/50/5/055010](https://doi.org/10.1088/0741-3335/50/5/055010). URL: <http://iopscience.iop.org/0741-3335/50/5/055010>.
- [220] R. Guirlet, D. Villegas, T. Parisot, C. Bourdelle, X. Garbet, F. Imbeaux, D. Mazon, and D. Pacella. “Anomalous transport of light and heavy impurities in Tore Supra ohmic, weakly sawtoothed plasmas”. In: *Nuclear Fusion* 49.5 (2009), 055007. DOI: [10.1088/0029-5515/49/5/055007](https://doi.org/10.1088/0029-5515/49/5/055007). URL: <http://iopscience.iop.org/0029-5515/49/5/055007>.
- [221] D. Villegas, R. Guirlet, C. Bourdelle, X. Garbet, F. Imbeaux, J.-L. Ségui, and R. Sabot. “Influence of the electron temperature gradient on impurity transport in Tore Supra”. In: *36<sup>th</sup> European Physical Society Conference on Plasma*

- Physics*. (Sofia, Bulgaria, June 29–July 3, 2009). Edited by M. Mateev and E. Benova. Volume 33E. European Conference Abstracts. European Physical Society, 2009, P-4.209. URL: [http://epsppd.epfl.ch/Sofia/pdf/P4\\_209.pdf](http://epsppd.epfl.ch/Sofia/pdf/P4_209.pdf).
- [222] R. Guirlet, A. Sirinelli, T. Parisot, R. Sabot, X. Garbet, J. F. Artaud, C. Bourdelle, P. Hennequin, G. T. Hoang, F. Imbeaux, J. L. Ségui, and D. Villegas. “Particle Transport in Vanishing Turbulence Conditions in the Tore Supra Plasma Core”. In: *Proceedings of the 23<sup>rd</sup> IAEA Fusion Energy Conference*. (Daejeon, South Korea, Oct. 11–16, 2010). Volume IAEA-CN-180. International Atomic Energy Agency. 2010, EXC/P8-06. URL: <http://www-naweb.iaea.org/napc/physics/FEC/FEC2010/html/node582.htm>.
- [223] R. Guirlet, A. Sirinelli, T. Parisot, R. Sabot, J. F. Artaud, C. Bourdelle, X. Garbet, P. Hennequin, G. T. Hoang, F. Imbeaux, J. L. Ségui, D. Mazon, and D. Villegas. “Particle transport in low core turbulence Tore-Supra plasmas”. In: *Nuclear Fusion* 50.9 (2010), 095009. DOI: [10.1088/0029-5515/50/9/095009](https://doi.org/10.1088/0029-5515/50/9/095009). URL: <http://iopscience.iop.org/0029-5515/50/9/095009>.
- [224] D. Villegas, R. Guirlet, C. Bourdelle, G. T. Hoang, X. Garbet, and R. Sabot. “Experimental Electron Temperature Gradient Dependence of Heavy Impurity Transport in Fusion Devices”. In: *Physical Review Letters* 105.3 (July 2010), 035002. DOI: [10.1103/PhysRevLett.105.035002](https://doi.org/10.1103/PhysRevLett.105.035002). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.105.035002>.
- [225] D. Villegas, R. Guirlet, C. Bourdelle, X. Garbet, G. T. Hoang, R. Sabot, F. Imbeaux, and J. L. Ségui. “Experimental and theoretical study of nickel transport dependence on gradients in Tore Supra”. In: *Nuclear Fusion* 54.7 (2014), 073011. DOI: [10.1088/0029-5515/54/7/073011](https://doi.org/10.1088/0029-5515/54/7/073011). URL: <http://iopscience.iop.org/0029-5515/54/7/073011>.
- [226] M. Bessenrodt-Weberpals, J. Hackmann, C. Nieswand, and J. Uhlenbusch. “Distribution of beryllium concentrations and fluxes in the tokamak UNITER”. In: *Plasma Physics and Controlled Fusion* 30.4 (1988), pages 407–413. DOI: [10.1088/0741-3335/30/4/010](https://doi.org/10.1088/0741-3335/30/4/010). URL: <http://iopscience.iop.org/0741-3335/30/4/010>.
- [227] K. Khlopenkov and S. Sudo. “New particle transport diagnostics with tracer-encapsulated solid pellet”. In: *Plasma Physics and Controlled Fusion* 43.11 (2001), pages 1547–1557. DOI: [10.1088/0741-3335/43/11/309](https://doi.org/10.1088/0741-3335/43/11/309). URL: <http://iopscience.iop.org/0741-3335/43/11/309>.

- [228] Y. Liang, K. Ida, J. E. Rice, T. Minami, H. Funaba, S. Kado, A. Fujisawa, Y. Yoshimura, S. Nishimura, M. Isobe, S. Okamura, and K. Matsuoka. “Observation of low impurity diffusivity inside the neoclassical transport barrier in the Compact Helical System”. In: *Physics of Plasmas* 9.10 (Oct. 2002), pages 4179–4187. DOI: [10.1063/1.1506160](https://doi.org/10.1063/1.1506160). URL: <http://scitation.aip.org/content/aip/journal/pop/9/10/10.1063/1.1506160>.
- [229] S. Sudo, N. Tamura, K. Khlopenkov, S. Muto, H. Funaba, I. Viniar, V. Sergeev, K. Sato, K. Ida, K. Kawahata, A. Komori, K. Matsuoka, K. Narihara, S. Okamura, N. Ohya, K. Tanaka, and O. Motojima. “Particle transport diagnostics on CHS and LHD with tracer-encapsulated solid pellet injection”. In: *Plasma Physics and Controlled Fusion* 44.1 (2002), pages 129–135. DOI: [10.1088/0741-3335/44/1/311](https://doi.org/10.1088/0741-3335/44/1/311). URL: <http://iopscience.iop.org/0741-3335/44/1/311>.
- [230] J. Sallander. “Impurity profiles and radial transport in the EXTRAP-T2 reversed field pinch”. In: *Plasma Physics and Controlled Fusion* 41.5 (1999), pages 679–691. DOI: [10.1088/0741-3335/41/5/308](https://doi.org/10.1088/0741-3335/41/5/308). URL: <http://iopscience.iop.org/0741-3335/41/5/308>.
- [231] M. Kuldkepp, P. R. Brunsell, M. Cecconello, R. Dux, S. Menmuir, and E. Rachlew. “Measurements and modeling of transport and impurity radial profiles in the EXTRAP T2R reversed field pinch”. In: *Physics of Plasmas* 13.9 (2006), 092506. DOI: [10.1063/1.2349304](https://doi.org/10.1063/1.2349304). URL: <http://scitation.aip.org/content/aip/journal/pop/13/9/10.1063/1.2349304>.
- [232] H. Kaneko, K. Kondo, O. Motojima, H. Zushi, T. Mizuuchi, S. Sudo, S. Besshou, M. Sato, H. Okada, Y. Takeiri, F. Sano, T. Mutoh, T. Obiki, A. Iiyoshi, K. Uo, J. E. Rice, J. L. Terry, and E. S. Marmor. “Transport analysis of injected impurities in currentless Heliotron E plasmas”. In: *Nuclear Fusion* 27.7 (1987), pages 1075–1090. DOI: [10.1088/0029-5515/27/7/002](https://doi.org/10.1088/0029-5515/27/7/002). URL: <http://iopscience.iop.org/0029-5515/27/7/002>.
- [233] J. E. Rice, J. L. Terry, E. S. Marmor, O. Motojima, H. Kaneko, K. Kondo, T. Mizuuchi, S. Besshou, T. Mutoh, F. Sano, A. Sasaki, M. Sato, S. Sudo, H. Zushi, M. Iima, K. Magome, T. Obiki, A. Iiyoshi, and K. Uo. “Transport of injected impurities in Heliotron E”. In: *Nuclear Fusion* 24.9 (1984), pages 1205–1211. DOI: [10.1088/0029-5515/24/9/010](https://doi.org/10.1088/0029-5515/24/9/010). URL: <http://iopscience.iop.org/0029-5515/24/9/010>.
- [234] Chunfeng Dong, Shigeru Morita, Motoshi Goto, Erhui Wang, Izumi Murakami, and Norimasa Yamamoto. “Accurate Determination of Iron Density in Large Helical Device Based on Absolute Intensity Profile Measure-

- ment of Extreme Ultraviolet Spectral Lines”. In: *Japanese Journal of Applied Physics* 51.1R (2011), 010205. DOI: [10.1143/JJAP.51.010205](https://doi.org/10.1143/JJAP.51.010205). URL: <http://iopscience.iop.org/1347-4065/51/1R/010205>.
- [235] S. Morita, M. Goto, S. Muto, H. Y. Zhou, C. F. Dong, and LHD Experiment Group. “Core Impurity Transport Studies from Different Diagnostic Approaches in LHD”. In: *Fusion Science and Technology* 58.1 (July–Aug. 2010), pages 91–102. URL: [http://www.ans.org/pubs/journals/fst/a\\_10796](http://www.ans.org/pubs/journals/fst/a_10796).
- [236] Shigeru Sudo. “A review of impurity transport characteristics in the LHD”. In: *Plasma Physics and Controlled Fusion* 58.4 (2016), 043001. DOI: [10.1088/0741-3335/58/4/043001](https://doi.org/10.1088/0741-3335/58/4/043001). URL: <http://iopscience.iop.org/0741-3335/58/4/043001>.
- [237] Shigeru Sudo, Naoki Tamura, Hisamichi Funaba, Sadatsugu Muto, Chihiro Suzuki, and Izumi Murakami. “Impurity Transport Study with TESPEL Injection and Simulation”. In: *Plasma and Fusion Research* 8 (2013), 2402059. DOI: [10.1585/pfr.8.2402059](https://doi.org/10.1585/pfr.8.2402059). URL: [https://www.jstage.jst.go.jp/article/pfr/8/0/8\\_2402059/\\_article](https://www.jstage.jst.go.jp/article/pfr/8/0/8_2402059/_article).
- [238] Shigeru Morita, Chunfeng Dong, Motoshi Goto, Masahiro Kobayashi, Sadatsugu Muto, Kozo Yamazaki, Masayuki Yokoyama, Hangyu Zhou, Tsuyoshi Akiyama, Naoko Ashikawa, Zhengying Cui, Keisuke Fujii, Xiang Gao, Masahiro Hasuo, Katumi Ida, Katsunori Ikeda, Atsushi Iwamae, Osamu Kaneko, Suguru Masuzaki, Junichi Miyazawa, Tomohiro Morisaki, Kenichi Nagaoka, Kazumichi Narihara, Satoshi Ohdachi, Masaki Osakabe, Satoru Sakakibara, Ryuichi Sakamoto, Mamoru Shoji, Yasuhiko Takeiri, Kenji Tanaka, Kazuo Toi, Masayuki Tokitani, Tokihiko Tokuzawa, Katsuyoshi Tsumori, Kiyomasa Watanabe, Hiroshi Yamada, Ichihiko Yamada, Jun Yanagibayashi, Mikiro Yoshinuma, Akio Komori, and LHD experiment group. “Progress in Impurity-Related Physics Experiments in LHD”. In: *Plasma and Fusion Research* 5 (2010), S2004. DOI: [10.1585/pfr.5.S2004](https://doi.org/10.1585/pfr.5.S2004). URL: [https://www.jstage.jst.go.jp/article/pfr/5/0/5\\_0\\_S2004/\\_article](https://www.jstage.jst.go.jp/article/pfr/5/0/5_0_S2004/_article).
- [239] M. Yoshinuma, K. Ida, M. Yokoyama, M. Osakabe, K. Nagaoka, S. Morita, M. Goto, N. Tamura, C. Suzuki, S. Yoshimura, H. Funaba, Y. Takeiri, K. Ikeda, K. Tsumori, O. Kaneko, and the LHD Experimental Group. “Observation of an impurity hole in the Large Helical Device”. In: *Nuclear Fusion* 49.6 (2009), 062002. DOI: [10.1088/0029-5515/49/6/062002](https://doi.org/10.1088/0029-5515/49/6/062002). URL: <http://iopscience.iop.org/0029-5515/49/6/062002>.

- [240] Shigeru Morita, Motoshi Goto, Masahiro Kobayashi, Sadatsugu Muto, Malay Bikas Chowdhuri, Chunfeng Dong, Hangyu Zhou, Zhengying Cui, Keisuke Fujii, Akihiro Furuzawa, Masahiro Hasuo, Atsushi Iwamae, Yinxian Jie, Mohammed Koubiti, Ikuya Sakurai, Yuzuru Tawara, Baonian Wan, Zhenwei Wu, and Naohiro Yamaguchi. “Edge and Core Impurity Transport Study with Spectroscopic Instruments in LHD”. In: *Plasma Science and Technology* 11.4 (Aug. 2009), pages 402–408. DOI: [10.1088/1009-0630/11/4/07](https://doi.org/10.1088/1009-0630/11/4/07). URL: <http://iopscience.iop.org/1009-0630/11/4/07>.
- [241] K. Ida, M. Yoshinuma, M. Osakabe, K. Nagaoka, M. Yokoyama, H. Funaba, C. Suzuki, T. Ido, A. Shimizu, I. Murakami, N. Tamura, H. Kasahara, Y. Takeiri, K. Ikeda, K. Tsumori, O. Kaneko, S. Morita, M. Goto, K. Tanaka, K. Narihara, T. Minami, I. Yamada, and LHD Experimental Group. “Observation of an impurity hole in a plasma with an ion internal transport barrier in the Large Helical Device”. In: *Physics of Plasmas* 16.5 (2009), 056111. DOI: [10.1063/1.3111097](https://doi.org/10.1063/1.3111097). URL: <http://scitation.aip.org/content/aip/journal/pop/16/5/10.1063/1.3111097>.
- [242] Naoki Tamura, Kuninori Sato, Pavel R. Goncharov, Hisamichi Funaba, Yuri L. Igitkhanov, Shigeru Sudo, Ichihiko Yamada, Kazumichi Narihara, Kenji Tanaka, Clive Michael, Kazuo Kawahata, Shinji Yoshimura, Byron J. Peterson, Ryuichi Sakamoto, Masahiro Kobayashi, Hiroshi Yamada, Nobuyoshi Oyabu, Akio Komori, and LHD Experimental Group. “Tracer impurity transport inside an IDB plasma of LHD”. In: *Proceedings of the 17<sup>th</sup> International Toki Conference on Physics of Flows and Turbulence in Plasmas and the 16<sup>th</sup> International Stellarator/Heliotron Workshop*. (Ceratopia Toki, Gifu, Japan, Oct. 15–19, 2007). National Institute for Fusion Science. 2007. URL: [http://www.nifs.ac.jp/itc/itc17/file/PDF\\_proceedings/P1-069.pdf](http://www.nifs.ac.jp/itc/itc17/file/PDF_proceedings/P1-069.pdf).
- [243] Sadatsugu Muto, Shigeru Morita, and LHD Experimental Group. “Impurity Transport Study of Medium-Z Argon by Means of Soft X-Ray Pulse Height Analyzer in LHD”. In: *Plasma and Fusion Research* 2 (2007), S1069. DOI: [10.1585/pfr.2.S1069](https://doi.org/10.1585/pfr.2.S1069). URL: [https://www.jstage.jst.go.jp/article/pfr/2/0/2\\_0\\_S1069/\\_article](https://www.jstage.jst.go.jp/article/pfr/2/0/2_0_S1069/_article).
- [244] Yu. Igitkhanov, P. Goncharov, S. Sudo, N. Tamura, D. Kalinina, H. Funaba, M. Yokoyama, S. Murakami, K. Kawahata, T. Kato, T. Morisaki, B. Peterson, Y. Hakamura, H. Yamada, and O. Motojima. “Impurity Transport Studies on LHD”. In: *Plasma and Fusion Research* 2 (2007), S1131. DOI: [10.1585/pfr.2.S1131](https://doi.org/10.1585/pfr.2.S1131). URL: [https://www.jstage.jst.go.jp/article/pfr/2/0/2\\_0\\_S1131/\\_article](https://www.jstage.jst.go.jp/article/pfr/2/0/2_0_S1131/_article).

- [245] H. Nozato, S. Morita, M. Goto, Y. Takase, A. Ejiri, T. Amano, K. Tanaka, S. Inagaki, and LHD Experimental Group. “A dependence of carbon impurity transport coefficients on fuel ions in hydrogen and helium plasmas of Large Helical Device”. In: *Physics of Plasmas* 13.9 (2006), 092502. DOI: [10.1063/1.2337790](https://doi.org/10.1063/1.2337790). URL: <http://scitation.aip.org/content/aip/journal/pop/13/9/10.1063/1.2337790>.
- [246] Shigeru Morita, Motoshi Goto, Sadatsugu Muto, Ryuji Katai, Hisamichi Yamazaki, Hideaki Nozato, Atsushi Iwamae, Makoto Atake, Takshi Fujimoto, Atsushi Sakaue, Hiroaki Nishimura, Ikuya Sakurai, Chiho Matsumoto, Akihiro Furuzawa, Yuzuru Tawara, Mitsutoshi Aramaki, Yuji Okumura, Koichi Sasaki, Gong Xianzu, Li Jiangang, Wan Baonian, and Cui Zhengying. “Spectroscopic Studies on Impurity Transport of Core and Edge Plasmas in LHD”. In: *Plasma Science and Technology* 8.1 (Jan. 2006), pages 55–60. DOI: [10.1088/1009-0630/8/1/13](https://doi.org/10.1088/1009-0630/8/1/13). URL: <http://iopscience.iop.org/1009-0630/8/1/13>.
- [247] H. Nozato, S. Morita, M. Goto, Y. Takase, A. Ejiri, T. Amano, K. Tanaka, S. Inagaki, and LHD Experimental Group. “A study of charge dependence of particle transport using impurity pellet injection and high-spatial resolution bremsstrahlung measurement on the Large Helical Device”. In: *Physics of Plasmas* 11.5 (May 2004), pages 1920–1930. DOI: [10.1063/1.1695355](https://doi.org/10.1063/1.1695355). URL: <http://scitation.aip.org/content/aip/journal/pop/11/5/10.1063/1.1695355>.
- [248] N. Tamura, S. Sudo, K. V. Khlopenkov, S. Kato, V. Yu. Sergeev, S. Muto, K. Sato, H. Funaba, K. Tanaka, T. Tokuzawa, I. Yamada, K. Narihara, Y. Nakamura, K. Kawahata, N. Ohyaabu, O. Motojima, and LHD experimental groups. “Impurity transport studies by means of tracer-encapsulated solid pellet injection in neutral beam heated plasmas on LHD”. In: *Plasma Physics and Controlled Fusion* 45.1 (2003), pages 27–41. DOI: [10.1088/0741-3335/45/1/302](https://doi.org/10.1088/0741-3335/45/1/302). URL: <http://iopscience.iop.org/0741-3335/45/1/302>.
- [249] J. E. Rice. *Simulated Impurity Transport in LHD from MIST*. Technical report NIFS-548. Nagoya, Japan: National Institute for Fusion Science, May 1998. URL: <http://www.nifs.ac.jp/report/nifs548.html>.
- [250] B. Zurro, M. A. Ochando, K. J. McCarthy, F. Medina, A. Baciero, R. Dux, E. Ascasíbar, A. López-Fraguas, A. López-Sánchez, T. Estrada, E. de la Luna, J. Vega, and TJ-II Team. “Study of impurity transport injected by laser ablation in the TJ-II”. In: *29<sup>th</sup> European Physical Society Conference on Plasma Physics and Controlled Fusion*. (Montreux, Switzerland, June 17–21, 2002). Edited



- by R. Behn and C. Varandas. Volume 26B. European Conference Abstracts. European Physical Society, 2002, P-5.025. URL: [http://crpppc42.epfl.ch/Montreux/pdf/P5\\_025.pdf](http://crpppc42.epfl.ch/Montreux/pdf/P5_025.pdf).
- [251] B. Zurro, M. A. Ochando, A. Baciero, R. Dux, K. J. McCarthy, F. Medina, A. López-Sánchez, D. Rapisarda, A. Fernández, I. Pastor, J. Herranz, and TJ-II Team. “Transport Analysis of Impurities Injected by Laser Ablation in the TJ-II Stellarator”. In: *30<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (St. Petersburg, Russia, July 7–11, 2003). Edited by R. Koch and S. Lebedev. Volume 27A. European Conference Abstracts. European Physical Society, 2003, P-2.79. URL: [http://epsppd.epfl.ch/StPetersburg/PDF/P2\\_079.PDF](http://epsppd.epfl.ch/StPetersburg/PDF/P2_079.PDF).
- [252] B. Zurro, M. A. Ochando, A. Baciero, K. J. McCarthy, F. Medina, A. López-Sánchez, D. Rapisarda, D. Jimenez, A. Fernández, I. Pastor, J. Herranz, and R. Dux. “Method to deduce local impurity transport quantities from the evolution of tomographically reconstructed bolometer signals during tracer injection at TJ-II”. In: *Review of Scientific Instruments* 75.10 (Oct. 2004), pages 4231–4233. DOI: [10.1063/1.1787585](https://doi.org/10.1063/1.1787585). URL: <http://scitation.aip.org/content/aip/journal/rsi/75/10/10.1063/1.1787585>.
- [253] B. Zurro, M. A. Ochando, A. Baciero, A. Fernández, A. Cappa, V. Tribaldos, K. J. McCarthy, F. Medina, A. López-Sánchez, D. Jiménez Rey, D. Rapisarda, I. Pastor, J. Herranz, and TJ-II Team. “Impurity transport and confinement in the TJ-II Stellarator”. In: *20<sup>th</sup> IAEA Fusion Energy Conference*. (Vilamoura, Portugal, Nov. 1–6, 2004). Volume IAEA-CN-116. International Atomic Energy Agency. 2004, EX/P6-32. URL: [http://www-pub.iaea.org/mtcd/meetings/PDFplus/fusion-20-preprints/EX\\_P6-32.pdf](http://www-pub.iaea.org/mtcd/meetings/PDFplus/fusion-20-preprints/EX_P6-32.pdf).
- [254] B. Zurro, A. Baciero, E. Hollmann, M. Tillack, M. Liniers, A. Fernández, A. Cappa, F. Medina, A. López-Sánchez, I. Pastor, J. Herranz, and TJ-II Team. “Confinement of impurities injected by laser blow-off in the ECRH and NBI regimes of the TJ-II stellarator”. In: *36<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Sofia, Bulgaria, June 29–July 3, 2009). Edited by M. Matsev and E. Benova. Volume 33E. European Conference Abstracts. European Physical Society, 2009, P-4.191. URL: [http://epsppd.epfl.ch/Sofia/pdf/P4\\_191.pdf](http://epsppd.epfl.ch/Sofia/pdf/P4_191.pdf).
- [255] B. Zurro, E. Hollmann, A. Baciero, M. A. Ochando, F. Medina, K. J. McCarthy, E. Blanco, E. de la Cal, D. Carralero, M. A. Pedrosa, and TJ-II team. “Transport studies using laser blow-off injection of low-Z trace impurities injected into the TJ-II stellarator”. In: *Nuclear Fusion* 51.6 (2011), 063015. DOI:



- 10.1088/0029-5515/51/6/063015. URL: <http://iopscience.iop.org/0029-5515/51/6/063015>.
- [256] B. Zurro, E. M. Hollmann, A. Baciero, M. A. Ochando, K. J. McCarthy, F. Medina, J. L. Velasco, I. Pastor, D. Baião, E. de la Cal, D. Rapisarda, and the TJ-II Team. “Studying the impurity charge and main ion mass dependence of impurity confinement in ECR-heated TJ-II stellarator”. In: *Plasma Physics and Controlled Fusion* 56.12 (2014), 124007. DOI: 10.1088/0741-3335/56/12/124007. URL: <http://iopscience.iop.org/0741-3335/56/12/124007>.
- [257] B. Zurro, J. L. Velasco, E. M. Hollmann, A. Baciero, M. A. Ochando, R. Dux, K. J. McCarthy, F. Medina, I. Pastor, R. J. Hajjar, J. M. García-Regaña, D. López-Bruna, A. V. Melnikov, L. G. Eliseev, HIBP Team, and TJ-II team. “Transport analysis of impurities injected by laser blow-off in ECRH and NBI heated plasmas of TJ-II”. In: *42<sup>nd</sup> European Physical Society Conference on Plasma Physics*. (Lisbon, Portugal, June 22–26, 2015). Edited by R. Bingham, W. Suttrop, S. Atzeni, R. Foest, K. McClements, B. Goncalves, C. Silva, and R. Coelho. Volume 39E. European Conference Abstracts. European Physical Society, 2015, P4.160. URL: <http://ocs.ciemat.es/EPS2015PAP/pdf/P4.160.pdf>.
- [258] W VII-A Team and NI Group. “Impurity transport in the Wendelstein VII-A stellarator”. In: *Nuclear Fusion* 25.11 (1985), pages 1593–1609. DOI: 10.1088/0029-5515/25/11/006. URL: <http://iopscience.iop.org/0029-5515/25/11/006>.
- [259] D. Hildebrandt, R. Brakel, A. Elsner, P. Grigull, H. Hacker, R. Burhenn, S. Fiedler, L. Giannone, C. Görner, H. J. Hartfuß, G. Herre, A. Herrmann, J. V. Hofmann, G. Kühner, D. Naujoks, F. Sardei, A. Weller, R. Wolf, w7-AS Team, ECRH Team, and NBI Team. “Plasma response on impurity injection in w7-AS”. In: *Journal of Nuclear Materials* 241–243 (Feb. 1997), pages 950–955. DOI: 10.1016/S0022-3115(97)80172-9. URL: <http://www.sciencedirect.com/science/article/pii/S0022311597801729>.
- [260] R. Burhenn, A. Weller, w7-AS Team, NI Group, and The ECRH Group. “Derivation of local impurity transport quantities from soft-x radiation evolution during tracer injection at w7-AS”. In: *Review of Scientific Instruments* 70.1 (Jan. 1999), pages 603–607. DOI: 10.1063/1.1149357. URL: <http://scitation.aip.org/content/aip/journal/rsi/70/1/10.1063/1.1149357>.
- [261] R. Burhenn, K. Ida, R. Brakel, L. Giannone, P. Grigull, J. P. Knauer, F. Kunkel, H. Maassberg, K. McCormick, E. Pasch, A. Weller, w7-AS team, ECRH group,

- and NI group. “Characterization of impurity transport in the w7-AS stellarator during the transition to the improved confinement regime”. In: *29<sup>th</sup> European Physical Society Conference on Plasma Physics and Controlled Fusion*. (Montreux, Switzerland, June 17–21, 2002). Edited by R. Behn and C. Varandas. Volume 26B. European Conference Abstracts. European Physical Society, 2002, P-4.043. URL: [http://epsppd.epfl.ch/Montreux/pdf/P4\\_043.pdf](http://epsppd.epfl.ch/Montreux/pdf/P4_043.pdf).
- [262] K. Ida, R. Burhenn, K. McCormick, E. Pasch, H. Yamada, M. Yoshinuma, S. Inagaki, S. Murakami, M. Osakabe, Y. Liang, R. Brakel, H. Ehmler, L. Giannone, P. Grigull, J. P. Knauer, H. Maassberg, and A. Weller. “Impurity transport model for the normal confinement and high density H-mode discharges in Wendelstein 7-AS”. In: *Plasma Physics and Controlled Fusion* 45.10 (2003), pages 1931–1938. DOI: [10.1088/0741-3335/45/10/006](https://doi.org/10.1088/0741-3335/45/10/006). URL: <http://iopscience.iop.org/0741-3335/45/10/006>.
- [263] Rainer Burhenn, Jürgen Baldzuhn, Rudolf Brakel, Hartmut Ehmler, Louis Giannone, Peter Eckhard Grigull, Jens Knauer, Maciej Krychowiak, Matthias Hirsch, Katsumi Ida, Henning Maassberg, Gerald Kent McCormick, Ekkehard Pasch, Henning Thomsen, Arthur Weller, w7-AS Team, ECRH Group, and NI Group. “Impurity Transport Studies in the Wendelstein 7-AS Stellarator”. In: *Fusion Science and Technology* 46.1 (July 2004), pages 115–128. URL: [http://www.ans.org/pubs/journals/fst/a\\_547](http://www.ans.org/pubs/journals/fst/a_547).
- [264] D. Zhang, R. Dux, R. Burhenn, H. Thomsen, H. Jenzsch, R. König, and T. Sunn Pedersen. “Feasibility Assessment of Bolometry as Impurity Transport Study Tool for the Stellarator w7-x”. In: *41<sup>st</sup> European Physical Society Conference on Plasma Physics*. (Berlin, Germany, June 23–27, 2014). Edited by S. Ratynskaia, P. Mantica, A. Benuzzi-Mounaix, G. Dilecce, R. Bingham, M. Hirsch, B. Kemnitz, T. Klinger, B. Ph. van Milligen, O. Scholten, and P. Helfenstein. Volume 38F. European Conference Abstracts. European Physical Society, 2014, P1.068. URL: <http://ocs.ciemat.es/EPS2014PAP/pdf/P1.068.pdf>.
- [265] T. Barbui, L. Carraro, D. J. Den Hartog, S. T. A. Kumar, and M. Nornberg. “Impurity transport studies in the Madison Symmetric Torus reversed-field pinch during standard and pulsed poloidal current drive regimes”. In: *Plasma Physics and Controlled Fusion* 56.7 (2014), 075012. DOI: [10.1088/0741-3335/56/7/075012](https://doi.org/10.1088/0741-3335/56/7/075012). URL: <http://iopscience.iop.org/0741-3335/56/7/075012>.
- [266] L. Carraro, M. E. Puiatti, F. Sattin, P. Scarin, M. Valisa, and M. Mattioli. “Carbon and oxygen behaviour in the reversed field pinch RFX”. In: *Nuclear*

- Fusion* 36.12 (1996), pages 1623–1632. DOI: [10.1088/0029-5515/36/12/I03](https://doi.org/10.1088/0029-5515/36/12/I03). URL: <http://iopscience.iop.org/0029-5515/36/12/I03>.
- [267] L. Carraro, S. Costa, M. E. Puiatti, F. Sattin, P. Scarin, and M. Valisa. “Reconstruction of the radiation emitted by the intrinsic impurities in the RFX reversed field pinch”. In: *Plasma Physics and Controlled Fusion* 42.6 (2000), pages 731–741. DOI: [10.1088/0741-3335/42/6/308](https://doi.org/10.1088/0741-3335/42/6/308). URL: <http://iopscience.iop.org/0741-3335/42/6/308>.
- [268] L. Carraro, M. E. Puiatti, F. Sattin, P. Scarin, and M. Valisa. “Impurity transport during pulsed poloidal current drive experiment in the reversed field pinch experiment RFX”. In: *Plasma Physics and Controlled Fusion* 44.10 (2002), pages 2135–2148. DOI: [10.1088/0741-3335/44/10/305](https://doi.org/10.1088/0741-3335/44/10/305). URL: <http://iopscience.iop.org/0741-3335/44/10/305>.
- [269] S. Menmuir, L. Carraro, A. Alfier, F. Bonomo, A. Fassina, G. Spizzo, and N. Vianello. “Impurity transport studies in RFX-mod multiple helicity and enhanced confinement QSH regimes”. In: *Plasma Physics and Controlled Fusion* 52.9 (2010), 095001. DOI: [10.1088/0741-3335/52/9/095001](https://doi.org/10.1088/0741-3335/52/9/095001). URL: <http://iopscience.iop.org/0741-3335/52/9/095001>.
- [270] T. Barbui, L. Carraro, P. Franz, P. Innocente, S. Munaretto, and G. Spizzo. “Light impurity transport studies with solid pellet injections in the RFX-mod reversed-field pinch”. In: *Plasma Physics and Controlled Fusion* 57.2 (2014), 025006. DOI: [10.1088/0741-3335/57/2/025006](https://doi.org/10.1088/0741-3335/57/2/025006). URL: <http://iopscience.iop.org/0741-3335/57/2/025006>.
- [271] R. M. Mayo, F. M. Levinton, D. D. Meyerhofer, T. K. Chu, S. F. Paul, and M. Yamada. “Measurement of the local carbon diffusion coefficient in the s-1 spheromak”. In: *Nuclear Fusion* 29.9 (1989), pages 1493–1504. DOI: [10.1088/0029-5515/29/9/006](https://doi.org/10.1088/0029-5515/29/9/006). URL: <http://iopscience.iop.org/0029-5515/29/9/006>.
- [272] G. Cunningham. “Impurities and impurity transport in the spheromak SPHEX”. In: *Plasma Physics and Controlled Fusion* 39.9 (1997), pages 1339–1354. DOI: [10.1088/0741-3335/39/9/006](https://doi.org/10.1088/0741-3335/39/9/006). URL: <http://iopscience.iop.org/0741-3335/39/9/006>.
- [273] J. Bucalossi, G. Martin, R. Dejarnac, A. Géraud, C. Gil, R. Guirlet, J. Gunn, T. Loarer, B. Pégourié, F. Saint-Laurent, and E. Tsitrone. “Plasma Fuelling by Pulsed Supersonic Gas Injection on Tore Supra”. In: *29<sup>th</sup> European Physical Society Conference on Plasma Physics and Controlled Fusion*. (Montreux, Switzerland, June 17–21, 2002). Edited by R. Behn and C. Varandas. Vol-

- ume 26B. European Conference Abstracts. European Physical Society, 2002, O-2.07. URL: [http://epsppd.epfl.ch/Montreux/pdf/02\\_07.pdf](http://epsppd.epfl.ch/Montreux/pdf/02_07.pdf).
- [274] J. F. Friichtenicht. “Laser-generated pulsed atomic beams”. In: *Review of Scientific Instruments* 45.1 (Jan. 1974), pages 51–56. DOI: [10.1063/1.1686447](https://doi.org/10.1063/1.1686447). URL: <http://scitation.aip.org/content/aip/journal/rsi/45/1/10.1063/1.1686447>.
- [275] K. V. Khlopenkov and S. Sudo. “Production and acceleration of tracer encapsulated solid pellets for particle transport diagnostics”. In: *Review of Scientific Instruments* 69.9 (Sept. 1998), pages 3194–3198. DOI: [10.1063/1.1149083](https://doi.org/10.1063/1.1149083). URL: <http://scitation.aip.org/content/aip/journal/rsi/69/9/10.1063/1.1149083>.
- [276] Ralph Dux. *STRAHL User Manual*. Technical report IPP 10/30. Max-Planck-Institut für Plasmaphysik, Sept. 2006. URL: <http://hdl.handle.net/11858/00-001M-0000-0027-0DB8-4>.
- [277] Russell A. Hulse. “Numerical Studies of Impurities in Fusion Plasmas”. In: *Nuclear Technology/Fusion* 3.2 (Mar. 1983), pages 259–272. URL: [http://www.ans.org/pubs/journals/fst/a\\_20849](http://www.ans.org/pubs/journals/fst/a_20849).
- [278] K. W. Gentle, B. Richards, and F. Waelbroeck. “A measurement of hydrogen ion transport parameters in tokamak discharges”. In: *Plasma Physics and Controlled Fusion* 29.9 (1987), pages 1077–1092. DOI: [10.1088/0741-3335/29/9/003](https://doi.org/10.1088/0741-3335/29/9/003). URL: <http://iopscience.iop.org/0741-3335/29/9/003>.
- [279] K. Nagashima, A. Sakasai, and T. Fukuda. “Gas puff modulation experiments on JT-60U”. In: *Nuclear Fusion* 33.11 (1993), pages 1677–1683. DOI: [10.1088/0029-5515/33/11/I08](https://doi.org/10.1088/0029-5515/33/11/I08). URL: <http://iopscience.iop.org/0029-5515/33/11/I08>.
- [280] J. O’Rourke, C. Gowers, G. J. Kramer, P. D. Morgan, R. Simonini, and A. C. C. Sips. “Measurements of the electron source distribution and particle transport coefficients in JET”. In: *Plasma Physics and Controlled Fusion* 35.5 (1993), pages 585–594. DOI: [10.1088/0741-3335/35/5/004](https://doi.org/10.1088/0741-3335/35/5/004). URL: <http://iopscience.iop.org/0741-3335/35/5/004>.
- [281] Ralph Dux. *Impurity Transport in Tokamak Plasmas*. Technical report IPP 10/27. Max-Planck-Institut für Plasmaphysik, Dec. 2004. URL: <http://hdl.handle.net/11858/00-001M-0000-0027-2036-E>.

- [282] Philip R. Bevington and D. Keith Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. 3<sup>rd</sup> edition. 1221 Avenue of the Americas, New York, NY 10020: McGraw-Hill, 2003. ISBN: 978-0072472271. URL: <http://higher.ed.mheducation.com/sites/0072472278/index.html>.
- [283] D. S. Sivia and J. Skilling. *Data Analysis: A Bayesian Tutorial*. 2<sup>nd</sup> edition. Oxford Science Publications. Great Clarendon Street, Oxford OX2 6DP, United Kingdom: Oxford University Press, 2006. ISBN: 978-0198568322. URL: <https://global.oup.com/academic/product/data-analysis-9780198568322>.
- [284] Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability*. 2<sup>nd</sup> edition. Belmont, Massachusetts: Athena Scientific, 2008. ISBN: 978-1886529236. URL: <http://www.athenasc.com/probbook.html>.
- [285] Udo von Toussaint. “Bayesian inference in physics”. In: *Reviews of Modern Physics* 83.3 (July–Sept. 2011), pages 943–999. DOI: [10.1103/RevModPhys.83.943](https://doi.org/10.1103/RevModPhys.83.943). URL: <http://journals.aps.org/rmp/abstract/10.1103/RevModPhys.83.943>.
- [286] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. 3<sup>rd</sup> edition. Texts in Statistical Science. Boca Raton, FL: CRC Press, 2014. ISBN: 978-1439840955. URL: <http://www.stat.columbia.edu/~gelman/book/>.
- [287] George S. Fishman. *Monte Carlo*. Springer Series in Operations Research. New York, NY: Springer Science+Business Media, 1996. ISBN: 978-1-4419-2847-4. DOI: [10.1007/978-1-4757-2553-7](https://doi.org/10.1007/978-1-4757-2553-7). URL: <http://link.springer.com/book/10.1007%2F978-1-4757-2553-7>.
- [288] James E. Gentle. *Random Number Generation and Monte Carlo Methods*. 2<sup>nd</sup> edition. Statistics and Computing. New York, NY: Springer Science+Business Media, 2003. ISBN: 978-0-387-00178-4. DOI: [10.1007/b97336](https://doi.org/10.1007/b97336). URL: <http://link.springer.com/book/10.1007/b97336>.
- [289] Wolfgang Hörmann, Josef Leydold, and Gerhard Derflinger. *Automatic Nonuniform Random Variate Generation*. Statistics and Computing. Berlin: Springer, 2004. ISBN: 978-3-642-07372-4. DOI: [10.1007/978-3-662-05946-3](https://doi.org/10.1007/978-3-662-05946-3). URL: <http://link.springer.com/book/10.1007/978-3-662-05946-3>.
- [290] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. 2<sup>nd</sup> edition. Springer Texts in Statistics. New York, NY: Springer Science+Business Media, 2004. ISBN: 978-1-4419-1939-7. DOI: [10.1007/978-1-4757-4145-2](https://doi.org/10.1007/978-1-4757-4145-2). URL: <http://link.springer.com/book/10.1007%2F978-1-4757-4145-2>.

- [291] Christiane Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer Series in Statistics. New York, NY: Springer Science+Business Media, 2009. ISBN: 978-0387781648. DOI: [10.1007/978-0-387-78165-5](https://doi.org/10.1007/978-0-387-78165-5). URL: <http://link.springer.com/book/10.1007/978-0-387-78165-5>.
- [292] Nick T. Thomopoulos. *Essentials of Monte Carlo Simulation*. New York, NY: Springer Science+Business Media, 2013. ISBN: 978-1-4614-6021-3. DOI: [10.1007/978-1-4614-6022-0](https://doi.org/10.1007/978-1-4614-6022-0). URL: <http://link.springer.com/book/10.1007%2F978-1-4614-6022-0>.
- [293] E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001. URL: <http://www.scipy.org/>.
- [294] *NIST Digital Library of Mathematical Functions*. URL: <http://dlmf.nist.gov/>.
- [295] *NIST Digital Library of Mathematical Functions: 5.2 Definitions*. URL: <http://dlmf.nist.gov/5.2>.
- [296] Y. L. Tong. *The Multivariate Normal Distribution*. Springer Series in Statistics. New York, NY: Springer, 1990. ISBN: 978-1-4613-9657-4. DOI: [10.1007/978-1-4613-9655-0](https://doi.org/10.1007/978-1-4613-9655-0). URL: <http://link.springer.com/book/10.1007/978-1-4613-9655-0>.
- [297] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006. ISBN: 978-0262182539. URL: <http://www.gaussianprocess.org/gpml/>.
- [298] Allan Gut. *An Intermediate Course in Probability*. 2<sup>nd</sup> edition. Springer Texts in Statistics. 223 Spring Street, New York, NY 10013, USA: Springer Science+Business Media, 2009. ISBN: 978-1441901613. DOI: [10.1007/978-1-4419-0162-0](https://doi.org/10.1007/978-1-4419-0162-0). URL: <http://link.springer.com/book/10.1007/978-1-4419-0162-0>.
- [299] A. O'Hagan. "Curve Fitting and Optimal Design for Prediction". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 40.1 (1978), pages 1–42. URL: <http://www.jstor.org/stable/2984861>.
- [300] David J. C. MacKay. "Introduction to Gaussian Processes". In: *Neural Networks and Machine Learning*. Edited by C. M. Bishop. NATO ASI Series. Kluwer Academic Press, 1998, pages 133–166. URL: <http://www.inference.phy.cam.ac.uk/mackay/gpB.pdf>.

- [301] Michael L. Stein. *Interpolation of Spatial Data*. Springer Series in Statistics. New York, NY: Springer, 1999. ISBN: 978-1-4612-7166-6. DOI: [10.1007/978-1-4612-1494-6](https://doi.org/10.1007/978-1-4612-1494-6). URL: <http://link.springer.com/book/10.1007/978-1-4612-1494-6>.
- [302] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. 3<sup>rd</sup> edition. Cambridge: Cambridge University Press, 2007. ISBN: 978-0521880688. URL: <http://numerical.recipes/>.
- [303] David Barber. *Bayesian Reasoning and Machine Learning*. The Edinburgh Building, Cambridge CB2 8RU, UK: Cambridge University Press, 2012. ISBN: 978-0521518147. URL: <http://www.cs.ucl.ac.uk/staff/d.barber/brml/>.
- [304] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. ISBN: 978-0262018029. URL: <https://mitpress.mit.edu/books/machine-learning-0>.
- [305] M. A. Chilenski, M. Greenwald, Y. Marzouk, N. T. Howard, A. E. White, J. E. Rice, and J. R. Walk. “Improved profile fitting and quantification of uncertainty in experimental measurements of impurity transport coefficients using Gaussian process regression”. In: *Nuclear Fusion* 55.2 (2015), 023012. DOI: [10.1088/0029-5515/55/2/023012](https://doi.org/10.1088/0029-5515/55/2/023012). URL: <http://iopscience.iop.org/0029-5515/55/2/023012>.
- [306] M. A. Chilenski. *gptools: Gaussian processes with arbitrary derivative constraints and predictions*. 2015. URL: <https://github.com/markchil/gptools>.
- [307] M. A. Chilenski. *gptools: Gaussian processes with arbitrary derivative constraints and predictions (online documentation)*. 2015. URL: <http://gptools.readthedocs.org/>.
- [308] M. A. Chilenski. *profiletools: Classes for working with profile data of arbitrary dimension*. 2015. URL: <https://github.com/markchil/profiletools>.
- [309] M. A. Chilenski. *profiletools: Classes for working with profile data of arbitrary dimension (online documentation)*. 2015. URL: <http://profiletools.readthedocs.org/>.
- [310] Grace Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. 3600 University City Science Center, Philadelphia, PA 19104-2688: Society for Industrial and Applied Mathematics, 1990. ISBN: 978-0-89871-244-5. DOI: [10.1137/1.9781611970128](https://doi.org/10.1137/1.9781611970128). URL: <http://epubs.siam.org/doi/book/10.1137/1.9781611970128>.



- [311] Paul Dierckx. *Curve and Surface Fitting with Splines*. Numerical Mathematics and Scientific Computation. Great Clarendon Street, Oxford OX2 6DP, United Kingdom: Oxford University Press, 1993. ISBN: 978-0198534402. URL: <https://global.oup.com/academic/product/curve-and-surface-fitting-with-splines-9780198534402>.
- [312] G. T. von Nessi, M. J. Hole, J. Svensson, and L. Appel. “Evidence cross-validation and Bayesian inference of MAST plasma equilibria”. In: *Physics of Plasmas* 19.1 (2012), 012506. DOI: [10.1063/1.3677362](https://doi.org/10.1063/1.3677362). URL: <http://scitation.aip.org/content/aip/journal/pop/19/1/10.1063/1.3677362>.
- [313] Dong Li, J. Svensson, H. Thomsen, F. Medina, A. Werner, and R. Wolf. “Bayesian soft X-ray tomography using non-stationary Gaussian Processes”. In: *Review of Scientific Instruments* 84.8 (2013), 083506. DOI: [10.1063/1.4817591](https://doi.org/10.1063/1.4817591). URL: <http://scitation.aip.org/content/aip/journal/rsi/84/8/10.1063/1.4817591>.
- [314] J. A. Romero and J. Svensson. “Optimization of out-vessel magnetic diagnostics for plasma boundary reconstruction in tokamaks”. In: *Nuclear Fusion* 53.3 (2013), 033009. DOI: [10.1088/0029-5515/53/3/033009](https://doi.org/10.1088/0029-5515/53/3/033009). URL: <http://iopscience.iop.org/0029-5515/53/3/033009>.
- [315] G. T. von Nessi, M. J. Hole, and the MAST Team. “A unified method for inference of tokamak equilibria and validation of force-balance models based on Bayesian analysis”. In: *Journal of Physics A: Mathematical and Theoretical* 46.18 (2013), 185501. DOI: [10.1088/1751-8113/46/18/185501](https://doi.org/10.1088/1751-8113/46/18/185501). URL: <http://iopscience.iop.org/1751-8121/46/18/185501>.
- [316] G. T. von Nessi and M. J. Hole. “Using Bayesian analysis and Gaussian processes to infer electron temperature and density profiles on the Mega-Ampere Spherical Tokamak experiment”. In: *Review of Scientific Instruments* 84.6 (2013), 063505. DOI: [10.1063/1.4811378](https://doi.org/10.1063/1.4811378). URL: <http://scitation.aip.org/content/aip/journal/rsi/84/6/10.1063/1.4811378>.
- [317] A. Langenberg, H. Thomsen, R. Burhenn, O. Marchuk, J. Svensson, T. S. Pedersen, and R. C. Wolf. “Forward Modeling of a High Resolution X-ray Imaging Crystal Spectrometer for the Wendelstein 7-x Stellarator”. In: *41<sup>st</sup> European Physical Society Conference on Plasma Physics*. (Berlin, Germany, June 23–27, 2014). Edited by S. Ratynskaia, P. Mantica, A. Benuzzi-Mounaix, G. Dilecce, R. Bingham, M. Hirsch, B. Kemnitz, T. Klinger, B. Ph. van Milligen, O. Scholten, and P. Helfenstein. Volume 38F. European Conference Ab-



- stracts. European Physical Society, 2014, P1.074. URL: <http://ocs.ciemat.es/EPS2014PAP/pdf/P1.074.pdf>.
- [318] Sehyun Kwak, J. Svensson, M. Brix, Y.-c. Ghim, and JET Contributors. *Bayesian modelling of the emission spectrum of the JET Li-BES system*. Oct. 2015. arXiv: [1510.01189](https://arxiv.org/abs/1510.01189). URL: <http://arxiv.org/abs/1510.01189>.
- [319] Otto J. W. F. Kardaun. *Classical Methods of Statistics*. Berlin: Springer, 2005. ISBN: 978-3642439797. DOI: [10.1007/3-540-29288-8](https://doi.org/10.1007/3-540-29288-8). URL: <http://link.springer.com/book/10.1007/3-540-29288-8>.
- [320] A. E. White, L. Schmitz, W. A. Peebles, T. L. Rhodes, T. A. Carter, G. R. McKee, M. W. Shafer, G. M. Staebler, K. H. Burrell, J. C. DeBoo, and R. Prater. “Simultaneous measurement of core electron temperature and density fluctuations during electron cyclotron heating on DIII-D”. In: *Physics of Plasmas* 17.2 (2010), 020701. DOI: [10.1063/1.3318469](https://doi.org/10.1063/1.3318469). URL: <http://scitation.aip.org/content/aip/journal/pop/17/2/10.1063/1.3318469>.
- [321] Carl de Boor. *A Practical Guide to Splines*. 2<sup>nd</sup> edition. Applied Mathematical Sciences 27. New York, NY: Springer, 2001. ISBN: 978-0387903569. URL: <https://www.springer.com/us/book/9780387953663>.
- [322] Larry L. Schumaker. *Spline Functions: Basic Theory*. 3<sup>rd</sup> edition. The Edinburgh Building, Cambridge CB2 8RU, UK: Cambridge University Press, 2007. ISBN: 978-0521705127. URL: <http://www.cambridge.org/us/academic/subjects/mathematics/numerical-analysis/spline-functions-basic-theory-3rd-edition>.
- [323] David L. B. Jupp. “Approximation to Data by Splines with Free Knots”. In: *SIAM Journal on Numerical Analysis* 15.2 (1978), pages 328–343. DOI: [10.1137/0715022](https://doi.org/10.1137/0715022). URL: <http://epubs.siam.org/doi/abs/10.1137/0715022>.
- [324] Thomas C. M. Lee. “Smoothing parameter selection for smoothing splines: a simulation study”. In: *Computational Statistics and Data Analysis* 42.1-2 (Feb. 2003), pages 139–148. DOI: [10.1016/S0167-9473\(02\)00159-7](https://doi.org/10.1016/S0167-9473(02)00159-7). URL: <http://www.sciencedirect.com/science/article/pii/S0167947302001597>.
- [325] David L. B. Jupp. “The ‘Lethargy’ Theorem – A Property of Approximation by  $\gamma$ -Polynomials”. In: *Journal of Approximation Theory* 14.3 (July 1975), pages 204–217. DOI: [10.1016/0021-9045\(75\)90056-8](https://doi.org/10.1016/0021-9045(75)90056-8). URL: <http://www.sciencedirect.com/science/article/pii/0021904575900568>.

- [326] Grace Wahba. “Bayesian ‘Confidence Intervals’ for the Cross-validated Smoothing Spline”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 45.1 (1983), pages 133–150. URL: <https://www.jstor.org/stable/2345632>.
- [327] Douglas Nychka. “Bayesian Confidence Intervals for Smoothing Splines”. In: *Journal of the American Statistical Association* 83.404 (Dec. 1988), pages 1134–1143. DOI: [10.2307/2290146](https://doi.org/10.2307/2290146). URL: <http://www.jstor.org/stable/2290146>.
- [328] Ker-Chau Li. “Honest Confidence Regions for Nonparametric Regression”. In: *The Annals of Statistics* 17.3 (1989), pages 1001–1008. URL: <http://www.jstor.org/stable/2241707>.
- [329] Yuedong Wang and Grace Wahba. “Bootstrap confidence intervals for smoothing splines and their comparison to Bayesian confidence intervals”. In: *Journal of Statistical Computation and Simulation* 51.2-4 (1995), pages 263–279. DOI: [10.1080/00949659508811637](https://doi.org/10.1080/00949659508811637). URL: <http://www.tandfonline.com/doi/abs/10.1080/00949659508811637>.
- [330] S. Zhou, X. Shen, and D. A. Wolfe. “Local asymptotics for regression splines and confidence regions”. In: *The Annals of Statistics* 26.5 (1998), pages 1760–1782. DOI: [10.1214/aos/1024691356](https://doi.org/10.1214/aos/1024691356). URL: <https://projecteuclid.org/euclid.aos/1024691356>.
- [331] Wenxin Mao and Linda H. Zhao. “Free-knot polynomial splines with confidence intervals”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.4 (2003), pages 901–919. DOI: [10.1046/j.1369-7412.2003.00422.x](https://doi.org/10.1046/j.1369-7412.2003.00422.x). URL: <http://onlinelibrary.wiley.com/doi/10.1046/j.1369-7412.2003.00422.x/abstract>.
- [332] Shanggang Zhou and Douglas A. Wolfe. “On derivative estimation in spline regression”. In: *Statistica Sinica* 10 (2000), pages 93–108. URL: <http://www3.stat.sinica.edu.tw/statistica/j10n1/j10n14/j10n14.htm>.
- [333] Gengsheng Qin and Min Tsao. “Empirical likelihood based inference for the derivative of the nonparametric regression function”. In: *Bernoulli* 11.4 (2005), pages 715–735. URL: <http://www.jstor.org/stable/3318892>.
- [334] Guanqun Cao, Jing Wang, Li Wang, and David Todem. “Spline confidence bands for functional derivatives”. In: *Journal of Statistical Planning and Inference* 142.6 (June 2012), pages 1557–1570. DOI: [10.1016/j.jspi.2012.01.009](https://doi.org/10.1016/j.jspi.2012.01.009). URL: <http://www.sciencedirect.com/science/article/pii/S0378375812000183>.

- [335] V. Dose and R. Fischer. “Function Estimation Employing Exponential Splines”. In: *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 25<sup>th</sup> International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. (San Jose, CA, USA, Aug. 7–12, 2005). Edited by Kevin H. Knuth, Ali E. Abbas, Robin D. Morris, and J. Patrick Castle. Volume 803. AIP Conference Proceedings. American Institute of Physics. AIP Publishing, 2005, pages 67–71. DOI: [10.1063/1.2149780](https://doi.org/10.1063/1.2149780). URL: <http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.2149780>.
- [336] R. Fischer and V. Dose. “Flexible and reliable profile estimation using exponential splines”. In: *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. (Paris, France, July 8–13, 2006). Edited by Ali Mohammad-Djafari. Volume 872. AIP Conference Proceedings. AIP Publishing, 2006, pages 296–303. DOI: [10.1063/1.2423287](https://doi.org/10.1063/1.2423287). URL: <http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.2423287>.
- [337] R. Fischer, A. Dinklage, and Y. Turkin. “Non-parametric profile gradient estimation”. In: *33<sup>rd</sup> European Physical Society Conference on Plasma Physics*. (Rome, Italy, June 19–23, 2006). Edited by F. De Marco and G. Vlad. Volume 301. European Conference Abstracts. European Physical Society, 2006, p–2.127. URL: [http://epsppd.epfl.ch/Roma/pdf/P2\\_127.pdf](http://epsppd.epfl.ch/Roma/pdf/P2_127.pdf).
- [338] O. P. Le Maître and O. M. Knio. *Spectral Methods for Uncertainty Quantification*. Scientific Computation. Springer Science+Business Media, 2010. ISBN: 978-9048135196. DOI: [10.1007/978-90-481-3520-2](https://doi.org/10.1007/978-90-481-3520-2). URL: <http://link.springer.com/book/10.1007/978-90-481-3520-2>.
- [339] G. Matheron. “The Intrinsic Random Functions and Their Applications”. In: *Advances in Applied Probability* 5.3 (Dec. 1973), pages 439–438. DOI: [10.2307/1425829](https://doi.org/10.2307/1425829). URL: <http://www.jstor.org/stable/1425829>.
- [340] Sébastien Da Veiga and Amandine Marrel. “Gaussian process modeling with inequality constraints”. In: *Annales de la Faculté des Sciences de Toulouse* 21.3 (2012), pages 529–555. DOI: [10.5802/afst.1344](https://doi.org/10.5802/afst.1344). URL: [http://afst.cedram.org/item?id=AFST\\_2012\\_6\\_21\\_3\\_529\\_0](http://afst.cedram.org/item?id=AFST_2012_6_21_3_529_0).
- [341] E. Solak, R. Murray-Smith, W.E. Leithead, D.J. Leith, and C.E. Rasmussen. “Derivative observations in Gaussian Process models of dynamic systems”. In: *Conference on Neural Information Processing Systems*. (Vancouver, Canada, Dec. 9–14, 2002). Edited by S. Becker, S. Thrun, and K. Obermayer. Advances

- in neural information processing systems 15. MIT Press, 2003. URL: <http://eprints.gla.ac.uk/3119/>.
- [342] Emmanuel Vazquez and Eric Walter. “Estimating derivatives and integrals with Kriging”. In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC ’05. 44<sup>th</sup> IEEE Conference on*. IEEE, Dec. 2005, pages 8156–8161. DOI: [10.1109/CDC.2005.1583482](https://doi.org/10.1109/CDC.2005.1583482). URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1583482](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1583482).
- [343] Phillip Boyle. “Gaussian Processes for Regression and Optimisation”. PhD thesis. Victoria University of Wellington, 2007. URL: <http://hdl.handle.net/10063/421>.
- [344] Brian A. Lockwood and Mihai Anitescu. “Gradient-Enhanced Universal Kriging for Uncertainty Propagation”. In: *Nuclear Science and Engineering* 170.2 (Feb. 2012), pages 168–195. DOI: [10.13182/NSE10-86](https://doi.org/10.13182/NSE10-86). URL: <http://epubs.ans.org/?a=13359>.
- [345] Roderick Murray-Smith and Barak A. Pearlmutter. “Transformations of Gaussian Process Priors”. In: *Deterministic and Statistical Methods in Machine Learning*. Edited by J. Winkler, N. Lawrence, and M. Niranjan. Volume 3635. Lecture Notes in Computer Science. Springer, 2005, pages 110–123. DOI: [10.1007/11559887\\_7](https://doi.org/10.1007/11559887_7). URL: [http://link.springer.com/chapter/10.1007%2F11559887\\_7](http://link.springer.com/chapter/10.1007%2F11559887_7).
- [346] Jeffrey P. Freidberg. *Functional constraints on plasma profiles*. Personal communication. Aug. 2014.
- [347] Eric W. Weisstein. *MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/>.
- [348] Eric W. Weisstein. “Pappus’s Centroid Theorem.” *From MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/PappusCentroidTheorem.html>.
- [349] Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. “Hybrid Monte Carlo”. In: *Physics Letters B* 195.2 (Sept. 1987), pages 216–222. DOI: [10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X). URL: <http://www.sciencedirect.com/science/article/pii/037026938791197X>.
- [350] Radford M. Neal. “An Improved Acceptance Procedure for the Hybrid Monte Carlo Algorithm”. In: *Journal of Computational Physics* 111.1 (Mar. 1994), pages 194–203. DOI: [10.1006/jcph.1994.1054](https://doi.org/10.1006/jcph.1994.1054). URL: <http://www.sciencedirect.com/science/article/pii/S0021999184710540>.

- [351] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (June 1953), pages 1087–1092. DOI: [10.1063/1.1699114](https://doi.org/10.1063/1.1699114). URL: <http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114>.
- [352] W. K. Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1 (1970), pages 97–109. DOI: [10.2307/2334940](https://doi.org/10.2307/2334940). URL: <http://www.jstor.org/stable/2334940>.
- [353] Jonathan Goodman and Jonathan Weare. “Ensemble samplers with affine invariance”. In: *Communications in Applied Mathematics and Computational Science* 5.1 (2010), pages 65–80. DOI: [10.2140/camcos.2010.5.65](https://doi.org/10.2140/camcos.2010.5.65). URL: <http://msp.org/camcos/2010/5-1/p04.xhtml>.
- [354] Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. “emcee: The MCMC Hammer”. In: *Publications of the Astronomical Society of the Pacific* 125.925 (Mar. 2013), pages 306–312. DOI: [10.1086/670067](https://doi.org/10.1086/670067). URL: <http://www.jstor.org/stable/10.1086/670067>.
- [355] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (1979), pages 239–245. DOI: [10.2307/1268522](https://doi.org/10.2307/1268522). URL: <http://www.jstor.org/stable/1268522>.
- [356] J. H. Irby, E. S. Marmor, E. Sevillano, and S. M. Wolfe. “Two-color interferometer system for Alcator C-Mod”. In: *Review of Scientific Instruments* 59.8 (1988), pages 1568–1570. DOI: [10.1063/1.1140199](https://doi.org/10.1063/1.1140199). URL: <http://scitation.aip.org/content/aip/journal/rsi/59/8/10.1063/1.1140199>.
- [357] Thomas Cho-Tak Luke. “Measurement of particle transport coefficients on Alcator C-Mod”. PhD thesis. Massachusetts Institute of Technology, Oct. 1994. URL: <http://hdl.handle.net/1721.1/11755>.
- [358] C. P. Kasten, J. H. Irby, R. Murray, A. E. White, and D. C. Pace. “A new interferometry-based electron density fluctuation diagnostic on Alcator C-Mod”. In: *Review of Scientific Instruments* 83.10 (2012), 10E301. DOI: [10.1063/1.4728090](https://doi.org/10.1063/1.4728090). URL: <http://scitation.aip.org/content/aip/journal/rsi/83/10/10.1063/1.4728090>.
- [359] Cale Phillip Kasten. “Two-Color interferometry as a fluctuation diagnostic on Alcator C-Mod”. Master’s thesis. Cambridge, MA: Massachusetts Institute of Technology, June 2013. URL: <http://hdl.handle.net/1721.1/82457>.

- [360] A. Ince-Cushman, J. E. Rice, M. Bitter, M. L. Reinke, K. W. Hill, M. F. Gu, E. Eikenberry, Ch. Broennimann, S. Scott, Y. Podpaly, S. G. Lee, and E. S. Marmor. “Spatially resolved high resolution x-ray spectroscopy for magnetically confined fusion plasmas (invited)”. In: *Review of Scientific Instruments* 79.10 (2008), 10E302. DOI: [10.1063/1.2968707](https://doi.org/10.1063/1.2968707). URL: <http://scitation.aip.org/content/aip/journal/rsi/79/10/10.1063/1.2968707>.
- [361] R.S. Granetz and L. Wang. “Design of the X-ray tomography system on Alcator C-Mod”. In: *International School of Plasma Physics. Diagnostics for Contemporary Fusion Experiments*. (Villa Monastero, Aug. 27–Sept. 6, 1991). Edited by P. E. Stott, D. K. Akulina, G. Gorini, and E. Sindoni. Editrice Compositori Bologna, 1991, pages 425–437.
- [362] Ian C. Faust. *Toroidal Radiation Inversion Protocol (Python)*. URL: <https://github.com/icfaust/TRIPPy>.
- [363] M. Valovič, R. Budny, L. Garzotti, X. Garbet, A. A. Korotkov, J. Rapp, R. Neu, O. Sauter, P. de Vries, B. Alper, M. Beurskens, J. Brzozowski, D. McDonald, H. Leggate, C. Giroud, V. Parail, I. Voitsekhovitch, and JET EFDA contributors. “Density peaking in low collisionality ELMY H-mode in JET”. In: *Plasma Physics and Controlled Fusion* 46.12 (2004), pages 1877–1889. DOI: [10.1088/0741-3335/46/12/006](https://doi.org/10.1088/0741-3335/46/12/006). URL: <http://iopscience.iop.org/0741-3335/46/12/006>.
- [364] H. Weisen, A. Zabolotsky, C. Angioni, I. Furno, X. Garbet, C. Giroud, H. Leggate, P. Mantica, D. Mazon, J. Weiland, L. Zabeo, K.-D. Zastrow, and JET-EFDA contributors. “Collisionality and shear dependences of density peaking in JET and extrapolation to ITER”. In: *Nuclear Fusion* 45.2 (2005), L1–L4. DOI: [10.1088/0029-5515/45/2/L01](https://doi.org/10.1088/0029-5515/45/2/L01). URL: <http://iopscience.iop.org/0029-5515/45/2/L01>.
- [365] H. Weisen, A. Zabolotsky, M. Maslov, M. Beurskens, C. Giroud, D. Mazon, and JET-EFDA contributors. “Scaling of density peaking in JET H-modes and implications for ITER”. In: *Plasma Physics and Controlled Fusion* 48.5A (2006), A457–A466. DOI: [10.1088/0741-3335/48/5A/S47](https://doi.org/10.1088/0741-3335/48/5A/S47). URL: <http://iopscience.iop.org/0741-3335/48/5A/S47>.
- [366] C. Angioni, H. Weisen, O. J. W. F. Kardaun, M. Maslov, A. Zabolotsky, C. Fuchs, L. Garzotti, C. Giroud, B. Kurzan, P. Mantica, A. G. Peeters, J. Stober, the ASDEX Upgrade Team, and contributors to the EFDA-JET Workprogramme. “Scaling of density peaking in H-mode plasmas based on a combined database of AUG and JET observations”. In: *Nuclear Fusion* 47.9 (2007),

- pages 1326–1335. DOI: [10.1088/0029-5515/47/9/033](https://doi.org/10.1088/0029-5515/47/9/033). URL: <http://iopscience.iop.org/0029-5515/47/9/033>.
- [367] M. Greenwald, C. Angioni, J. W. Hughes, J. Terry, and H. Weisen. “Density profile peaking in low collisionality H-modes: comparison of Alcator C-Mod data to ASDEX Upgrade/JET scalings”. In: *Nuclear Fusion* 47.9 (2007), L26–L29. DOI: [10.1088/0029-5515/47/9/L03](https://doi.org/10.1088/0029-5515/47/9/L03). URL: <http://iopscience.iop.org/0029-5515/47/9/L03>.
- [368] M. Valovič, L. Garzotti, I. Voitsekhovitch, M. Beurskens, X. Garbet, C. Giroud, D. Keeling, and JET EFDA contributors. “On the correlation between density profile and particle flux in H-mode tokamak plasmas and the implication for ITER”. In: *Nuclear Fusion* 47.3 (2007), pages 196–200. DOI: [10.1088/0029-5515/47/3/005](https://doi.org/10.1088/0029-5515/47/3/005). URL: <http://iopscience.iop.org/0029-5515/47/3/005>.
- [369] A. Dinklage, R. Fischer, G. Kühner, H. Maaßberg, E. Pasch, and J. Svensson. “Steps towards an integrated data analysis: Basic concepts and Bayesian analysis of Thomson scattering data”. In: *29<sup>th</sup> European Physical Society Conference on Plasma Physics and Controlled Fusion*. (Montreux, Switzerland, June 17–21, 2002). Edited by R. Behn and C. Varandas. Volume 26B. European Conference Abstracts. European Physical Society, 2002, P–5.103. URL: [http://epsppd.epfl.ch/Montreux/pdf/P5\\_103.pdf](http://epsppd.epfl.ch/Montreux/pdf/P5_103.pdf).
- [370] R. Fischer, C. Wendland, A. Dinklage, S. Gori, V. Dose, and the W7-AS team. “Thomson scattering analysis with the Bayesian probability theory”. In: *Plasma Physics and Controlled Fusion* 44.8 (2002), pages 1501–1519. DOI: [10.1088/0741-3335/44/8/306](https://doi.org/10.1088/0741-3335/44/8/306). URL: <http://iopscience.iop.org/0741-3335/44/8/306>.
- [371] A. Dinklage, R. Fischer, J. Geiger, S. Gori, S. Kühner, H. Maaßberg, H. Svensson, and Udo von Toussaint. “From Off-line to Real-time Analysis: Accelerating Bayesian Analysis Codes”. In: *30<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (St. Petersburg, Russia, July 7–11, 2003). Edited by R. Koch and S. Lebedev. Volume 27A. European Conference Abstracts. European Physical Society, 2003, P–4.80. URL: [http://epsppd.epfl.ch/StPetersburg/PDF/P4\\_080.PDF](http://epsppd.epfl.ch/StPetersburg/PDF/P4_080.PDF).
- [372] A. Dinklage, R. Fischer, and J. Svensson. “Bayesian Methods for Integrated Data Analysis”. In: *International Conference on Research and Applications of Plasmas (PLASMA-2003)*. (Warsaw, Poland, Sept. 9–12, 2003). 2003. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.4094&rep=rep1&type=pdf>.



- [373] R. Fischer, A. Dinklage, and E. Pasch. “Bayesian modelling of fusion diagnostics”. In: *Plasma Physics and Controlled Fusion* 45.7 (2003), pages 1095–1111. DOI: [10.1088/0741-3335/45/7/304](https://doi.org/10.1088/0741-3335/45/7/304). URL: <http://iopscience.iop.org/0741-3335/45/7/304>.
- [374] J. Svensson, A. Dinklage, J. Geiger, and R. Fischer. “An Integrated Data Analysis Model for the w7-AS Stellarator”. In: *30<sup>th</sup> European Physical Society Conference on Controlled Fusion and Plasma Physics*. (St. Petersburg, Russia, July 7–11, 2003). Edited by R. Koch and S. Lebedev. Volume 27A. European Conference Abstracts. European Physical Society, 2003, P-1.065. URL: [http://epsppd.epfl.ch/StPetersburg/PDF/P1\\_065.PDF](http://epsppd.epfl.ch/StPetersburg/PDF/P1_065.PDF).
- [375] R. Fischer and A. Dinklage. “Integrated data analysis of fusion diagnostics by means of the Bayesian probability theory”. In: *Review of Scientific Instruments* 75.10 (2004), pages 4237–4239. DOI: [10.1063/1.1787607](https://doi.org/10.1063/1.1787607). URL: <http://scitation.aip.org/content/aip/journal/rsi/75/10/10.1063/1.1787607>.
- [376] A. Dinklage, R. Fischer, H. Dreier, J. Svensson, and Yu. Turkin. “Integrated Approaches in Fusion Data Analysis”. In: *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 24<sup>th</sup> International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. (Garching, Germany, July 25–30, 2004). Edited by Rainer Fischer, Roland Preuss, and Udo von Toussaint. Volume 735. AIP Conference Proceedings. American Institute of Physics, Nov. 2004, pages 43–51. DOI: [10.1063/1.1835196](https://doi.org/10.1063/1.1835196). URL: <http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.1835196>.
- [377] A. Dinklage, C. D. Beidler, R. Fischer, H. Maaßberg, J. Svensson, and Yu. A. Turkin. “Integrated Interpretive Transport Modelling”. In: *31<sup>st</sup> European Physical Society Conference on Plasma Physics*. (London, UK, June 28–July 2, 2004). Edited by P. Norreys and H. Hutchinson. Volume 28G. European Conference Abstracts. European Physical Society, 2004, O-2.03. URL: [http://epsppd.epfl.ch/London/pdf/02\\_03.pdf](http://epsppd.epfl.ch/London/pdf/02_03.pdf).
- [378] Andreas Dinklage, Rainer Fischer, and Jakob Svensson. “Topics and methods for data validation by means of Bayesian probability theory”. In: *Fusion Science and Technology* 46.2 (Sept. 2004), pages 355–364. URL: [http://www.ans.org/pubs/journals/fst/a\\_575](http://www.ans.org/pubs/journals/fst/a_575).
- [379] J. Svensson and R. W. T. König. “Bayesian Modelling of Spectrometer Systems”. In: *32<sup>nd</sup> European Physical Society Conference on Plasma Physics and*



- Controlled Fusion combined with the 8<sup>th</sup> International Workshop on Fast Ignition of Fusion Targets*. (Tarragona, Spain, June 26–July 1, 2005). Edited by C Hidalgo and B. Ph. van Milligen. Volume 29C. European Conference Abstracts. European Physical Society, 2005, P–5.087. URL: [http://epsppd.epfl.ch/Tarragona/pdf/P5\\_087.pdf](http://epsppd.epfl.ch/Tarragona/pdf/P5_087.pdf).
- [380] G. Verdoolaege, M. G. Von Hellermann, R. Jaspers, M. M. Ichir, and G. Van Oost. “Integrated Bayesian Estimation of  $Z_{\text{eff}}$  in the TEXTOR Tokamak from Bremsstrahlung and CX Impurity Density Measurements”. In: *Twenty sixth International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, CNRS. (Paris, France, July 8–13, 2006). 2006, 71. URL: [http://djafari.free.fr/maxent2006/Finals/071\\_Verdoolaege.pdf](http://djafari.free.fr/maxent2006/Finals/071_Verdoolaege.pdf).
- [381] R. Fischer and A. Dinklage. “The concept of Integrated Data Analysis of complementary experiments”. In: *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. (Saratoga Springs, NY, USA, July 8–13, 2007). Edited by Kevin H. Knuth, Ariel Caticha, Julian L. Center, Adom Griffin, and Carlos C. Rodríguez. Volume 954. AIP Conference Proceedings. American Institute of Physics, 2007, pages 195–202. DOI: [10.1063/1.2821262](https://doi.org/10.1063/1.2821262). URL: <http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.2821262>.
- [382] J. Svensson and A. Werner. “Large Scale Bayesian Data Analysis for Nuclear Fusion Experiments”. In: *IEEE International Symposium on Intelligent Signal Processing*. (Alcala de Henares, Spain, Oct. 3–5, 2007). IEEE, 2007. DOI: [10.1109/WISP.2007.4447579](https://doi.org/10.1109/WISP.2007.4447579). URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4447579>.
- [383] S. A. Arshad, J. G. Cordey, D. C. McDonald, J. Farthing, E. Joffrin, M. von Hellermann, C. M. Roach, and J. Svensson. “Data validation, analysis, and applications for fusion plasmas”. In: *Fusion Science and Technology* 53.2 (Feb. 2008), pages 667–698. URL: [http://www.ans.org/pubs/journals/fst/a\\_1683](http://www.ans.org/pubs/journals/fst/a_1683).
- [384] Andreas Dinklage, Helko Dreier, Rainer Fischer, Silvio Gori, Roland Preuss, and Udo von Toussaint. “Integrated Data Analysis for Fusion: A Bayesian Tutorial for Fusion Diagnosticians”. In: *Burning Plasma Diagnostics: An International Conference*. (Varenna, Italy, Sept. 24–28, 2007). Edited by Giuseppe Gorini, Francesco P. Orsitto, Elio Sindoni, and Marco Tardocchi. Volume 988. AIP Conference Proceedings. American Institute of Physics, Mar. 2008, pages 471–480. DOI: [10.1063/1.2905117](https://doi.org/10.1063/1.2905117).

- URL: <http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.2905117>.
- [385] Dirk Dodt, Andreas Dinklage, Rainer Fischer, Klaus Bartschat, Oleg Zatsarinny, and Detlef Loffhagen. “Reconstruction of an electron energy distribution function using integrated data analysis”. In: *Journal of Physics D: Applied Physics* 41.20 (2008), 205207. DOI: [10.1088/0022-3727/41/20/205207](https://doi.org/10.1088/0022-3727/41/20/205207). URL: <http://iopscience.iop.org/0022-3727/41/20/205207>.
- [386] R. Fischer, E. Wolfrum, Ch. Fuchs, and ASDEX Upgrade Team. “Integrated density profile analysis in ASDEX Upgrade H-modes”. In: *35<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Hersonissos, Greece, June 9–13, 2008). Edited by P. Lalouis and S. Moustazis. Volume 32D. European Conference Abstracts. European Physical Society, 2008, P-4.010. URL: [http://epsppd.epfl.ch/Hersonissos/pdf/P4\\_010.pdf](http://epsppd.epfl.ch/Hersonissos/pdf/P4_010.pdf).
- [387] J. Svensson, A. Werner, and JET-EFDA Contributors. “Current tomography for axisymmetric plasmas”. In: *Plasma Physics and Controlled Fusion* 50.8 (2008), 085002. DOI: [10.1088/0741-3335/50/8/085002](https://doi.org/10.1088/0741-3335/50/8/085002). URL: <http://iopscience.iop.org/0741-3335/50/8/085002>.
- [388] R. Fischer, A. Burckhart, N. Hicks, B. Kurzan, E. Wolfrum, and ASDEX Upgrade Team. “Multiple diagnostic data analysis of density and temperature profiles in ASDEX Upgrade”. In: *36<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Sofia, Bulgaria, June 29–July 3, 2009). Edited by M. Matsev and E. Benova. Volume 33E. European Conference Abstracts. European Physical Society, 2009, P-1.159. URL: [http://epsppd.epfl.ch/Sofia/pdf/P1\\_159.pdf](http://epsppd.epfl.ch/Sofia/pdf/P1_159.pdf).
- [389] R. Fischer, C. J. Fuchs, B. Kurzan, W. Suttrop, E. Wolfrum, and ASDEX Upgrade Team. “Integrated data analysis of profile diagnostics at ASDEX Upgrade”. In: *Fusion Science and Technology* 58.2 (Oct. 2010), pages 675–684. DOI: [10.13182/FST10-110](https://doi.org/10.13182/FST10-110). URL: [http://www.ans.org/pubs/journals/fst/a\\_10892](http://www.ans.org/pubs/journals/fst/a_10892).
- [390] S. K. Rathgeber, R. Fischer, S. Fietz, J. Hobirk, A. Kallenbach, H. Meister, T. Pütterich, F. Ryter, G. Tardini, E. Wolfrum, and ASDEX Upgrade Team. “Estimation of profiles of the effective ion charge at ASDEX Upgrade with Integrated Data Analysis”. In: *Plasma Physics and Controlled Fusion* 52.9 (2010), 095008. DOI: [10.1088/0741-3335/52/9/095008](https://doi.org/10.1088/0741-3335/52/9/095008). URL: <http://iopscience.iop.org/0741-3335/52/9/095008>.

- [391] Geert Verdoolaege, Rainer Fischer, Guido Van Oost, and JET-EFDA Contributors. “Potential of a Bayesian Integrated Determination of the Ion Effective Charge via Bremsstrahlung and Charge Exchange Spectroscopy in Tokamak Plasmas”. In: *IEEE Transactions on Plasma Science* 38.11 (Nov. 2010), pages 3168–3196. DOI: [10.1109/TPS.2010.2071884](https://doi.org/10.1109/TPS.2010.2071884). URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5585786>.
- [392] G. Verdoolaege, R. Fischer, and G. Van Oost. “Integrated analysis and consistency measurement of bremsstrahlung and charge exchange spectroscopy data for the determination of the ion effective charge”. In: *Review of Scientific Instruments* 81.10 (2010), 10D703. DOI: [10.1063/1.3464466](https://doi.org/10.1063/1.3464466). URL: <http://scitation.aip.org/content/aip/journal/rsi/81/10/10.1063/1.3464466>.
- [393] S. Schmuck, J. Svensson, E. de la Luna, L. Figini, T. Johnson, B. Alper, M. Beurskens, J. Fessey, T. Gerbaud, A. Sirinelli, and JET EFDA Contributors. “Bayesian derivation of electron temperature profile using JET ECE diagnostics”. In: *38<sup>th</sup> European Physical Society Conference on Plasma Physics*. (Strasbourg, France, June 27–July 1, 2011). Edited by A. Becoulet, T. Hoang, and U. Stroth. Volume 35G. European Conference Abstracts. European Physical Society, 2011, P5.046. URL: <http://ocs.ciemat.es/EPS2011PAP/pdf/P5.046.pdf>.
- [394] B. Ph. van Milligen, T. Estrada, E. Ascasíbar, D. Tafalla, D. López-Bruna, A. López-Fraguas, J. A. Jiménez, I. García-Cortés, A. Dinklage, and R. Fischer. “Integrated data analysis at TJ-II: The density profile”. In: *Review of Scientific Instruments* 82.7 (2011), 073503. DOI: [10.1063/1.3608551](https://doi.org/10.1063/1.3608551). URL: <http://scitation.aip.org/content/aip/journal/rsi/82/7/10.1063/1.3608551>.
- [395] M. N. Gibbs. “Bayesian Gaussian Processes for Regression and Classification”. PhD thesis. University of Cambridge, 1997. URL: <http://www.inference.phy.cam.ac.uk/mng10/GP/thesis.ps.gz>.
- [396] Jasper Snoek, Kevin Swersky, Richard S. Zemel, and Ryan P. Adams. “Input Warping for Bayesian Optimization of Non-stationary Functions”. In: *Proceedings of the 31<sup>st</sup> International Conference on Machine Learning*. (Beijing, China, June 21–26, 2014). Edited by Eric P. Xing and Tony Jebara. Volume 32. JMLR: Workshop and Conference Proceedings. June 2014, pages 1674–1682. arXiv: [1402.0929](https://arxiv.org/abs/1402.0929). URL: <http://jmlr.org/proceedings/papers/v32/snoek14.html>.

- [397] E. S. Marmor and Alcator C-Mod Group. “The Alcator C-Mod Program”. In: *Fusion Science and Technology* 51.3 (Apr. 2007), pages 261–265. URL: [http://www.ans.org/pubs/journals/fst/a\\_1421](http://www.ans.org/pubs/journals/fst/a_1421).
- [398] N. P. Basse, A. Dominguez, E. M. Edlund, C. L. Fiore, R. S. Granetz, A. E. Hubbard, J. W. Hughes, I. H. Hutchinson, J. H. Irby, B. LaBombard, L. Lin, Y. Lin, B. Lipschultz, J. E. Liptac, E. S. Marmor, D. A. Mossessian, R. R. Parker, M. Porkolab, J. E. Rice, J. A. Snipes, V. Tang, J. L. Terry, S. M. Wolfe, S. J. Wukitch, K. Zhurovich, R. V. Bravenec, P. E. Phillips, W. L. Rowan, G. J. Kramer, G. Schilling, S. D. Scott, and S. J. Zweben. “Diagnostic Systems on Alcator C-Mod”. In: *Fusion Science and Technology* 51.3 (Apr. 2007), pages 476–507. URL: [http://www.ans.org/pubs/journals/fst/a\\_1434](http://www.ans.org/pubs/journals/fst/a_1434).
- [399] J. W. Heard, C. Watts, R. F. Gandy, P. E. Phillips, G. Cima, R. Chatterjee, A. Blair, A. Hubbard, C. W. Domier, and N. C. Luhmann Jr. “High resolution electron cyclotron emission temperature profile and fluctuation diagnostic for Alcator C-Mod”. In: *Review of Scientific Instruments* 70.1 (Jan. 1999), pages 1011–1013. DOI: [10.1063/1.1149368](https://doi.org/10.1063/1.1149368). URL: <http://scitation.aip.org/content/aip/journal/rsi/70/1/10.1063/1.1149368>.
- [400] Jerry W. Hughes. “Edge Transport Barrier Studies on the Alcator C-Mod Tokamak”. PhD thesis. Massachusetts Institute of Technology, 2005. URL: <http://hdl.handle.net/1721.1/34436>.
- [401] *NIST/SEMATECH e-Handbook of Statistical Methods: 1.3.3.22. Probability Plot*. URL: <http://www.itl.nist.gov/div898/handbook/eda/section3/probplot.htm>.
- [402] D. Kraft. *A software package for sequential quadratic programming*. Technical report DFVLR-FB 88-28. Institute for Flight Mechanics, Koln, Germany: DLR German Aerospace Center, 1988.
- [403] K. Ida, M. Yoshinuma, K. Nagaoka, M. Osakabe, S. Morita, M. Goto, M. Yokoyama, H. Funaba, S. Murakami, K. Ikeda, H. Nakano, K. Tsumori, Y. Takeiri, O. Kaneko, and LHD experiment group. “Spontaneous toroidal rotation driven by the off-diagonal term of momentum and heat transport in the plasma with the ion internal transport barrier in LHD”. In: *Nuclear Fusion* 50.6 (2010), 064007. DOI: [10.1088/0029-5515/50/6/064007](https://doi.org/10.1088/0029-5515/50/6/064007). URL: <http://iopscience.iop.org/0029-5515/50/6/064007>.
- [404] Felix I. Parra, Michael Barnes, and Peter J. Catto. “Sources of intrinsic rotation in the low-flow ordering”. In: *Nuclear Fusion* 51.11 (2011), 113001. DOI: [10.1088/0029-5515/51/11/113001](https://doi.org/10.1088/0029-5515/51/11/113001). URL: <http://iopscience.iop.org/0029-5515/51/11/113001>.

- [405] Felix I. Parra, Michael Barnes, and Arthur G. Peeters. “Up-down symmetry of the turbulent transport of toroidal angular momentum in tokamaks”. In: *Physics of Plasmas* 18.6 (2011), 062501. DOI: [10.1063/1.3586332](https://doi.org/10.1063/1.3586332). URL: <http://scitation.aip.org/content/aip/journal/pop/18/6/10.1063/1.3586332>.
- [406] Y. Camenen, Y. Idomura, S. Jolliet, and A. G. Peeters. “Consequences of profile shearing on toroidal momentum transport”. In: *Nuclear Fusion* 51.7 (2011), 073039. DOI: [10.1088/0029-5515/51/7/073039](https://doi.org/10.1088/0029-5515/51/7/073039). URL: <http://iopscience.iop.org/0029-5515/51/7/073039>.
- [407] Felix I. Parra, Michael Barnes, Iván Calvo, and Peter J. Catto. “Intrinsic rotation with gyrokinetic models”. In: *Physics of Plasmas* 19.5 (2012), 056116. DOI: [10.1063/1.3699186](https://doi.org/10.1063/1.3699186). URL: <http://scitation.aip.org/content/aip/journal/pop/19/5/10.1063/1.3699186>.
- [408] F. I. Parra, M. F. F. Nave, A. A. Schekochihin, C. Giroud, J. S. de Grassie, J. H. F. Severo, P. de Vries, and K.-D. Zastrow. “Scaling of Spontaneous Rotation with Temperature and Plasma Current in Tokamaks”. In: *Physical Review Letters* 108.9 (Mar. 2012), 095001. DOI: [10.1103/PhysRevLett.108.095001](https://doi.org/10.1103/PhysRevLett.108.095001). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.108.095001>.
- [409] M. Barnes, F. I. Parra, J. P. Lee, E. A. Belli, M. F. F. Nave, and A. E. White. “Intrinsic Rotation Driven by Non-Maxwellian Equilibria in Tokamak Plasmas”. In: *Physical Review Letters* 111.5 (Aug. 2013), 055005. DOI: [10.1103/PhysRevLett.111.055005](https://doi.org/10.1103/PhysRevLett.111.055005). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.111.055005>.
- [410] Michael Barnes. *Dependence of intrinsic rotation on higher-order profile gradients*. Personal communication. Aug. 2014.
- [411] Jungpyo Lee, Felix I. Parra, and Michael Barnes. “Turbulent momentum pinch of diamagnetic flows in a tokamak”. In: *Nuclear Fusion* 54.2 (2014), 022002. DOI: [10.1088/0029-5515/54/2/022002](https://doi.org/10.1088/0029-5515/54/2/022002). URL: <http://iopscience.iop.org/0029-5515/54/2/022002>.
- [412] J. P. Lee, M. Barnes, F. I. Parra, E. A. Belli, and J. Candy. “The effect of diamagnetic flows on turbulent driven ion toroidal rotation”. In: *Physics of Plasmas* 21.5 (2014), 056106. DOI: [10.1063/1.4872322](https://doi.org/10.1063/1.4872322). URL: <http://scitation.aip.org/content/aip/journal/pop/21/5/10.1063/1.4872322>.

- [413] Felix I. Parra and Michael Barnes. “Intrinsic rotation in tokamaks: theory”. In: *Plasma Physics and Controlled Fusion* 57.4 (2015), 045002. DOI: [10.1088/0741-3335/57/4/045002](https://doi.org/10.1088/0741-3335/57/4/045002). URL: <https://iopscience.iop.org/article/10.1088/0741-3335/57/4/045002/>.
- [414] J. C. Hillesheim, F. I. Parra, M. Barnes, N. A. Crocker, H. Meyer, W. A. Peebles, R. Scannell, A. Thornton, and the MAST Team. “Dependence of intrinsic rotation reversals on collisionality in MAST”. In: *Nuclear Fusion* 55.3 (2015), 032003. DOI: [10.1088/0029-5515/55/3/032003](https://doi.org/10.1088/0029-5515/55/3/032003). URL: <http://iopscience.iop.org/0029-5515/55/3/032003>.
- [415] Neil H. Timm. *Applied Multivariate Analysis*. Springer Texts in Statistics. Springer, 2002. ISBN: 978-0-387-95347-2. DOI: [10.1007/b98963](https://doi.org/10.1007/b98963). URL: <https://link.springer.com/book/10.1007/b98963>.
- [416] R. J. Groebner, D. R. Baker, K. H. Burrell, T. N. Carlstrom, J. R. Ferron, P. Gohil, L. L. Lao, T. H. Osborne, D. M. Thomas, W. P. West, J. A. Boedo, R. A. Moyer, G. R. McKee, R. D. Deranian, E. J. Doyle, C. L. Rettig, T. L. Rhodes, and J. C. Rost. “Progress in quantifying the edge physics of the H mode regime in DIII-D”. In: *Nuclear Fusion* 41.12 (2001), pages 1789–1802. DOI: [10.1088/0029-5515/41/12/306](https://doi.org/10.1088/0029-5515/41/12/306). URL: <http://iopscience.iop.org/0029-5515/41/12/306>.
- [417] John Reel Walk Jr. “Pedestal Structure and Stability in High-Performance Plasmas on Alcator C-Mod”. PhD thesis. Massachusetts Institute of Technology, Sept. 2014. URL: <http://hdl.handle.net/1721.1/95524>.
- [418] Jerry Hughes. *Quick TS data question*. Personal communication. Sept. 2016.
- [419] R. J. Hawryluk. “An empirical approach to tokamak transport”. In: *Physics of Plasmas Close to Thermonuclear Conditions: Proceedings of the course held at the International School of Plasma Physics*. (Villa Monastero, Varenna, Italy, Aug. 27–Sept. 8, 1979). Edited by B. Coppi, G. G. Leotta, D. Pfirsch, R. Pozzoli, and E. Sindoni. Volume 1. Brussels: CEC, 1980, pages 19–46. URL: <http://w3.pppl.gov/transp/papers/Hawryluk.pdf>.
- [420] J. E. Rice, M. L. Reinke, J. M. A. Ashbourn, C. Gao, M. M. Victora, M. A. Chilenski, L. Delgado-Aparicio, N. T. Howard, A. E. Hubbard, J. W. Hughes, and J. H. Irby. “X-ray observations of Ca<sup>19+</sup>, Ca<sup>18+</sup> and satellites from Alcator C-Mod tokamak plasmas”. In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 47.7 (2014), 075701. DOI: [10.1088/0953-4075/47/7/075701](https://doi.org/10.1088/0953-4075/47/7/075701). URL: <http://iopscience.iop.org/0953-4075/47/7/075701>.

- [421] M. L. Reinke, P. Beiersdorfer, N. T. Howard, E. W. Magee, Y. Podpaly, J. E. Rice, and J. L. Terry. “Vacuum ultraviolet impurity spectroscopy on the Alcator C-Mod tokamak”. In: *Review of Scientific Instruments* 81.10 (2010), 10D736. DOI: [10.1063/1.3494380](https://doi.org/10.1063/1.3494380). URL: <http://scitation.aip.org/content/aip/journal/rsi/81/10/10.1063/1.3494380>.
- [422] Jorge J. Moré. “The Levenberg-Marquardt algorithm: Implementation and theory”. In: *Numerical Analysis*. Edited by G. A. Watson. Volume 630. Lecture Notes in Mathematics. Springer, 1978, pages 105–116. ISBN: 978-3-540-08538-6. DOI: [10.1007/BFb0067700](https://doi.org/10.1007/BFb0067700). URL: <http://link.springer.com/chapter/10.1007%2FBFb0067700>.
- [423] C. B. Markwardt. “Non-linear Least-squares Fitting in IDL with MPFIT”. In: *Astronomical Data Analysis Software and Systems xviii*. Edited by D. A. Bohlander, D. Durand, and P. Dowler. Volume 411. Astronomical Society of the Pacific Conference Series. Sept. 2009, pages 251–254. arXiv: [0902.2850](https://arxiv.org/abs/0902.2850) [astro-ph.IM]. URL: <http://aspbooks.org/custom/publications/paper/411-0251.html>.
- [424] M. L. Reinke, I. H. Hutchinson, J. E. Rice, N. T. Howard, A. Bader, S. Wukitch, Y. Lin, D. C. Pace, A. Hubbard, J. W. Hughes, and Y. Podpaly. “Poloidal variation of high-Z impurity density due to hydrogen minority ion cyclotron resonance heating on Alcator C-Mod”. In: *Plasma Physics and Controlled Fusion* 54.4 (2012), 045004. DOI: [10.1088/0741-3335/54/4/045004](https://doi.org/10.1088/0741-3335/54/4/045004). URL: <http://iopscience.iop.org/0741-3335/54/4/045004>.
- [425] J. E. Rice, J. L. Terry, E. S. Marmor, and F. Bombarda. “X-ray observations of up-down impurity density asymmetries in Alcator C-Mod plasmas”. In: *Nuclear Fusion* 37.2 (1997), pages 241–249. DOI: [10.1088/0029-5515/37/2/I09](https://doi.org/10.1088/0029-5515/37/2/I09). URL: <http://iopscience.iop.org/0029-5515/37/2/I09>.
- [426] M. L. Reinke, J. E. Rice, I. H. Hutchinson, M. Greenwald, N. T. Howard, J. W. Hughes, J. Irby, Y. Podpaly, J. L. Terry, and A. White. “Non-neoclassical up/down asymmetry of impurity emission on Alcator C-Mod”. In: *Nuclear Fusion* 53.4 (2013), 043006. DOI: [10.1088/0029-5515/53/4/043006](https://doi.org/10.1088/0029-5515/53/4/043006). URL: <http://iopscience.iop.org/0029-5515/53/4/043006>.
- [427] J. L. Terry, E. S. Marmor, K. I. Chen, and H. W. Moos. “Observation of Poloidal Asymmetry in Impurity-Ion Emission Due to  $\nabla B$  Drifts”. In: *Physical Review Letters* 39.25 (Dec. 1977), pages 1615–1618. DOI: [10.1103/PhysRevLett.39.1615](https://doi.org/10.1103/PhysRevLett.39.1615). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.39.1615>.



- [428] L. C. Ingesson, H. Chen, P. Helander, and M. J. Mantsinen. “Comparison of basis functions in soft x-ray tomography and observation of poloidal asymmetries in impurity density”. In: *Plasma Physics and Controlled Fusion* 42.2 (2000), pages 161–180. DOI: [10.1088/0741-3335/42/2/308](https://doi.org/10.1088/0741-3335/42/2/308). URL: <http://iopscience.iop.org/0741-3335/42/2/308>.
- [429] H. Chen, N. C. Hawkes, L. C. Ingesson, M. von Hellermann, K.-D. Zastrow, M. G. Haines, M. Romanelli, and N. J. Peacock. “Poloidally asymmetric distribution of impurities in Joint European Torus plasmas”. In: *Physics of Plasmas* 7.11 (Nov. 2000), pages 4567–4572. DOI: [10.1063/1.1311806](https://doi.org/10.1063/1.1311806). URL: <http://scitation.aip.org/content/aip/journal/pop/7/11/10.1063/1.1311806>.
- [430] K. Brau, S. Suckewer, and S. K. Wong. “Vertical poloidal asymmetries of low-Z element radiation in the PDX tokamak”. In: *Nuclear Fusion* 23.12 (1983), pages 1657–1668. DOI: [10.1088/0029-5515/23/12/009](https://doi.org/10.1088/0029-5515/23/12/009). URL: <http://iopscience.iop.org/0029-5515/23/12/009>.
- [431] T. Fülöp and H. Nordman. “Turbulent and neoclassical impurity transport in tokamak plasmas”. In: *Physics of Plasmas* 16.3 (2009), 032306. DOI: [10.1063/1.3083299](https://doi.org/10.1063/1.3083299). URL: <http://scitation.aip.org/content/aip/journal/pop/16/3/10.1063/1.3083299>.
- [432] Fredrick H. Seguin, Richard Petrasso, and Earl S. Marmor. “Effects of Internal Disruptions on Impurity Transport in Tokamaks”. In: *Physical Review Letters* 51.6 (Aug. 1983), pages 455–458. DOI: [10.1103/PhysRevLett.51.455](https://doi.org/10.1103/PhysRevLett.51.455). URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.51.455>.
- [433] G. Fussmann. “Analytical modelling of impurity transport in toroidal devices”. In: *Nuclear Fusion* 26.8 (1986), pages 983–1002. DOI: [10.1088/0029-5515/26/8/001](https://doi.org/10.1088/0029-5515/26/8/001). URL: <http://iopscience.iop.org/0029-5515/26/8/001>.
- [434] William E. Lorensen and Harvey E. Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. In: *Computer Graphics* 21.4 (July 1987), pages 163–169. DOI: [10.1145/37401.37422](https://doi.org/10.1145/37401.37422). URL: <https://dl.acm.org/citation.cfm?id=37422>.
- [435] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. “scikit-image: image processing in Python”. In: *PeerJ* 2 (2014), e453. DOI: [10.7717/peerj.453](https://doi.org/10.7717/peerj.453). URL: <https://peerj.com/articles/453/>.



- [436] E. C. Fieller. “The Distribution of the Index in a Normal Bivariate Population”. In: *Biometrika* 24.3/4 (Nov. 1932), pages 428–440. DOI: [10.2307/2331976](https://doi.org/10.2307/2331976). URL: <http://www.jstor.org/stable/2331976>.
- [437] D. V. Hinkley. “On the Ratio of Two Correlated Normal Random Variables”. In: *Biometrika* 56.3 (Dec. 1969), pages 635–639. DOI: [10.2307/2334671](https://doi.org/10.2307/2334671). URL: <http://www.jstor.org/stable/2334671>.
- [438] Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2005. ISBN: 978-0-89871-572-9. DOI: [10.1137/1.9780898717921](https://doi.org/10.1137/1.9780898717921). URL: <http://epubs.siam.org/doi/book/10.1137/1.9780898717921>.
- [439] Klaus Mosegaard and Albert Tarantola. “International Handbook of Earthquake and Engineering Seismology (Part A)”. In: New York, NY: Academic Press, 2002. Chapter Probabilistic Approach to Inverse Problems, pages 237–265. URL: [http://www.ipgp.fr/~tarantola/Files/Professional/Papers\\_PDF/InverseProblemHandbk.pdf](http://www.ipgp.fr/~tarantola/Files/Professional/Papers_PDF/InverseProblemHandbk.pdf).
- [440] J. L. Fernández-Martínez, Z. Fernández-Muñiz, J. L. G. Pallero, and L. M. Pedruelo-González. “From Bayes to Tarantola: New insights to understand uncertainty in inverse problems”. In: *Journal of Applied Geophysics* 98 (2013), pages 62–72. DOI: [10.1016/j.jappgeo.2013.07.005](https://doi.org/10.1016/j.jappgeo.2013.07.005). URL: <http://www.sciencedirect.com/science/article/pii/S0926985113001444?>
- [441] Albert Tarantola and Bernard Valette. “Inverse Problems = Quest for Information”. In: *Journal of Geophysics* 50 (1982), pages 159–170. URL: [http://www.ipgp.jussieu.fr/~tarantola/Files/Professional/Papers\\_PDF/IP\\_QI\\_latex.pdf](http://www.ipgp.jussieu.fr/~tarantola/Files/Professional/Papers_PDF/IP_QI_latex.pdf).
- [442] H. P. Summers. *The ADAS User Manual, version 2.6*. 2004. URL: <http://www.adas.ac.uk>.
- [443] B. L. Henke, E. M. Gullikson, and J. C. Davis. “X-ray interactions: photoabsorption, scattering, transmission, and reflection at  $E = 50\text{--}30\,000$  eV,  $Z = 1\text{--}92$ ”. In: *Atomic Data and Nuclear Data Tables* 54.2 (July 1993), pages 181–342. DOI: [10.1006/adnd.1993.1013](https://doi.org/10.1006/adnd.1993.1013). URL: <http://www.sciencedirect.com/science/article/pii/S0092640X83710132>.
- [444] *Filter Transmission*. URL: [http://henke.lbl.gov/optical\\_constants/filter2.html](http://henke.lbl.gov/optical_constants/filter2.html).
- [445] Anne Thorne, Ulf Litzén, and Svereric Johansson. *Spectrophysics: Principles and Applications*. Berlin: Springer, 1999. ISBN: 3-540-65117-9.

- [446] R. Mewe. “Interpolation Formulae for the Electron Impact Excitation of Ions in the H-, He-, Li-, and Ne-Sequences”. In: *Astronomy and Astrophysics* 20 (1972), pages 215–221. URL: <http://adsabs.harvard.edu/full/1972A%26A...20..215M>.
- [447] Hans-Joachim Kunze. *Introduction to Plasma Spectroscopy*. Springer Series on Atomic, Optical and Plasma Physics 56. Berlin: Springer, 2009. DOI: [10.1007/978-3-642-02233-3](https://doi.org/10.1007/978-3-642-02233-3). URL: <https://link.springer.com/book/10.1007/978-3-642-02233-3>.
- [448] Takashi Fujimoto. *Plasma Spectroscopy*. International Series of Monographs on Physics 123. Great Clarendon Street, Oxford OX2 6DP, United Kingdom: Oxford University Press, 2004. ISBN: 0-19-853028-5. URL: <https://global.oup.com/academic/product/plasma-spectroscopy-9780198530282>.
- [449] TFR Group. “Impurity ion transport modifications due to sawtooth internal disruptions during neutral-beam heating in TFR”. In: *Nuclear Fusion* 25.8 (1985), pages 981–986. DOI: [10.1088/0029-5515/25/8/011](https://doi.org/10.1088/0029-5515/25/8/011). URL: <http://iopscience.iop.org/0029-5515/25/8/011>.
- [450] David J. Earl and Michael W. Deem. “Parallel tempering: Theory, applications, and new perspectives”. In: *Physical Chemistry Chemical Physics* 7.23 (2005), pages 3910–3916. DOI: [10.1039/B509983H](https://doi.org/10.1039/B509983H). URL: <http://pubs.rsc.org/en/content/articlehtml/2005/cp/b509983h>.
- [451] W. D. Vousden, W. M. Farr, and I. Mandel. “Dynamic temperature selection for parallel-tempering in Markov chain Monte Carlo simulations”. In: *Monthly Notices of the Royal Astronomical Society* 455.2 (2016), pages 1919–1937. DOI: [10.1093/mnras/stv2422](https://doi.org/10.1093/mnras/stv2422). URL: <https://mnras.oxfordjournals.org/content/455/2/1919.full>.
- [452] F. Feroz and M. P. Hobson. “Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses”. In: *Monthly Notices of the Royal Astronomical Society* 384.2 (2008), pages 449–463. DOI: [10.1111/j.1365-2966.2007.12353.x](https://doi.org/10.1111/j.1365-2966.2007.12353.x). URL: <http://mnras.oxfordjournals.org/content/384/2/449>.
- [453] F. Feroz, M. P. Hobson, and M. Bridges. “MULTINEST: an efficient and robust Bayesian inference tool for cosmology and particle physics”. In: *Monthly Notices of the Royal Astronomical Society* 398.4 (2009), pages 1601–1614. DOI: [10.1111/j.1365-2966.2009.14548.x](https://doi.org/10.1111/j.1365-2966.2009.14548.x). URL: <http://mnras.oxfordjournals.org/content/398/4/1601.short>.

- [454] F. Feroz, M. P. Hobson, E. Cameron, and A. N. Pettitt. *Importance Nested Sampling and the MULTINEST Algorithm*. Jan. 2014. arXiv: [1306.2144](https://arxiv.org/abs/1306.2144). URL: <http://arxiv.org/abs/1306.2144>.
- [455] J. Buchner, A. Georgakakis, K. Nandra, L. Hsu, C. Rangel, M. Brightman, A. Merloni, M. Salvato, J. Donley, and D. Kocevski. “X-ray spectral modelling of the AGN obscuring region in the CDFS: Bayesian model selection and catalogue”. In: *Astronomy and Astrophysics* 564 (Apr. 2014), A125. DOI: [10.1051/0004-6361/201322971](https://doi.org/10.1051/0004-6361/201322971). URL: <http://www.aanda.org/articles/aa/abs/2014/04/aa22971-13/aa22971-13.html>.
- [456] Rupert Allison and Joanna Dunkley. “Comparison of sampling techniques for Bayesian parameter estimation”. In: *Monthly Notices of the Royal Astronomical Society* 437.4 (Feb. 2014), pages 3918–1928. DOI: [10.1093/mnras/stt2190](https://doi.org/10.1093/mnras/stt2190). URL: <http://mnras.oxfordjournals.org/content/437/4/3918>.
- [457] C. P. Ballance, S. D. Loch, A. R. Foster, R. K. Smith, M. C. Witthoefft, and T. R. Kallman. “Uncertainties on Atomic Data”. In: *Fusion Science and Technology* 63.3 (May 2013), pages 358–362. URL: [http://www.ans.org/pubs/journals/fst/a\\_16442](http://www.ans.org/pubs/journals/fst/a_16442).
- [458] Gideon Schwarz. “Estimating the Dimension of a Model”. In: *The Annals of Statistics* 6.2 (1978), pages 461–464. DOI: [10.1214/aos/1176344136](https://doi.org/10.1214/aos/1176344136). URL: <https://projecteuclid.org/euclid.aos/1176344136>.
- [459] D. G. T. Denison, B. K. Mallick, and A. F. M. Smith. “Automatic Bayesian curve fitting”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 60.2 (1998), pages 333–350. URL: <http://www.jstor.org/stable/2985943>.
- [460] Anne E. White. *Quick GYRO question*. Personal communication. Aug. 2016.
- [461] C. Angioni, R. M. McDermott, E. Fable, R. Fischer, T. Pütterich, F. Ryter, G. Tardini, and the ASDEX Upgrade Team. “Gyrokinetic modelling of electron and boron density profiles of H-mode plasmas in ASDEX Upgrade”. In: *Nuclear Fusion* 51.2 (2011), 023006. DOI: [10.1088/0029-5515/51/2/023006](https://doi.org/10.1088/0029-5515/51/2/023006). URL: <http://iopscience.iop.org/0029-5515/51/2/023006>.
- [462] F. J. Casson, C. Angioni, R. Dux, B. Geiger, R. M. McDermott, L. Menchero, A. G. Peeters, C. Veth, and ASDEX Upgrade team. “Progress in the theoretical description and experimental characterisation of impurity transport in ASDEX Upgrade”. In: *39<sup>th</sup> European Physical Society Conference on Plasma Physics and 16<sup>th</sup> International Congress on Plasma Physics*. (Stockholm, Swe-

- den, July 2–6, 2012). Edited by S. Ratynskaya, L. Blomberg, and A. Fasoli. Volume 36F. European Conference Abstracts. European Physical Society, 2012, p1.080. URL: <http://ocs.ciemat.es/EPSICPP2012PAP/pdf/P1.080.pdf>.
- [463] F. M. Poli, P. T. Bonoli, M. Chilenski, R. Mumgaard, S. Shiraiwa, G. M. Wallace, R. Andre, L. Delgado-Aparicio, S. Scott, J. R. Wilson, R. M. Harvey, Yu. V. Petrov, M. Reinke, I. Faust, R. Granetz, J. Hughes, and J. Rice. “Experimental and modeling uncertainties in the validation of lower hybrid current drive”. In: *Plasma Physics and Controlled Fusion* 58.9 (2016), 095001. DOI: [10.1088/0741-3335/58/9/095001](https://doi.org/10.1088/0741-3335/58/9/095001). URL: <http://iopscience.iop.org/0741-3335/58/9/095001>.
- [464] George B. Arfken and Hans J. Weber. *Mathematical Methods for Physicists*. 6<sup>th</sup> edition. 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA: Elsevier Academic Press, 2005. ISBN: 978-0123846549. URL: <http://www.sciencedirect.com/science/book/9780123846549>.
- [465] *NIST Digital Library of Mathematical Functions: 18.5 Explicit Representations*. URL: <http://dlmf.nist.gov/18.5>.
- [466] *NIST Digital Library of Mathematical Functions: 18.9 Recurrence Relations and Derivatives*. URL: <http://dlmf.nist.gov/18.9>.
- [467] *NIST Digital Library of Mathematical Functions: 4.7: Derivatives and Differential Equations*. URL: <http://dlmf.nist.gov/4.7>.
- [468] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/>.
- [469] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/06.10.02.0002.01>.
- [470] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/06.10.02.0001.01>.
- [471] Eric W. Weisstein. “Pochhammer Symbol.” *From MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/PochhammerSymbol.html>.
- [472] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/06.10.02.0003.01>.
- [473] Michael Hardy. “Combinatorics of Partial Derivatives”. In: *The Electronic Journal of Combinatorics* 13.1 (2006), R1. URL: [http://www.emis.ams.org/journals/EJC/Volume\\_13/PDF/v13i1r1.pdf](http://www.emis.ams.org/journals/EJC/Volume_13/PDF/v13i1r1.pdf).

- [474] Donald E. Knuth. *The Art of Computer Programming*. Volume 4A: Combinatorial Algorithms Part 1. Upper Saddle River, NJ: Pearson Addison Wesley, 2011. ISBN: 978-0-201-03804-0. URL: <https://cs.stanford.edu/~uno/taocp.html>.
- [475] NIST Digital Library of Mathematical Functions: 10.25 Definitions. URL: <http://dlmf.nist.gov/10.25>.
- [476] Eric W. Weisstein. "Faà di Bruno's Formula." From *MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/FaadiBrunosFormula.html>.
- [477] Eric W. Weisstein. "Bell Polynomial." From *MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/BellPolynomial.html>.
- [478] SymPy Development Team. *SymPy: Python library for symbolic mathematics*. URL: <http://www.sympy.org>.
- [479] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/03.04.20.0014.02>.
- [480] Eric W. Weisstein. "Leibniz Identity." From *MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/LeibnizIdentity.html>.
- [481] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/03.04.06.0030.01>.
- [482] NIST Digital Library of Mathematical Functions: 10.31: Power Series. URL: <http://dlmf.nist.gov/10.31>.
- [483] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/03.02.20.0015.02>.
- [484] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/01.04.20.0012.01>.
- [485] Eric W. Weisstein. "Stirling Number of the First Kind." From *MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/StirlingNumberoftheFirstKind.html>.
- [486] R. B. Gramacy. "Bayesian Treed Gaussian Process Models". PhD thesis. University of California Santa Cruz, Dec. 2005. URL: <http://search.proquest.com/docview/305004297>.
- [487] Robert B. Gramacy. "tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models". In: *Journal of Statistical Software* 19.9 (2007), pages 1–46. DOI: 10.18637/jss.v019.i09. URL: <http://www.jstatsoft.org/v19/i09/>.

- [488] Robert B. Gramacy and Matthew Alan Taddy. “Categorical Inputs, Sensitivity Analysis, Optimization and Importance Tempering with *tgpr* Version 2, an R Package for Treed Gaussian Process Models”. In: *Journal of Statistical Software* 33.6 (2010), pages 1–48. DOI: [10.18637/jss.v033.i06](https://doi.org/10.18637/jss.v033.i06). URL: <http://www.jstatsoft.org/v33/i06/>.
- [489] H. B. Curry and I. J. Schoenberg. “On Pólya frequency functions IV: The fundamental spline functions and their limits”. In: *Journal d’Analyse Mathématique* 17.1 (Dec. 1966), pages 71–107. DOI: [10.1007/BF02788653](https://doi.org/10.1007/BF02788653). URL: <http://link.springer.com/article/10.1007%2FBF02788653>.
- [490] J. O. Ramsay. “Monotone Regression Splines in Action”. In: *Statistical Science* 3.4 (1988), pages 425–441. DOI: [10.1214/ss/1177012761](https://doi.org/10.1214/ss/1177012761). URL: <https://projecteuclid.org/euclid.ss/1177012761>.
- [491] Paul D. Sampson and Peter Guttorp. “Nonparametric Estimation of Nonstationary Spatial Covariance Structure”. In: *Journal of the American Statistical Association* 87.417 (Mar. 1992), pages 108–119. DOI: [10.1080/01621459.1992.10475181](https://doi.org/10.1080/01621459.1992.10475181). URL: <http://www.tandfonline.com/doi/abs/10.1080/01621459.1992.10475181>.
- [492] Edward Snelson, Carl E. Rasmussen, and Zoubin Ghahramani. “Warped Gaussian Processes”. In: *Seventeenth Annual Conference on Neural Information Processing Systems (NIPS)*. (British Columbia, Canada, Dec. 8–13, 2003). Edited by S. Thrun, L. K. Saul, and B. Schölkopf. Advances in neural information processing systems 16. MIT Press, 2004, pages 337–344. URL: <http://papers.nips.cc/paper/2481-warped-gaussian-processes.pdf>.
- [493] *NIST Digital Library of Mathematical Functions: 8.17 Incomplete Beta Functions*. URL: <http://dlmf.nist.gov/8.17>.
- [494] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/06.21.20.0012.01>.
- [495] *The Wolfram Functions Site*. URL: <http://functions.wolfram.com/06.21.20.0001.01>.
- [496] *NIST Digital Library of Mathematical Functions: 5.12 Beta Function*. URL: <http://dlmf.nist.gov/5.12>.
- [497] Mary C. Meyer. “Inference using shape-restricted regression splines”. In: *The Annals of Applied Statistics* 2.3 (2008), pages 1013–1033. DOI: [10.1214/08-A0AS167](https://doi.org/10.1214/08-A0AS167). URL: <https://projecteuclid.org/euclid.aoas/1223908050>.
- [498] David Duvenaud. *The Kernel Cookbook: Advice on Covariance functions*. URL: <http://people.seas.harvard.edu/~dduvenaud/cookbook/>.

- [499] David Kristjanson Duvenaud. “Automatic Model Construction with Gaussian Processes”. PhD thesis. University of Cambridge, June 2014. URL: <http://people.seas.harvard.edu/~dduvenaud/thesis.pdf>.
- [500] M. A. Chilenski, I. C. Faust, and J. R. Walk. “eqtools: Modular, Extensible, Open-Source, Cross-Machine Python Tools for Working with Magnetic Equilibria”. In: *Computer Physics Communications (in press)* (2016). DOI: [10.1016/j.cpc.2016.09.011](https://doi.org/10.1016/j.cpc.2016.09.011). URL: <http://www.sciencedirect.com/science/article/pii/S001046551630282X>.
- [501] M. A. Chilenski, I. C. Faust, and J. R. Walk. *eqtools: Python tools for magnetic equilibria in tokamak plasmas*. 2016. URL: <https://github.com/PSFCPlasmaTools/eqtools>.
- [502] M. A. Chilenski, I. C. Faust, and J. R. Walk. *eqtools: Tools for interacting with magnetic equilibria*. 2016. URL: <http://eqtools.readthedocs.org/>.
- [503] L. L. Lao, H. St. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer. “Reconstruction of current profile parameters and plasma shapes in tokamaks”. In: *Nuclear Fusion* 25.11 (1985), pages 1611–1622. DOI: [10.1088/0029-5515/25/11/007](https://doi.org/10.1088/0029-5515/25/11/007). URL: <http://iopscience.iop.org/0029-5515/25/11/007>.
- [504] Jeffrey P. Freidberg. *Ideal Magnetohydrodynamics*. Modern Perspectives in Energy. 233 Spring Street, New York, NY 10013: Plenum Press, 1987. ISBN: 0-306-42512-2.
- [505] Eric W. Weisstein. “Cylindrical Coordinates.” *From MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/CylindricalCoordinates.html>.
- [506] Ernst Hairer, Gerhard Wanner, and Syvert P. Nørsett. *Solving Ordinary Differential Equations I*. 2<sup>nd</sup> edition. Springer Series in Computational Mathematics 8. Berlin: Springer, 1993. ISBN: 978-3-540-56670-0. DOI: [10.1007/978-3-540-78862-1](https://doi.org/10.1007/978-3-540-78862-1). URL: <http://link.springer.com/book/10.1007/978-3-540-78862-1>.
- [507] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (Oct. 2011), pages 2825–2830. URL: <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>.



- [508] *scikit-learn: 1.7. Gaussian Processes*. URL: [http://scikit-learn.org/stable/modules/gaussian\\_process.html](http://scikit-learn.org/stable/modules/gaussian_process.html).
- [509] *PyGP*. URL: <https://pythonhosted.org/pygp/>.
- [510] *GPy*. URL: <https://sheffieldml.github.io/GPy/>.
- [511] Sivaram Ambikasaran, Daniel Foreman-Mackey, Leslie Greengard, David W. Hogg, and Michael O’Neil. *Fast Direct Methods for Gaussian Processes*. Apr. 2015. arXiv: [1403.6015](https://arxiv.org/abs/1403.6015). URL: <http://arxiv.org/abs/1403.6015>.
- [512] *George*. URL: <http://dan.iel.fm/george/current/>.
- [513] *Gaussian Process in Python*. URL: <http://www.acgc.uct.ac.za/~seikel/GAPP/index.html>.
- [514] *The Gaussian Processes Web Site: Software*. URL: <http://www.gaussianprocess.org/#code>.
- [515] *Gaussian Process Regression Models*. URL: <https://www.mathworks.com/help/stats/gaussian-process-regression-models.html>.
- [516] *Documentation for GPML Matlab Code version 3.6*. URL: <http://www.gaussianprocess.org/gpml/code/matlab/doc/>.
- [517] *gptk: Gaussian Processes Tool-Kit*. URL: <https://cran.r-project.org/web/packages/gptk/index.html>.
- [518] *GPfit: Gaussian Process Modeling*. URL: <https://cran.r-project.org/web/packages/GPfit/index.html>.
- [519] Blake MacDonald, Pritam Ranjan, and Hugh Chipman. “GPfit: An R Package for Fitting a Gaussian Process Model to Deterministic Simulator Outputs”. In: *Journal of Statistical Software* 64.12 (2015). DOI: [10.18637/jss.v064.i12](https://doi.org/10.18637/jss.v064.i12). URL: <https://www.jstatsoft.org/article/view/v064i12/0>.
- [520] *C++ implementation of Gaussian process regression*. URL: <https://github.com/jonfink/GP>.
- [521] *Gaussian process library for machine learning*. URL: <https://github.com/mblum/libgp>.
- [522] *Gaussian process code in C++ including some implementations of GP-LVM and IVM*. URL: <https://github.com/SheffieldML/GPc>.
- [523] *Matlab implementations of Gaussian processes and other machine learning tools*. URL: <https://github.com/SheffieldML/GPmat>.



- [524] Marina Seikel, Chris Clarkson, and Mathew Smith. “Reconstruction of dark energy and expansion dynamics using Gaussian processes”. In: *Journal of Cosmology and Astroparticle Physics* 2012.06 (June 2012), 036. DOI: [10.1088/1475-7516/2012/06/036](https://doi.org/10.1088/1475-7516/2012/06/036). URL: <http://iopscience.iop.org/1475-7516/2012/06/036>.
- [525] Malte Kuss, Tobias Pfingsten, Lehel Csató, and Carl E. Rasmussen. *Approximate Inference for Robust Gaussian Process Regression*. Technical report 136. Max-Planck-Institut für biologische Kybernetik, Mar. 2005. URL: <https://ei.is.tuebingen.mpg.de/publications/3265>.
- [526] Malte Kuß. “Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning”. PhD thesis. Technische Universität Darmstadt, Mar. 2006. URL: <http://ei.is.tuebingen.mpg.de/publications/4050>.
- [527] Jarno Vanhatalo, Pasi Jylänki, and Aki Vehtari. “Gaussian process regression with Student- $t$  likelihood”. In: *Advances in Neural Information Processing Systems (NIPS)*. Edited by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta. Volume 22. Curran Associates, Inc., 2009, pages 1910–1918. URL: <http://papers.nips.cc/paper/3806-gaussian-process-regression-with-student-t-likelihood>.
- [528] Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. “Robust Gaussian Process Regression with a Student- $t$  Likelihood”. In: *The Journal of Machine Learning Research* 12 (Nov. 2011), pages 3227–3257. URL: <https://dl.acm.org/citation.cfm?id=2078209>.
- [529] Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. “The NumPy Array: A Structure for Efficient Numerical Computation”. In: *Computing in Science and Engineering* 13.2 (2011), pages 22–30. DOI: [10.1109/MCSE.2011.37](https://doi.org/10.1109/MCSE.2011.37). URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5725236>.
- [530] 9.7. *itertools* – Functions creating iterators for efficient looping. URL: <https://docs.python.org/2/library/itertools.html>.
- [531] Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic*. URL: <http://mpmath.org/>.
- [532] R. Bradshaw, S. Behnel, D. S. Seljebotn, G. Ewing, et al. *The Cython compiler*. URL: <http://cython.org>.
- [533] Robert McGibbon. *Optimized Matern 5/2 Kernel*. URL: <https://github.com/markchil/gptools/pull/2>.

- [534] Robert McGibbon. [WIP] *BetaCDF-warped Matern52 kernel*. URL: <https://github.com/markchil/gptools/pull/3>.
- [535] Robert McGibbon. *BetaCDF*. URL: <https://github.com/markchil/gptools/pull/4>.
- [536] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9<sup>th</sup> Python in Science Conference*. Edited by Stéfan van der Walt and Jarrod Millman. 2010, pages 51–56. URL: <http://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf>.
- [537] Wes McKinney. “pandas: a Foundational Python Library for Data Analysis and Statistics”. In: *Python for High Performance and Scientific Computing (PyHPC2011)*. 2011. URL: [http://www.dlr.de/sc/Portaldata/15/Resources/dokumente/pyhpc2011/submissions/pyhpc2011\\_submission\\_9.pdf](http://www.dlr.de/sc/Portaldata/15/Resources/dokumente/pyhpc2011/submissions/pyhpc2011_submission_9.pdf).
- [538] David Freedman and Persi Diaconis. “On the Histogram as a Density Estimator:  $L_2$  Theory”. In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57.4 (Dec. 1981), pages 453–476. DOI: [10.1007/BF01025868](https://doi.org/10.1007/BF01025868). URL: <http://link.springer.com/article/10.1007%2FBF01025868>.
- [539] M. A. Chilenski. *bayesimp: Bayesian analysis of impurity transport data*. 2016. URL: <https://github.com/markchil/bayesimp>.
- [540] Steven G. Johnson. *The NLOpt nonlinear-optimization package*. URL: <http://ab-initio.mit.edu/nlopt>.
- [541] Per A Brodtkorb and John D’Errico. *numdifftools 0.9.16*. URL: <https://github.com/pbrod/numdifftools>.
- [542] Bengt Fornberg. “Generation of Finite Difference Formulas on Arbitrarily Spaced Grids”. In: *Mathematics of Computation* 51.184 (Oct. 1988), pages 699–706. DOI: [10.1090/S0025-5718-1988-0935077-0](https://doi.org/10.1090/S0025-5718-1988-0935077-0). URL: <http://www.ams.org/journals/mcom/1988-51-184/S0025-5718-1988-0935077-0/home.html>.
- [543] Christophe Andrieu and Gareth O. Roberts. “The pseudo-marginal approach for efficient Monte Carlo computations”. In: *The Annals of Statistics* 37.2 (2009), pages 697–725. DOI: [10.1214/07-AOS574](https://doi.org/10.1214/07-AOS574). URL: <https://projecteuclid.org/euclid.aos/1236693147>.

- [544] I. M. Sobol. “On the distribution of points in a cube and the approximate evaluation of integrals”. In: *USSR Computational Mathematics and Mathematical Physics* 7.4 (1967), pages 86–112. DOI: [10 . 1016 / 0041 - 5553 \(67 \) 90144 - 9](https://doi.org/10.1016/0041-5553(67)90144-9). URL: <http://www.sciencedirect.com/science/article/pii/0041555367901449>.
- [545] Thomas Harvey Rowan. “Functional stability analysis of numerical algorithms”. PhD thesis. The University of Texas at Austin, May 1990. URL: <http://search.proquest.com/docview/303865032>.
- [546] G. van Rossum. *Python tutorial*. Technical report CS-R9526. Amsterdam: Centrum voor Wiskunde en Informatica (CWI), May 1995. URL: <http://oai.cwi.nl/oai/asset/5007/05007D.pdf>.
- [547] Python Software Foundation. *Python Language Reference, version 2.7*. URL: <http://www.python.org>.
- [548] Fernando Pérez and Brian E. Granger. “IPython: a System for Interactive Scientific Computing”. In: *Computing in Science and Engineering* 9.3 (May 2007), pages 21–29. ISSN: 1521-9615. DOI: [10 . 1109 / MCSE . 2007 . 53](https://doi.org/10.1109/MCSE.2007.53). URL: <http://ipython.org>.
- [549] John D. Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science and Engineering* 9.3 (2007), pages 90–95. DOI: [10 . 1109 / MCSE . 2007 . 55](https://doi.org/10.1109/MCSE.2007.55). URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4160265>.
- [550] Victor Ng, John Burkardt, and Corrado Chisari. *SOBOL quasi random number sequence generator*. URL: [https://github.com/crankycoder/sobol\\_seq](https://github.com/crankycoder/sobol_seq).

# A

## Supporting material from chapter 2: Profile fitting with Gaussian process regression

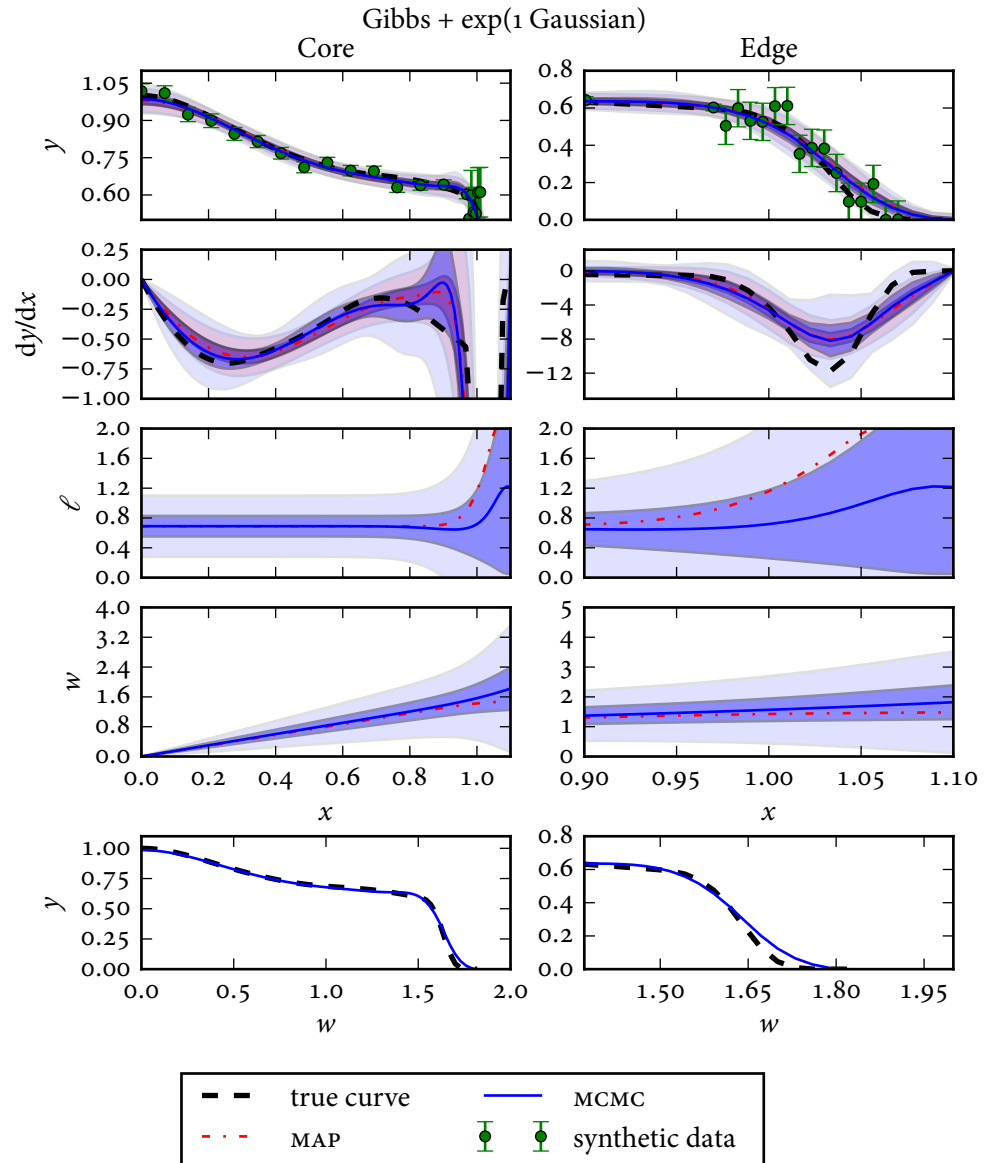
This chapter contains figures and tables from the analysis presented in [chapter 2](#) which were considered to distract from the main discussion, but which are included here for completeness.

### A.1 Material from [section 2.4.3.2](#): Nonstationary covariance kernels applied to entire profiles

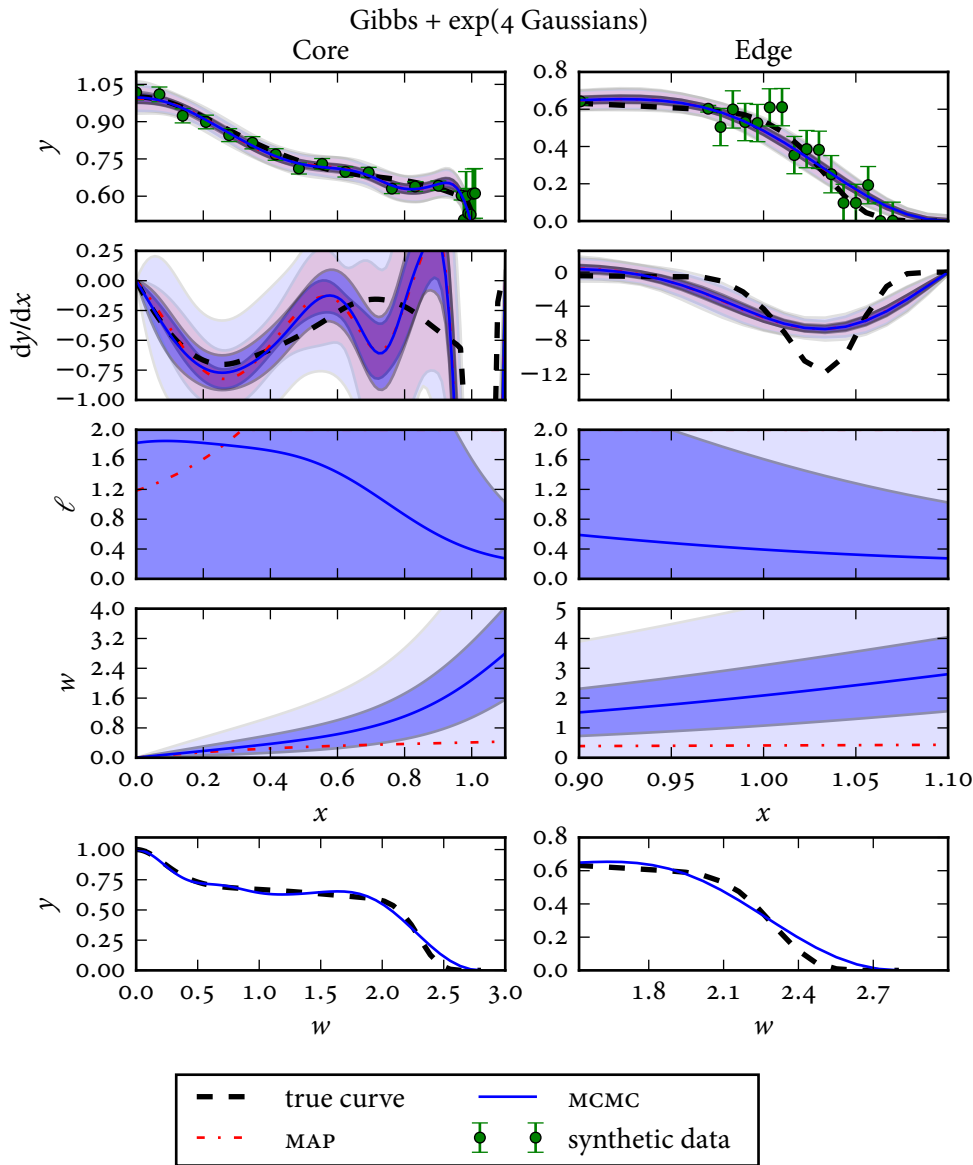
This section contains the following material:

- [Figure A.1](#) shows a plot of the synthetic data fit using the Gibbs covariance kernel with a covariance length scale function which is an exponential of a single Gaussian basis function.
- [Figure A.2](#) shows a plot of the synthetic data fit using the Gibbs covariance kernel with a covariance length scale function which is an exponential of four Gaussian basis functions.
- [Figure A.3](#) shows a plot of the synthetic data fit using the SE covariance kernel with beta-CDF input warping function.
- [Figure A.4](#) shows a plot of the synthetic data fit using the SE covariance kernel with 2-knot I-spline input warping function.

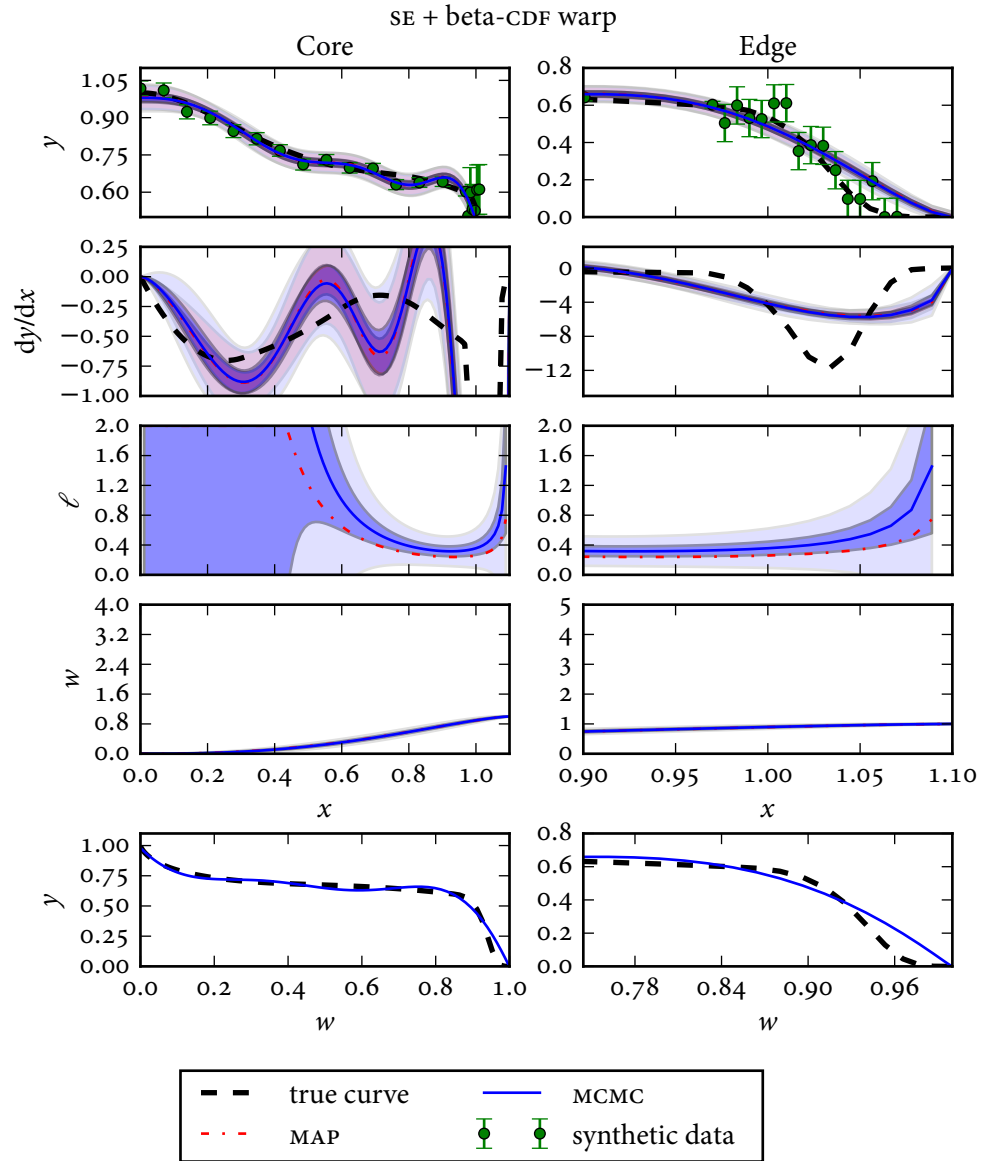
- [Figure A.5](#) shows the marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with tanh covariance length scale function.
- [Figure A.6](#) shows the marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with covariance length scale function which is an exponential of a single Gaussian basis function.
- [Figure A.7](#) shows the marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with covariance length scale function which is an exponential of a sum of four Gaussian basis functions.
- [Figure A.8](#) shows the marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with beta-CDF input warping function.
- [Figure A.9](#) shows the marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with 2-knot I-spline input warping function.
- [Figure A.10](#) shows the marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with 3-knot I-spline input warping function.
- [Table A.1](#) gives the summary statistics of the posterior distributions for the hyperparameters of the various non-stationary fits to synthetic data.



**Figure A.1:** Synthetic data and fits performed using the Gibbs covariance kernel with a covariance length scale function which is an exponential of a single Gaussian basis function, presented as in figure 2.16. The gradient in the core has some extra structure relative to the Gibbs + tanh case and the MAP estimate has converged to a rather unphysical mode. This figure was produced using `synthetic_test_nonstationary.py`.

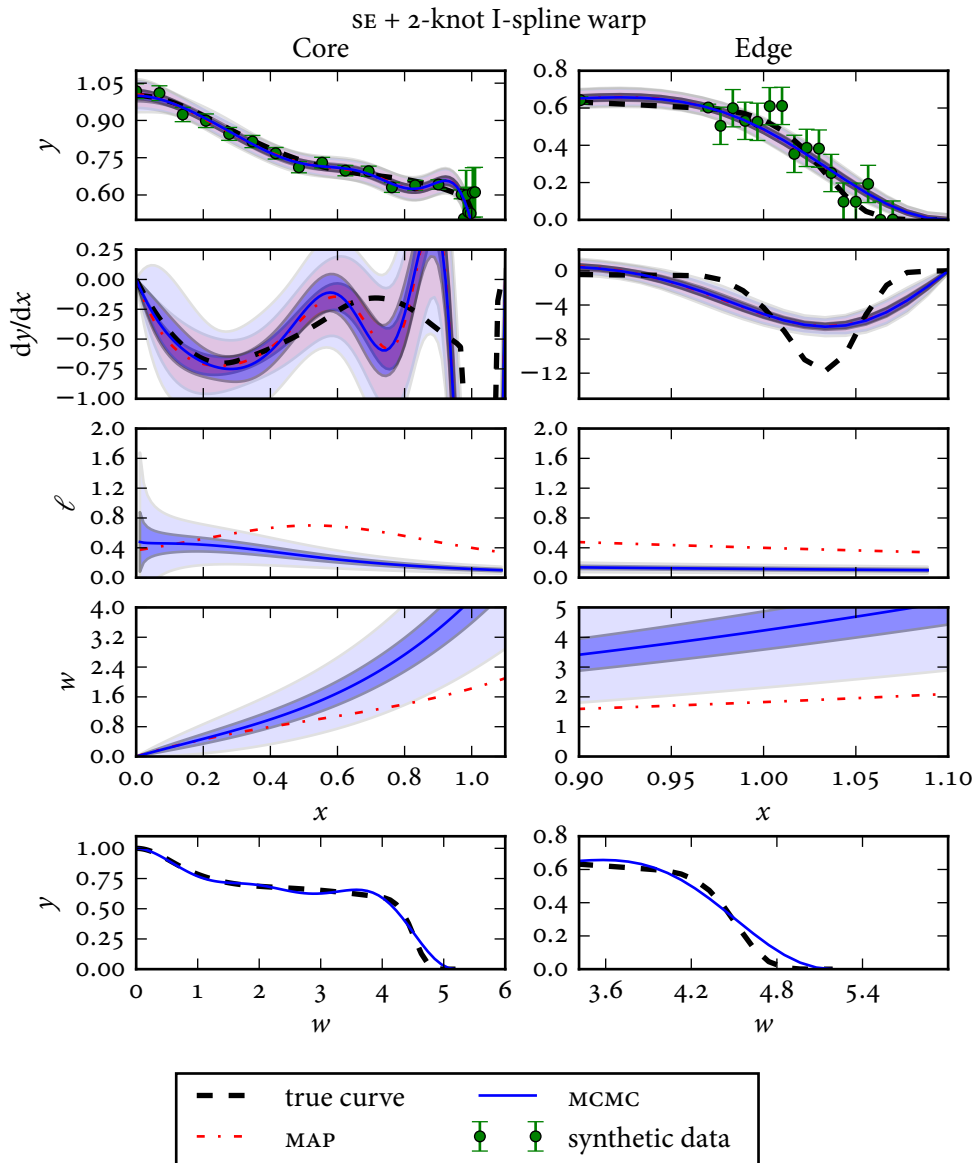


**Figure A.2:** Synthetic data and fits performed using the Gibbs covariance kernel with a covariance length scale function which is an exponential of four Gaussian basis functions, presented as in figure 2.16. There is a very high uncertainty in the covariance length scale function and lots of extra structure in both the core value and gradient; this is an indication that the prior distribution is insufficiently informative. The effective warping function stretches out the edge slightly more than the Gibbs + tanh and Gibbs + exponential of one Gaussian cases. This figure was produced using `synthetic_test_nonstationary.py`.

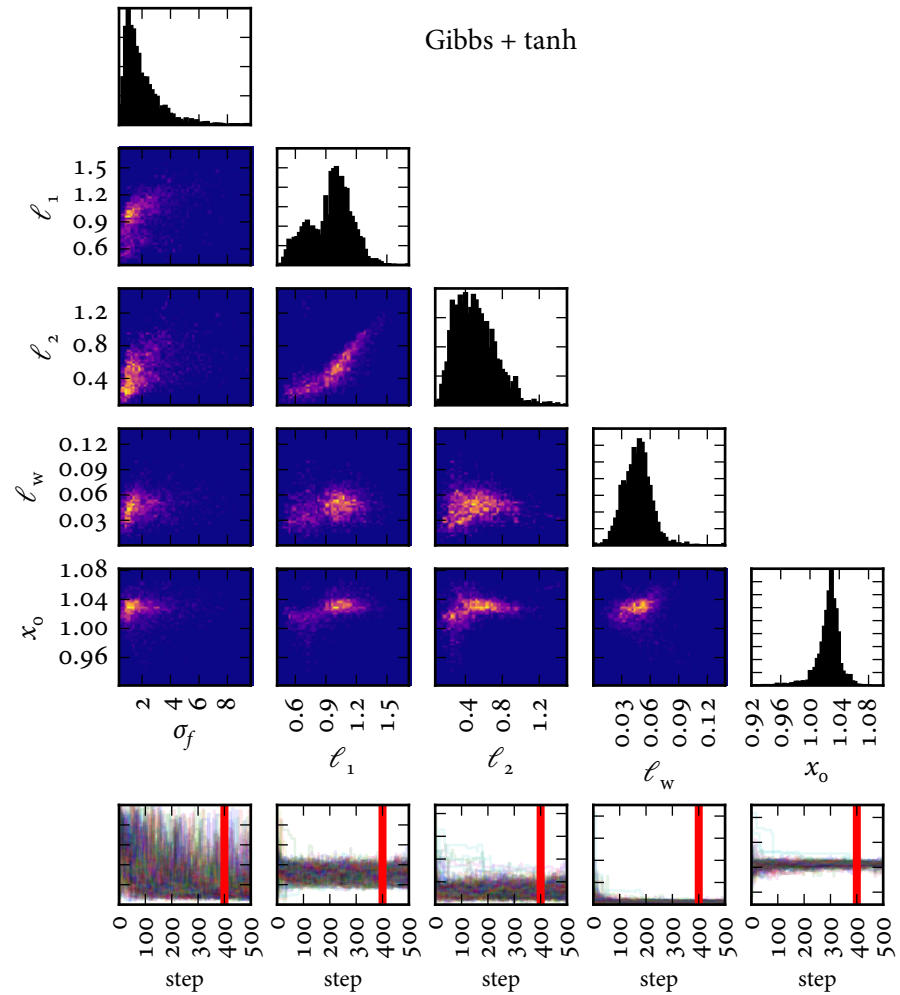


**Figure A.3:** Synthetic data and fits performed using the SE covariance kernel with beta-CDF input warping function, presented as in [figure 2.16](#). The effective length scale function  $\ell$  was computed from [equation \(B.93\)](#). The first and last points in the plots of  $\ell$  and  $w$  were removed because they go to infinity when  $\alpha$  or  $\beta < 1$ . There is too much additional structure in the value and gradient for this approach to be useful for tokamak profile fitting. The effective length scale is very long in the core and somewhat long in the flat region outside of the pedestal, with a short value at and inside the pedestal. This short length scale region is what is responsible for the wiggles in the profile over the region  $0.6 \leq x \leq 1$ . This figure was produced using `synthetic_test_nonstationary.py`.

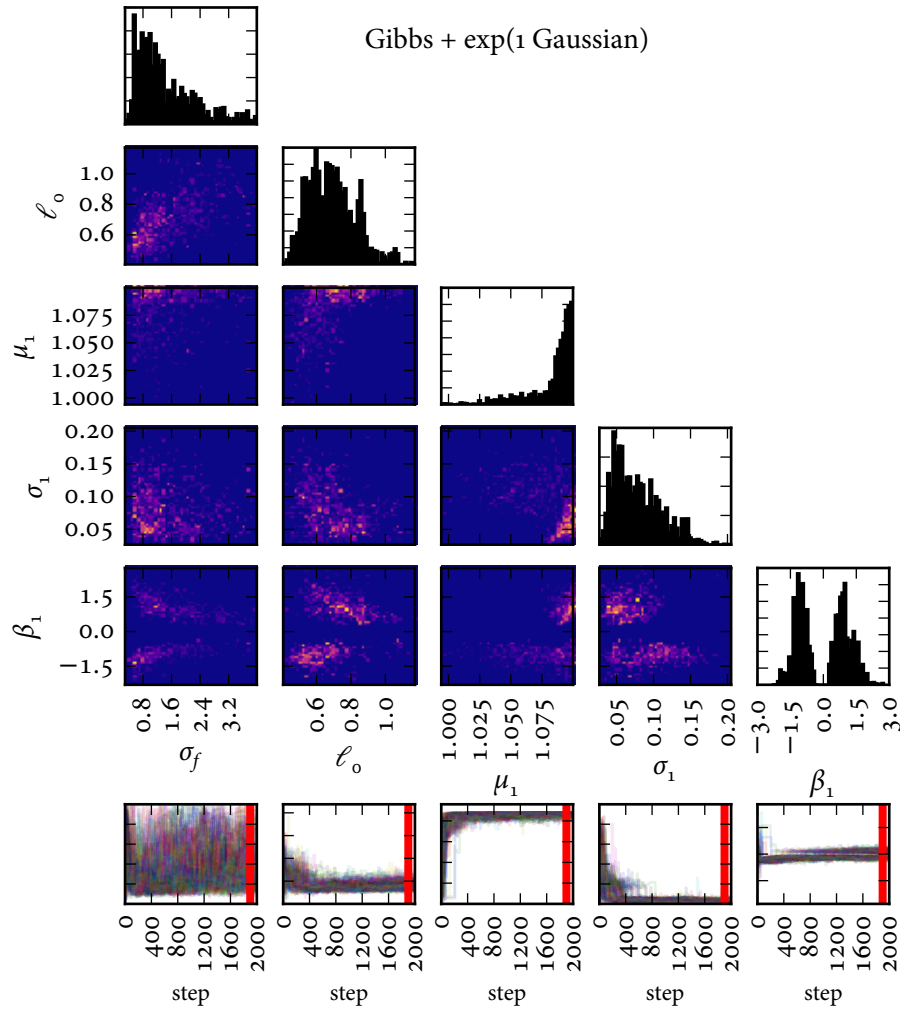




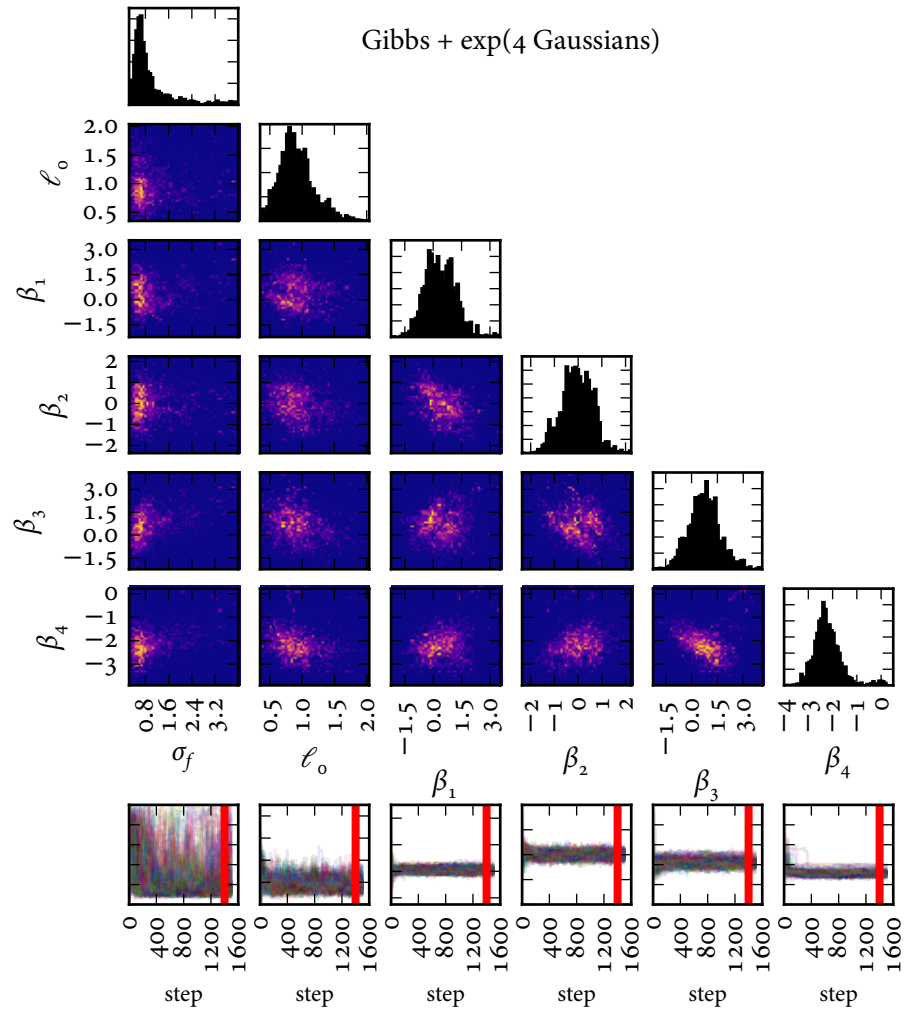
**Figure A.4:** Synthetic data and fits performed using the SE covariance kernel with 2-knot I-spline input warping function, presented as in figure A.3. There is additional structure in the value and gradient, comparable to the level of the Gibbs + exponential of four Gaussians case. The effective covariance length scale is fairly low throughout the profile which is responsible for the wiggles in the profile over the region  $0.5 \leq x \leq 1$ . This figure was produced using `synthetic_test_nonstationary.py`.



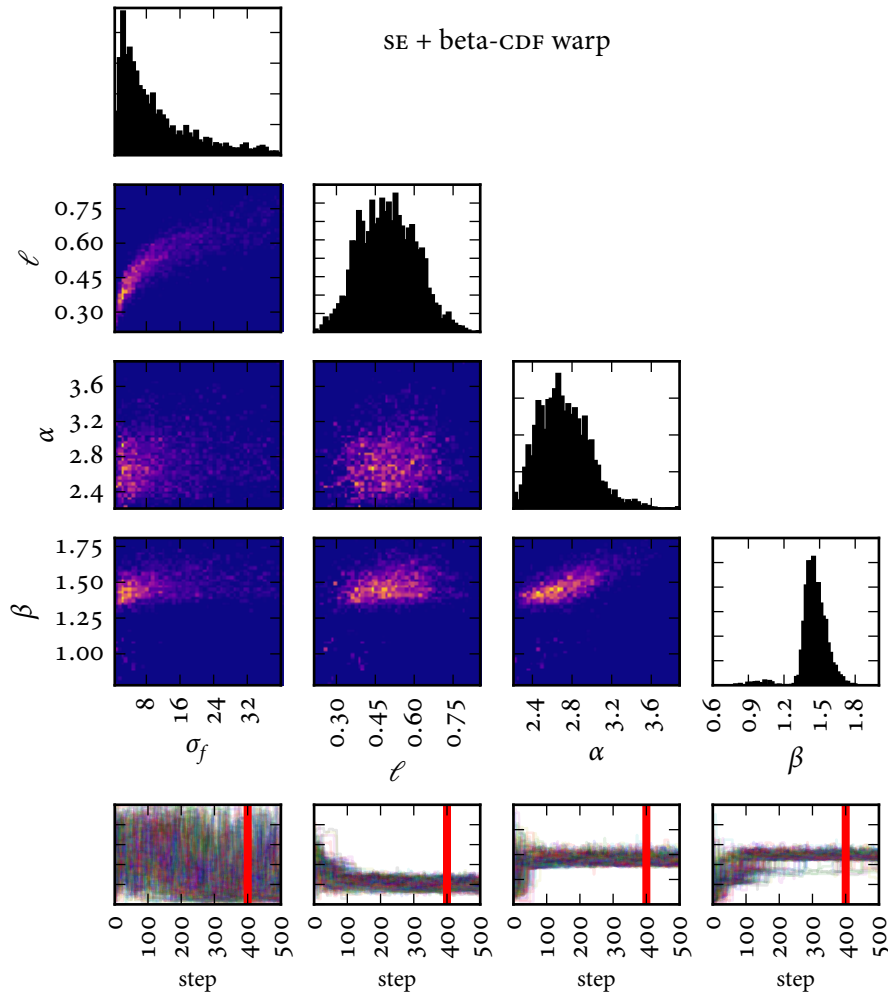
**Figure A.5:** Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with tanh covariance length scale function. The distribution appears to have two modes which are blurred together: one centered around  $\ell_1 \approx 0.6$  and one centered around  $\ell_1 \approx 1$ . This figure was produced using `synthetic_test_nonstationary.py`.



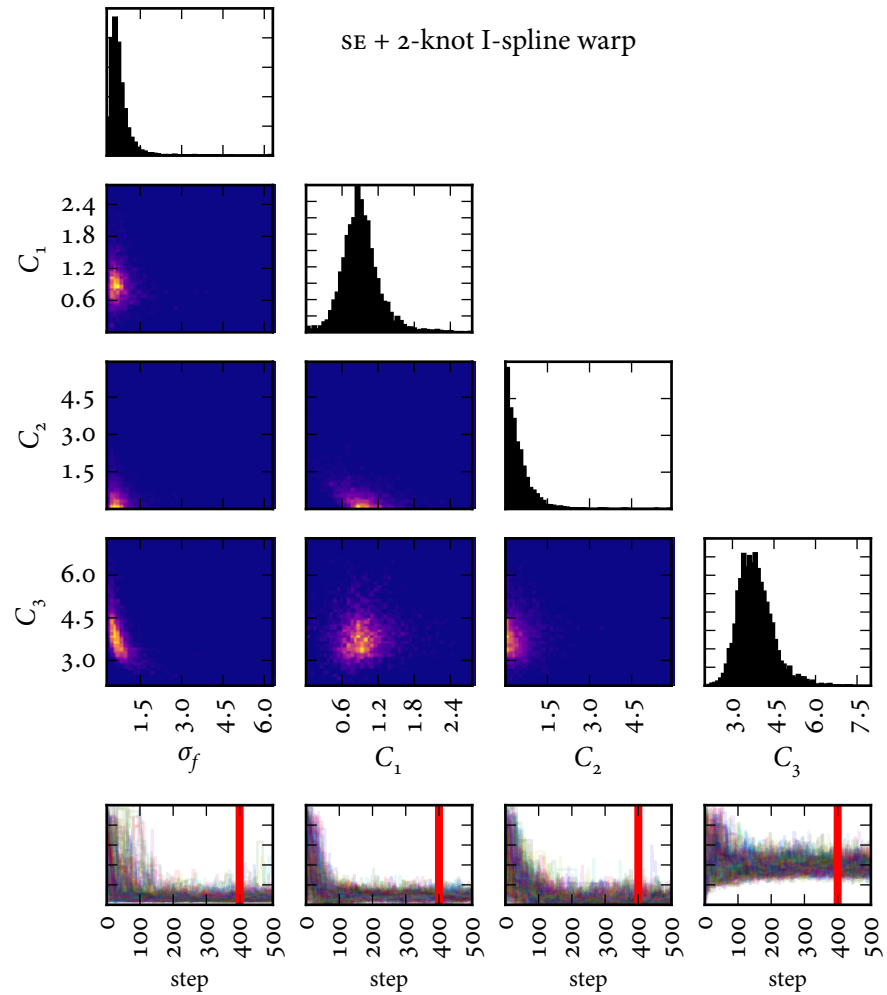
**Figure A.6:** Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with covariance length scale function which is an exponential of a single Gaussian basis function. The center of the Gaussian,  $\mu_1$ , was constrained to lie in the interval  $[0, 1.1]$  where the data are present which is why the distribution terminates there. This figure was produced using `synthetic_test_nonstationary.py`.



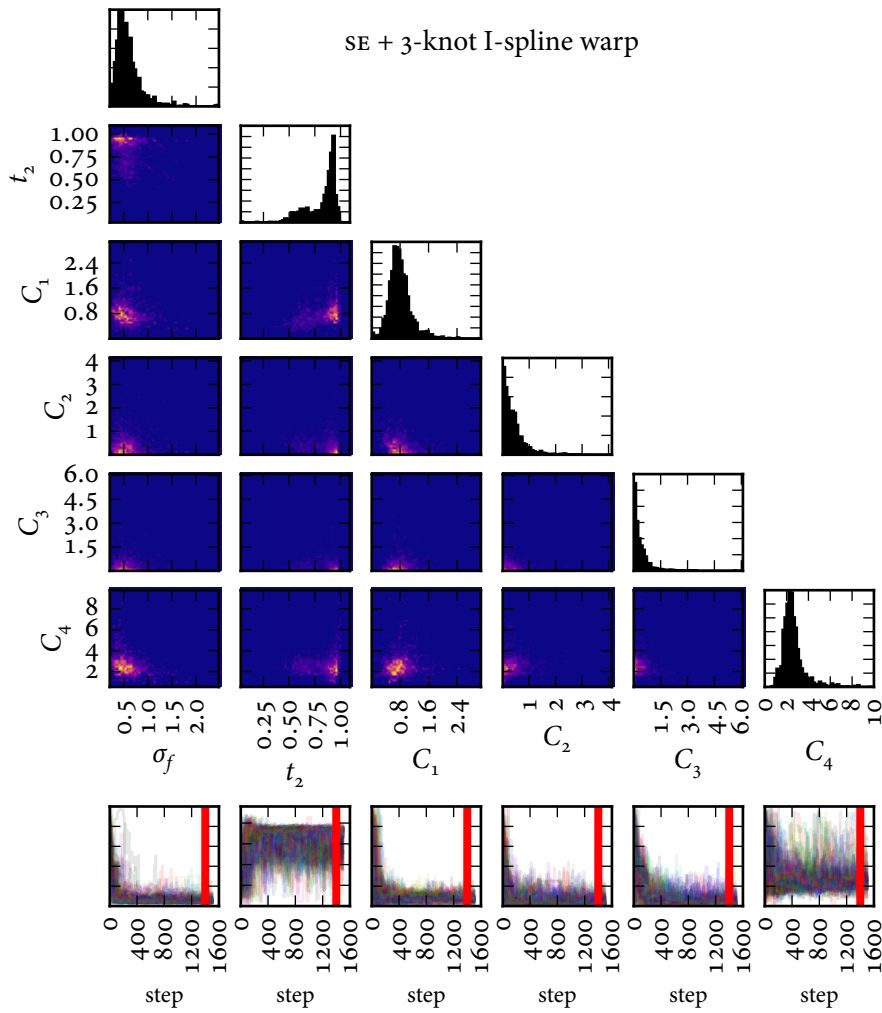
**Figure A.7:** Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the Gibbs covariance kernel with covariance length scale function which is an exponential of a sum of four Gaussian basis functions. The distributions for the coefficients  $\beta_1$  through  $\beta_3$  are very similar to the  $\mathcal{N}(0, 1)$  prior distributions. The final weight,  $\beta_4$ , however, is very clearly negative which causes the covariance length scale to decrease at the edge. This figure was produced using `synthetic_test_nonstationary.py`.



**Figure A.8:** Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with beta-CDF input warping function. Despite the fact that  $\beta < 1$  would be what is needed to stretch out the edge of the domain, the posterior distribution for  $\beta$  very clearly favors positive values. Therefore, both the edge and the core are compressed, effectively stretching the area around midradius. This is likely because the pedestal is too far from the edge of the fitting domain  $0 \leq x \leq 1.1$  for the intuitive interpretation of  $\beta$  to apply. This figure was produced using `synthetic_test_nonstationary.py`.



**Figure A.9:** Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with 2-knot I-spline input warping function. The coefficients were restricted to be positive to ensure a monotonic input warping function, which is why the distribution for  $C_2$  terminates at that edge of the domain. The high value of  $C_3$  causes the edge to be stretched out, whereas the low value of  $C_2$  causes the region just outside of midradius to be compressed. This figure was produced using `synthetic_test_nonstationary.py`.



**Figure A.10:** Marginal posterior distributions for the hyperparameters of the fit to synthetic data using the SE covariance kernel with 3-knot I-spline input warping function. The coefficients were restricted to be positive to ensure a monotonic input warping function, which is why the distributions for  $C_2$  and  $C_3$  terminate at that edge of the domain. The very low values of  $C_2$  and  $C_3$  are what account for the compression of the flat region just outside of midradius and the comparatively high value of  $C_4$  causes the edge to be stretched out. This figure was produced using `synthetic_test_nonstationary.py`.

**Table A.1:** Summary statistics of the posterior distributions for the hyperparameters of the various non-stationary fits to synthetic data.

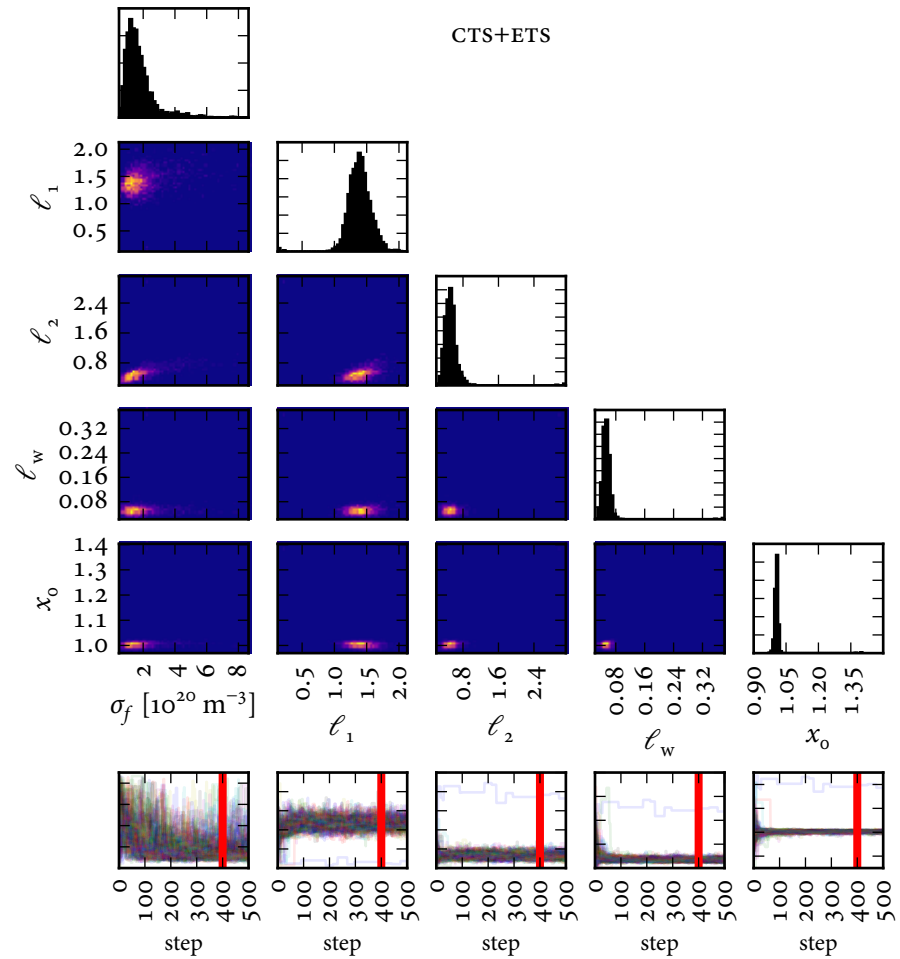
Parameter	Mode	Mean	95% interval
<i>Gibbs covariance kernel + tanh covariance length scale function</i>			
$\sigma_f$	1.16	1.99	[ 0.615 , 5.54 ]
$\ell_1$	1.00	0.947	[ 0.537 , 1.30 ]
$\ell_2$	0.592	0.505	[ 0.180 , 0.962 ]
$\ell_w$	0.0473	0.0462	[ 0.0206, 0.0738]
$x_0$	1.03	1.03	[ 0.988 , 1.05 ]
<i>Gibbs covariance kernel + exponential of one Gaussian</i>			
$\sigma_f$	0.788	1.44	[ 0.520 , 3.58 ]
$\ell_0$	0.689	0.690	[ 0.473 , 1.02 ]
$\mu_1$	1.10	1.08	[ 1.03 , 1.10 ]
$\sigma_1$	0.0725	0.0782	[ 0.0364, 0.148 ]
$\beta_1$	1.30	-0.00551	[-1.65 , 1.82 ]
<i>Gibbs covariance kernel + exponential of four Gaussians</i>			
$\sigma_f$	0.640	0.972	[ 0.353 , 3.28 ]
$\ell_0$	0.847	0.913	[ 0.484 , 1.54 ]
$\beta_1$	0	0.382	[-1.07 , 2.00 ]
$\beta_2$	0.367	-0.0390	[-1.28 , 1.19 ]
$\beta_3$	0.733	0.732	[-0.945 , 2.70 ]
$\beta_4$	1.10	-2.21	[-3.10 , -0.482 ]
<i>SE covariance kernel + beta-CDF input warping function</i>			
$\sigma_f$	2.07	10.2	[ 1.20 , 34.2 ]
$\ell$	0.376	0.498	[ 0.302 , 0.712 ]
$\alpha$	2.59	2.72	[ 2.32 , 3.28 ]
$\beta$	1.41	1.46	[ 1.15 , 1.66 ]
<i>SE covariance kernel + 2-knot I-spline input warping function</i>			
$\sigma_f$	0.565	0.740	[ 0.355 , 1.62 ]
$C_1$	1	0.911	[ 0.363 , 1.62 ]
$C_2$	0	0.448	[ 0.0121, 1.56 ]
$C_3$	1.10	3.83	[ 2.81 , 5.36 ]
<i>SE covariance kernel + 3-knot I-spline input warping function</i>			
$\sigma_f$	0.508	0.599	[ 0.300 , 1.27 ]
$t_2$	1	0.816	[ 0.496 , 0.965 ]
$C_1$	0	0.796	[ 0.305 , 1.56 ]
$C_2$	0.930	0.391	[ 0.0114, 1.47 ]
$C_3$	1.10	0.374	[ 0.0102, 1.68 ]
$C_4$	0.618	2.56	[ 1.20 , 5.27 ]



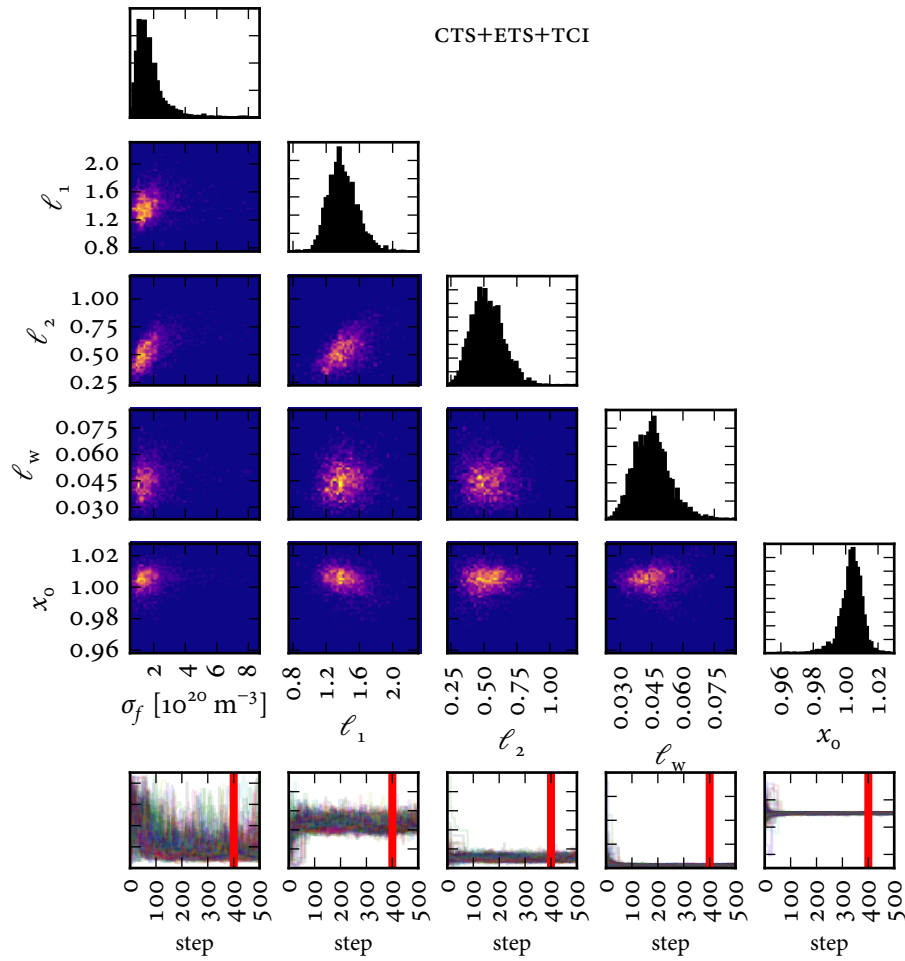
## **A.2 Material from section 2.5.2: Fit including TCI data**

This section contains the following material:

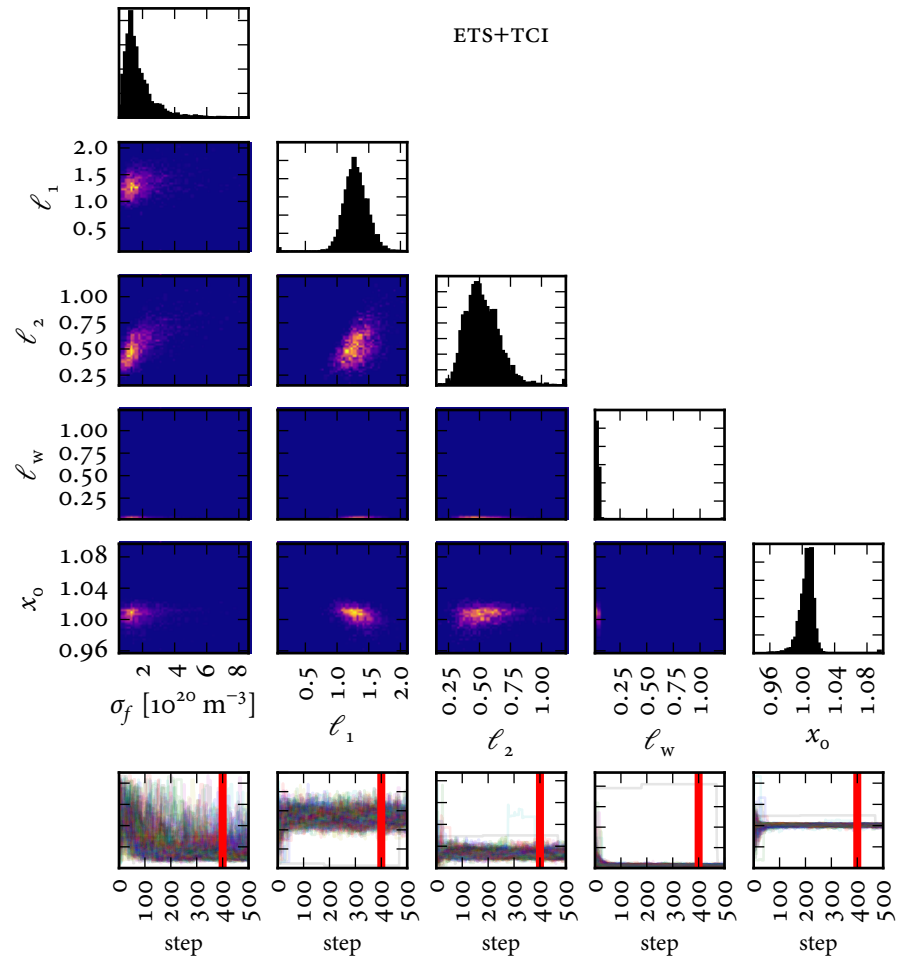
1. [Figure A.11](#) shows the marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using only the Thomson scattering data.
2. [Figure A.12](#) shows the marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using the core and edge Thomson scattering data combined with the TCI data.
3. [Figure A.13](#) shows the marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using only the edge Thomson scattering and TCI data.



**Figure A.11:** Marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using only the Thomson scattering data. This figure was produced using `ne_TCI_test.py`.



**Figure A.12:** Marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using the core and edge Thomson scattering data combined with the TCI data. This figure was produced using `ne_TCI_test.py`.



**Figure A.13:** Marginal posterior distributions for the hyperparameters of the fit to Alcator C-Mod shot 1120907032 using only the edge Thomson scattering and rci data. This figure was produced using `ne_TCI_test.py`.

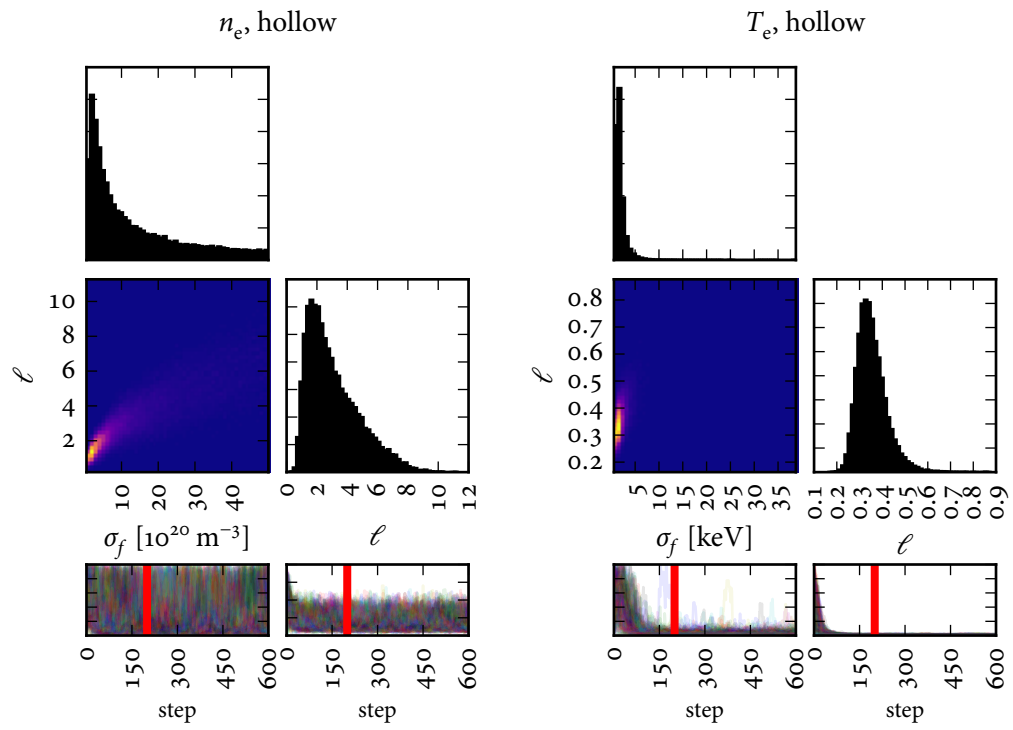
### A.3 Material from section 2.5.3: Computing second derivatives to investigate rotation reversals

This section contains the following material:

- [Table A.2](#) gives the summary statistics of the posterior distributions for the hyperparameters of the fits for Alcator C-Mod shots 1120221011 and 1120221012.
- [Figure A.14](#) shows the marginal posterior distributions for the hyperparameters of the fits to the profiles for Alcator C-Mod shots 1120221011 and 1120221012.

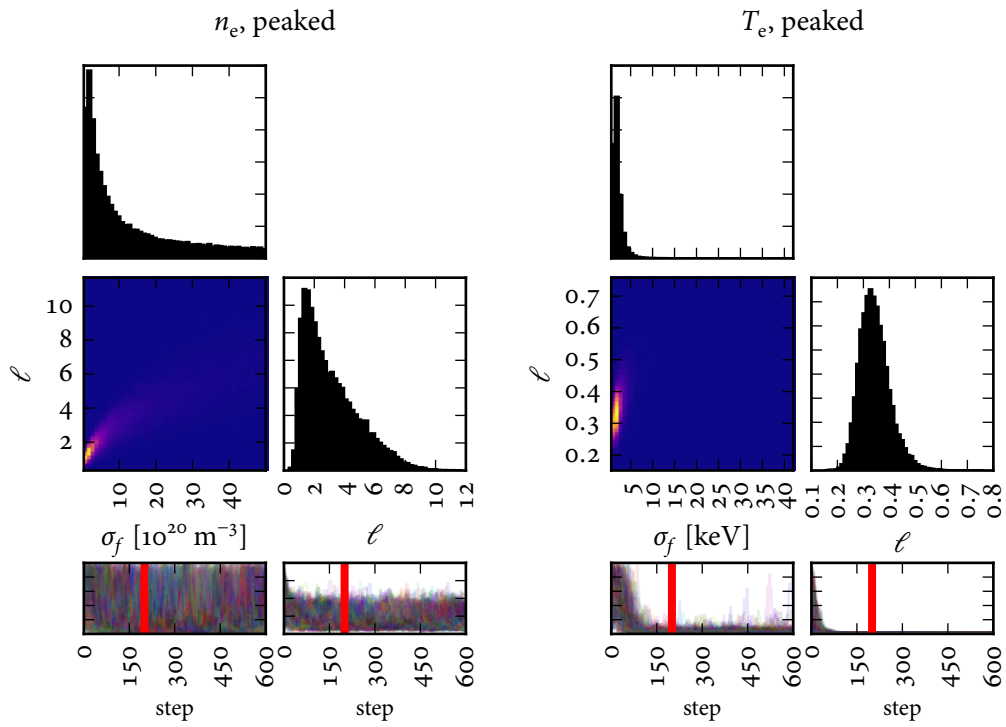
**Table A.2:** Summary statistics of the posterior distributions for the hyperparameters of the fits for Alcator C-Mod shots 1120221011 and 1120221012.

Quantity	Case	Parameter [units]	Mode	Mean	95% interval
$n_e$	hollow	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	1.35	13.6	[1.07, 45.9]
		$\ell$	1.23	3.10	[0.929, 7.19]
$n_e$	peaked	$\sigma_f$ [ $10^{20} \text{ m}^{-3}$ ]	1.09	13.1	[0.854, 46.1]
		$\ell$	1.18	3.13	[0.913, 7.40]
$T_e$	hollow	$\sigma_f$ [keV]	1.20	1.92	[0.835, 5.16]
		$\ell$	0.317	0.355	[0.255, 0.510]
$T_e$	peaked	$\sigma_f$ [keV]	1.33	2.09	[0.933, 5.17]
		$\ell$	0.313	0.344	[0.247, 0.477]



(a) 1120221011 (higher density, hollow rotation profile)

**Figure A.14:** (Continues on the facing page.)



(b) 1120221012 (lower density, peaked rotation profile)

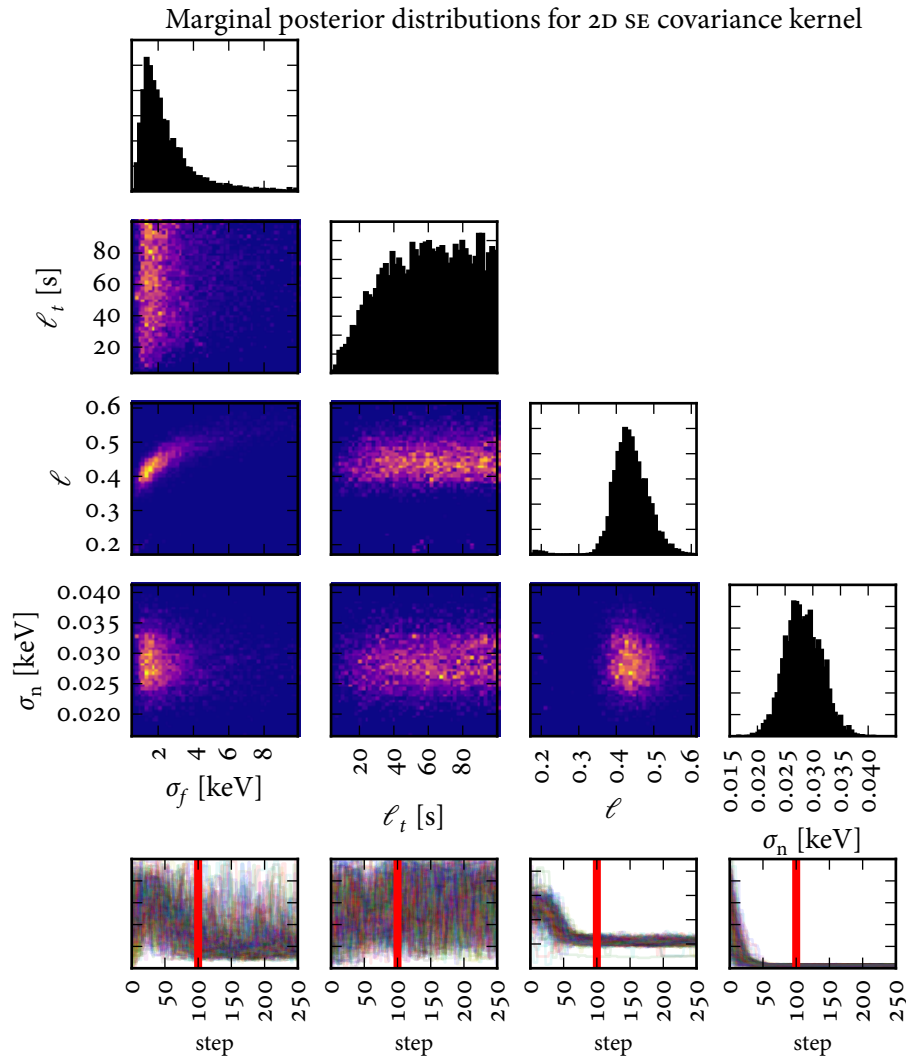
**Figure A.14:** (Continued from the facing page.) Marginal posterior distributions for the hyperparameters of the fits to the profiles for Alcator C-Mod shots (a) 1120221011 and (b) 1120221012. Both pairs of profiles ended up with nearly identical posterior distributions. These figures were produced using `ne_rotation.py` and `Te_rotation.py`.

#### A.4 Material from **section 2.5.4: 2D fitting of sawtooth-free data**

This section contains the following material:

- [Figure A.15](#) shows the univariate and bivariate marginal posterior distributions and MCMC sampler chains for the hyperparameters of the 2D SE covariance function with added homoscedastic noise.
- [Figure A.16](#) shows the marginal posterior distributions for the hyperparameters for the evaluation of the sawtooth-free  $T_e$  profiles at various time points.
- [Figure A.17](#) shows the marginal posterior distributions for the hyperparameters of the 1D sawtooth-free  $T_e$  profiles with various averaging schemes.





**Figure A.15:** Univariate and bivariate marginal posterior distributions and MCMC sampler chains for the hyperparameters of the 2D SE covariance function with added homoscedastic noise. The temporal covariance length scale is  $\ell_t$  and the spatial covariance length scale is  $\ell$ . The results are presented as in figure 2.21. Approximately 30 eV of additional noise appears to be necessary to explain the observed scatter in the data points. The data clearly favor long temporal covariance length scales, but the time window is not long enough nor is the prior distribution informative enough for the posterior distribution to develop a clear mode. This figure was produced using `LH_2d.py`.

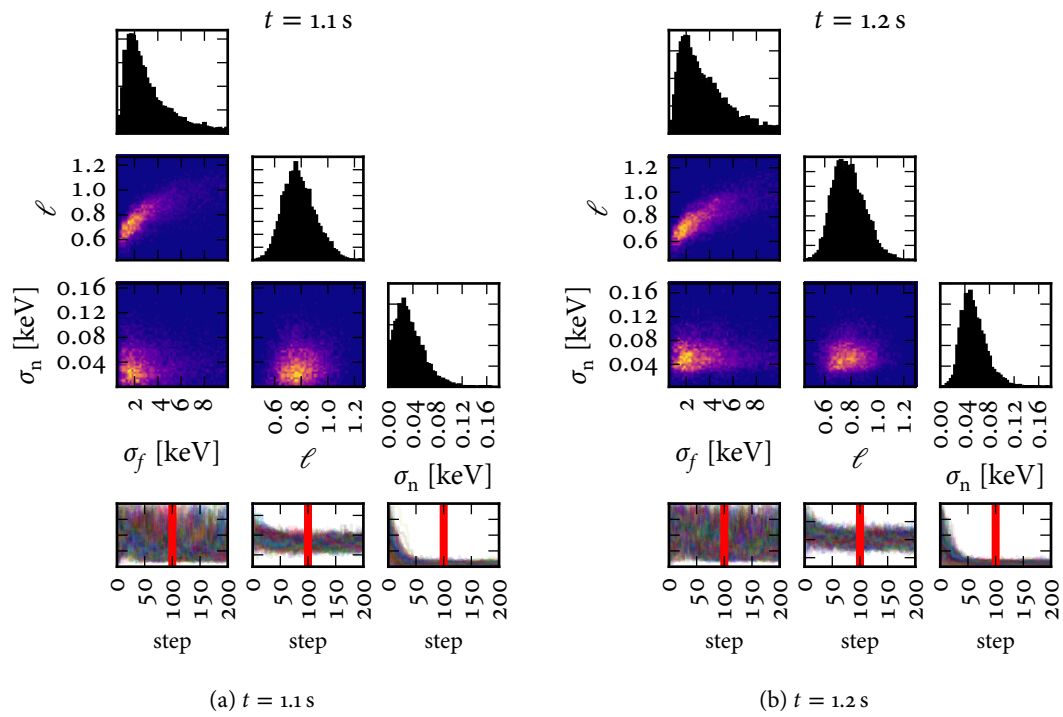
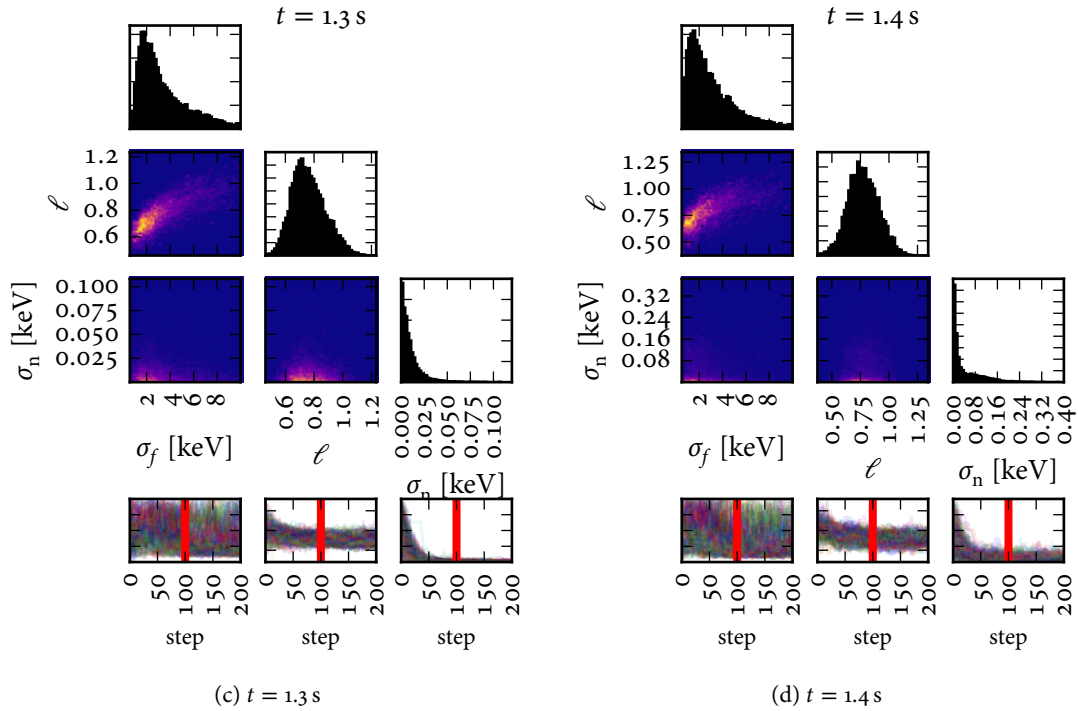
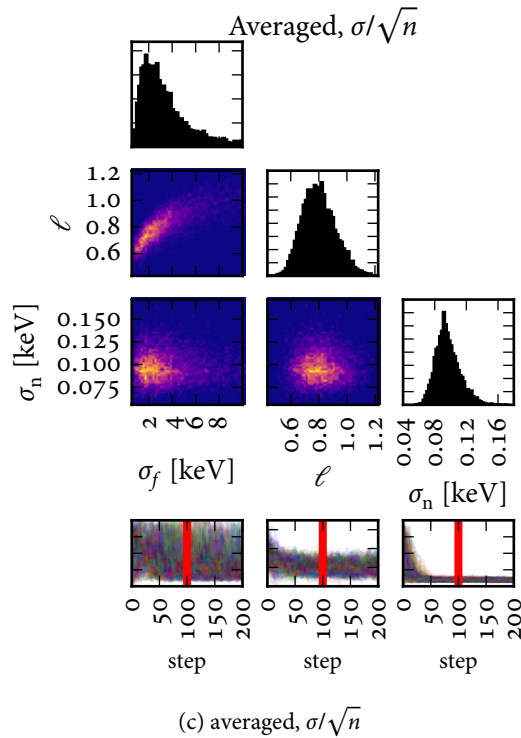


Figure A.16: (Continues on the facing page.)



**Figure A.16:** (Continued from the facing page.) Marginal posterior distributions for the hyperparameters for the evaluation of the sawtooth-free  $T_e$  profiles at various time points. The posterior distributions are fairly similar between all four cases, with the biggest difference being in the additional homoscedastic noise inferred,  $\sigma_n$ . These figures were produced using `LH_2d.py`.





**Figure A.17:** (Continued from the facing page.) Marginal posterior distributions for the hyperparameters of the 1D sawtooth-free  $T_e$  profiles with various averaging schemes. With all of the points included, a second mode appears for the spatial covariance length scale  $\ell$ . This second mode is in the same position as the dominant mode in the posterior distribution from the 2D fit (figure A.15). If a prior distribution which favors shorter length scales were used instead, this mode would likely have been the dominant one. The prior distribution was chosen to prevent unphysical fine scale structure from occurring, however. With the “ $\sigma/\sqrt{n}$ ” averaging, a slightly higher level of additional homoscedastic noise  $\sigma_n$  is inferred. These figures were produced using `LH_2d.py`.

# B

## Covariance kernels

This chapter explores a variety of covariance kernels which have potential applications to tokamak profile fitting, as well as other, general-purpose applications of Gaussian process regression. Everything described in this chapter is supported by the `gptools` code [306, 307] described in [appendix E](#). Two key design objectives of `gptools` were to support data of arbitrary dimension and derivatives of arbitrary order where possible, which is why this section carries out its calculations to such a high degree of generality. Recall from [section 2.3.1](#) that a covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j) \equiv \text{cov}[y(\mathbf{x}_i), y(\mathbf{x}_j)]$  is a function which maps from a pair of coordinates to a real number (i.e.,  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  where  $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$  and  $\mathcal{X}$  can be any set). In this work, each of the coordinates  $\mathbf{x}_i, \mathbf{x}_j$  will be real vectors with  $D$  elements (i.e.,  $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$ ), though in general one can construct covariance kernels which operate on more general objects such as matrices or strings. The introduction given in [section 2.3](#) focussed on stationary covariance kernels which are a function of  $\boldsymbol{\tau} = \mathbf{x}_i - \mathbf{x}_j$  only (and hence are invariant to translations) and isotropic covariance kernels which are a function of  $r = |\boldsymbol{\tau}| = |\mathbf{x}_i - \mathbf{x}_j|$  only (and hence vary the same in all dimensions). Most isotropic covariance kernels only depend on the ratio  $r/\ell$ , where  $\ell$  is the covariance length scale. This formulation is too restrictive for general use, since different input dimensions can have dramatically different covariance length scales or even different units. In general, one can make the substitution  $r^2 \rightarrow \boldsymbol{\tau}^\top \mathbf{M} \boldsymbol{\tau}$  to introduce anisotropy into an isotropic covariance kernel, where  $\mathbf{M}$  is any positive semidefinite matrix. This work takes  $\mathbf{M}$  to be diagonal and uses the substitution

$$\frac{r^2}{\ell^2} \rightarrow \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} = \sum_{d=1}^D \frac{(x_{i_d} - x_{j_d})^2}{\ell_d^2}, \quad (\text{B.1})$$

where each dimension has its own covariance length scale  $\ell_d$ .

## B.1 The squared exponential covariance kernel

The squared exponential (SE) covariance kernel was introduced in [equation \(2.5\)](#), and is a very useful kernel because it encodes the assumption that the fitted curve should be infinitely differentiable. Substituting [equation \(B.1\)](#), the general form of the SE covariance kernel is

$$k_{\text{SE}}(\mathbf{x}_i, \mathbf{x}_j) = k_{\text{SE}}(\boldsymbol{\tau}) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2}\right) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{d=1}^D \frac{(x_{i_d} - x_{j_d})^2}{\ell_d^2}\right). \quad (\text{B.2})$$

While the functional form of this expression causes some authors refer to this as a ‘‘Gaussian’’ covariance kernel, this nomenclature is avoided as this does *not* represent a probability distribution, it represents the spatial structure of the covariance of a probability distribution.

Turning the sum in the middle expression in [equation \(B.2\)](#) into a product yields

$$k_{\text{SE}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \prod_{d=1}^D \exp\left(-\frac{\tau_d^2}{2\ell_d^2}\right). \quad (\text{B.3})$$

Recall the Rodrigues representation of the (physicists’) Hermite polynomials  $H_n(x)$  [464]:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}, \quad (\text{B.4})$$

where [294, 465]:

$$H_n(x) = n! \sum_{l=0}^{\lfloor n/2 \rfloor} \frac{(-1)^l (2x)^{n-2l}}{l! (n-2l)!}. \quad (\text{B.5})$$

Rearranging,

$$\frac{d^n}{dx^n} e^{-x^2} = (-1)^n H_n(x) e^{-x^2}. \quad (\text{B.6})$$

To find the derivatives of [equation \(B.3\)](#), first find

$$\begin{aligned} \frac{d^n}{d\tau_d^n} \exp\left(-\frac{\tau_d^2}{2\ell_d^2}\right) &= \left(\frac{1}{\sqrt{2}\ell_d}\right)^n \frac{d^n}{dx^n} e^{-x^2} = \\ &= \left(\frac{-1}{\sqrt{2}\ell_d}\right)^n H_n\left(\frac{\tau_d}{\sqrt{2}\ell_d}\right) \exp\left(-\frac{\tau_d^2}{2\ell_d^2}\right), \quad (\text{B.7}) \end{aligned}$$

where the substitution  $x = \tau_d/(\sqrt{2\ell_d})$  was used. The derivative of [equation \(B.3\)](#) to arbitrary orders for each dimension is then

$$\frac{\partial^n k_{\text{SE}}(\boldsymbol{\tau})}{\prod_{d=1}^D \partial \tau_d^{n_d}} = \sigma_f^2 \prod_{d=1}^D \left( \frac{-1}{\sqrt{2\ell_d}} \right)^{n_d} \text{H}_{n_d} \left( \frac{\tau_d}{\sqrt{2\ell_d}} \right) \exp \left( -\frac{\tau_d^2}{2\ell_d^2} \right) \quad (\text{B.8})$$

$$= k_{\text{SE}}(\boldsymbol{\tau}) \prod_{d=1}^D \left( \frac{-1}{\sqrt{2\ell_d}} \right)^{n_d} \text{H}_{n_d} \left( \frac{\tau_d}{\sqrt{2\ell_d}} \right), \quad (\text{B.9})$$

where  $n = \sum_{d=1}^D n_d$  and  $n_d$  is the order of derivative with respect to dimension  $d$ . What is needed to apply [equation \(2.29\)](#) and [equation \(2.30\)](#), however, are the derivatives with respect to  $x_{i_d}$  and  $x_{j_d}$ . Note that

$$\tau_d = x_{i_d} - x_{j_d}, \quad \frac{\partial \tau_d}{\partial x_{i_d}} = 1, \quad \frac{\partial \tau_d}{\partial x_{j_d}} = -1 \quad (\text{B.10})$$

$$\frac{\partial^{n_i+n_j} \tau_d}{\partial x_{i_d}^{n_i} \partial x_{j_d}^{n_j}} = 0, \quad \text{for } n_i + n_j \geq 2. \quad (\text{B.11})$$

Applying the chain rule to [equation \(B.9\)](#) then gives

$$\frac{\partial^n k_{\text{SE}}(\boldsymbol{\tau}(\mathbf{x}_i, \mathbf{x}_j))}{\prod_{d=1}^D \partial x_{i_d}^{n_{i_d}} \prod_{d=1}^D \partial x_{j_d}^{n_{j_d}}} = \frac{\partial^n k_{\text{SE}}(\boldsymbol{\tau})}{\prod_{d=1}^D \partial \tau_d^{n_{i_d}+n_{j_d}}} \prod_{d=1}^D \left( \frac{\partial \tau_d}{\partial x_{i_d}} \right)^{n_{i_d}} \prod_{d=1}^D \left( \frac{\partial \tau_d}{\partial x_{j_d}} \right)^{n_{j_d}} \quad (\text{B.12})$$

$$= k_{\text{SE}}(\boldsymbol{\tau}) \prod_{d=1}^D \left( \frac{-1}{\sqrt{2\ell_d}} \right)^{n_{i_d}+n_{j_d}} \text{H}_{n_{i_d}+n_{j_d}} \left( \frac{\tau_d}{\sqrt{2\ell_d}} \right) \cdot \prod_{d=1}^D (-1)^{n_{j_d}} \quad (\text{B.13})$$

$$= k_{\text{SE}}(\boldsymbol{\tau}) \prod_{d=1}^D \frac{(-1)^{n_{i_d}+2n_{j_d}}}{(\sqrt{2\ell_d})^{n_{i_d}+n_{j_d}}} \text{H}_{n_{i_d}+n_{j_d}} \left( \frac{\tau_d}{\sqrt{2\ell_d}} \right), \quad (\text{B.14})$$

where  $n = \sum_{d=1}^D n_{i_d} + \sum_{d=1}^D n_{j_d}$ ,  $n_{i_d}$  is the order of derivative with respect to dimension  $d$  of  $\mathbf{x}_i$  and  $n_{j_d}$  is the order of derivative with respect to dimension  $d$  of  $\mathbf{x}_j$ . [Equation \(B.14\)](#) enables `gptools` to compute derivatives of *arbitrary* order for data of *arbitrary* dimensionality. A plot of the 1D SE covariance kernel and its first few derivatives is provided in [figure B.1](#) and plots of the 2D SE covariance kernel and its first few derivatives are provided in [figure B.2](#).

One final piece of useful information is the derivative of the SE covariance kernel with respect to the hyperparameters, which enables the use of gradient-based



optimizers and advanced types of MCMC sampling. The derivative with respect to  $\sigma_f$  is trivial:

$$\frac{\partial k_{\text{SE}}(\boldsymbol{\tau})}{\partial \sigma_f} = 2\sigma_f \prod_{d=1}^D \exp\left(-\frac{\tau_d^2}{2\ell_d^2}\right) \quad (\text{B.15})$$

$$\frac{\partial}{\partial \sigma_f} \frac{\partial^n k_{\text{SE}}(\boldsymbol{\tau}(\mathbf{x}_i, \mathbf{x}_j))}{\prod_{d=1}^D \partial x_{i_d}^{n_{i_d}} \prod_{d=1}^D \partial x_{j_d}^{n_{j_d}}} = \frac{2}{\sigma_f} \frac{\partial^n k_{\text{SE}}(\boldsymbol{\tau}(\mathbf{x}_i, \mathbf{x}_j))}{\prod_{d=1}^D \partial x_{i_d}^{n_{i_d}} \prod_{d=1}^D \partial x_{j_d}^{n_{j_d}}}, \quad (\text{B.16})$$

where in the last expression the limit as  $\sigma_f \rightarrow 0$  is zero. To find the derivatives with respect to the covariance length scales  $\ell_d$ , first note

$$\frac{\partial k_{\text{SE}}(\boldsymbol{\tau})}{\partial \ell_e} = \sigma_f^2 \prod_{d=1}^D \exp\left(-\frac{\tau_d^2}{2\ell_d^2}\right) \frac{\tau_e^2}{\ell_e^3} = \frac{\tau_e^2}{\ell_e^3} k_{\text{SE}}(\boldsymbol{\tau}). \quad (\text{B.17})$$

Note that the derivative of a Hermite polynomial is [294, 466]

$$\frac{dH_n(x)}{dx} = 2nH_{n-1}(x). \quad (\text{B.18})$$

Therefore, the derivative of equation (B.9) with respect to covariance length scale  $\ell_e$  is

$$\begin{aligned} \frac{\partial}{\partial \ell_e} \frac{\partial^n k_{\text{SE}}(\boldsymbol{\tau})}{\prod_{d=1}^D \partial \tau_d^{n_d}} &= \frac{\partial k_{\text{SE}}(\boldsymbol{\tau})}{\partial \ell_e} \prod_{d=1}^D \left(\frac{-1}{\sqrt{2\ell_d}}\right)^{n_d} H_{n_d}\left(\frac{\tau_d}{\sqrt{2\ell_d}}\right) \\ &+ \mathbb{I}(n_e > 0) k_{\text{SE}}(\boldsymbol{\tau}) n_e \left(\frac{-1}{\sqrt{2\ell_e}}\right)^{n_e-1} \frac{1}{\sqrt{2\ell_e^2}} H_{n_e}\left(\frac{\tau_e}{\sqrt{2\ell_e}}\right) \\ &\quad \cdot \prod_{\substack{d=1 \\ d \neq e}}^D \left(\frac{-1}{\sqrt{2\ell_d}}\right)^{n_d} H_{n_d}\left(\frac{\tau_d}{\sqrt{2\ell_d}}\right) \\ &+ \mathbb{I}(n_e > 0) k_{\text{SE}}(\boldsymbol{\tau}) \left(\frac{-1}{\sqrt{2\ell_e}}\right)^{n_e} 2n_e H_{n_e-1}\left(\frac{\tau_e}{\sqrt{2\ell_e}}\right) \\ &\quad \cdot \frac{-\tau_e}{\sqrt{2\ell_e^2}} \prod_{\substack{d=1 \\ d \neq e}}^D \left(\frac{-1}{\sqrt{2\ell_d}}\right)^{n_d} H_{n_d}\left(\frac{\tau_d}{\sqrt{2\ell_d}}\right) \quad (\text{B.19}) \end{aligned}$$

$$\begin{aligned}
&= k_{\text{SE}}(\boldsymbol{\tau}) \prod_{d=1}^D \left( \frac{-1}{\sqrt{2}\ell_d} \right)^{n_d} \text{H}_{n_d} \left( \frac{\tau_d}{\sqrt{2}\ell_d} \right) \left( \frac{\tau_e^2}{\ell_e^3} - \mathbb{I}(n_e > 0) \frac{n_e}{\ell_e} \right. \\
&\quad \left. - \mathbb{I}(n_e > 0) \frac{\sqrt{2n_e}\tau_e}{\ell_e^2} \frac{\text{H}_{n_e-1}(\tau_e/(\sqrt{2}\ell_e))}{\text{H}_{n_e}(\tau_e/(\sqrt{2}\ell_e))} \right), \quad (\text{B.20})
\end{aligned}$$

where

$$\mathbb{I}(n_e > 0) = \begin{cases} 1, & n_e > 0 \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.21})$$

is an indicator function. This then gets multiplied by another factor of  $\prod_{d=1}^D (-1)^{n_d}$  to arrive at the derivative of [equation \(B.14\)](#) with respect to  $\ell_e$ .

## B.2 The rational quadratic covariance kernel

The rational quadratic (RQ) covariance kernel is a stationary, isotropic covariance kernel with the basic form

$$k_{\text{RQ}}(r) = \sigma_f^2 \left( 1 + \frac{r^2}{2\alpha\ell^2} \right)^{-\alpha}, \quad (\text{B.22})$$

where  $\alpha > 0$  and  $\sigma_f$  and  $\ell$  have similar interpretations as for the SE covariance kernel. As noted in [297], this covariance kernel can be obtained as an infinite sum over SE covariance kernels with different covariance length scales, the mixture of which is determined by  $\alpha$ . Note that as  $\alpha \rightarrow \infty$  the RQ covariance kernel reduces to the SE covariance kernel with covariance length scale  $\ell$ . Making the substitution from [equation \(B.1\)](#) yields

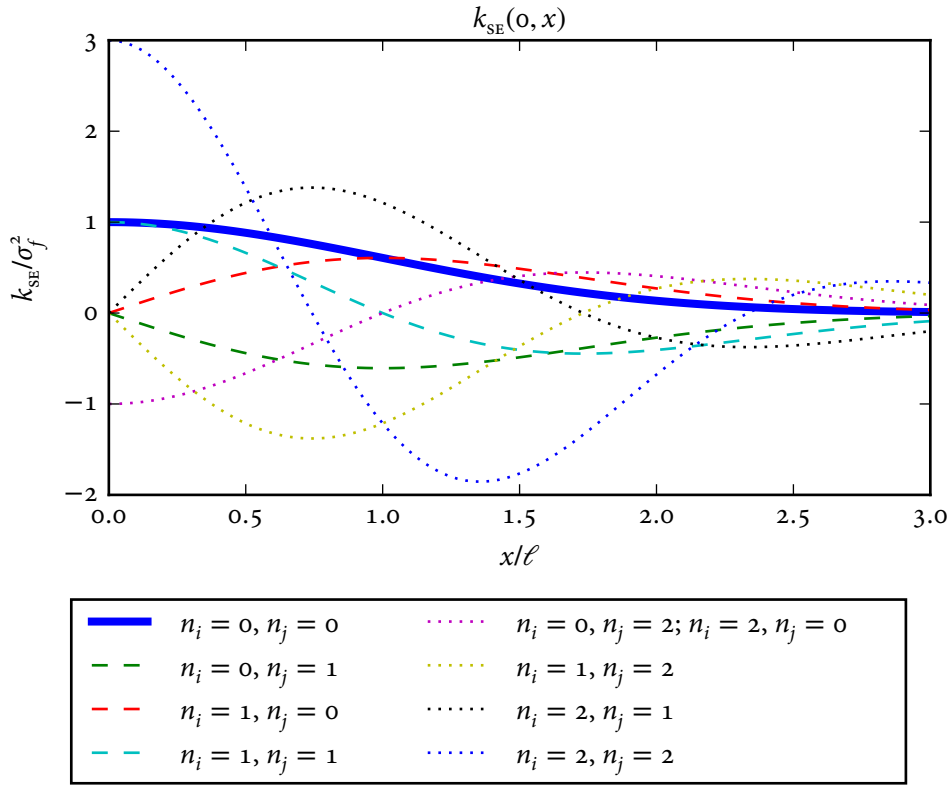
$$k_{\text{RQ}}(\mathbf{x}_i, \mathbf{x}_j) = k_{\text{RQ}}(\boldsymbol{\tau}) = \sigma_f^2 \left( 1 + \frac{1}{2\alpha} \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \right)^{-\alpha}. \quad (\text{B.23})$$

In order to get this into a form which can be differentiated, define

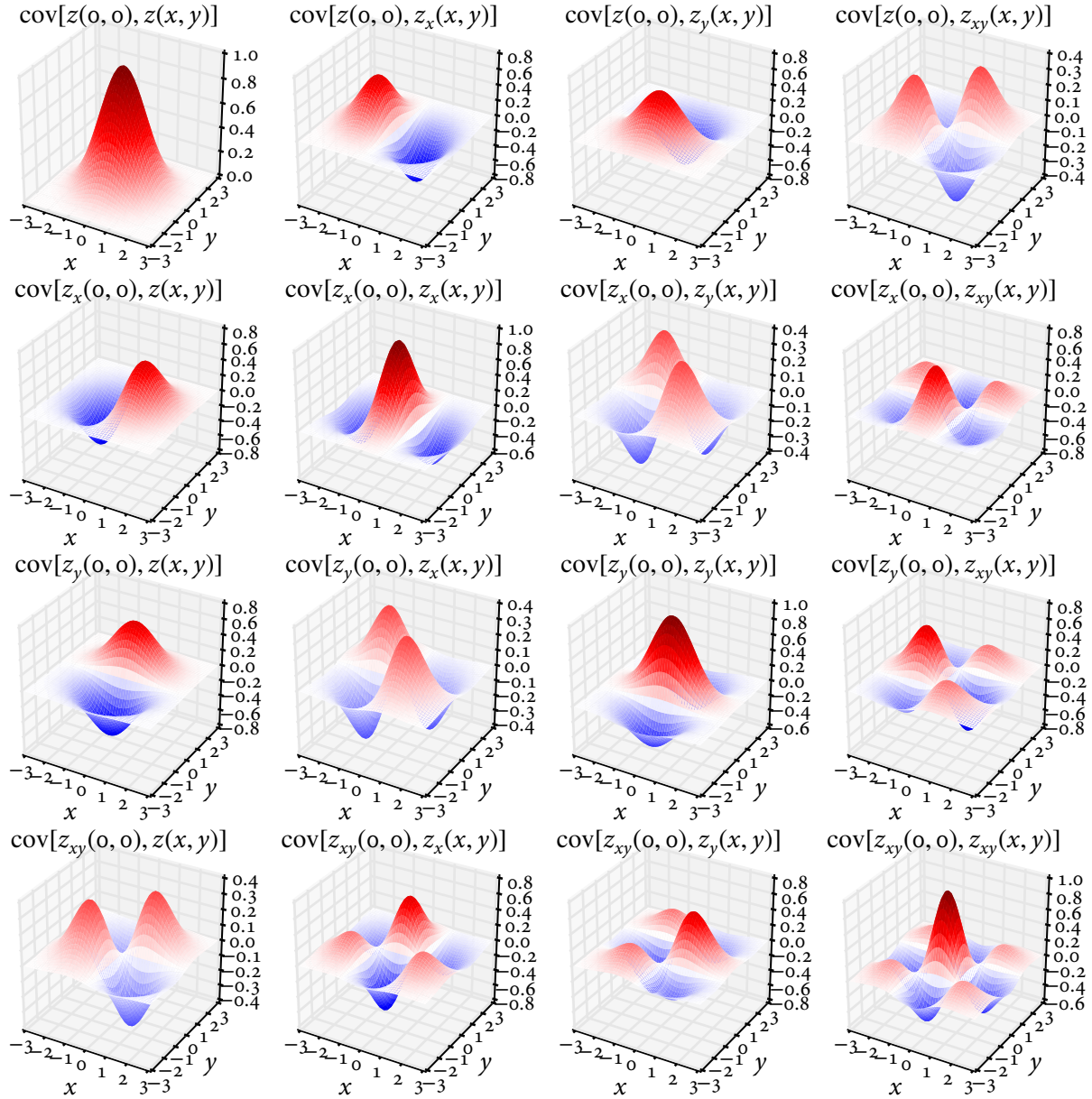
$$y = 1 + \frac{1}{2\alpha} \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2}. \quad (\text{B.24})$$

The RQ kernel is then simply

$$k_{\text{RQ}}(y) = \sigma_f^2 y^{-\alpha}, \quad (\text{B.25})$$



**Figure B.1:** One-dimensional squared exponential covariance kernel and a few of its derivatives. The horizontal scale has been scaled by  $\ell$  and the vertical scale has been scaled by  $\sigma_f^2$  to remove the dependence on the hyperparameters. The **solid blue curve** gives  $k_{SE}(o, x)$  itself. Notice that this curve is infinitely differentiable and has dropped in value significantly by  $x/\ell = 1$ . The long-dashed curves give the functions necessary for dealing with first derivatives. The **green dashed curve** is  $\partial k_{SE}/\partial x_j = \text{cov}[y(o), y'(x)]$ . The **red dashed curve** is  $\partial k_{SE}/\partial x_i = \text{cov}[y'(o), y(x)]$ . The opposite signs of the **red** and **green** curves may be disturbing on first glance, given that the covariance kernel must give rise to a symmetric positive definite covariance matrix. But, the symmetry that matters here is of locations *and* derivatives:  $\text{cov}[y(o), y'(x)] = \text{cov}[y'(x), y(o)]$ , which *does* hold. The **teal dashed curve** is  $\partial^2 k_{SE}/\partial x_i \partial x_j = \text{cov}[y'(o), y'(x)]$ . The dotted curves give the functions necessary for dealing with second derivatives. The **magenta dotted curve** is  $\partial^2 k_{SE}/\partial x_i^2 = \partial^2 k_{SE}/\partial x_j^2 = \text{cov}[y''(o), y(x)] = \text{cov}[y(o), y''(x)]$ . The **yellow dotted curve** is  $\partial^3 k_{SE}/\partial x_i \partial x_j^2 = \text{cov}[y'(o), y''(x)]$ . The **black dotted curve** is  $\partial^3 k_{SE}/\partial x_i^2 \partial x_j = \text{cov}[y''(o), y'(x)]$ . The **blue dotted curve** is  $\partial^4 k_{SE}/\partial x_i^2 \partial x_j^2 = \text{cov}[y''(o), y''(x)]$ . This figure was produced using `kernel_demo.py`.



**Figure B.2:** Two-dimensional squared exponential covariance kernel and a few of its derivatives. To remove the dependence on the hyperparameters, the  $x$  axis has been scaled by  $\ell_x$ , the  $y$  axis by  $\ell_y$  and the vertical axis by  $\sigma_f^2$ . Positive values are red, negative values are blue and zero values are white. The subscripts in the titles indicate partial derivatives:  $z_{xy} = \partial^2 z / (\partial x \partial y)$ , for instance. This figure was produced using `kernel_demo_2d.py`.

the derivatives of which are [294, 467]

$$\frac{d^n k_{\text{RQ}}(y)}{dy^n} = \sigma_f^2 (-\alpha)(-\alpha - 1) \dots (-\alpha - n + 1) y^{-\alpha-n}. \quad (\text{B.26})$$

This can be simplified<sup>1</sup> through the introduction of the Pochhammer symbol for the rising factorial [294, 295, 347, 468–471]:

$$(a)_n = \prod_{k=0}^{n-1} (a+k) = \frac{\Gamma(a+n)}{\Gamma(a)}, \quad (\text{B.27})$$

where  $\Gamma(a)$  is the gamma function [294, 295]. For the case of integer  $a \leq 0$  and  $n \leq -a$ , the latter expression with the gamma function breaks down, and the correct expression is [468, 472]:

$$(a)_n = \frac{(-1)^n (-a)!}{(-a-n)!}. \quad (\text{B.28})$$

Expressing equation (B.26) in terms of the Pochhammer symbol yields

$$\frac{d^n k_{\text{RQ}}(y)}{dy^n} = \sigma_f^2 (1 - \alpha - n)_n y^{-\alpha-n}. \quad (\text{B.29})$$

Applying the chain rule to this is far more involved than for the SE covariance kernel, as  $y$  does not factor out nicely into a series of independent products like equation (B.3). Instead, apply Faà di Bruno's formula, given here in the very useful combinatorial form presented in [473]:

$$\frac{\partial^n}{\partial x_{[1]} \dots \partial x_{[n]}} f(y(\mathbf{x})) = \sum_{\pi \in \Pi} f^{(|\pi|)}(y(\mathbf{x})) \prod_{B \in \pi} \frac{\partial^{|\mathcal{B}|} y(\mathbf{x})}{\prod_{j \in B} \partial x_{[j]}}. \quad (\text{B.30})$$

The notation for the partial derivatives here is necessarily somewhat more involved than what has been used elsewhere in this section. The variables  $x_{[1]}$  through  $x_{[n]}$  are placeholders for which of the elements of  $\mathbf{x} = [x_1, x_2, \dots, x_d]^T$  to differentiate with respect to. In other words, to take the second derivative with respect to  $x_3$  (i.e.,  $\partial^2 f / \partial x_3^2$ ), in equation (B.30) one would set  $n = 2$ ,  $x_{[1]} = x_3$ ,  $x_{[2]} = x_3$ . To compute  $\partial^4 f / \partial x_3^2 \partial x_4 \partial x_{10}$ , one would set  $n = 4$ ,  $x_{[1]} = x_3$ ,  $x_{[2]} = x_3$ ,  $x_{[3]} = x_4$ ,  $x_{[4]} = x_{10}$ . The other non-trivial nomenclature in equation (B.30) is as follows:

1. The Pochhammer symbol is in fact terrible notation, and is used to denote the rising or the falling factorial depending on the context. But SciPy [293] provides an implementation of the rising factorial as defined in equation (B.27) and resources such as [294, 347, 468] use it extensively, so this notation is used to make it clear what is used in the `gptools` implementation.

- $\Pi$  is the set of all possible partitions of the set  $\{1, \dots, n\}$ . A *partition* is a division of a set into subsets, where each element appears in one and only one subset [474]. Each subset in a partition is referred to as a *block*.
- $\pi$  is one such partition.
- $|\pi|$  is the number of blocks in partition  $\pi$ .
- $B$  is a block in the partition  $\pi$ .
- $|B|$  is the number of elements in block  $B$ .

In order to use this expression, first note

$$\frac{\partial y}{\partial \tau_d} = \frac{\tau_d}{\alpha \ell_d^2}, \quad \frac{\partial^2 y}{\partial \tau_d^2} = \frac{1}{\alpha \ell_d^2}, \quad \frac{\partial^3 y}{\partial \tau_d^3} = 0, \quad \frac{\partial^2 y}{\partial \tau_d \partial \tau_e} = 0. \quad (\text{B.31})$$

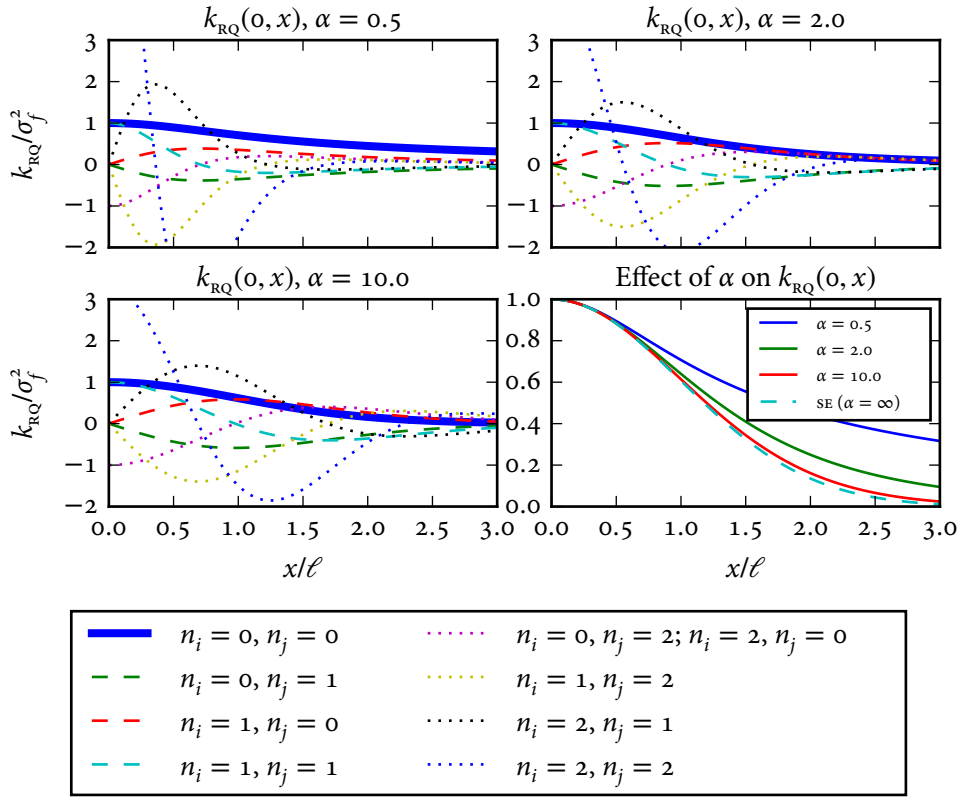
This means that only blocks of length one and length two with both elements referring to the same dimension of  $\boldsymbol{\tau}$  need to be considered. The derivatives of  $\tau_d$  with respect to  $x_{i_d}$  and  $x_{j_d}$  are as in equation (B.10) and equation (B.11). It is not possible to write quite so direct of a formula as equation (B.14), but these pieces are what are used to implement the rational quadratic covariance kernel in `gptools`. Refer to section E.4.1.3 and section E.4.4 for more details on the actual implementation. Plots of the 1D RQ covariance kernel and its first few derivatives for a few values of  $\alpha$  are provided in figure B.3 and plots of the 2D RQ covariance kernel and its first few derivatives for  $\alpha = 1/2$  are provided in figure B.4.

### B.3 The Matérn covariance kernel

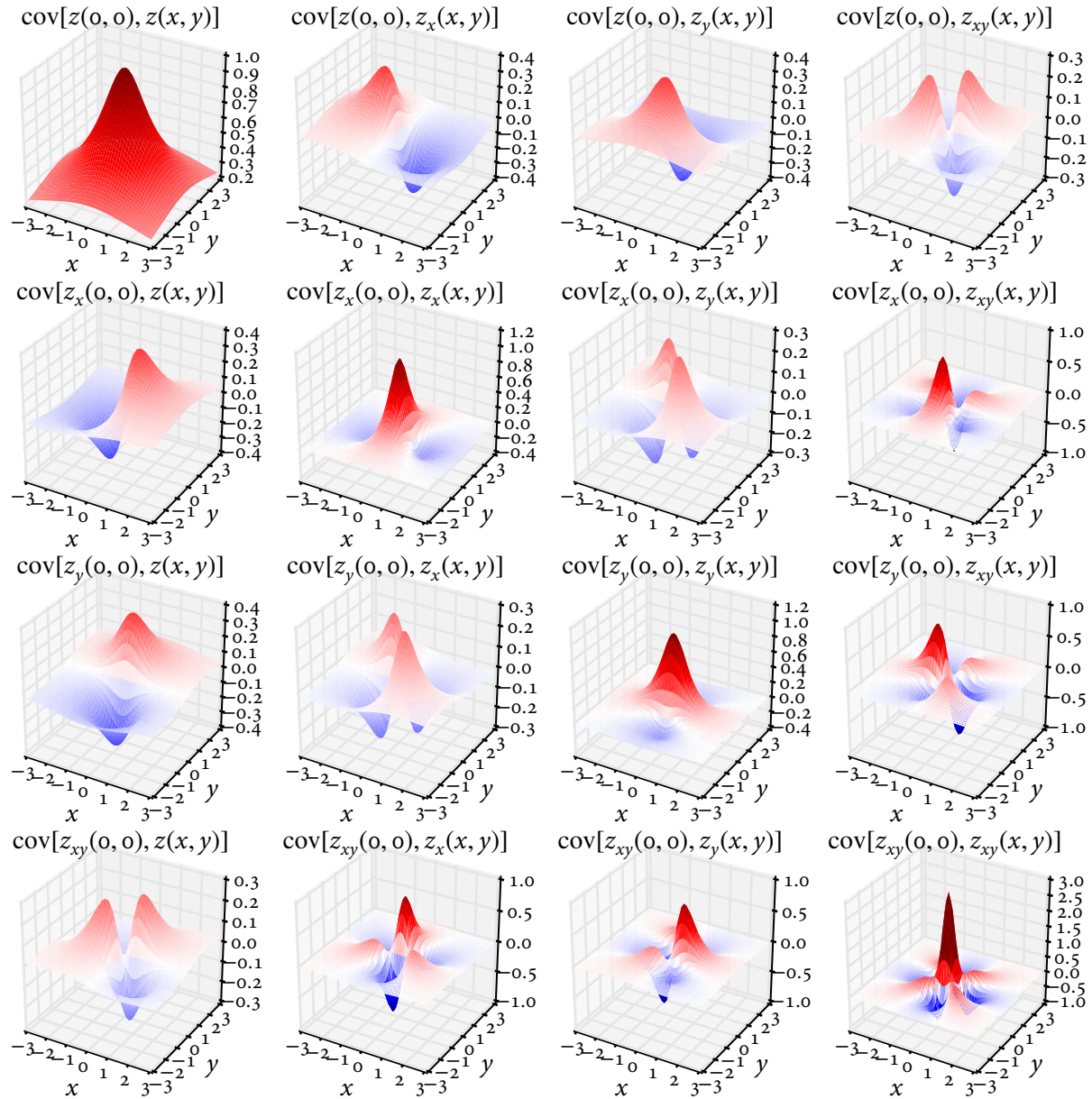
The Matérn covariance kernel is a stationary, isotropic covariance kernel with the basic form

$$k_M(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu}r}{\ell} \right)^\nu K_\nu \left( \frac{\sqrt{2\nu}r}{\ell} \right), \quad (\text{B.32})$$

where  $K_\nu(z)$  is the modified Bessel function of the second kind [294, 475]. The Matérn covariance function yields curves which are only differentiable up to order  $n < \nu$ . As  $\nu \rightarrow \infty$ , the Matérn covariance kernel reduces to the SE covariance kernel with covariance length scale  $\ell$ . It has been suggested that the Matérn covariance kernel is better suited to describe real data than the SE covariance kernel, as the infinite differentiability implied by the SE covariance kernel is a rather strong constraint [301].



**Figure B.3:** One-dimensional rational quadratic covariance kernel and its derivatives for  $\alpha \in \{1/2, 2, 10\}$  and a comparison of  $k_{\text{RQ}}$  for  $\alpha \in \{1/2, 2, 10, \infty\}$ . The plots for the various values of  $\alpha$  are presented in the same manner as figure B.1, and have the same scale. Notice that the curves have very long tails for small  $\alpha$ , which then move down towards the limiting SE case as  $\alpha \rightarrow \infty$ . This figure was produced using `kernel_demo.py`.



**Figure B.4:** Two-dimensional rational quadratic covariance kernel and a few of its derivatives for  $\alpha = 1/2$ , presented as in figure B.2. This figure was produced using `kernel_demo_2d.py`.



Making the substitution from [equation \(B.1\)](#) yields

$$k_M(\mathbf{x}_i - \mathbf{x}_j) = k_M(\boldsymbol{\tau}) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( 2\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \right)^{\nu/2} K_\nu \left( \left( 2\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \right)^{1/2} \right). \quad (\text{B.33})$$

To get this into a form which can be differentiated, define

$$y = 2\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2}. \quad (\text{B.34})$$

The Matérn covariance kernel is then

$$k_M(y) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} y^{\nu/2} K_\nu(y^{1/2}). \quad (\text{B.35})$$

To differentiate this, note that Faà di Bruno's formula can be expressed in a particularly useful form for single variables using the partial Bell polynomials  $B_{n,k}$  [[347](#), [476](#), [477](#)]:

$$\frac{d^n}{dx^n} f(g(x)) = \sum_{k=1}^n f^{(k)}(g(x)) B_{n,k}(g'(x), g''(x), \dots, g^{(n-k+1)}(x)) \quad (\text{B.36})$$

$$B_{n,k}(x_1, x_2, \dots, x_{n-k+1}) = \sum n! \prod_{i=1}^{n-k+1} \frac{1}{j_i!} \left( \frac{x_i}{i!} \right)^{j_i}, \quad (\text{B.37})$$

where the sum is over all sets of  $j = \{j_i\}$  such that  $\sum_{i=1}^{n-k+1} j_i = k$  and  $\sum_{i=1}^{n-k+1} ij_i = n$ . The partial Bell polynomials can be written in a useful recursive form as [[478](#)]:

$$B_{n,k}(x_1, x_2, \dots, x_{n-k+1}) = \sum_{i=1}^{n-k+1} \binom{n-1}{i-1} x_i B_{n-i,k-1} \quad (\text{B.38})$$

$$B_{0,0} = 1, \quad B_{n,0} = \delta_n, \quad B_{0,k} = \delta_k, \quad (\text{B.39})$$

where

$$\delta_n = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.40})$$

is the Kronecker delta. Note that the derivative of the modified Bessel function of the second kind is [[468](#), [479](#)]:

$$\frac{d^n}{dy^n} K_\nu(y) = \frac{(-1)^n}{2^n} \sum_{k=0}^n \binom{n}{k} K_{2k-n+\nu}(y). \quad (\text{B.41})$$

Also recall that [294, 467]

$$\frac{d^n}{dy^n} y^a = (1 + a - n)_n y^{a-n}, \quad (\text{B.42})$$

so

$$\frac{d^n y^{1/2}}{dy^n} = (3/2 - n)_n y^{1/2-n} \quad (\text{B.43})$$

The derivatives of  $K_\nu(y^{1/2})$  are then

$$\begin{aligned} \frac{d^n}{dy^n} K_\nu(y^{1/2}) &= \sum_{k=1}^n \left( \frac{(-1)^k}{2^k} \sum_{j=0}^k \binom{k}{j} K_{2j-k+\nu}(y^{1/2}) \right) \\ &\cdot \mathbf{B}_{n,k} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - n + k - 1)_{n-k+1} y^{1/2-n+k-1} \right) \end{aligned} \quad (\text{B.44})$$

Apply the general Leibniz rule [347, 480]

$$\frac{d^n}{dy^n} (a(y)b(y)) = \sum_{k=0}^n \binom{n}{k} \frac{d^k a(y)}{dy^k} \frac{d^{n-k} b(y)}{dy^{n-k}} \quad (\text{B.45})$$

to obtain the derivatives of  $k_M(y)$ :

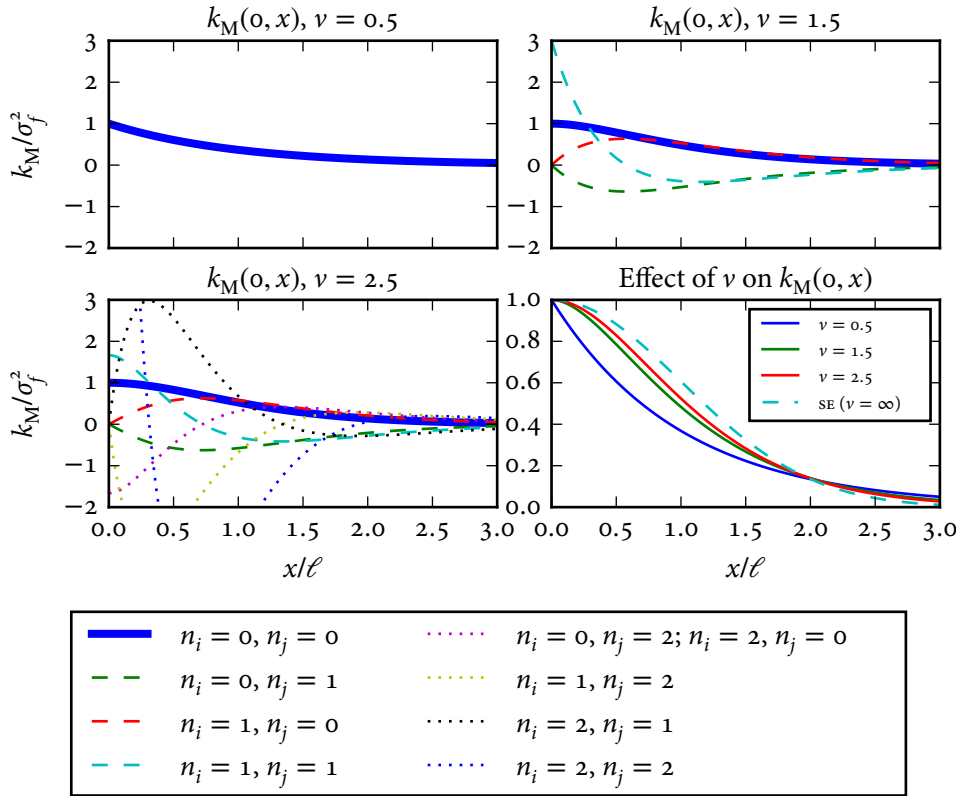
$$\frac{d^n k_M(y)}{dy^n} = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \sum_{k=0}^n \binom{n}{k} \frac{d^k y^{\nu/2}}{dy^k} \frac{d^{n-k} K_\nu(y^{1/2})}{dy^{n-k}} \quad (\text{B.46})$$

$$\begin{aligned} &= \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \sum_{k=0}^n \binom{n}{k} (1 + \nu/2 - k)_k y^{\nu/2-k} \sum_{j=1}^{n-k} \left( \frac{(-1)^j}{2^j} \sum_{i=0}^j \binom{j}{i} K_{2i-j+\nu}(y^{1/2}) \right) \right. \\ &\quad \left. \cdot \mathbf{B}_{n-k,j} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - n + k + j - 1)_{n-k-j+1} y^{1/2-n+k+j-1} \right) \right). \end{aligned} \quad (\text{B.47})$$

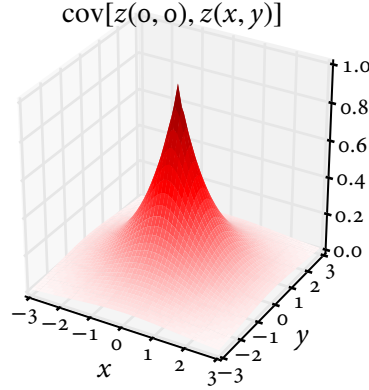
Finally, note that

$$\frac{\partial y}{\partial \tau_d} = \frac{4\nu\tau_d}{\ell_d^2}, \quad \frac{\partial^2 y}{\partial \tau_d^2} = \frac{4\nu}{\ell_d^2}, \quad \frac{\partial^3 y}{\partial \tau_d^3} = 0, \quad \frac{\partial^2 y}{\partial \tau_d \partial \tau_e} = 0. \quad (\text{B.48})$$

These expressions can then be used with [equation \(B.30\)](#) to compute the derivatives of the Matérn covariance kernel. Refer to [section E.4.1.3](#) and [section E.4.5](#) for more details on the actual implementation. Plots of the 1D Matérn covariance kernel for various values of  $\nu$  are given in [figure B.5](#) and plots of the 2D Matérn covariance kernel with various values of  $\nu$  are given in [figure B.6](#) through [figure B.8](#).



**Figure B.5:** One-dimensional Matérn covariance kernel and its derivatives for  $\nu \in \{1/2, 3/2, 5/2\}$  and a comparison of  $k_M$  for  $\nu \in \{1/2, 3/2, 5/2, \infty\}$ . The plots for the various values of  $\nu$  are presented in the same manner as figure B.1, and have the same scale. Only the derivatives needed for  $n < \nu$  are shown. Notice that the curves for small  $\nu$  have wider tails but narrower cores than the SE covariance kernel, and gradually approach the SE covariance kernel's shape as  $\nu \rightarrow \infty$ . This figure was produced using `kernel_demo.py`.



**Figure B.6:** Two-dimensional Matérn covariance kernel for  $\nu = 1/2$ , presented as in figure B.2. Just  $\text{cov}[z(o, o), z(x, y)]$  is shown. Notice the cusp at the origin – this corresponds to the fact that a Gaussian process with a Matérn covariance kernel with  $\nu = 1/2$  is not differentiable. This figure was produced using `kernel_demo_2d.py`.

### B.3.1 Handling the behavior as $y \rightarrow 0$

Equation (B.47) is very poorly behaved near the origin, as  $K_\nu(y) \propto y^{-\nu}$  as  $y \rightarrow 0$ . Therefore, successful implementation of equation (B.47) requires careful attention to the behavior as  $y \rightarrow 0$ . Because the modified Bessel function of the second kind for integer order  $\nu$  is defined with a limit, the cases where  $\nu$  is an integer versus not require separate handling.

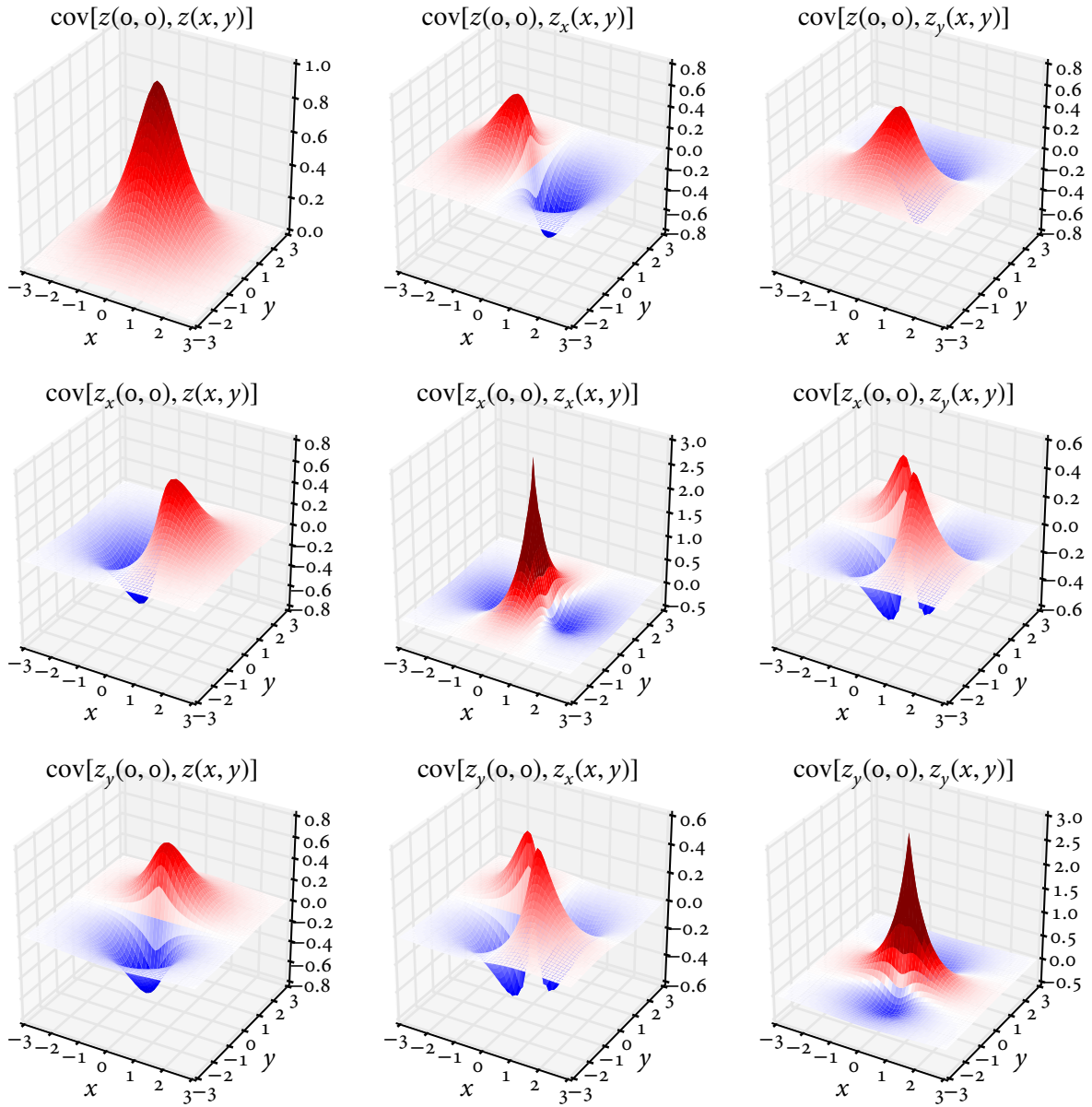
#### B.3.1.1 $\nu$ is not an integer

In all of the following, assume  $\nu \geq 1/2$ . The power series for  $K_\nu(z)$  for  $\nu \notin \mathbb{Z}$  is [468, 481]:

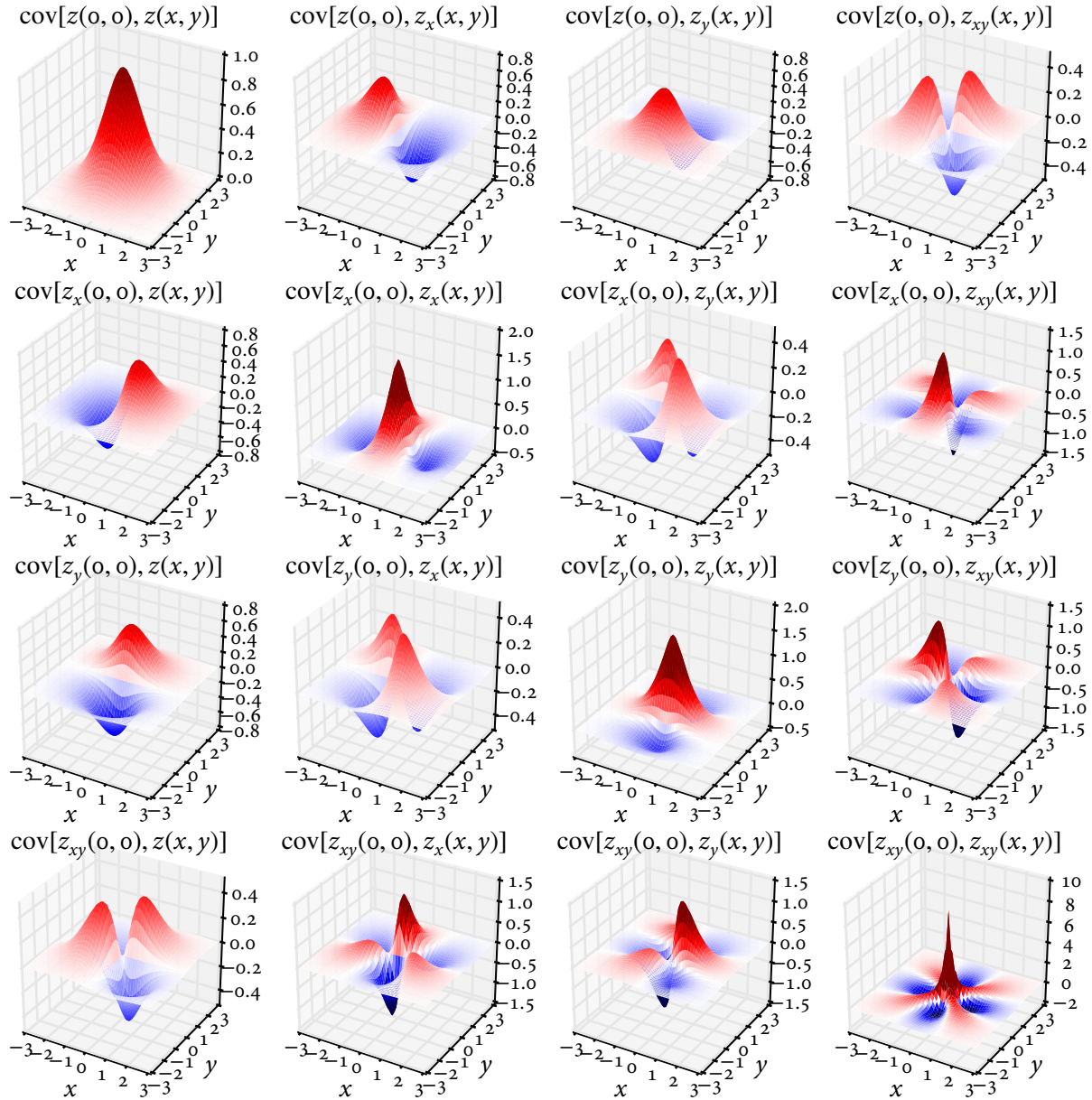
$$K_\nu(z) = \frac{\Gamma(\nu)}{2^{1-\nu}} z^{-\nu} \sum_{k=0}^{\infty} \frac{z^{2k}}{2^{2k}(1-\nu)_k k!} + \frac{\Gamma(-\nu)}{2^{1+\nu}} z^\nu \sum_{k=0}^{\infty} \frac{z^{2k}}{2^{2k}(1+\nu)_k k!}. \quad (\text{B.49})$$

Now substitute  $z = y^{1/2}$  and multiply this by  $y^{v/2}$ :

$$y^{v/2} K_\nu(y^{1/2}) = \Gamma(\nu) \sum_{k=0}^{\infty} \frac{y^k}{2^{1-\nu+2k}(1-\nu)_k k!} + \Gamma(-\nu) \sum_{k=0}^{\infty} \frac{y^{v+k}}{2^{1+\nu+2k}(1+\nu)_k k!}. \quad (\text{B.50})$$



**Figure B.7:** Two-dimensional Matérn covariance kernel and a few of its derivatives for  $\nu = 3/2$ , presented as in figure B.2. Notice the cusp at the origin of the  $n_{i_x} = n_{j_x} = 1$  and  $n_{i_y} = n_{j_y} = 1$  plots – this corresponds to the fact that a Gaussian process with a Matérn covariance kernel with  $\nu = 3/2$  can only be differentiated once. This figure was produced using `kernel_demo_2d.py`.



**Figure B.8:** Two-dimensional Matérn covariance kernel and a few of its derivatives for  $\nu = 5/2$ , presented as in figure B.2. Notice the cusp at the origin of the  $n_x = n_y = 1$  plot – this corresponds to the fact that a Gaussian process with a Matérn covariance kernel with  $\nu = 5/2$  can only be differentiated twice. This figure was produced using `kernel_demo_2d.py`.

Applying equation (B.42), the derivative of  $y^{\nu/2}\mathbf{K}_\nu(y^{1/2})$  is

$$\frac{d^n}{dy^n}(y^{\nu/2}\mathbf{K}_\nu(y^{1/2})) = \Gamma(\nu) \sum_{k=n}^{\infty} \frac{(1+k-n)_n y^{k-n}}{2^{1-\nu+2k}(1-\nu)_k k!} + \Gamma(-\nu) \sum_{k=0}^{\infty} \frac{(1+\nu+k-n)_n y^{\nu+k-n}}{2^{1+\nu+2k}(1+\nu)_k k!}, \quad (\text{B.51})$$

where the first sum can start from  $k = n$  because  $(1+k-n)_n = 0$  for  $n > k$  when  $n$  and  $k$  are nonnegative integers. The first sum can never have any terms which diverge, since  $k$  and  $n$  are both integers. So, the second term is what determines if the function is finite or not. It will diverge first (and most strongly) for  $k = 0$ , which means that the criterion for the sum to diverge is  $n > \nu$ . For  $n \leq \nu$ , the limit is determined by the first surviving term in each sum. For the case where  $\nu$  is not an integer, the second sum can never have an exponent of zero, so we only need to consider the first sum. The first term which survives is  $k = n$ , so the limit for  $\nu \notin \mathbb{Z} \wedge n \leq \nu$  is

$$\lim_{y \rightarrow 0} \frac{d^n}{dy^n}(y^{\nu/2}\mathbf{K}_\nu(y^{1/2})) = \frac{\Gamma(\nu)(1)_n}{2^{1-\nu+2n}(1-\nu)_n n!} = \frac{\Gamma(\nu)\Gamma(n+1)}{2^{1-\nu+2n}(1-\nu)_n n!}, \quad (\text{B.52})$$

where the substitution  $(1)_n = \Gamma(n+1)/\Gamma(1) = \Gamma(n+1)$  was made.

There is one final piece necessary to handle this case: note that this result will be used in equation (B.30), where it may be multiplied by one or more factors of the form given in equation (B.48). So, near  $y = 0$ ,

$$\begin{aligned} \frac{d^{|\pi|}}{dy^{|\pi|}}(y^{\nu/2}\mathbf{K}_\nu(y^{1/2})) \prod_{B \in \pi} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j} \approx \\ \left( \frac{\Gamma(\nu)\Gamma(n+1)}{2^{1-\nu+2n}(1-\nu)_n n!} + \frac{\Gamma(-\nu)(1+\nu-n)_n y^{\nu-n}}{2^{1+\nu}} \right) \prod_{\substack{B \in \pi \\ |B|=1}} \left( \frac{4\nu\tau_b}{\ell_b^2} \right) \prod_{\substack{B \in \pi \\ |B| \neq 1}} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j}, \end{aligned} \quad (\text{B.53})$$

where  $b \in B$  refers to the single element of the block of size one. If there are any blocks with  $|B| = 2$  and the indices in the block are not equal or  $|B| > 2$  then this is zero. Otherwise, the behavior as  $\boldsymbol{\tau}$  becomes the zero vector must be handled carefully. Write equation (B.53) in terms of  $\tau_d$  and ignore the first term inside the parenthesis (which cannot diverge) to obtain:

$$\frac{\Gamma(-\nu)(1+\nu-n)_n}{2^{1+\nu}} \left( 2\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \right)^{\nu-n} \prod_{\substack{B \in \pi \\ |B|=1}} \left( \frac{4\nu\tau_b}{\ell_b^2} \right) \prod_{\substack{B \in \pi \\ |B| \neq 1}} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j}. \quad (\text{B.54})$$

Now, consider two possible paths to take the limit as  $\tau \rightarrow 0$ . Let  $T_{|B|=1}$  denote the set of variables which are included in a block of size one and let  $n_1$  be the number of blocks in  $\pi$  which are size one. (Note that  $n_1 \geq |T_{|B|=1}|$  because there may be multiple single derivatives with respect to a given variable.) For the first path, hold all  $\tau_d \in T_{|B|=1}$  fixed at non-zero values while taking the other  $\tau_d \notin T_{|B|=1}$  to zero. Then take the  $\tau_d \in T_{|B|=1}$  to zero along the path  $\tau_d = \tau$  for all  $\tau_d \in T_{|B|=1}$ . Mathematically, the first step of this path gives

$$\frac{\Gamma(-\nu)(1 + \nu - n)_n}{2^{1+\nu}} \left( 2^\nu \sum_{\substack{B \in \pi \\ |B|=1}} \frac{\tau_b^2}{\ell_b^2} \right)^{\nu-n} \prod_{\substack{B \in \pi \\ |B|=1}} \left( \frac{4\nu\tau_b}{\ell_b^2} \right) \prod_{\substack{B \in \pi \\ |B| \neq 1}} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j}, \quad (\text{B.55})$$

and the second step gives

$$\frac{\Gamma(-\nu)(1 + \nu - n)_n}{2^{1+\nu}} \left( 2^\nu \tau^2 \sum_{\substack{B \in \pi \\ |B|=1}} \frac{1}{\ell_b^2} \right)^{\nu-n} \prod_{\substack{B \in \pi \\ |B|=1}} \left( \frac{4\nu\tau}{\ell_b^2} \right) \prod_{\substack{B \in \pi \\ |B| \neq 1}} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j} \quad (\text{B.56})$$

$$= \tau^{2\nu-2n+n_1} \frac{\Gamma(-\nu)(1 + \nu - n)_n}{2^{1+\nu}} \left( 2^\nu \sum_{\substack{B \in \pi \\ |B|=1}} \frac{1}{\ell_b^2} \right)^{\nu-n} \prod_{\substack{B \in \pi \\ |B|=1}} \frac{4\nu}{\ell_b^2} \prod_{\substack{B \in \pi \\ |B| \neq 1}} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j}. \quad (\text{B.57})$$

This goes to 0 if  $2\nu - 2n + n_1 > 0$ , diverges if  $2\nu - 2n + n_1 < 0$  and can obtain a finite value if  $2\nu - 2n + n_1 = 0$ . But now consider the path which takes *all* variables to zero along the path  $\tau_d = \tau$  at the same time. Mathematically, this gives

$$\begin{aligned} & \frac{\Gamma(-\nu)(1 + \nu - n)_n}{2^{1+\nu}} \left( 2^\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \right)^{\nu-n} \prod_{\substack{B \in \pi \\ |B|=1}} \left( \frac{4\nu\tau}{\ell_b^2} \right) \prod_{\substack{B \in \pi \\ |B| \neq 1}} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j} \quad (\text{B.58}) \\ & = \tau^{2\nu-2n+n_1} \frac{\Gamma(-\nu)(1 + \nu - n)_n}{2^{1+\nu}} \left( 2^\nu \sum_{d=1}^D \frac{1}{\ell_d^2} \right)^{\nu-n} \prod_{\substack{B \in \pi \\ |B|=1}} \left( \frac{4\nu}{\ell_b^2} \right) \prod_{\substack{B \in \pi \\ |B| \neq 1}} \frac{\partial^{|B|} y(\boldsymbol{\tau})}{\prod_{j \in B} \partial \tau_j}. \quad (\text{B.59}) \end{aligned}$$

This again goes to zero if  $2\nu - 2n + n_1 > 0$ , which implies that the limit can exist in this case. But, when  $2\nu - 2n + n_1 = 0$ , the limiting value includes a sum over *all*  $1/\ell_d^2$  instead of just the  $1/\ell_b^2$  for which  $b$  are the elements of the blocks with size one. Therefore, the limit does not exist in this case.



**B.3.1.2  $\nu$  is an integer**

The case for  $\nu \in \mathbb{Z}$  is much more involved, as the convenient power series given in equation (B.49) breaks down. Instead,  $K_\nu(z)$  is given by the far more cumbersome series [294, 482]

$$\begin{aligned} K_\nu(z) &= \frac{1}{2^{1-\nu}} z^{-\nu} \sum_{k=0}^{\nu-1} \frac{(-1)^k (\nu - k - 1)!}{2^{2k} k!} z^{2k} \\ &\quad + (-1)^{\nu+1} \ln\left(\frac{z}{2}\right) I_\nu(z) \\ &\quad + \frac{(-1)^\nu}{2^{1+\nu}} z^\nu \sum_{k=0}^{\infty} \frac{\Psi(k+1) + \Psi(\nu + k + 1)}{2^{2k} k! (\nu + k)!} z^{2k}, \end{aligned} \quad (\text{B.60})$$

where  $I_\nu(z)$  is the modified Bessel function of the first kind [294, 475] and  $\psi(z) = \Gamma'(z)/\Gamma(z)$  is the digamma function [294, 295]. Substituting  $z = y^{1/2}$  and multiplying by  $y^{\nu/2}$  gives

$$\begin{aligned} y^{\nu/2} K_\nu(y^{1/2}) &= \sum_{k=0}^{\nu-1} \frac{(-1)^k (\nu - k - 1)!}{2^{1-\nu+2k} k!} y^k \\ &\quad + (-1)^{\nu+1} y^{\nu/2} \ln\left(\frac{y^{1/2}}{2}\right) I_\nu(y^{1/2}) \\ &\quad + (-1)^\nu \sum_{k=0}^{\infty} \frac{\Psi(k+1) + \Psi(\nu + k + 1)}{2^{1+\nu+2k} k! (\nu + k)!} y^{\nu+k}. \end{aligned} \quad (\text{B.61})$$

In order to differentiate this series, note the derivative of the modified Bessel function of the first kind [468, 483]:

$$\frac{d^n I_\nu(z)}{dz^n} = 2^{-n} \sum_{k=0}^n \binom{n}{k} I_{2k-n+\nu}(z). \quad (\text{B.62})$$

Applying equation (B.36), the derivatives of  $I_\nu(y^{1/2})$  are

$$\begin{aligned} \frac{d^n I_\nu(y^{1/2})}{dy^n} &= \sum_{k=1}^n 2^{-k} \left( \sum_{j=0}^k \binom{k}{j} I_{2j-k+\nu}(y^{1/2}) \right) \\ &\quad \cdot B_{n,k} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - n + k - 1)_{n-k+1} y^{1/2-n+k-1} \right). \end{aligned} \quad (\text{B.63})$$

The derivatives of the logarithm in the middle term of equation (B.61) are [468, 484]

$$\frac{d^n}{dy^n} \ln\left(\frac{y^{1/2}}{2}\right) = \frac{d^n}{dy^n} \left( \frac{1}{2} \ln y - \ln 2 \right) = \frac{S_n^{(1)}}{2y^n} + \delta_n \ln\left(\frac{y^{1/2}}{2}\right), \quad (\text{B.64})$$

where

$$S_n^{(1)} = \begin{cases} 0, & n < 1 \\ (-1)^{n-1}(n-1)!, & \text{otherwise} \end{cases} \quad (\text{B.65})$$

is the first Stirling number of the first kind [347, 485]. Using the general Leibniz rule, the derivatives of  $y^{v/2} \ln(y^{1/2}/2)$  are

$$\frac{d^n}{dy^n} \left( y^{v/2} \ln \left( \frac{y^{1/2}}{2} \right) \right) = \sum_{k=0}^n \binom{n}{k} (1 + v/2 - k)_k y^{v/2-k} \left( \frac{S_{n-k}^{(1)}}{2y^{n-k}} + \delta_{n-k} \ln \left( \frac{y^{1/2}}{2} \right) \right) \quad (\text{B.66})$$

$$= \sum_{k=0}^n \binom{n}{k} (1 + v/2 - k)_k \left( \frac{S_{n-k}^{(1)}}{2y^{n-v/2}} + \delta_{n-k} y^{v/2-k} \ln \left( \frac{y^{1/2}}{2} \right) \right). \quad (\text{B.67})$$

But the first term in the parenthesis is only nonzero for  $k < n$  and the last term is only nonzero for  $k = n$ , so the sum simplifies to

$$= (1 + v/2 - n)_n y^{v/2-n} \ln \left( \frac{y^{1/2}}{2} \right) + \sum_{k=0}^{n-1} \binom{n}{k} \frac{(1 + v/2 - k)_k S_{n-k}^{(1)}}{2y^{n-v/2}}. \quad (\text{B.68})$$

Using this, the derivatives of  $y^{v/2} \ln(y^{1/2}/2) I_\nu(y^{1/2})$  are

$$\begin{aligned} \frac{d^n}{dy^n} \left( y^{v/2} \ln \left( \frac{y^{1/2}}{2} \right) I_\nu(y^{1/2}) \right) &= \sum_{k=0}^n \binom{n}{k} \frac{d^k}{dy^k} \left( y^{v/2} \ln \left( \frac{y^{1/2}}{2} \right) \right) \frac{d^{n-k}}{dy^{n-k}} (I_\nu(y^{1/2})) \\ &= \sum_{k=0}^{\infty} \binom{n}{k} \left( (1 + v/2 - k)_k y^{v/2-k} \ln \left( \frac{y^{1/2}}{2} \right) + \sum_{j=0}^{k-1} \binom{k}{j} \frac{(1 + v/2 - j)_j S_{k-j}^{(1)}}{2y^{k-v/2}} \right) \\ &\quad \cdot \left( \sum_{j=1}^k 2^{-j} \left( \sum_{i=0}^j \binom{j}{i} I_{2i-j+\nu}(y^{1/2}) \right) \right) \\ &\quad \cdot \mathbf{B}_{k,j} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - k + j - 1)_{k-j+1} y^{1/2-k+j-1} \right) \end{aligned} \quad (\text{B.69})$$

$$\begin{aligned}
 &= \sum_{k=0}^{\infty} \binom{n}{k} (1 + \nu/2 - k)_k y^{\nu/2-k} \ln\left(\frac{y^{1/2}}{2}\right) \left( \sum_{j=1}^k \frac{1}{2^j} \left( \sum_{i=0}^j \binom{j}{i} I_{2i-j+\nu}(y^{1/2}) \right) \right) \\
 &\quad \cdot B_{k,j} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - k + j - 1)_{k-j+1} y^{1/2-k+j-1} \right) \\
 &+ \sum_{k=0}^{\infty} \binom{n}{k} \left( \sum_{j=0}^{k-1} \binom{k}{j} \frac{(1 + \nu/2 - j)_j S_{k-j}^{(1)}}{y^{k-\nu/2}} \right) \left( \sum_{j=1}^k \frac{1}{2^{j+1}} \left( \sum_{i=0}^j \binom{j}{i} I_{2i-j+\nu}(y^{1/2}) \right) \right) \\
 &\quad \cdot B_{k,j} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - k + j - 1)_{k-j+1} y^{1/2-k+j-1} \right). \quad (\text{B.70})
 \end{aligned}$$

Therefore, the derivatives of equation (B.61) are

$$\begin{aligned}
 \frac{d^n}{dy^n} (y^{\nu/2} K_{\nu}(y^{1/2})) &= \\
 &\sum_{k=n}^{\nu-1} \frac{(-1)^k (\nu - k - 1)! (1 + k - n)_n}{2^{1-\nu+2k} k!} y^{k-n} \\
 &+ (-1)^{\nu+1} \sum_{k=0}^{\infty} \binom{n}{k} (1 + \nu/2 - k)_k y^{\nu/2-k} \ln\left(\frac{y^{1/2}}{2}\right) \left( \sum_{j=1}^k \frac{1}{2^j} \left( \sum_{i=0}^j \binom{j}{i} I_{2i-j+\nu}(y^{1/2}) \right) \right) \\
 &\quad \cdot B_{k,j} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - k + j - 1)_{k-j+1} y^{1/2-k+j-1} \right) \\
 &+ (-1)^{\nu+1} \sum_{k=0}^{\infty} \binom{n}{k} \left( \sum_{j=0}^{k-1} \binom{k}{j} \frac{(1 + \nu/2 - j)_j S_{k-j}^{(1)}}{y^{k-\nu/2}} \right) \left( \sum_{j=1}^k \frac{1}{2^{j+1}} \left( \sum_{i=0}^j \binom{j}{i} I_{2i-j+\nu}(y^{1/2}) \right) \right) \\
 &\quad \cdot B_{k,j} \left( \frac{1}{2y^{1/2}}, \frac{-1}{4y^{3/2}}, \dots, (3/2 - k + j - 1)_{k-j+1} y^{1/2-k+j-1} \right) \\
 &+ (-1)^{\nu} \sum_{k=\max(0, n-\nu)}^{\infty} \frac{(\psi(k+1) + \psi(\nu + k + 1)) (1 + \nu + k - n)_n}{2^{1+\nu+2k} k! (\nu + k)!} y^{\nu+k-n}. \quad (\text{B.71})
 \end{aligned}$$

This expression is extremely unwieldy, so what is done in practice is to interpolate between  $\nu - \varepsilon$  and  $\nu + \varepsilon$  for some small value of  $\varepsilon$ :

$$\frac{d^n}{dy^n} (y^{\nu/2} K_{\nu}(y^{1/2})) \approx \frac{1}{2} \left( \frac{d^n}{dy^n} (y^{(\nu-\varepsilon)/2} K_{\nu-\varepsilon}(y^{1/2})) + \frac{d^n}{dy^n} (y^{(\nu+\varepsilon)/2} K_{\nu+\varepsilon}(y^{1/2})) \right). \quad (\text{B.72})$$

## B.4 Handling the edge: nonstationary covariance kernels

The covariance kernels presented to this point have all been *stationary*: they were functions of  $\boldsymbol{\tau} = \mathbf{x}_i - \mathbf{x}_j$  only and are hence invariant to translations in space. In the context of the covariance length scale, a stationary covariance kernel has a constant covariance length scale throughout all of space. In a tokamak plasma, however, the assumption of stationarity is *not* justified: the profile is expected to be relatively macroscopically smooth in the core, then have a rapid change and go to zero at the edge. Other behavior such as an internal transport barrier or strongly localized heating can lead to other localized regions of rapid change. This section presents two possible ways of introducing non-stationarity into a Gaussian process: the Gibbs covariance kernel which is an SE covariance kernel with spatially-varying covariance length scale and warping of the inputs which allows any stationary covariance kernel to be made nonstationary. Most of the work in this thesis uses the Gibbs covariance kernel described in [section B.4.1](#). Note that another way to accomplish a nonstationary fit is to use a parametric mean function  $m(\mathbf{x})$  which fits the extra structure in the edge, and then use a stationary covariance kernel to fit the residuals from the fit. This approach was applied to fitting H-mode pedestal data with a modified hyperbolic tangent mean function in [section 2.5.5](#). Schemes have been devised for efficiently partitioning the domain into regions governed by different models [[486–488](#)], but this level of sophistication was not attempted in the present work. It should also be noted that the treed Gaussian process approach as presented in [[486–488](#)] generates realizations which are only piecewise continuous, and hence would require extensions to generate the differentiable fits required for the profile fitting task.

### B.4.1 The Gibbs covariance kernel

Gibbs [[395](#)] obtained the following nonstationary version of the SE covariance kernel:

$$k_G(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left( \frac{2\ell(\mathbf{x}_i)\ell(\mathbf{x}_j)}{\ell^2(\mathbf{x}_i) + \ell^2(\mathbf{x}_j)} \right)^{1/2} \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\ell^2(\mathbf{x}_i) + \ell^2(\mathbf{x}_j)}\right), \quad (\text{B.73})$$

where the covariance length scale  $\ell(\mathbf{x})$  is now an arbitrary function of  $\mathbf{x}$  and  $\sigma_f^2$  is the signal variance as before. It is important to note that the functional form of  $\ell(\mathbf{x})$  does *not* correspond to the functional form of the profile – it merely sets how fast the profile can vary in space. Letting  $\ell$  be a function of  $\mathbf{x}$  allows the profile to have regions with slowly varying spatial structure smoothly joined to regions with more rapidly varying spatial structure. Because of the complexity of [equation \(B.73\)](#), only the first few derivatives were worked out, and only the univariate case was handled.

First, define the following shorthands:

$$\tau = x_i - x_j, \quad \ell_i = \ell(x_i), \quad \ell_j = \ell(x_j), \quad \ell'_i = \left. \frac{d\ell(x)}{dx} \right|_{x=x_i}, \quad \ell'_j = \left. \frac{d\ell(x)}{dx} \right|_{x=x_j}. \quad (\text{B.74})$$

The derivatives relevant to the present work are:

$$\frac{\partial k_G(x_i, x_j)}{\partial x_i} = \frac{\ell_j \exp(-\tau^2/(\ell_i^2 + \ell_j^2))}{\sqrt{2\ell_i \ell_j (\ell_i^2 + \ell_j^2)^{5/2}}} (\ell'_i (-\ell_i^4 + \ell_j^4) - 4\tau \ell_i (\ell_i^2 + \ell_j^2) + 4\tau^2 \ell_i^2 \ell'_i) \quad (\text{B.75})$$

$$\frac{\partial k_G(x_i, x_j)}{\partial x_j} = \frac{\ell_i \exp(-\tau^2/(\ell_i^2 + \ell_j^2))}{\sqrt{2\ell_i \ell_j (\ell_i^2 + \ell_j^2)^{5/2}}} (\ell'_j (\ell_i^4 - \ell_j^4) + 4\tau \ell_j (\ell_i^2 + \ell_j^2) + 4\tau^2 \ell_j^2 \ell'_j) \quad (\text{B.76})$$

$$\begin{aligned} \frac{\partial^2 k_G(x_i, x_j)}{\partial x_i \partial x_j} &= \frac{\exp(-\tau^2/(\ell_i^2 + \ell_j^2))}{2\sqrt{2\ell_i \ell_j (\ell_i^2 + \ell_j^2)^{9/2}}} \\ &\cdot \left( -\ell_i^8 \ell'_i \ell'_j - \ell_j^8 \ell'_i \ell'_j + 8\ell_i^7 \ell_j + 8\ell_i \ell_j^7 + 8\ell_i^6 \ell_j^2 \ell'_i \ell'_j + 8\ell_i^2 \ell_j^6 \ell'_i \ell'_j \right. \\ &\quad \left. + 24\ell_i^5 \ell_j^3 + 24\ell_i^3 \ell_j^5 + 18\ell_i^4 \ell_j^4 \ell'_i \ell'_j \right. \\ &+ 4\tau (-\ell_i^7 \ell'_j + \ell_j^7 \ell'_i - 5\ell_i^6 \ell_j \ell'_i + 5\ell_i \ell_j^6 \ell'_j + 3\ell_i^5 \ell_j^2 \ell'_j - 3\ell_i^2 \ell_j^5 \ell'_i - 9\ell_i^4 \ell_j^3 \ell'_i + 9\ell_i^3 \ell_j^4 \ell'_j) \\ &\quad + 4\tau^2 (\ell_i^6 \ell'_i \ell'_j + \ell_j^6 \ell'_i \ell'_j - 4\ell_i^5 \ell_j - 4\ell_i \ell_j^5 - 9\ell_i^4 \ell_j^2 \ell'_i \ell'_j - 9\ell_i^2 \ell_j^4 \ell'_i \ell'_j - 8\ell_i^3 \ell_j^3) \\ &\quad \left. + 16\tau^3 (\ell_i^4 \ell_j \ell'_i - \ell_i \ell_j^4 \ell'_j - \ell_i^3 \ell_j^2 \ell'_j + \ell_i^2 \ell_j^3 \ell'_i) + 16\tau^4 \ell_i^2 \ell_j^2 \ell'_i \ell'_j \right). \quad (\text{B.77}) \end{aligned}$$

#### B.4.1.1 tanh covariance length scale function for tokamak profile fitting

In order to model a tokamak profile, we need a covariance length scale function with a core saturation value, a shorter edge saturation value to allow the rapid drop at the edge, and a smooth transition between the two. These requirements motivated the use of a hyperbolic tangent:

$$\ell(x) = \frac{\ell_1 + \ell_2}{2} - \frac{\ell_1 - \ell_2}{2} \tanh \frac{x - x_0}{\ell_w} \quad (\text{B.78})$$

$$\frac{d\ell(x)}{dx} = -\frac{\ell_1 - \ell_2}{2\ell_w} \operatorname{sech}^2 \frac{x - x_0}{\ell_w}, \quad (\text{B.79})$$

where  $\ell_1$  is the core saturation value,  $\ell_2$  is the edge saturation value,  $x_0$  is the location of the center of the transition between the two covariance length scales and  $\ell_w$  is the characteristic width of the transition. In light of the popularity of tanh-like functions for fitting pedestal data it is very important to recall that this is *not* in any way forcing the fitted curve to follow a tanh function – it merely dictates the spatial correlation length as described above. This formulation has the advantage that it yields a curve which is infinitely differentiable. Plots of the Gibbs covariance kernel with tanh covariance length scale function for hyperparameters relevant to tokamak profile fitting are given in [figure B.9](#). Covariance length scale functions consisting of two constant regions joined with either cubic or quintic polynomials were also tested, but were found to not produce fits as satisfactory as those using the hyperbolic tangent. This formulation was tested with synthetic data in [section 2.4](#). This formulation can easily be extended to include an arbitrary number of regions, for instance adding an extra region to fit a profile exhibiting an internal transport barrier (ITB).

#### B.4.1.2 More flexible covariance length scale functions

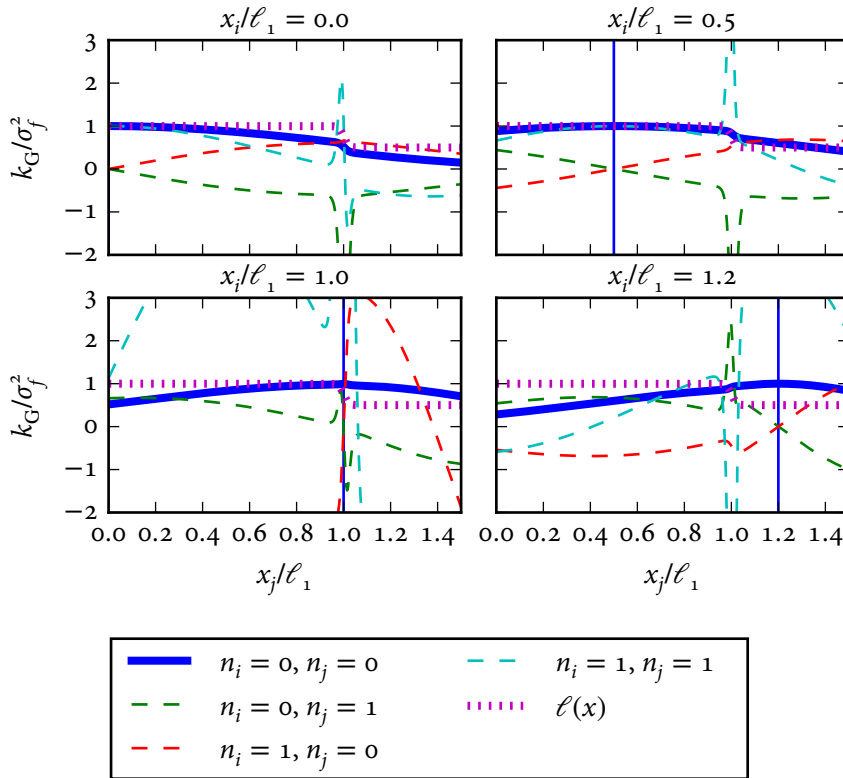
When sufficient data are available, a less restrictive covariance length scale function can be used. Gibbs [395] uses a set of Gaussian basis functions to construct the logarithm of the covariance length scale function, which in light of [equation \(2.19\)](#) is equivalent to constructing a Gaussian process to describe the logarithm of the covariance length scale function. Li [313] uses a Gaussian process to interpolate values of the covariance length scale (not its logarithm) given at a small number of specific points. Several approaches were tested as part of this thesis, and the results of the benchmarking with synthetic data are given in [section 2.4.3.2](#).

**B.4.1.2.1 Exponential of Gaussians** First, consider a slight modification of the exponential of Gaussian basis functions used in [395]:

$$\ell(x) = \ell_o \exp\left(\sum_{i=1}^N \beta_i \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right)\right) \quad (\text{B.80})$$

$$\frac{d\ell(x)}{dx} = -\ell_o \exp\left(\sum_{i=1}^N \beta_i \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right)\right) \sum_{i=1}^N \frac{\beta_i(x - \mu_i)}{\sigma_i^2} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right), \quad (\text{B.81})$$

where  $\ell_o$  is a positive value which sets the value of the covariance length scale function far from the basis functions,  $\mu_i$  are the centers of the  $N$  Gaussians,  $\sigma_i$  are the widths (i.e., standard deviations) of the Gaussians and  $\beta_i$  are the amplitude of each



**Figure B.9:** One-dimensional Gibbs covariance kernel  $k_G(x_i, x_j)$  with tanh covariance length scale function and a few of its derivatives for various values of  $x_i$ . The hyperparameters used in these figures are similar to those encountered for L-mode density profiles:  $\sigma_f = 1$ ,  $\ell_1 = 1$ ,  $\ell_2 = 0.5$ ,  $\ell_w = 0.2$ ,  $x_0 = 1.0$ . In each subfigure,  $x_i$  is indicated with the vertical blue line and  $x_j$  is given on the horizontal axis. As expected, the covariance between points drops noticeably across the pedestal. This figure was produced using `gibbs_demo.py`.

Gaussian basis function. While Gibbs [395] used a fixed grid of many basis functions with the same width, it was found that this often produced unphysical oscillations in the gradient. Instead, a small number of Gaussians was used and the centers  $\mu_i$  and widths  $\sigma_i$  were treated as free hyperparameters along with  $\ell_o$  and the weights  $\beta_i$ . This allows the inference to put changes in length scale where the data demand them while still keeping the curve smooth and with constant length scale elsewhere.

**B.4.1.2.2 B-splines** Another option is to use B-spline basis functions [321, 322]. These functions have mathematical properties which enable them to represent a wide variety of functions with a minimum number of parameters. The basis functions are piecewise polynomials of degree<sup>2</sup>  $d$ , and are represented using a nondecreasing set of  $n_t$  knots  $t_i$ . In practice,  $d$  copies of the edge knots are appended at each end of the domain. So, for  $d = 3$  (a cubic B-spline), the set of knots would end up as

$$\mathbf{t} = \{t_1, t_1, t_1, t_1, t_2, \dots, t_{n_t-1}, t_{n_t}, t_{n_t}, t_{n_t}, t_{n_t}\}, \quad (\text{B.82})$$

for a total of  $n_t + 2d$  knots. The basis functions are defined recursively [321]:

$$\mathbf{B}_{i,0}(x|\mathbf{t}) = \begin{cases} 1, & t_i \leq x < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.83})$$

$$\mathbf{B}_{i,d}(x|\mathbf{t}) = \frac{x - t_i}{t_{i+d} - t_i} \mathbf{B}_{i,d-1}(x|\mathbf{t}) + \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} \mathbf{B}_{i+1,d-1}(x|\mathbf{t}), \quad (\text{B.84})$$

where  $\mathbf{B}_{i,d}(x|\mathbf{t})$  is the  $i^{\text{th}}$  B-spline basis function of polynomial degree  $d$  with knot set  $\mathbf{t}$ . For a given set of  $n_t$  interior knots, this then defines  $n_t + d - 1$  non-zero basis functions. The derivatives of the basis functions are [321]

$$\frac{d\mathbf{B}_{i,d}(x|\mathbf{t})}{dx} = d \left( \frac{\mathbf{B}_{i,d-1}(x|\mathbf{t})}{t_{i+d} - t_i} - \frac{\mathbf{B}_{i+1,d-1}(x|\mathbf{t})}{t_{i+d+1} - t_{i+1}} \right) = \mathbf{M}_{i,d-1}(x|\mathbf{t}) - \mathbf{M}_{i+1,d-1}(x|\mathbf{t}), \quad (\text{B.85})$$

where

$$\mathbf{M}_{i,d}(x|\mathbf{t}) = \frac{d+1}{t_{i+d+1} - t_i} \mathbf{B}_{i,d}(x|\mathbf{t}) \quad (\text{B.86})$$

is the  $i^{\text{th}}$  M-spline basis function of polynomial degree  $d$  [321, 489, 490], an alternate normalization of the B-spline basis functions. The length scale is then represented

2. Note that the literature on splines commonly defines B-splines in terms of “order”  $k = d + 1$ . This is very unfortunate notation as it masks the polynomial degree of the spline, and so the clearer but less conventional notation in terms of the polynomial degree  $d$  is used here.



as a weighted sum

$$\ell(x) = \sum_{i=1}^{n_i+d-1} C_i \mathbf{B}_{i,d}(x|\mathbf{t}). \quad (\text{B.87})$$

Both the knot locations  $t_i$  and the spline coefficients  $C_i$  can be treated as free parameters.

### B.4.2 Warped inputs

As noted in [297] and implemented in [395, 396, 491], another way of introducing non-stationarity to a Gaussian process is to pass the inputs  $\mathbf{x}$  through some non-linear warping function  $\mathbf{w} = f(\mathbf{x})$  before fitting the Gaussian process. In general,  $\mathbf{w}$  and  $\mathbf{x}$  could have different dimensions and  $f(\mathbf{x})$  could be many-to-one. But, for the tokamak profile fitting task, the most useful warping function maps single inputs  $x$  to single outputs  $w$ . Furthermore, the mapping function should<sup>3</sup> be monotonic and one-to-one so that there is a unique  $w$  for every  $x$ . A useful way of thinking about this procedure is that the warping function stretches the input space such that a function which was originally nonstationary becomes stationary.

It is important to distinguish between a Gaussian process with warped inputs and a *warped Gaussian process* as described in [492]: a so-called “warped Gaussian process” is the result of passing the *output* of a Gaussian process through a non-linear function, which may have parameters which are inferred from the data. Warped Gaussian processes are not supported by `gptools` at present.

The implementation in `gptools` uses a warping function  $f : \mathbb{R}^D \rightarrow \mathbb{R}^D$  where each element of  $\mathbf{w} = f(\mathbf{x})$  can be expressed as  $w_d = f_d(x_d)$ , where  $f_d : \mathbb{R} \rightarrow \mathbb{R}$ . In other words,  $\mathbf{w}$  is the same dimension as  $\mathbf{x}$  and each component of  $\mathbf{w}$  only depends on the corresponding component of  $\mathbf{x}$ . Let  $\mathbf{w}_i = f(\mathbf{x}_i)$ ,  $\mathbf{w}_j = f(\mathbf{x}_j)$ ,  $y_i = y(f(\mathbf{x}_i))$  and

3. An otherwise stationary covariance kernel can be imbued with useful properties such as periodicity by relaxing these restrictions, see [297] for details.

$y_j = y(f(\mathbf{x}_j))$ . Then,

$$\text{cov}[y_i, y_j] = k(\mathbf{w}_i, \mathbf{w}_j) \quad (\text{B.88})$$

$$\text{cov} \left[ \frac{\partial y_i}{\partial x_{d_i}}, y_j \right] = \frac{\partial k(\mathbf{w}_i, \mathbf{w}_j)}{\partial x_{d_i}} = \frac{\partial k(\mathbf{w}_i, \mathbf{w}_j)}{\partial w_{d_i}} \frac{\partial w_{d_i}}{\partial x_{d_i}} \quad (\text{B.89})$$

$$\text{cov} \left[ y_i, \frac{\partial y_j}{\partial x_{d_j}} \right] = \frac{\partial k(\mathbf{w}_i, \mathbf{w}_j)}{\partial x_{d_j}} = \frac{\partial k(\mathbf{w}_i, \mathbf{w}_j)}{\partial w_{d_j}} \frac{\partial w_{d_j}}{\partial x_{d_j}} \quad (\text{B.90})$$

$$\text{cov} \left[ \frac{\partial y_i}{\partial x_{d_i}}, \frac{\partial y_j}{\partial x_{d_j}} \right] = \frac{\partial^2 k(\mathbf{w}_i, \mathbf{w}_j)}{\partial x_{d_i} \partial x_{d_j}} = \frac{\partial^2 k(\mathbf{w}_i, \mathbf{w}_j)}{\partial w_{d_i} \partial w_{d_j}} \frac{\partial w_{d_i}}{\partial x_{d_i}} \frac{\partial w_{d_j}}{\partial x_{d_j}}. \quad (\text{B.91})$$

This procedure can be extended to arbitrary transformations  $\mathbf{w}(\mathbf{x})$  and arbitrary orders of differentiation through appropriate application of the chain rule. The final extension supported by `gptools` is nested warpings:  $w_d = g_d(f_d(x_d))$ . Letting  $u_d = f_d(x_d)$ , the warping function derivatives are found through application of the chain rule:

$$\frac{\partial w_d}{\partial x_d} = \frac{\partial w_d}{\partial u_d} \frac{\partial u_d}{\partial x_d} = \frac{\partial g_d(u_d)}{\partial u_d} \frac{\partial f_d(x_d)}{\partial x_d}. \quad (\text{B.92})$$

This can be extended to arbitrary depth of nesting and arbitrary order of differentiation through appropriate application of the chain rule.

In order to put covariance kernels with warped inputs and covariance kernels with spatially-varying length scales on the same footing, note the following identity given in [395]:

$$\ell(x) = \frac{1}{df(x)/dx}. \quad (\text{B.93})$$

Therefore, a stationary covariance kernel with input warping function  $f(x)$  can be thought of as a nonstationary covariance kernel with spatially varying covariance length scale  $\ell(x)$ . Solving for  $f(x)$  gives

$$f(x) = \int_L^x \frac{1}{\ell(u)} du, \quad (\text{B.94})$$

where  $L \leq x$  is the lower bound of the domain. Therefore, a nonstationary covariance kernel with spatially varying covariance length scale  $\ell(x)$  can be thought of as a stationary covariance kernel with nonlinear input warping function  $f(x)$ .

### B.4.2.1 Warping to the unit hypercube

It is often convenient to employ a warping of the form  $f : \mathbb{R}^D \rightarrow [0, 1]^D$  such that all of the warped variables lie within the unit hypercube. This is a prerequisite to use the beta warping discussed in section B.4.2.2, for example. The linear warping

$$w = f(x) = \frac{x - a}{b - a} \quad (\text{B.95})$$

$$\frac{\partial f(x)}{\partial x} = \frac{1}{b - a}, \quad (\text{B.96})$$

accomplishes this as long as  $a \leq x \leq b$  for all points of interest. Here,  $x$  refers to the specific  $x_d$  in  $\mathbf{x}$  which is to be warped.

### B.4.2.2 Beta-CDF warping

Snoek et al. [396] advocates for the use of the cumulative distribution function (CDF) of the beta distribution as the warping function. This function is the regularized incomplete beta integral [294, 493]:

$$f(x) = \text{I}(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_0^x t^{\alpha-1} (1-t)^{\beta-1} dt, \quad (\text{B.97})$$

where  $\alpha > 0$ ,  $\beta > 0$  are the shape parameters of the beta distribution and  $0 \leq x \leq 1$ . The derivatives of this function are [468, 494, 495]:

$$\frac{d^n \text{I}(x; \alpha, \beta)}{dx^n} = -\frac{(1-x)^{\beta-1} x^{\alpha-n}}{\text{B}(\alpha, \beta)} \sum_{k=0}^{n-1} (-1)^{n-k} \binom{n-1}{k} (1-\beta)_k (1-\alpha)_{n-k-1} \left(\frac{x}{1-x}\right)^k \quad (\text{B.98})$$

$$\frac{d \text{I}(x; \alpha, \beta)}{dx} = \frac{(1-x)^{\beta-1} x^{\alpha-1}}{\text{B}(\alpha, \beta)}, \quad (\text{B.99})$$

where  $\text{B}(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$  is the beta function [294, 496]. Note that the slope diverges at  $x = 0$  if  $\alpha < 1$  and at  $x = 1$  if  $\beta < 1$ . This causes a problem with handling the slope at the magnetic axis and in the scrape-off layer when fitting a tokamak profile. To get around this in practice, the data are mapped to lie just *inside* the unit hypercube using the linear warping discussed in section B.4.2.1.

Any reasonable distribution's CDF will be an acceptable monotonic and one-to-one warping function, and in fact Gibbs [395] uses a sum of shifted Gaussian CDFs to construct a warping function. The beta distribution's CDF has the advantage that it can acquire many different shapes by varying the two parameters  $\alpha$  and  $\beta$ . Specifically, if  $\alpha = \beta = 1$ , this function just maps  $\mathbf{x}$  to itself. If  $\alpha > 1$ , the slope gets

shallower near  $x = 0$  and large changes in  $x$  produce only small changes in  $w$ : the space is compressed around  $x = 0$ . If  $\alpha < 1$ , the slope gets steeper near  $x = 0$  and small changes in  $x$  produce large changes in  $w$ : the space is expanded around  $x = 0$ . The same reasoning holds for the behavior near  $x = 1$  as  $\beta$  is varied. Therefore, to fit a tokamak profile, having  $\alpha \gtrsim 1$  and  $\beta < 1$  will compress the core and stretch the edge out to produce the desired non-stationarity.

### B.4.2.3 I-spline warping

When beta warping is insufficient, the I-splines [489, 490, 497]:

$$I_{i,d}(x|\mathbf{t}) = \int_L^x M_{i,d-1}(u|\mathbf{t}) du \quad (\text{B.100})$$

$$w(x) = \sum_{i=1}^{n_t+k-2} C_i I_{i,k}(x|\mathbf{t}), \quad (\text{B.101})$$

where  $M_{i,k}(x|\mathbf{t})$  are the M-spline basis functions given in equation (B.86) and  $L$  is the lower limit of the domain, offer a more flexible alternative. Because the M-spline basis functions are always positive, the I-spline basis functions will always be monotonic. So, the resulting warping function will always be monotonic if  $C_i \geq 0$ . Through appropriate selection of the knot grid, this approach does not require that the data be mapped to the unit hypercube before applying the warping. The derivative of the I-spline basis functions are obviously

$$\frac{dI_{i,d}(x|\mathbf{t})}{dx} = M_{i,d-1}(x|\mathbf{t}). \quad (\text{B.102})$$

## B.5 Transformations of covariance kernels

Since any symmetric, positive definite function is a valid covariance kernel, any transformation of existing covariance kernels which preserves symmetric positive definiteness will also be a valid covariance kernel [297]. Useful illustrations of some transformations are provided in [498, 499].

### B.5.1 Sums of covariance kernels

The sum of two covariance kernels is a valid covariance kernel. This construction is useful to represent data which have multiple length scales. A model could have a covariance kernel with a long covariance length scale to capture the macroscopic behavior plus a short length scale to represent local variations or noise. In fact,

the RQ covariance kernel discussed in [section B.2](#) is an infinite sum (i.e., integral) over SE covariance kernels with different covariance length scales. Sum kernels are particularly simple to include in our framework:

$$k_{\text{sum}}(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j) + k_2(\mathbf{x}_i, \mathbf{x}_j) \quad (\text{B.103})$$

$$\frac{\partial}{\partial x_{i_d}} k_{\text{sum}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\partial}{\partial x_{i_d}} k_1(\mathbf{x}_i, \mathbf{x}_j) + \frac{\partial}{\partial x_{i_d}} k_2(\mathbf{x}_i, \mathbf{x}_j), \quad (\text{B.104})$$

and so on for higher-order and mixed partial derivatives.

### B.5.2 Products of covariance kernels

The product of two covariance kernels is a valid covariance kernel. This is a useful construction for combining covariance kernels which operate on different dimensions. The SE covariance kernel can in fact be seen as a product of 1D SE covariance kernels, one for each dimension. As a more interesting example, the product of a Gibbs covariance kernel operating on  $\tau_1 = \rho_i - \rho_j$  and an SE covariance kernel operating on  $\tau_2 = t_i - t_j$  would be useful for performing time-dependent fits of (sawtooth-free) profile data with a strong pedestal. To utilize covariance kernels of the general form

$$k_{\text{prod}}(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j)k_2(\mathbf{x}_i, \mathbf{x}_j) \quad (\text{B.105})$$

in our framework requires some care with respect to differentiation. Note that the Leibniz rule of [equation \(B.45\)](#) generalizes to partial derivatives as

$$\frac{\partial^n}{\prod_{i=1}^n \partial x_{[i]}} (k_1(\mathbf{x}_i, \mathbf{x}_j)k_2(\mathbf{x}_i, \mathbf{x}_j)) = \sum_{S \in \mathcal{P}(\{1, \dots, n\})} \frac{\partial^{|S|} k_1}{\prod_{i \in S} \partial x_{[i]}} \frac{\partial^{n-|S|} k_2}{\prod_{i \notin S} \partial x_{[i]}}, \quad (\text{B.106})$$

where the notation for indicating which variable to differentiate with respect to is the same as was used in [equation \(B.30\)](#),  $\mathcal{P}(\{1, \dots, n\})$  is the power set (i.e., the set of all possible subsets) of the set of variables to differentiate with respect to,  $S$  is one of the subsets in  $\mathcal{P}(\{1, \dots, n\})$  and  $|S|$  is the number of elements in  $S$ .

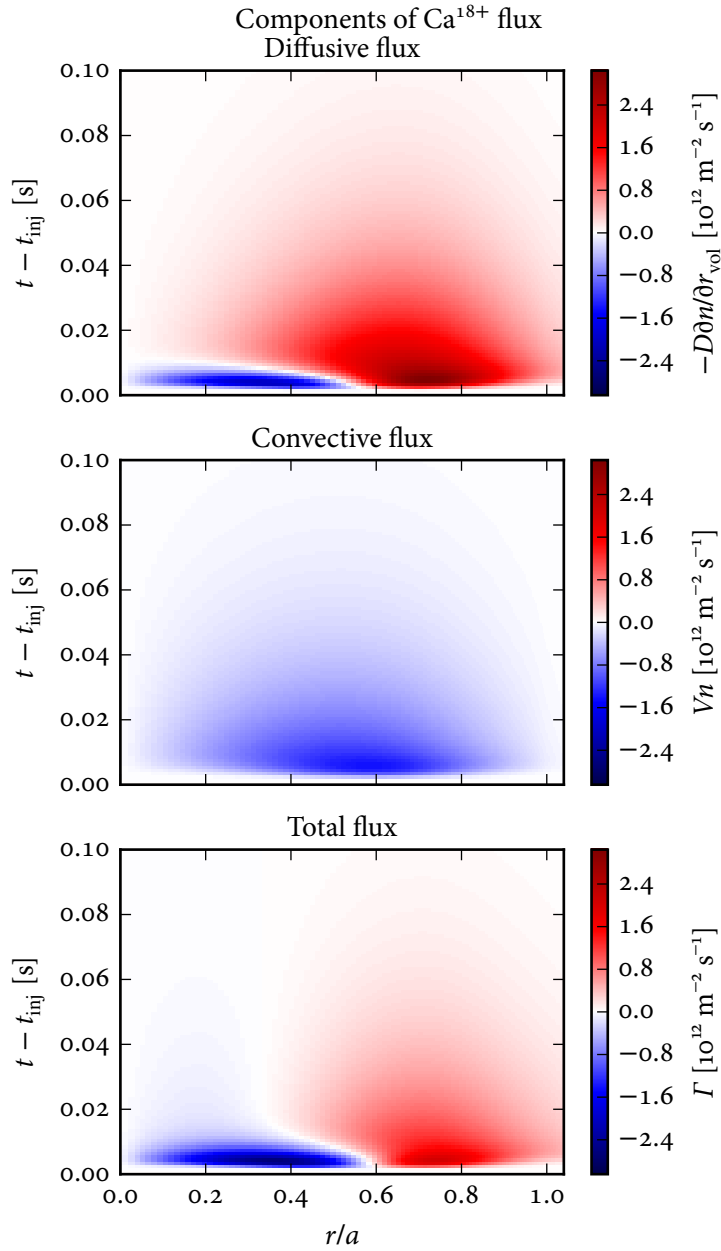
# C

## Supporting material from chapter 3: Inference of experimental impurity transport coefficient profiles

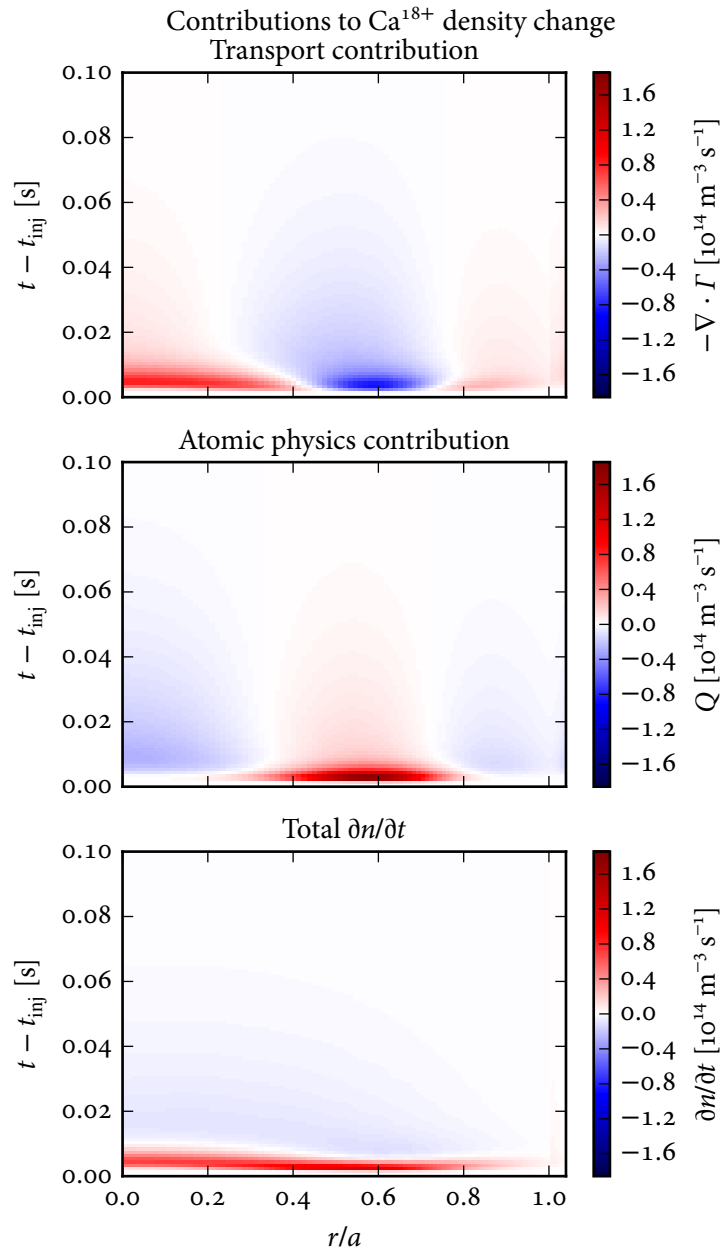
### C.1 Relative contributions of diffusion, convection and atomic physics

Consider the spatial and temporal evolution of He-like calcium as governed by the impurity continuity equation [equation \(3.1\)](#) with the transport coefficients given in [equation \(3.14\)](#). [Figure C.1](#) gives the contributions of diffusion and convection to the flux. Diffusion dominates the behavior in the outer portion of the plasma, but convection plays a role in the core region. The total impurity ion flux exhibits similar behavior.

[Figure C.2](#) gives the contributions of the transport flux  $-1/r_{\text{vol}} \cdot \partial(r_{\text{vol}} \Gamma_{Z,i,r_{\text{vol}}})/\partial r_{\text{vol}}$  and ionization/recombination source/sink term  $Q_{Z,i}$  to the time derivative of the He-like calcium density  $\partial n_{Z,i}/\partial t$ . The atomic physics term dominates the behavior early in the injection, and is of comparable magnitude to the transport term throughout. This complicates use of the flux-gradient approach to reconstruct the transport coefficients from measurements of only He-like calcium.



**Figure c.1:** Contributions from diffusion and convection to the flux of He-like calcium. All plots have the same color scale. This figure was produced using `analyze_rates.py`.



**Figure c.2:** Contributions from transport and atomic physics to the change in the He-like calcium density. All plots have the same color scale. This figure was produced using `analyze_rates.py`.

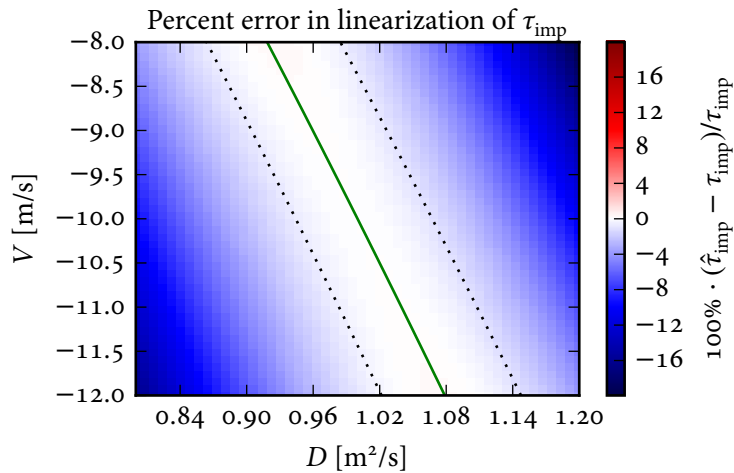


## c.2 Discrepancies between linearized and exact figures of merit

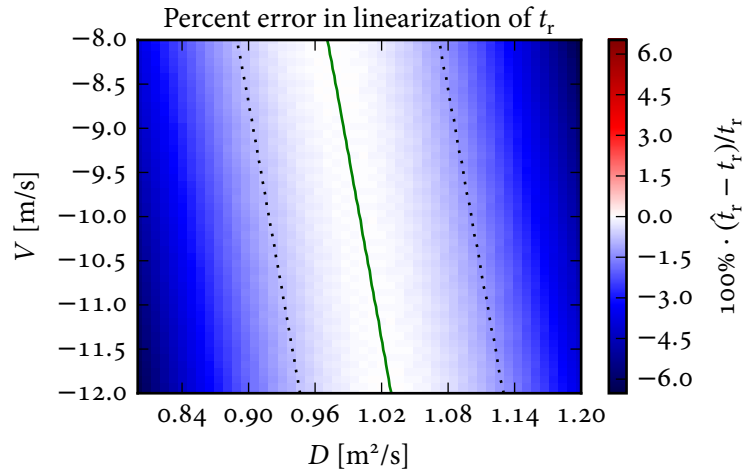
Figure c.3 through figure c.5 show the discrepancy between the linearizations and the actual STRAHL results for the figures of merit used in the linearized Bayesian analysis given in section 3.4.2.

## c.3 Posterior distributions for complex synthetic data

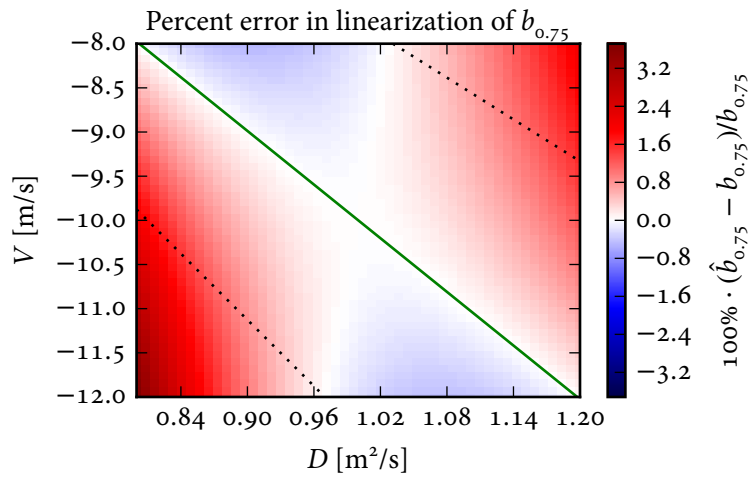
Figure c.6 through figure c.13 show posterior distributions for the coefficients of the transport coefficient profiles fit to the synthetic data in section 3.6.7. The posterior distribution for the seven coefficient per profile case is given in figure 3.59.



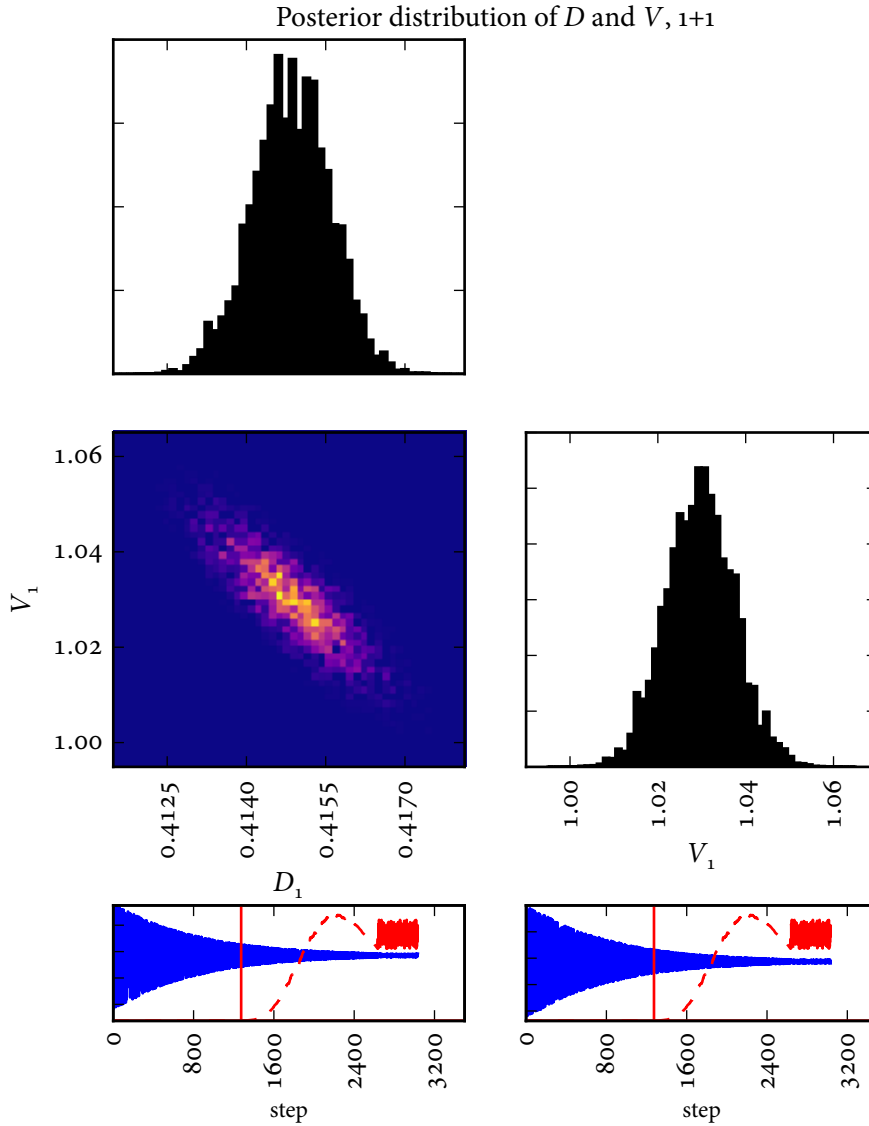
**Figure c.3:** Discrepancy between the actual  $\tau_{imp}$  obtained from the STRAHL outputs and the linearized  $\hat{\tau}_{imp}$ . The green solid contour corresponds to the value of  $\tau_{imp}$  at  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m/s}$  and the black dotted contours represent  $\pm 10\%$  around this value. This figure was produced using `make_bayes_time_res_plot.py`.



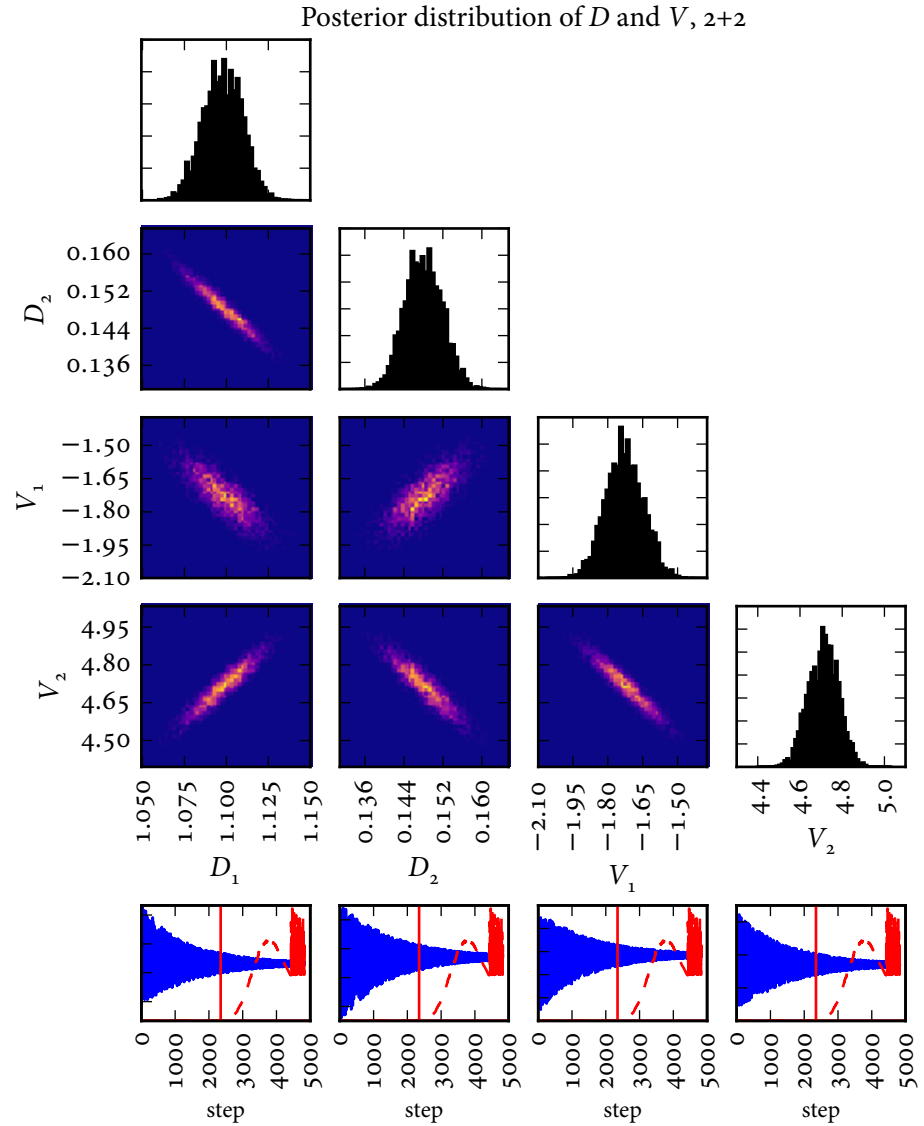
**Figure c.4:** Discrepancy between the actual  $t_r$  obtained from the STRAHL outputs and the linearized  $\hat{t}_p$ . The green solid contour corresponds to the value of  $t_r$  at  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m}/\text{s}$  and the black dotted contours represent  $\pm 10\%$  around this value. This figure was produced using `make_bayes_time_res_plot.py`.



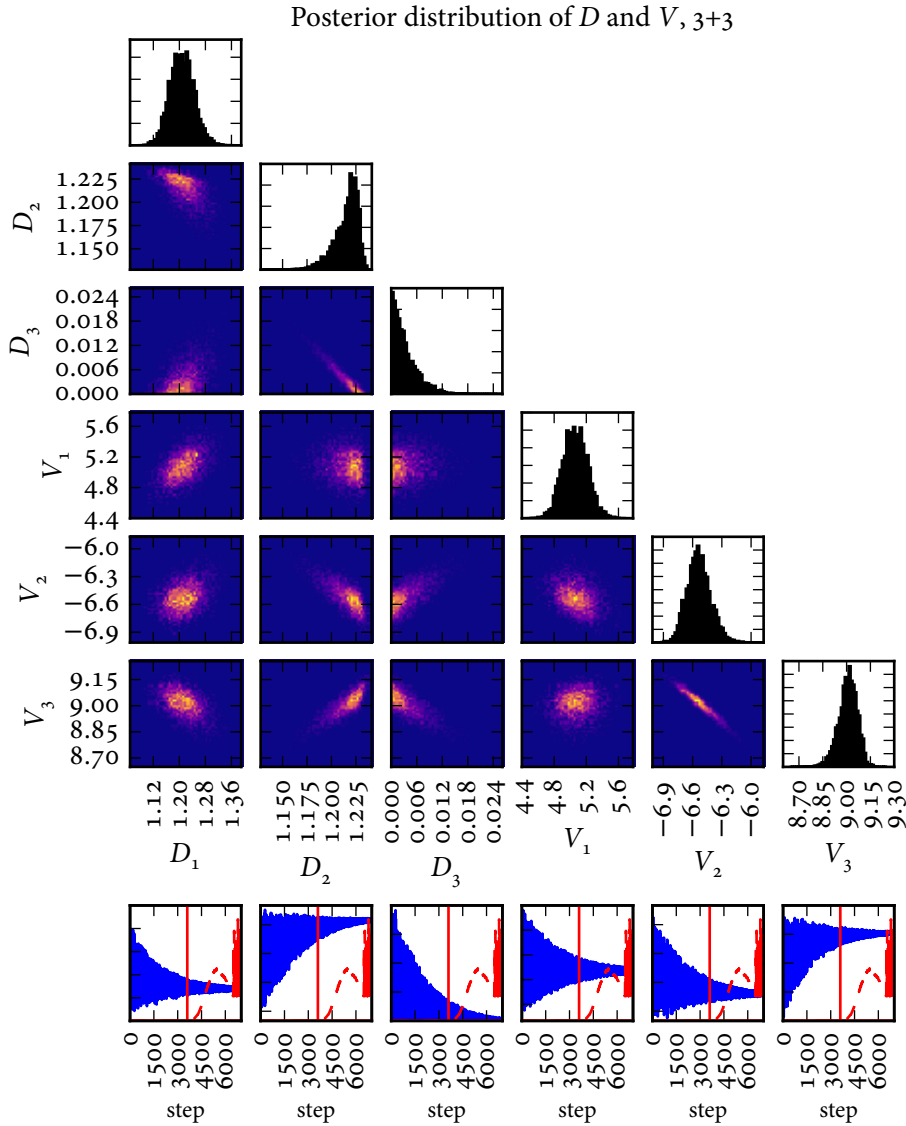
**Figure c.5:** Discrepancy between the actual  $b_{0.75}$  obtained from the STRAHL outputs and the linearized  $\hat{b}_{0.75}$ . The green solid contour corresponds to the value of  $b_{0.75}$  at  $D = 1 \text{ m}^2/\text{s}$ ,  $V = -10 \text{ m}/\text{s}$  and the black dotted contours represent  $\pm 10\%$  around this value. This figure was produced using `make_bayes_time_res_plot.py`.



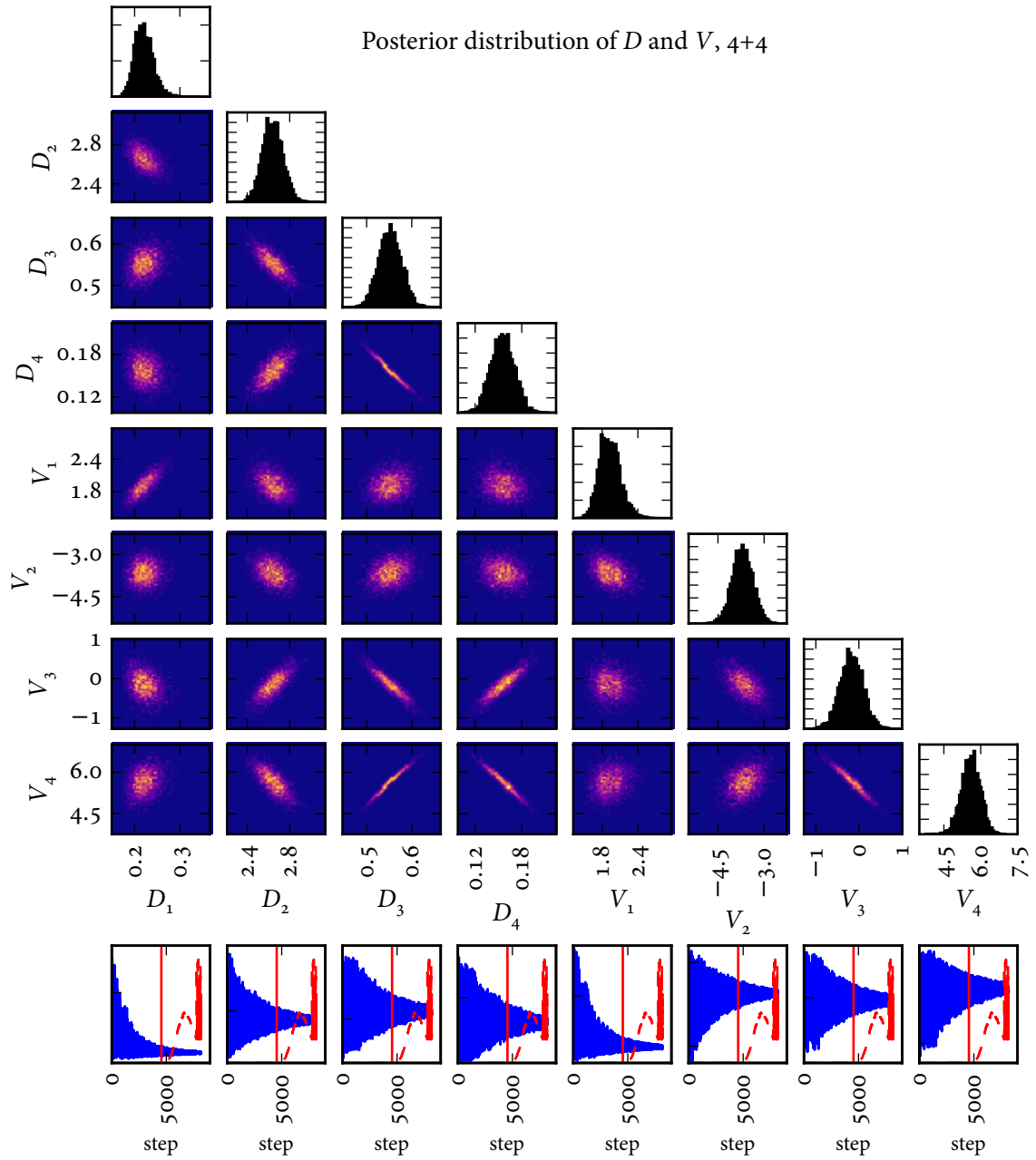
**Figure c.6:** Posterior distribution for  $D$  and  $V$  for the one coefficient per profile case. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.



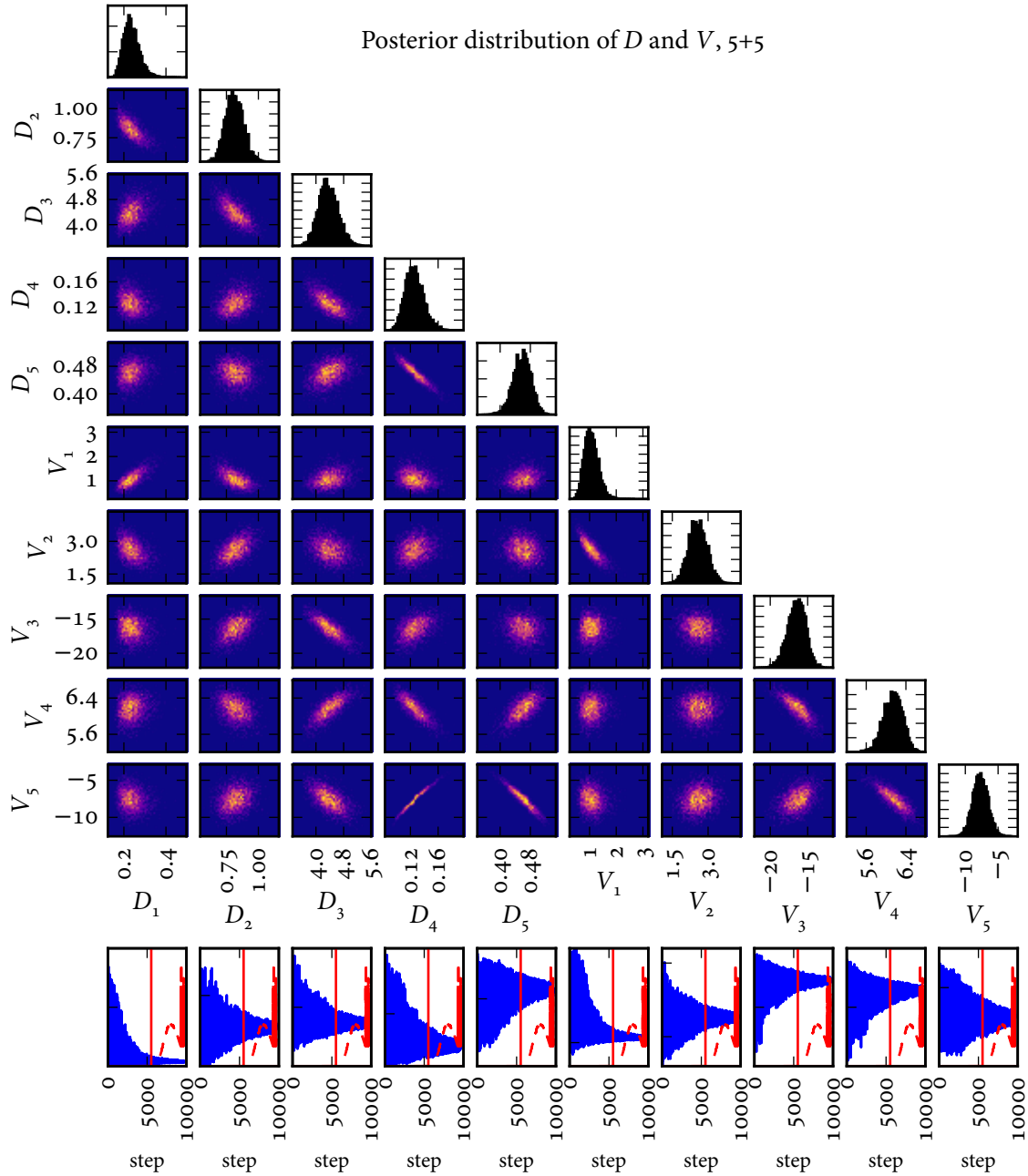
**Figure c.7:** Posterior distribution for  $D$  and  $V$  for the two coefficient per profile case. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.



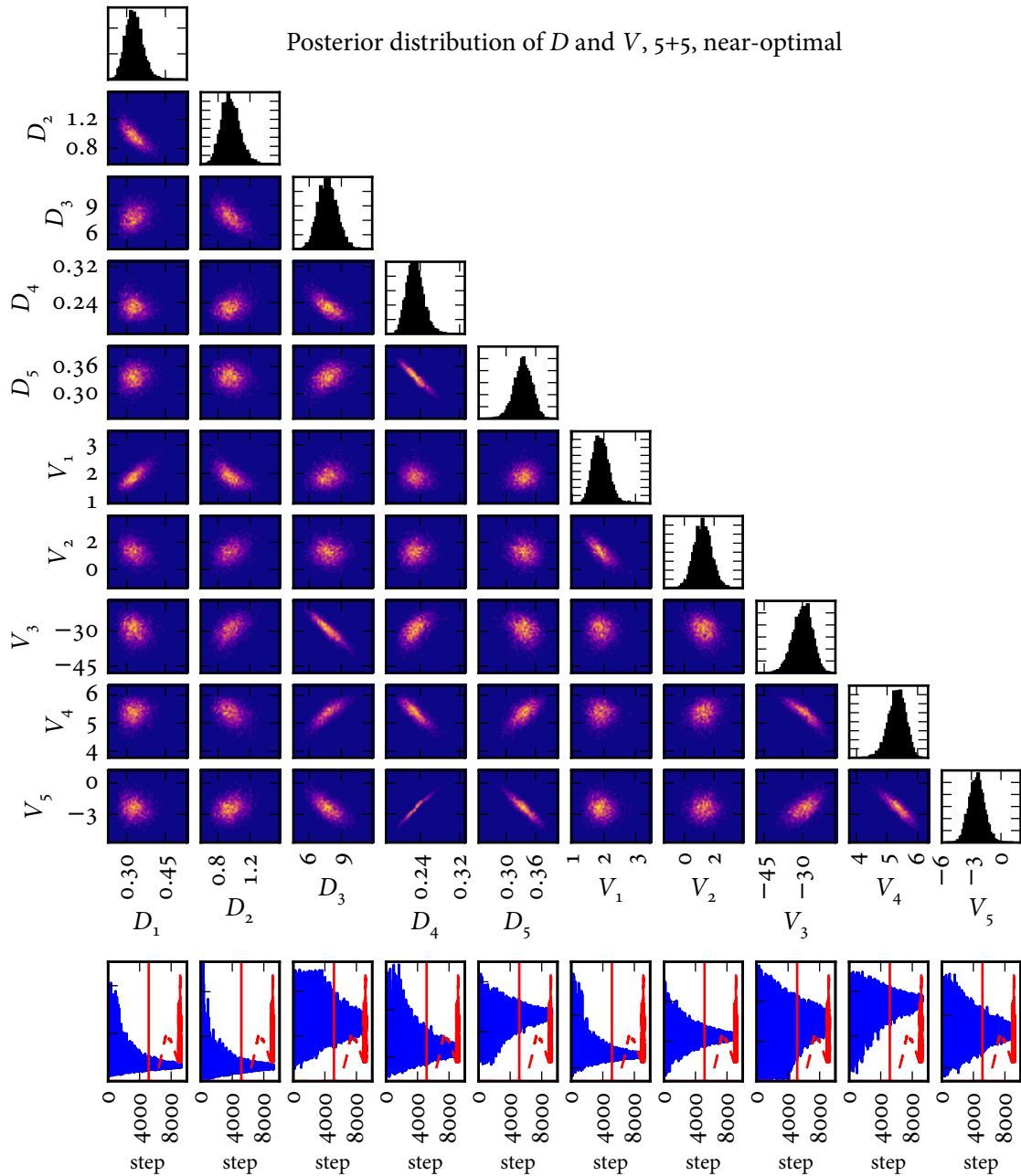
**Figure c.8:** Posterior distribution for  $D$  and  $V$  for the three coefficient per profile case. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.



**Figure c.9:** Posterior distribution for  $D$  and  $V$  for the four coefficient per profile case. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.

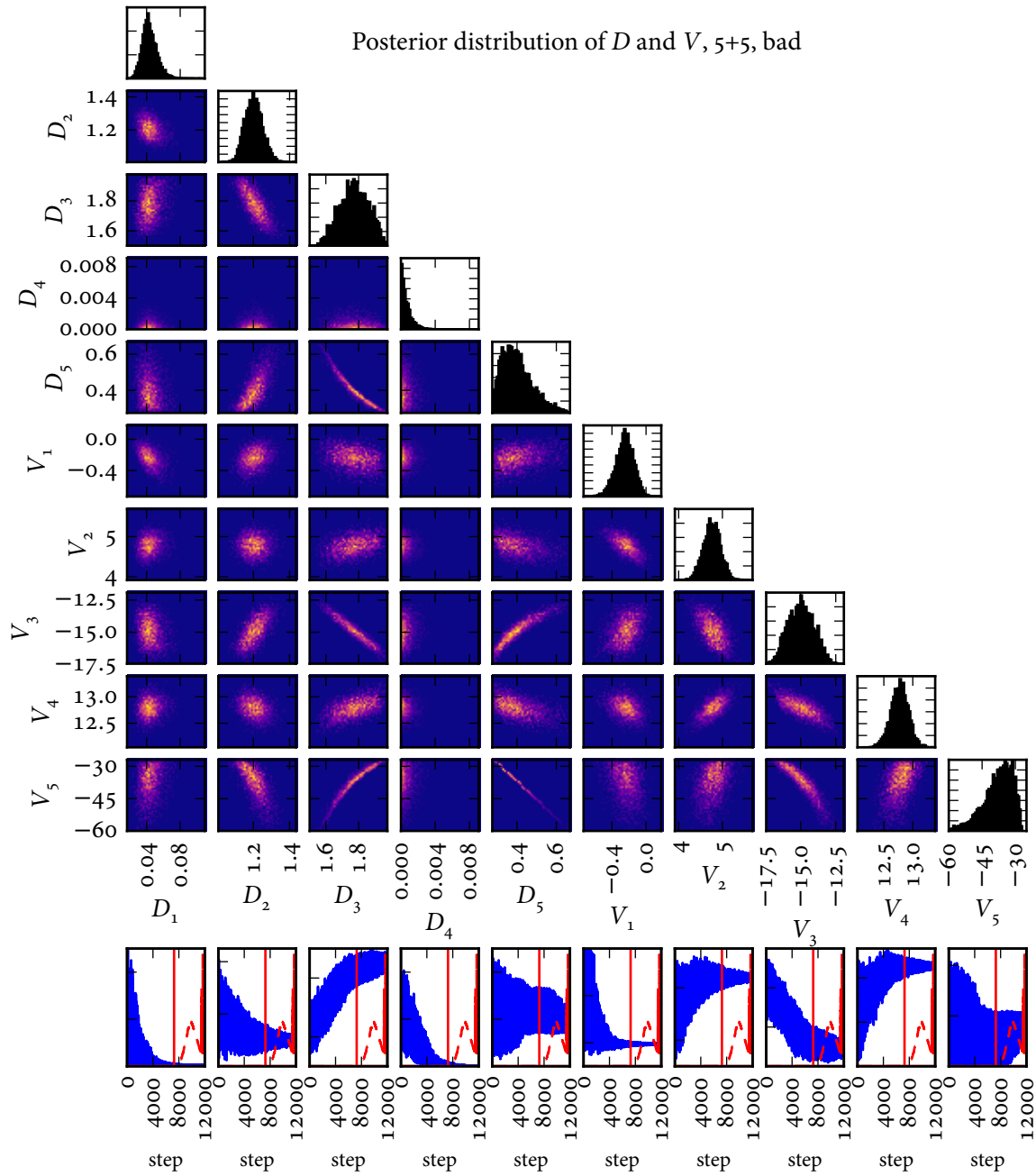


**Figure c.10:** Posterior distribution for  $D$  and  $V$  for the five coefficient per profile case with linearly-spaced knots. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.

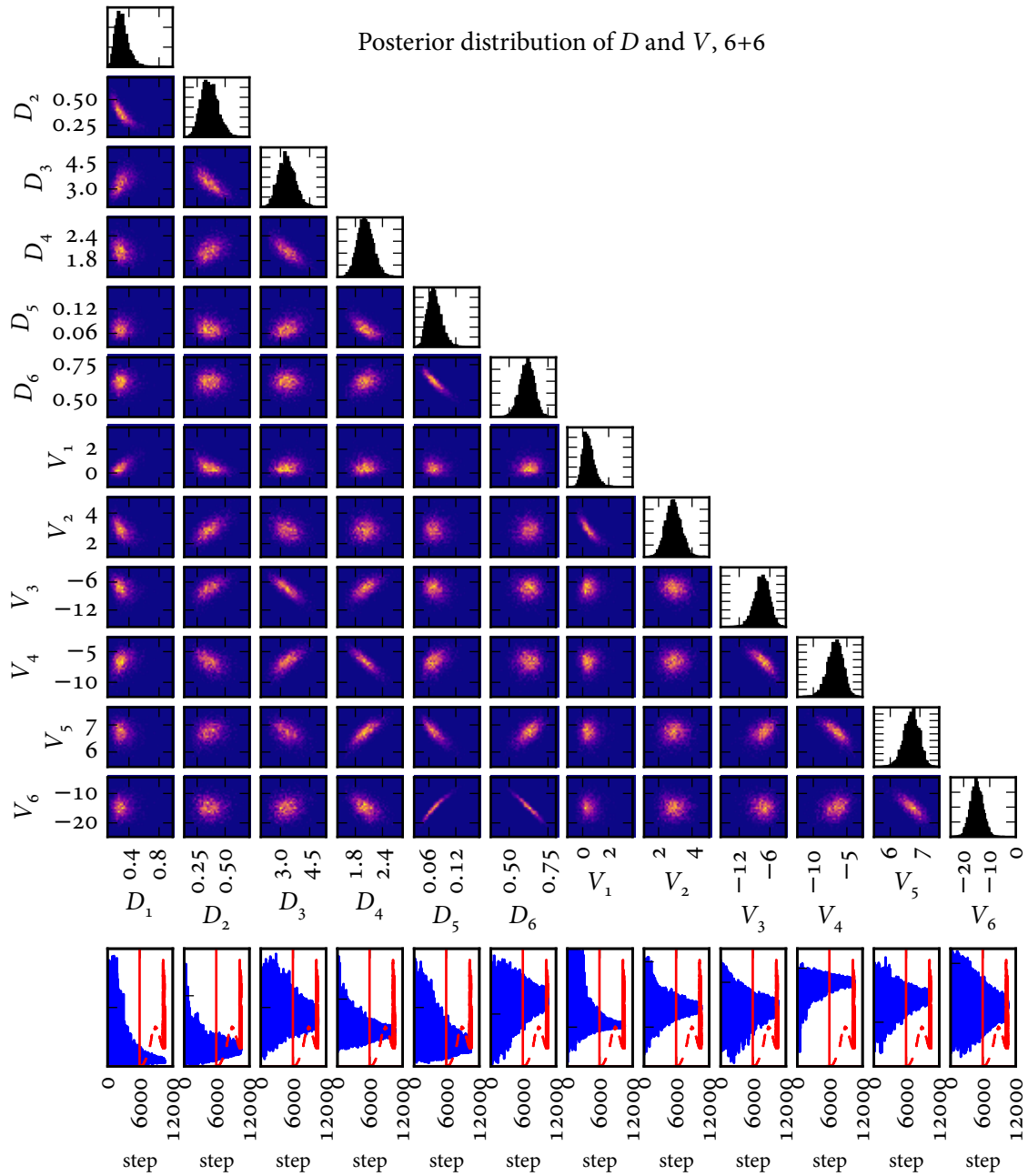


**Figure C.11:** Posterior distribution for  $D$  and  $V$  for the five coefficient per profile case with near-optimal knots. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.





**Figure c.12:** Posterior distribution for  $D$  and  $V$  for the five coefficient per profile case with purposefully bad knots. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.



**Figure c.13:** Posterior distribution for  $D$  and  $V$  for the six coefficient per profile case. This figure was produced using `settings_1101014006_22_NTH_as_true.py` and `post_process_NTH_as_true_MCMC.py`.

# Software for data analysis

A key contribution of this thesis was the development of extensive open-source code for data analysis, both inside and outside the domain of plasma physics. Because the user documentation and source code for all of the software described in the following chapters are available online, the focus is primarily on the implementation details which are not necessarily of interest to an end-user, but which constitute a key part of this thesis work. The following appendices should be viewed as a bridge between the math and physics discussed in this thesis and the user-facing interface described in the documentation.

All of the software presented here was implemented in the Python programming language. A list of the software configuration in use at the time this thesis was written is given in [table H.1](#). While some illustrative code snippets are given, the primary reference on how to use these codes are the scripts given at <https://github.com/markchil/thesiscode>, which are the actual data analysis scripts used for this thesis.

When what a code is doing is described mathematically, the equations will follow the mathematical convention of starting the indexing of elements in vectors and matrices at one and not zero as is used in Python. Ranges of elements may be indicated with a “slice index” where  $v_{2:5}$  indicates the second through fifth element of vector  $v$ , inclusive.

# D

## eqtools: an open-source Python package for handling magnetic equilibrium data

### D.1 Package overview

eqtools is a Python package which provides classes for working with magnetic equilibrium reconstruction data from multiple tokamaks, including support for all of the standard flux surface coordinate systems. The software was developed in collaboration with I. Faust and J. Walk. E. Davis helped with the DIII-D implementation and N. Vianello contributed the TCV implementation. eqtools is described in [500], the source code is available in [501] and the documentation is available in [502]. The objective of eqtools is to provide both simple, unified access to data related to the magnetic equilibrium reconstruction and a comprehensive set of coordinate mapping routines. The idea is to provide a set of tools which can easily be made to work on data from any tokamak using any magnetic reconstruction tool. By providing a unified interface, analysis programs written using eqtools can be made to work at a variety of facilities with minimal modification. An example of fetching the  $q$  profile with the standard MDSplus routines is given in [code listing D.1](#) and example of the same operation using eqtools is given in [code listing D.2](#). The example using MDSplus directly requires the user to remember the site-specific path to the node, while the eqtools example uses clear, easy to understand (and remember) names. Furthermore, the path to the node will almost certainly be different at a different facility, but the means of interacting with eqtools will be the same. So, when moving a code from one facility to another, instead of having to change *all* of the paths in an analysis script (or possibly completely change away from an MDSplus-based system), the user will simply have to change which subclass of `Equilibrium` is used.

---

**Code Listing D.1:** Fetching the  $q$  profile with MDSplus directly

---

```
1 import MDSplus
2 tree = MDSplus.Tree('ANALYSIS', 1101014006)
3 n_q = tree.getNode(r'\analysis::top.efit.results.g_eqdsk:qpsi')
4 q = n_q.data()
5 t = n_q.dim_of().data()
```

---

---

**Code Listing D.2:** Fetching the  $q$  profile with eqtools

---

```
1 import eqtools
2 e = eqtools.CModEFITTree(1101014006)
3 q = e.getQProfile()
4 t = e.getTimeBase()
```

---

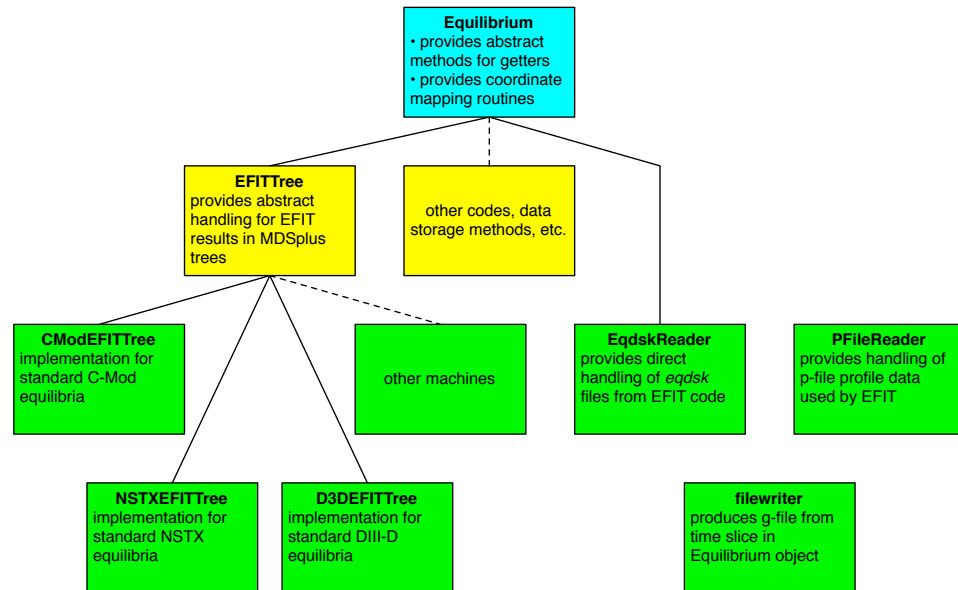
An outline of the package is given in [figure D.1](#). The core class is the `Equilibrium` abstract class which contains all of the coordinate mapping routines and abstract methods for all of the getter methods. Implementing subclasses are responsible for implementing getter methods which put the data into the format that the mapping routines require. The `EFITTree` abstract class implements most of these methods for facilities using the EFIT code [503]. Classes which implement `EFITTree` such as `CModEFITTree` and `NSTXEFITTree` then require very little additional code, thereby allowing easy porting of `eqtools` to any facility using MDSplus and EFIT.

### A note about coding conventions in eqtools

Because of its collaborative development, `eqtools` uses slightly different coding conventions from `gptools` and `profiletools`. In `eqtools`, class names are still indicated in upper camel case (`CModEFITTree`), but public method and attribute names use lower camel case (`getQProfile()`). Instead of providing direct access to the equilibrium data, `eqtools` provides getter methods such as `getQProfile()` which return a copy of the internal (private<sup>1</sup>) array. The idea is to prevent novice users from accidentally modifying the data in-place and hence causing the data to become inconsistent. This adds some overhead to copy the array each time, however. The mix-in `PropertyAccessMixin` provides property-like access to all of the getter

---

1. Python does not provide direct support for public and private attributes and methods. Instead, the standard Python convention of putting a leading underscore on the names of attributes and methods which are not meant for external use is employed. For instance, the internal variable holding the  $q$  profile is `_qpsi`.



**Figure D.1:** Structure of the *eqtools* package. This figure was originally published in [500] and is a revised version of a figure originally created by J. Walk.

methods, but does not save any of the overhead from copying the arrays.

## D.2 Coordinate systems supported and the coordinate mapping routines

*eqtools* supports transformations between a wide variety of coordinates in common use including real-space  $(R, Z)$  coordinates, mapped outboard midplane major radius  $R_{\text{mid}}$ , normalized minor radius  $r/a$ , unnormalized poloidal flux<sup>2</sup>  $\psi$ , normalized poloidal flux  $\psi_n$ , normalized toroidal flux  $\varphi_n$ , normalized flux surface volume  $V_n$  and the square roots of these quantities. The original routines were developed by S. Wolfe and were ported to Python as part of this thesis.

The most important transformation maps a given point  $(R, Z)$  at a given time  $t$  to the poloidal flux  $\psi$  at that location as computed with the magnetic reconstruction code. The default implementation uses a nearest-neighbor interpolation in time: the code first retrieves the flux reconstruction at the time closest to  $t$ , then a bivariate interpolating spline [293, 311] is used to map from  $(R, Z)$  to  $\psi$ . The bivariate spline coefficients from each time are stored in memory as they are computed in order to

2. What is used is actually the (Stokes) stream function  $\psi = \psi_p/(2\pi)$ , with units of Wb/rad. See section D.3.

speed up subsequent calculations at that time slice. A more advanced method using a tricubic interpolating spline to interpolate smoothly in space *and* time is described in [500].

Once the coordinate has been mapped to  $\psi$ , subsequent calculations are simpler. Normalized poloidal flux is defined as

$$\psi_n = \frac{\psi - \psi_o}{\psi_a - \psi_o}, \quad (\text{D.1})$$

where  $\psi_o$  is the poloidal flux at the magnetic axis and  $\psi_a$  is the poloidal flux at the last closed flux surface (LCFS). To ensure self-consistency, the default behavior is to use nearest-neighbor interpolation to get the values of  $\psi_o$  and  $\psi_a$  at the desired time. When a tricubic spline is used to map from  $(R, Z)$  to  $\psi$  a cubic spline is used to interpolate  $\psi_o$  and  $\psi_a$  in time.

Normalized toroidal flux is defined in terms of normalized poloidal flux as

$$\begin{aligned} \varphi(\psi) &= \int_{\psi_o}^{\psi} q(\psi') d\psi' \\ \varphi_n &= \frac{\varphi}{\varphi_a}, \end{aligned} \quad (\text{D.2})$$

where  $q(\psi) = d\varphi/d\psi$  is the safety factor profile (typically computed when EFIT is run) and  $\varphi_a = \int_{\psi_o}^{\psi_a} q(\psi') d\psi'$  is the toroidal flux at the last closed flux surface. The integral in equation (D.2) is numerically evaluated using the trapezoid rule.

The normalized flux surface volume is defined as

$$V_n(\psi) = \frac{V(\psi)}{V_a}, \quad (\text{D.3})$$

where  $V(\psi)$  is the volume enclosed by the (closed) flux surface with flux  $\psi$  and  $V_a$  is the volume enclosed by the last closed flux surface. In the case of Alcator C-Mod, the flux surface volume  $V(\psi)$  is computed automatically when EFIT is run, but it would be straightforward to override the `getFluxVol()` method of the `Equilibrium` class to compute this from the  $\psi(R, Z)$  grid when this quantity is not already available in the tree.

Mapping from  $\psi_n$  to  $R_{\text{mid}}$  is accomplished by forming a dense radial grid of  $R$  points that go from the magnetic axis to the edge of the grid the flux is reconstructed on, finding the vertical location of the magnetic axis at the desired time(s) (called  $Z_o$ ), then converting the resulting  $(R, Z_o)$  points to  $\psi_n$  with the routines described above. This one-to-one mapping between  $R_{\text{mid}}$  and  $\psi_n$  is then interpolated to give the desired conversion from  $\psi_n$  to  $R_{\text{mid}}$ .

By default,  $r/a$  is defined in terms of  $R_{\text{mid}}$  as

$$r/a = \frac{R_{\text{mid}} - R_0}{R_a - R_0}, \quad (\text{D.4})$$

where  $R_0$  is the major radius of the magnetic axis and  $R_a$  is the outboard midplane major radius of the last closed flux surface. Since other definitions of  $r/a$  are preferred when the Shafranov shift is high, the specific definition of  $r/a$  can be changed simply by overriding the methods `_rmid2roa()` and `_roa2rmid()` in the `Equilibrium` class. Variants on these basic routines are then used to map between other pairs of coordinates.

## D.3 Computing magnetic fields and current densities

### D.3.1 The magnetic field

The (*Stokes*) *stream function*  $\psi$  for the magnetic field is directly proportional to the (unnormalized) poloidal flux:  $\psi_p = 2\pi\psi$ . The poloidal field components are then given by [504]:

$$B_R = -\frac{1}{R} \frac{\partial \psi}{\partial Z}, \quad B_Z = \frac{1}{R} \frac{\partial \psi}{\partial R}. \quad (\text{D.5})$$

Because  $\psi$  is interpolated with a cubic bivariate spline, computing the derivatives necessary to compute the poloidal field is trivial. The toroidal component of the magnetic field is found from the flux function  $F = RB_\varphi$ . The sign convention for  $B_\varphi$  is right-handed such that the toroidal field of an Alcator C-Mod “forward-field” discharge (clockwise when viewed from above) is negative. Unfortunately, `EFIT` only stores  $F$  inside the LCFS. To compute the toroidal field outside of the LCFS, the vacuum component of the toroidal field at the magnetic axis,<sup>3</sup>  $B_{\varphi,\text{vac},0}$  is used:

$$B_{\varphi,\text{vac}} = B_{\varphi,\text{vac},0} \frac{R_0}{R}, \quad (\text{D.6})$$

where  $R_0$  is the major radius of the magnetic axis. This matches the toroidal field in the plasma quite well; see [figure D.2](#).

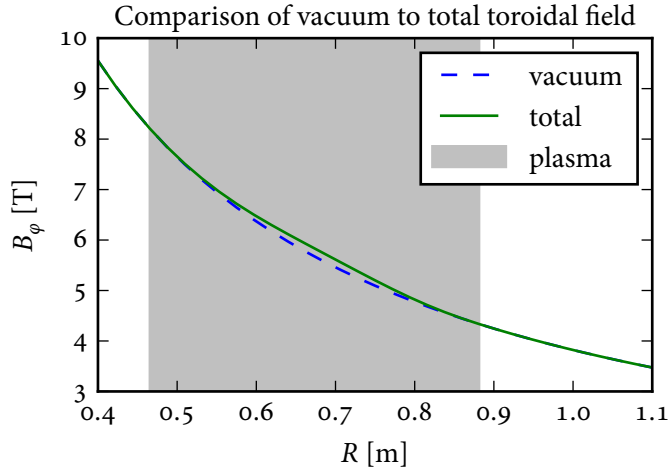
### D.3.2 The current density

The current density is

$$\mathbf{j} = \frac{1}{\mu_0} \left( \frac{1}{R} \frac{dF}{d\psi} \nabla\psi \times \hat{\boldsymbol{\phi}} - \frac{1}{R} \Delta^* \psi \hat{\boldsymbol{\phi}} \right), \quad (\text{D.7})$$

3. This is stored in the `EFIT` a-file as “btaxv.”





**Figure D.2:** Vacuum (blue dashed) and total (green solid) toroidal magnetic field at midplane as a function of major radius for  $t = 1$  s in Alcator C-Mod shot 1120907032, the reversed-field I-mode used for the profile analysis incorporating TCI data in section 2.5.2. The shaded region indicates the radial extent of the plasma; outside of this region the simple  $1/R$  dependence given in equation (D.6) is used. The joins to the paramagnetic region inside the LCFS are quite smooth, which verifies the accuracy of this assumption in the scrape-off layer. This figure was produced using `EFIT_Btor.py`.

where  $\hat{\phi}$  is the unit vector in the toroidal direction and

$$\Delta^* \psi = R^2 \nabla \cdot \left( \frac{\nabla \psi}{R^2} \right) = R \frac{\partial}{\partial R} \left( \frac{1}{R} \frac{\partial \psi}{\partial R} \right) + \frac{\partial^2 \psi}{\partial Z^2} \quad (\text{D.8})$$

is the elliptic operator. Using the Grad-Shafranov equation,

$$\Delta^* \psi = -\mu_0 R^2 \frac{dp}{d\psi} - F \frac{dF}{d\psi}, \quad (\text{D.9})$$

so the toroidal current density is

$$j_\phi = Rp' + \frac{FF'}{\mu_0 R}, \quad (\text{D.10})$$

where  $p' = dp/d\psi$  and  $F' = dF/d\psi$ . Expanding the cross product in [equation \(D.7\)](#) gives the poloidal components of the current density:

$$j_R = -\frac{1}{\mu_0 R} F' \frac{\partial \psi}{\partial Z} = \frac{F' B_R}{\mu_0} \quad (\text{D.11})$$

$$j_Z = \frac{1}{\mu_0 R} F' \frac{\partial \psi}{\partial R} = \frac{F' B_Z}{\mu_0}. \quad (\text{D.12})$$

Note that this is equivalent to

$$\mu_0 j_R = -\frac{1}{R} \frac{\partial F}{\partial Z}, \quad \mu_0 j_Z = \frac{1}{R} \frac{\partial F}{\partial R}, \quad (\text{D.13})$$

which means that  $F$  is the Stokes stream function for the axisymmetric vector field  $\mathbf{j}$ . In any case, EFIT computes the quantities  $p'$  and  $FF'$  inside of the LCFS, so the current density inside the plasma can be obtained directly from these expressions.

### D.3.3 Tracing field lines

Consider  $\mathbf{x}$  to be the path of a field line of vector field  $\mathbf{u}$ . Since the situation is axisymmetric, a field line can simply be parameterized by the toroidal angle,  $\varphi$ . By definition, the field line is everywhere parallel to the flux surface. Therefore, the field lines are defined by

$$\frac{d\mathbf{x}}{d\varphi} \times \mathbf{u} = \mathbf{0}. \quad (\text{D.14})$$

Note that a position vector can be expressed as

$$\mathbf{x} = R\hat{\mathbf{R}} + Z\hat{\mathbf{Z}}. \quad (\text{D.15})$$

The derivative of  $\hat{\mathbf{R}}$  with respect to  $\varphi$  is [\[347, 505\]](#):

$$\frac{\partial \hat{\mathbf{R}}}{\partial \varphi} = \hat{\boldsymbol{\phi}}. \quad (\text{D.16})$$

Therefore,

$$\frac{d\mathbf{x}}{d\varphi} = \frac{dR}{d\varphi} \hat{\mathbf{R}} + R\hat{\boldsymbol{\phi}} + \frac{dZ}{d\varphi} \hat{\mathbf{Z}}, \quad (\text{D.17})$$

and the field line is defined by

$$\left( Ru_Z - \frac{dZ}{d\varphi} u_\varphi \right) \hat{\mathbf{R}} + \left( \frac{dZ}{d\varphi} u_R - \frac{dR}{d\varphi} u_Z \right) \hat{\boldsymbol{\phi}} + \left( \frac{dR}{d\varphi} u_\varphi - Ru_R \right) \hat{\mathbf{Z}} = \mathbf{0}. \quad (\text{D.18})$$

One of these equations is redundant, so the field line is governed by the system of equations

$$\frac{dR}{d\varphi} = \frac{u_R}{u_\varphi} R, \quad \frac{dZ}{d\varphi} = \frac{u_Z}{u_\varphi} R. \quad (\text{D.19})$$

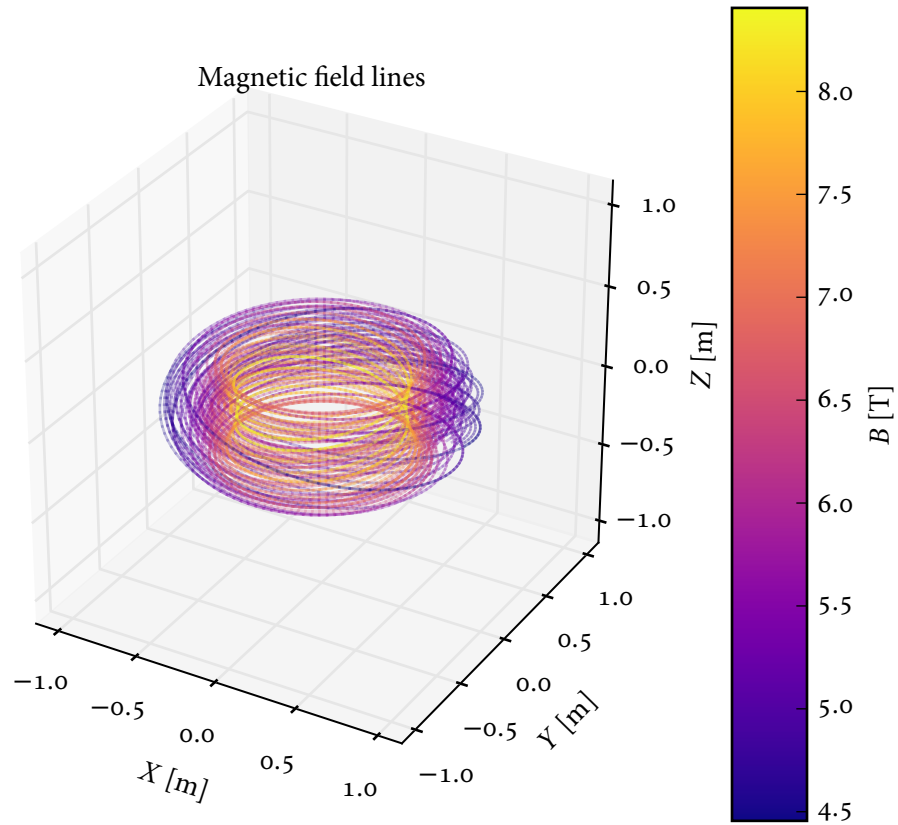
This system is solved numerically using the (4)5 Dormand-Prince integrator provided in SciPy [293, 506]. An example of magnetic field lines traced with this technique is given in figure D.3 and an example of current field lines traced with this technique is given in figure D.4.

## D.4 Verification and benchmarking

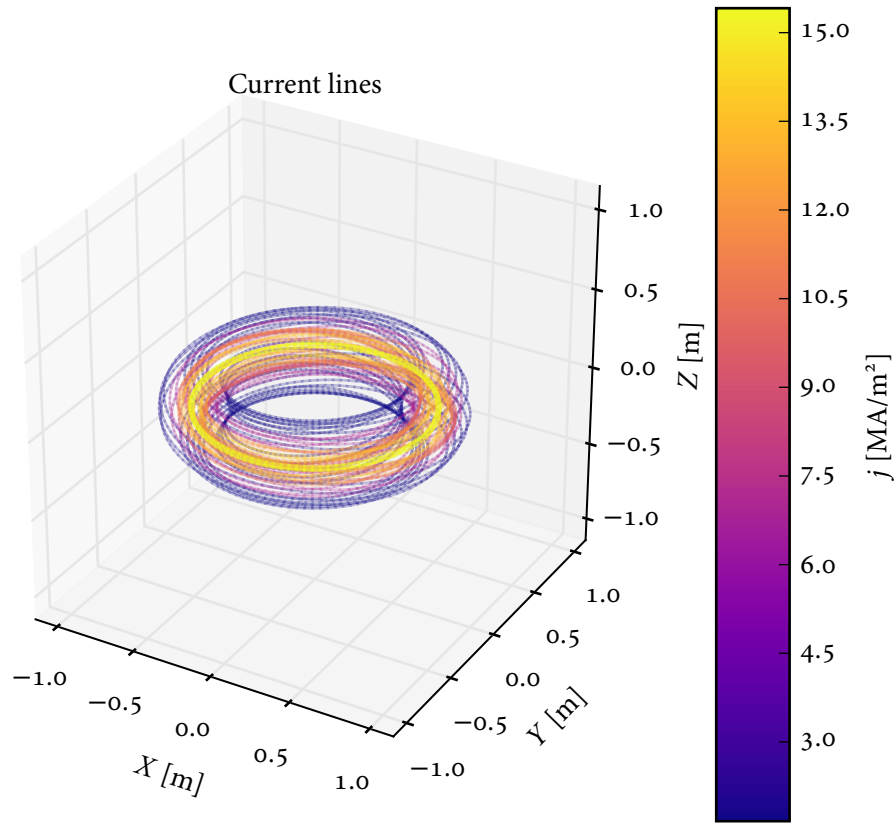
`eqtools` has been verified against the existing, thoroughly-tested IDL routines which are currently in use for handling coordinate mapping at Alcator C-Mod. Figure D.5 shows the discrepancy between the IDL routines and `eqtools` for the conversion of  $(R, Z)$  to  $\psi$ . The differences are small (of order  $10^{-7}$  Wb/rad, compared to a signal of order  $\pm 0.5$  Wb/rad), and are consistent with the fact that Python used double precision whereas IDL used single precision for the test.<sup>4</sup> Timing tests were conducted by converting a  $66 \times 66$  element  $(R, Z)$  grid (double the density of the grid EFIT provides the flux on) into each of the coordinates supported. This conversion was performed at 180 time slices (double the temporal resolution output by EFIT) and nearest-neighbor temporal interpolation was used. The test was run both with all points being processed at once (denoted “all” in table D.1) and with the routine being called inside a loop over all time points (denoted “loop” in table D.1). The conversion was performed twice with the same Equilibrium object in `eqtools` in order to assess the time savings from storing the spline coefficients. The IDL routines (with the exception of the conversion to  $R_{\text{mid}}$ ) support reuse of the spline coefficients from a single time slice, so a second run of the IDL code with the stored spline coefficients was performed when looping over time slices. The test was repeated 100 times, and the mean execution time to convert the  $(R, Z)$  grid for all time points is given in table D.1.

In all cases `eqtools` is faster than the IDL routines. Furthermore, the test results highlight the additional flexibility `eqtools` provides users. For the most efficient case where all of the points are passed at once, `eqtools` is only slightly faster for the simple conversions of  $(R, Z)$  to  $\psi$  and  $\psi_n$ . But, for the more complicated conversions

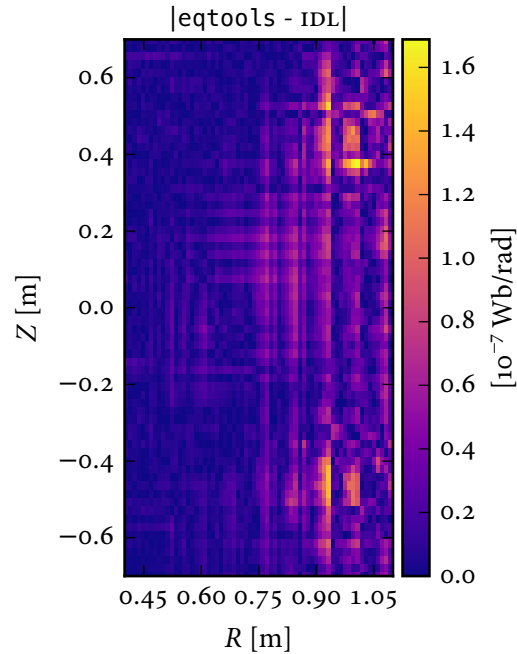
4. The data are retrieved from MDSplus in single precision but must be cast to double precision for the Python spline routines to work properly. While IDL can work in double precision, it defaults to single precision when given single precision input data.



**Figure D.3:** Magnetic field lines for Alcator C-Mod shot 1120907032, evaluated at  $t = 1$  s. Six lines were launched at the midplane at  $\varphi = 0$  at locations uniformly spaced in  $\psi$  over  $0.05 \leq \psi_n \leq 0.95$ . The lines were traced for 10 poloidal revolutions (i.e.,  $10q$  toroidal revolutions). Because the locations used are not rational surfaces the lines do not close on themselves in a finite number of orbits. The lines are drawn with 25% transparency and the color of the line gives the magnitude of the magnetic field – the  $1/R$  dependence can clearly be seen. While arrows are not shown on the field lines, note that this is a “reverse field” discharge, so the field lines are directed counterclockwise. This figure was produced using `EFIT_field_lines.py`.



**Figure D.4:** Current density field lines for Alcator C-Mod shot 1120907032, evaluated at  $t = 1$  s. Six lines were launched at the midplane at  $\varphi = 0$  at locations spaced uniformly in  $F$ . The lines were traced for the same number of revolutions as in figure D.3, and again do not close on themselves in a finite number of orbits because they were not launched on rational surfaces. The lines are again drawn with 25% transparency and the color of the line gives the magnitude of the current density – the concentration of the current in the core of the plasma can clearly be seen. While arrows are not shown on the field lines, note that this is a “reverse field” discharge, so the plasma current (and hence current density field lines) is directed counterclockwise. The fact that the poloidal current density curls in the same direction as the poloidal magnetic field is consistent with the fact that this discharge is paramagnetic, as shown in figure D.2. This figure was produced using `EFIT_field_lines.py`.



**Figure D.5:** Absolute difference between calculations of a mapping between the  $(R, Z)$  grid and poloidal flux  $\psi$  for `eqtools` and the current IDL implementation of the mapping routine for the analysis `EFIT` of Alcator C-Mod shot 1120914027. Relative differences of order  $10^{-7}$  Wb/rad are typical (the value of  $\psi$  is in the range  $\pm 0.5$  Wb/rad), and are consistent with the different float precisions used. This figure is an updated version of one which originally appeared in [500]. This figure was produced using `make_efit_compare_plot.py`.

to  $\varphi_n$ ,  $V_n$  and  $R_{\text{mid}}$ , `eqtools` is anywhere from 1.5 to 1.9 times faster than the IDL routines. Furthermore, the caching of intermediate results in `eqtools` accelerates the second call to a routine by as much as a factor of five (for the conversion to  $R_{\text{mid}}$ ). Where `eqtools` stands in stark contrast to the IDL routines is the case where the conversions are evaluated in a loop. While this case is not something which would be used directly as both codes support the evaluation of multiple time points at once, it is indicative of the performance to be expected should a user write a script which needs to evaluate each time point on a different grid, or otherwise needs the flexibility to split an operation up into steps applied to each time point. For this case, `eqtools` is anywhere from five to twenty times faster than the IDL routines. This is believed to be a result of the large overhead associated with function calls in

**Table D.1:** Time to convert all 180 time slices in milliseconds. “All” refers to passing all points to the routine at once, whereas “loop” refers to calling the routine in a loop over time points. “First” refers to the first call to the routine when data are collected from the server and spline coefficients are computed, “second” refers to the second call once the data and spline coefficients have been cached.

Conversion	IDL				eqtools			
	all		loop		all		loop	
	first	second	first	second	first	second	first	second
$(R, Z) \rightarrow \psi$	149	–	3147	3052	134	116	179	161
$(R, Z) \rightarrow \psi_n$	163	–	4024	3880	137	119	184	165
$(R, Z) \rightarrow \varphi_n$	473	–	5307	5196	254	165	303	215
$(R, Z) \rightarrow V_n$	468	–	5326	5220	253	164	302	215
$(R, Z) \rightarrow R_{\text{mid}}$	1400	–	5328	–	942	187	993	238

the IDL language. These test results highlight the improved performance and extra programming flexibility provided by eqtools.

# E

## gptools: an open-source Python package for general-purpose Gaussian process regression

### E.1 Package overview

`gptools` is a Python package which provides support for multivariate Gaussian process regression (GPR) with arbitrary derivative constraints and predictions. The source code is available in [306] and the user documentation is available in [307]. While there are several packages which support GPR available in Python [507–513] as well as many other programming languages [487, 488, 514–523], none of those available at the time that development of `gptools` started supported derivative constraints or predictions,<sup>1</sup> nor did any of the options available have much support for nonstationary data. Given that both of these capabilities are critical for the handling of tokamak profile data, it was necessary to implement a package to support Gaussian processes with derivative information from scratch. It was decided that the package should be as flexible as possible, including support for data of arbitrary dimension and both observations and predictions of derivatives of arbitrary order. The basic implementation follows [297], but with extensive additional development as described in [chapter 2](#) and [appendix B](#).

The core classes are `GaussianProcess` which represents the Gaussian process itself (including storing all the data and making predictions) and `Kernel` which represents a covariance kernel. Other classes for more advanced uses are described below and a diagram of the basic interactions between classes is given in [figure E.1](#).

---

1. GaPP [513, 524] is now available and also provides support for Gaussian process regression with derivative constraints and predictions, but not for nonstationary covariance kernels.



**Code Listing E.1:** Basic `gptools` example with univariate data. The use of the `param_bounds` keyword on line 8 when creating the `Kernel` instance sets a uniform prior distribution for  $\sigma_f$  and  $\ell$ .

---

```

1 # Import the package:
2 import gptools
3 # Data are assumed to be in 1D arrays X and y already.
4 # The uncertainty in y is in the 1D array err_y.
5 # The points you want to evaluate the curve at are in the 1D array Xstar.
6 # Create an SE covariance kernel:
7 k = gptools.SquaredExponentialKernel(
8     param_bounds=[(0, 2 * (max(y) - min(y))), (0, 2 * (max(X) - min(X)))]
9 )
10 # Create a Gaussian process with the SE covariance kernel but no data:
11 gp = gptools.GaussianProcess(k)
12 # Add the data :
13 gp.add_data(X, y, err_y=err_y)
14 # Add an exact zero-slope constraint at x=0:
15 gp.add_data(0, 0, n=0)
16 # Find the MAP estimate for the hyperparameters:
17 gp.optimize_hyperparameters()
18 # Make a prediction, including uncertainty estimate (standard deviation):
19 ystar, err_ystar = gp.predict(Xstar)

```

---

A basic example of using `gptools` is given in [code listing E.1](#).

The `gptools` package is organized into the following submodules and subpackages:

**gaussian\_process** Contains the core `GaussianProcess` class along with several helper classes.

**kernel** Subpackage with various covariance kernels.

**core** Provides the core `Kernel` class along with several helper classes.

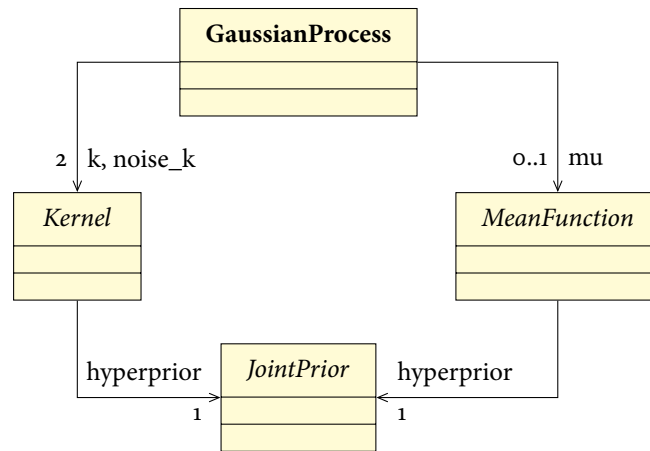
**squared\_exponential** Provides the squared exponential (SE) covariance kernel.

**rational\_quadratic** Provides the rational quadratic (RQ) covariance kernel.

**matern** Provides the Matérn covariance kernel.

**noise** Provides covariance kernels to model uncorrelated noise.

**gibbs** Provides the nonstationary Gibbs covariance kernel and various covariance length scale functions.



**Figure E.1:** Basic association structure of *gptools*. In diagrams such as this, classes are indicated with three-compartment boxes (see [figure E.2](#) for an explanation of class diagrams) and unidirectional associations are indicated with a solid line with a “v”-shaped arrowhead pointing towards the class which is known. The arrow is labeled with the multiplicity of the association and the name(s) of the attribute(s) in the containing class. So, the meaning of this diagram is that `GaussianProcess` contains two instances of `Kernel` (`k` and `noise_k`, see [figure E.2](#)) and zero or one instances of `MeanFunction`. `Kernel` and `MeanFunction` each contain an instance of `JointPrior`. Any given `Kernel`, `MeanFunction` or `JointPrior` instance has no record of what it is contained in, however. Note that `Kernel`, `MeanFunction` and `JointPrior` are all abstract classes.

**warping** Provides the `WarpedKernel` class and various input warping functions.

**mean** Provides the `MeanFunction` class and support for non-zero mean functions.

**utils** Provides helper classes and functions, including the `JointPrior` class used to construct prior distributions for hyperparameters.

**gp\_utils** Provides helper classes and functions which operate on `GaussianProcess` instances, and hence had to be in a separate module to avoid a circular import.

**splines** Provides the B-, M- and I-splines used in various parts of the package.

**error\_handling** Provides exception classes for error handling.

## E.2 Computational preliminaries: keeping track of hyperparameters in composite objects

A `GaussianProcess` instance has up to three attributes with hyperparameters: the covariance kernel `k`, the noise covariance kernel `noise_k` and the mean function `mu`. Each of these objects in turn may be composed of several objects combined through addition, multiplication, etc. In order to provide a seamless interface to the hyperparameters of each of these objects, `gptools` makes extensive use of “getter methods with property decorators.” This means that an attribute such as `gp.params` (the current values of the hyperparameters of the covariance kernel, noise covariance kernel and mean function of `GaussianProcess` instance `gp`) is in fact a method `params()` which has been marked with the Python decorator `@property` such that it is accessed with `gp.params` instead of `gp.params()`. The `params()` method then combines the hyperparameters from all three objects. If a simple array were returned, the user would be unable to modify the parameters using the array returned. Furthermore, each class with associated hyperparameters distinguishes between free and fixed hyperparameters. Therefore, properties like `free_params` are also implemented as getter methods with property decorators such that they return only the hyperparameters which are free. Again, it is desirable that the array which is returned supports modification of the elements. As such, the array-like classes `CombinedBounds` and `MaskedBounds` classes were implemented to solve this.

The `CombinedBounds` class represents the combination of the arrays from two objects. Its constructor takes two array-like objects `l1` and `l2` and provides a unified interface to them. Attempting to get an item retrieves the relevant item from the corresponding array. At present this is done internally by casting the two array-like objects to Python’s `list` data type and concatenating them before applying the array slice object, so only basic slice indexing is supported. Attempting to change the value of an item sets the relevant value of the relevant array. Because of how this is implemented at present, slice indexing is not supported.

The `MaskedBounds` class represents an array which has been masked, for instance the hyperparameters which are free to vary. Its constructor takes an array-like object `a` and a mask array `m`. The mask array should consist of the indices of the values which are to be included in the masked version of the array. If the arrays are stored internally as NumPy arrays, then the getting and setting of items will support the full complexity of Python’s slice indexing.

### E.3 Representing a Gaussian process computationally: the `GaussianProcess` class

The `GaussianProcess` class is the main class of *gptools*, and represents a Gaussian process. A class diagram is given in [figure E.2](#). This class is responsible both for storing the data as well as evaluating the covariance kernel and making predictions.

#### E.3.1 Attributes of the `GaussianProcess` class

The attributes of the `GaussianProcess` class are:

**num\_dim** Number of dimensions  $D$ :  $\mathbf{x} \in \mathbb{R}^D$ . This is actually a getter method with a property decorator which retrieves the value from `k`.

**k** Covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$ .

**noise\_k** Covariance kernel for the noise.

**mu** Parametric mean function  $m(\mathbf{x})$ .

**hyperprior** Prior distribution for the hyperparameters, returns  $\ln f_{\theta}(\theta)$  when called with the values for the hyperparameters  $\theta$ . This is actually a getter method with a property decorator which combines the prior distributions from `k`, `noise_k` and `mu`. Note that the parameters of `k`, `noise_k` and `mu` are all independent of each other, the classes have no knowledge of each other.

**X** Locations the training data are taken at, or the underlying quadrature points,  $\mathbf{X} \in \mathbb{R}^{n \times D}$  or  $\mathbf{X} \in \mathbb{R}^{n_Q \times D}$ .

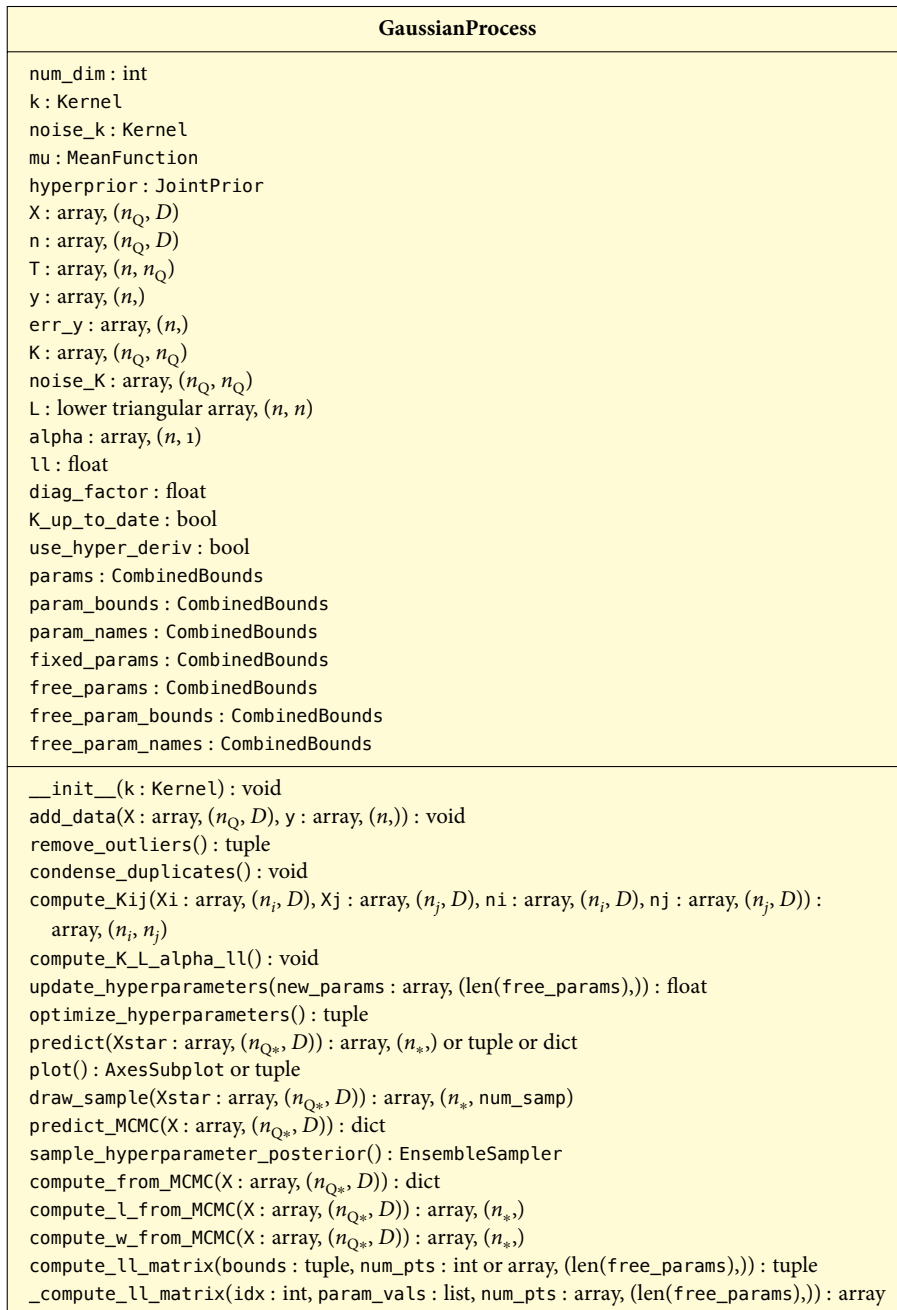
**n** Derivative orders for the training data. For each point in  $\mathbf{X}$ , this gives the order of derivative with respect to each dimension for the corresponding value  $y$ . By applying this to  $y$  and not  $\mathbf{z} = \mathbf{T}\mathbf{y}$  the Gaussian process can be constructed to include constraints on linear combinations of values and derivatives.

**T** Transformation matrix for the training data,  $\mathbf{T} \in \mathbb{R}^{n \times n_Q}$ .

**y** Observed (training) data,  $\mathbf{y} \in \mathbb{R}^n$  or  $\mathbf{z} \in \mathbb{R}^n$ .

**err\_y** Uncorrelated, possibly heteroscedastic noise on the observed data,  $\Sigma_{n,\text{obs}}$ .

**K** Covariance matrix,  $\mathbf{K}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n \times n}$  or  $\mathbf{K}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n_Q \times n_Q}$ .



**Figure E.2:** Class diagram for the GaussianProcess class. In this type of diagram, the class name is given in the top box, the attributes are given in the middle box and the methods are given in the bottom box. The data types of variables are indicated after the colon. For array data types, the shape is given as the following tuple. A GaussianProcess has  $n$  observations connected to  $n_Q$  quadrature points of dimension  $D$ . If there is no transformation matrix  $\mathbf{T}$  then  $n = n_Q$ .

- noise\_k** Noise covariance matrix,  $\Sigma_n \in \mathbb{R}^{n \times n}$  (less the observation noise given in `err_y`). Note that at present `gptools` applies the noise covariance kernel to the quadrature points  $\mathbf{X}$  and then transforms it according to  $\mathbf{T}\Sigma_n\mathbf{T}^\top$ . This is somewhat restrictive, because it does not allow for the inference of additional noise on the transformed quantities  $\mathbf{z}$ .
- L** Lower-triangular Cholesky decomposition of the total covariance matrix,  $\mathbf{L}\mathbf{L}^\top = \mathbf{K}_{\text{tot}} = \mathbf{T}(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \Sigma_n(\mathbf{X}, \mathbf{X}))\mathbf{T}^\top + \Sigma_{n, \text{obs}}$ , where  $\Sigma_{n, \text{obs}}$  is the observation noise on  $\mathbf{z}$  stored in `err_y`.
- alpha** Solution to  $\mathbf{K}_{\text{tot}}\alpha = \mathbf{z} - \mathbf{T}m(\mathbf{X})$ .
- ll** Log-posterior density of the fit,  $\ln f_{\theta|z}(\theta|z)$ .
- diag\_factor** Factor which is multiplied by the machine epsilon and added to  $\mathbf{K}_{\text{tot}}$  to stabilize the Cholesky decomposition.
- K\_up\_to\_date** Boolean used to keep track of whether or not  $\mathbf{K}$ ,  $\mathbf{L}$ ,  $\alpha$  and `ll` are up to date. Used to avoid having to recompute these each time `predict()` and the related methods called.
- use\_hyper\_deriv** Boolean used to indicate whether or not the derivatives with respect to the hyperparameters should be used when finding the MAP estimate.
- params, param\_bounds, param\_names** Hyperparameters, hyperparameter bounds and hyperparameter names from `k`, `noise_k` and `mu`. These are actually getter methods with property decorators which return `CombinedBounds` instances to facilitate getting and setting the parameters in the three objects all at once.
- fixed\_params** A list of boolean flags indicating which of the hyperparameters of `k`, `noise_k` and `mu` are to be held fixed while optimizing or sampling the hyperparameters. This is actually a getter method with a property decorator which returns a `CombinedBounds` instance to facilitate getting and setting the flags in the three objects all at once.
- free\_params, free\_param\_bounds, free\_param\_names** Equivalent to `params`, `param_bounds` and `param_names` but masked to have only the free hyperparameters.

### E.3.2 Creating a `GaussianProcess` and adding data

#### E.3.2.1 The `__init__()` method

All that is required to create a `GaussianProcess` is a covariance kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$  in the form of an instance of `Kernel`. If a noise covariance kernel is to be used, this must also be an instance of `Kernel`. A mean function  $m(\mathbf{x})$  can also be specified by providing an instance of `MeanFunction`.

#### E.3.2.2 The `add_data()` method

The `add_data()` method allows data to be added incrementally, not just at the time of creation. It accepts the independent variables  $\mathbf{X}$ , the data  $\mathbf{y}$  and optionally the data's standard deviation  $\sigma_y$ , the derivative orders of the observations  $\mathbf{n}$  and a transformation matrix  $\mathbf{T}$ . If no transformation matrix  $\mathbf{T}$  is passed but the `GaussianProcess` already has a  $\mathbf{T}$ , the  $\mathbf{T}$  for the data being added is automatically taken to be the identity matrix. If a transformation matrix  $\mathbf{T}$  is passed and the `GaussianProcess` already has a  $\mathbf{T}$ , then the new  $\mathbf{T}$  is given by the block diagonal matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\text{old}} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{\text{new}} \end{bmatrix}. \quad (\text{E.1})$$

#### E.3.2.3 The `condense_duplicates()` method

The way that the transformation matrices  $\mathbf{T}$  are combined in the `add_data()` method can make working with the `GaussianProcess` very slow if, for instance, `add_data()` is repeatedly called to add multiple channels from a line-integrated diagnostic which all share the same quadrature points  $\mathbf{X}_Q$ . The `condense_duplicates()` method finds the unique rows in  $\mathbf{X}$  and creates a new  $\mathbf{X}$  containing only these rows and constructs a new transformation matrix  $\mathbf{T}$  which utilizes this condensed version of  $\mathbf{X}$ . The `condense_duplicates()` method removes any columns of the new  $\mathbf{T}$  which are all zeros, indicating a quadrature point which does not actually enter into the calculation. This can happen when the weights of points outside of the shadow of the limiter are set to zero, for instance.

#### E.3.2.4 Automatically detecting suspicious data points: the `remove_outliers()` method

The `remove_outliers()` method implements a very primitive form of outlier detection as a way of enabling first-cut, automated processing of data. For rigorous analysis, one should of course always inspect their data and attempt to explain why

outlying points occur. For bulk processing of data, however, the ability to automatically fit many profiles without human intervention is invaluable. The method evaluates the mean  $\bar{z}_*$  at each of the data points and finds the points for which the weighted difference

$$\Delta = \frac{|\bar{z}_* - z|}{\sigma_z} \quad (\text{E.2})$$

is greater than a set threshold (by default the threshold is three). In this expression, the absolute value and division are both performed elementwise. Essentially, any points which lie more than three standard deviations away from the mean curve are thrown out. This method has a significant potential issue: the hyperparameters must be estimated before the mean curve  $\bar{z}_*$  can be predicted. If a point is sufficiently far off, it will cause the covariance length scale to become very small such that the curve passes close to *all* of the points. Therefore, this method is only appropriate for rough processing of the data, and will work best when good a priori estimates for the hyperparameters are available (such that the rough guesses can be used for this process and the MAP estimate does not need to be found with potential outliers still in the data set). In principle, a robust approach which incorporates the possibility of outliers into the probability model instead of simply throwing suspicious data out should be preferred [525–528], but this level of sophistication was not employed here.

### E.3.3 Formulating the pieces necessary to make an inference

The `compute_K_L_alpha_ll()` method is the fundamental method for training the Gaussian process once the data have been added using the `add_data()` method. The `update_hyperparameters()` method wraps the `compute_K_L_alpha_ll()` method in a way such that it can act as the objective function for an optimizer or MCMC sampler.

#### E.3.3.1 The `compute_Kij()` method

The `compute_Kij()` method is used to form a covariance matrix  $\mathbf{K}(\mathbf{X}_i, \mathbf{X}_j)$  by evaluating the covariance function  $k(\mathbf{x}_i, \mathbf{x}_j)$  between all possible pairs of points in  $\mathbf{X}_i$  and  $\mathbf{X}_j$ . It tiles the input arrays of locations  $\mathbf{X}_i, \mathbf{X}_j$  and derivative orders  $\mathbf{n}_i, \mathbf{n}_j$  into the appropriate shape to be passed to `Kernel.__call__()` and then evaluates the covariance kernel. A flag selects between evaluating the covariance kernel, the noise covariance kernel and a user-specified covariance kernel. One shortcoming of this method is



that it does not exploit the symmetry of  $\mathbf{K}(\mathbf{X}, \mathbf{X})$ , but an earlier implementation using a double for-loop which did exploit this symmetry was found to be much slower.

### E.3.3.2 The `compute_K_L_alpha_ll()` method

The algorithm for the `compute_K_L_alpha_ll()` method is based on algorithm 2.1 in [297], but has been extended to handle transformed quantities and a mean function according to the derivation in section 2.3.9.

Because the `compute_K_L_alpha_ll()` method is called by several other methods and in order to avoid repeated state-checking code, the method first checks to see if the attributes  $\mathbf{K}(\mathbf{X}, \mathbf{X})$ ,  $\mathbf{L}$ ,  $\alpha$  and  $\ln f_{\theta|z}(\theta|z)$  are up to date. If they are not up to date, the method computes  $\mathbf{K}(\mathbf{X}, \mathbf{X})$  and  $\Sigma_n$  (i.e., `K` and `noise_K`) using the `compute_Kij()` method. The noise kernel must be handled specially, as the `DiagonalNoiseKernel` class is meant to handle *uncorrelated* noise, but the way the `compute_Kij()` method works breaks this assumption when there are multiple observations at the same location (see section E.4.2). Therefore, if `noise_k` is an instance of `DiagonalNoiseKernel`,  $\Sigma_n$  is produced directly from the  $\sigma_n$  stored in `noise_k` without calling `compute_Kij()`. If  $\mathbf{T}$  is present, the product  $\mathbf{T}(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \Sigma_n)\mathbf{T}^T$  is performed. If  $\mathbf{T}$  is meant to be the identity matrix, this is represented as `T = None` so that this can be skipped. Once the component covariance matrices have been stored the total covariance matrix

$$\mathbf{K}_{\text{tot}} = \mathbf{T}(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \Sigma_n(\mathbf{X}, \mathbf{X}))\mathbf{T}^T + \Sigma_{n, \text{obs}} \quad (\text{E.3})$$

is formed and its lower triangular Cholesky decomposition  $\mathbf{L}\mathbf{L}^T = \mathbf{K}_{\text{tot}}$  is computed and stored. Next, if a mean function is present, the quantity  $z - \mathbf{T}\mathbf{m}(\mathbf{X})$  is computed, where again the product with  $\mathbf{T}$  is skipped if  $\mathbf{T}$  is meant to be the identity matrix. Then the vector

$$\alpha = \mathbf{K}_{\text{tot}}^{-1}(z - \mathbf{T}\mathbf{m}(\mathbf{X})) = (\mathbf{L}\mathbf{L}^T)^{-1}(z - \mathbf{T}\mathbf{m}(\mathbf{X})) = \mathbf{L}^T \backslash (\mathbf{L} \backslash (z - \mathbf{T}\mathbf{m}(\mathbf{X}))) \quad (\text{E.4})$$

(where<sup>2</sup>  $\mathbf{b} = \mathbf{A} \backslash \mathbf{c}$  is the solution to  $\mathbf{A}\mathbf{b} = \mathbf{c}$ ) is computed using an efficient solver for matrices which have been Cholesky-decomposed. The log-likelihood is then

2. The notation  $\mathbf{b} = \mathbf{A}^{-1}\mathbf{c}$  would have worked equally well, but the backslash notation is used to emphasize that the use of the Cholesky decomposition and efficient solver algorithms for triangular systems obviates the need to compute the matrix inverse directly.

computed from

$$\begin{aligned} \ln f_{Z|\Theta}(z|\theta) &= -\frac{1}{2}(z - \mathbf{T}\mathbf{m}(\mathbf{X}))^\top (\mathbf{T}(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)\mathbf{T}^\top + \boldsymbol{\Sigma}_{n,\text{obs}})^{-1}(z - \mathbf{T}\mathbf{m}(\mathbf{X})) \\ &\quad - \frac{1}{2} \ln |\mathbf{T}(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Sigma}_n)\mathbf{T}^\top + \boldsymbol{\Sigma}_{n,\text{obs}}| - \frac{n}{2} \ln 2\pi \end{aligned} \quad (\text{E.5})$$

$$= -\frac{1}{2}(z - \mathbf{T}\mathbf{m}(\mathbf{X}))\mathbf{K}_{\text{tot}}^{-1}(z - \mathbf{T}\mathbf{m}(\mathbf{X})) - \frac{1}{2} \ln |\mathbf{K}_{\text{tot}}| - \frac{n}{2} \ln 2\pi \quad (\text{E.6})$$

$$= -\frac{1}{2}(z - \mathbf{T}\mathbf{m}(\mathbf{X}))^\top \boldsymbol{\alpha} - \sum_{i=1}^n \ln L_{ii} - \frac{n}{2} \ln 2\pi, \quad (\text{E.7})$$

where in the last step the useful fact that the determinant of a matrix is equal to twice the product of the diagonal of the Cholesky decomposition has been used. This is then converted to the log-posterior density  $\ln f_{\Theta|Z}(\theta|z)$  by adding the log-prior density for the hyperparameters  $\ln f_{\Theta}(\theta)$ .

If the derivatives with respect to the hyperparameters are requested, the derivatives of the log-posterior with respect to each of the free parameters are computed according to [equation \(2.63\)](#):

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \ln f_{\Theta|Z}(\theta|z) &= \frac{1}{2} \left( \mathbf{T} \frac{\partial \mathbf{m}(\mathbf{X})}{\partial \theta_i} \right)^\top \mathbf{K}_{\text{tot}}^{-1}(z - \mathbf{T}\mathbf{m}(\mathbf{X})) \\ &\quad + \frac{1}{2} (z - \mathbf{T}\mathbf{m}(\mathbf{X}))^\top \mathbf{K}_{\text{tot}}^{-1} \frac{\partial \mathbf{K}_{\text{tot}}}{\partial \theta_i} \mathbf{K}_{\text{tot}}^{-1} (z - \mathbf{T}\mathbf{m}(\mathbf{X})) \\ &\quad + \frac{1}{2} (z - \mathbf{T}\mathbf{m}(\mathbf{X}))^\top \mathbf{K}_{\text{tot}}^{-1} \left( \mathbf{T} \frac{\partial \mathbf{m}(\mathbf{X})}{\partial \theta_i} \right) \\ &\quad - \frac{1}{2} \text{tr} \left( \mathbf{K}_{\text{tot}}^{-1} \frac{\partial \mathbf{K}_{\text{tot}}}{\partial \theta_i} \right) + \frac{\partial}{\partial \theta_i} \ln f_{\Theta}(\theta) \end{aligned} \quad (\text{E.8})$$

$$= \left( \mathbf{T} \frac{\partial \mathbf{m}(\mathbf{X})}{\partial \theta_i} \right)^\top \boldsymbol{\alpha} + \frac{1}{2} \boldsymbol{\alpha}^\top \frac{\partial \mathbf{K}_{\text{tot}}}{\partial \theta_i} \boldsymbol{\alpha} - \frac{1}{2} \text{tr} \left( (\mathbf{L}\mathbf{L}^\top)^{-1} \frac{\partial \mathbf{K}_{\text{tot}}}{\partial \theta_i} \right) + \frac{\partial}{\partial \theta_i} \ln f_{\Theta}(\theta). \quad (\text{E.9})$$

Note that the first term only enters when  $\theta_i$  is a parameter of the mean function and the second and third terms only enter when  $\theta_i$  is a hyperparameter of the covariance kernel or noise covariance kernel. Therefore, this expression is implemented by first looping over the free hyperparameters of the covariance kernel and noise covariance kernel and evaluating the second and third terms, then looping over the free parameters of the mean function (if it is present) and evaluating the first term, then finally evaluating the fourth term by looping over all of the parameters and hyperparameters and adding the log-derivative of the prior to the result previously obtained for each one.

Finally, when all of this has completed successfully, `K_up_to_date` is set to `True` so that these quantities do not have to be recomputed each time `predict()` or a related method is called.

### E.3.3.3 The `update_hyperparameters()` method

The `update_hyperparameters()` method takes the new values of the free hyperparameters as an argument, along with optional flags to determine what is returned and how edge cases and errors are handled. The method first updates the hyperparameters in the covariance kernel, noise covariance kernel and mean function. If the `exit_on_bounds` flag is `True` it checks to see if the log-prior density  $\ln f_{\theta}(\theta)$  is finite before calling `compute_K_L_alpha_ll()`. What is returned is one or both of  $-\ln f_{\theta|Z}(\theta|z)$  and  $-\nabla_{\theta} \ln f_{\theta|Z}(\theta|z)$ , depending on the state of the `use_hyper_deriv` attribute and the `hyper_deriv_handling` flag. The negative of the log-posterior density is used so that the `update_hyperparameters()` method can be used as the objective function of one of the many minimizers provided by SciPy.

### E.3.4 Finding the MAP estimate: the `optimize_hyperparameters()` method

The `optimize_hyperparameters()` method uses one of several optimization routines in SciPy [293] to find the values for the free hyperparameters which maximize the log-posterior. By default, the sequential quadratic programming routine `SLSQP` is used [402]. This optimizer was found to have acceptable performance both when analytic derivatives with respect to the hyperparameters are available and when these must be estimated using finite differences. Note that there is a slight issue with SciPy's implementation of finite differences: a forward difference is always used, so if a variable is at the upper limit of its bounds, SciPy will attempt to evaluate the log-likelihood at a point which is not allowed by the prior distribution. The optimizer can be started from the current value of `free_params` or a set of one or more points drawn from the prior distribution  $f_{\theta}(\theta)$ . When multiple points are used, the evaluations can proceed in parallel on as many processors as are available.

### E.3.5 Making a prediction and drawing random samples

#### E.3.5.1 The `predict()` method

The `predict()` method can proceed in one of two ways: if MCMC sampling of the hyperparameters is to be used, the relevant arguments are passed directly to the

`predict_MCMC()` method. Otherwise, the predictive mean and covariance are calculated according to equation (2.59). The covariance matrix  $\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$  is computed using the `compute_Kij()` method. The user can decide whether or not the noise kernel  $\Sigma_n$  is included in the prediction and its variance using the `noise` keyword; this is not shown in the following because it is not a very common use case. One can see the effect simply by considering the noise covariance kernel to be added to the signal covariance kernel when computing  $\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$  and  $\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$ .

The predictive mean is

$$\bar{\mathbf{z}}_* = \mathbf{T}_* \mathbf{m}(\mathbf{X}_*) + \mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top \mathbf{K}_{\text{tot}}^{-1} (\mathbf{z} - \mathbf{T} \mathbf{m}(\mathbf{X})) \quad (\text{E.10})$$

$$= \mathbf{T}_* (\mathbf{m}(\mathbf{X}_*) + \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top \boldsymbol{\alpha}), \quad (\text{E.11})$$

where the mean function  $\mathbf{m}(\mathbf{X}_*)$  and the product with the output transformation matrix  $\mathbf{T}_*$  are only computed if the respective attributes are present.

The predictive covariance matrix is only computed if it is requested, thereby allowing considerable time savings when uncertainty estimates are not required. To compute the predictive covariance matrix, first let

$$\mathbf{v} = \mathbf{L} \backslash \mathbf{T} \mathbf{K}(\mathbf{X}, \mathbf{X}_*). \quad (\text{E.12})$$

The predictive covariance matrix is then

$$\text{cov}[\mathbf{z}_*, \mathbf{z}_* | \mathbf{z}] = \mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \mathbf{T}_*^\top - \mathbf{T}_* \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top \mathbf{K}_{\text{tot}}^{-1} \mathbf{T} \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \mathbf{T}_*^\top \quad (\text{E.13})$$

$$= \mathbf{T}_* (\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top (\mathbf{L} \mathbf{L}^\top)^{-1} \mathbf{T} \mathbf{K}(\mathbf{X}, \mathbf{X}_*)) \mathbf{T}_*^\top \quad (\text{E.14})$$

$$= \mathbf{T}_* (\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \mathbf{T}^\top \mathbf{L}^\top \backslash \mathbf{L} \mathbf{T} \mathbf{K}(\mathbf{X}, \mathbf{X}_*)) \mathbf{T}_*^\top \quad (\text{E.15})$$

$$= \mathbf{T}_* (\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{v} \mathbf{v}^\top) \mathbf{T}_*^\top. \quad (\text{E.16})$$

If only the standard deviation is required, the square root of the diagonal of this matrix is then extracted.

If the user requests that samples be drawn from the distribution during this call,  $\bar{\mathbf{z}}_*$  and  $\text{cov}[\mathbf{z}_*, \mathbf{z}_* | \mathbf{z}]$  are then passed to the `draw_sample()` method so that they do not need to be re-computed. If a sample rejection function is provided in the `rejection_func` keyword, the samples are then filtered through this function. This enables the rejection of negative and/or non-monotonic samples, for example. The uncertainty and mean calculation can be overridden with the `full_MC` keyword. If this keyword is `True`, the mean and covariance matrix returned are the sample mean and covariance matrix of the *filtered* samples. If the constraint information present in the rejection function is called  $C$ , then the quantities returned then represent estimates of  $\mathbb{E}[\mathbf{z}_* | \mathbf{z}, \boldsymbol{\theta}, C]$  and  $\text{cov}[\mathbf{z}_*, \mathbf{z}_* | \mathbf{z}, \boldsymbol{\theta}, C]$ , where the dependence on the hyperparameters  $\boldsymbol{\theta}$  has also been made explicit. Care must be exercised when using

this formulation, however, as the MAP estimate of the hyperparameters  $\theta$  will be  $\theta = \arg \max_{\theta} f_{\theta|Z}(\theta|z)$  and *not*  $\arg \max_{\theta} f_{\theta|Z,C}(\theta|z, C)$ , nor will MCMC sampling be capable of producing samples from  $f_{\theta|Z,C}(\theta|z, C)$ . If this capability is necessary, more advanced techniques are needed [340], but such sophistication is not supported by `gptools`.

### E.3.5.2 The `draw_sample()` method

The `draw_sample()` method draws one or more samples of the profile at a fixed value of the hyperparameters,  $\tilde{z}_* \sim f_{z_*|Z,\theta}(z_*|z, \theta)$ , using equation (2.52), reproduced here for convenience:

$$\tilde{z}_* = \mathbf{A}\mathbf{u} + \bar{z}_* \quad (\text{E.17})$$

$$\mathbf{A}\mathbf{A}^T = \text{cov}[z_*, z_* | z] \quad (\text{E.18})$$

$$\mathbf{u} \sim \mathcal{N}(0_{n_*}, \mathbf{I}_{n_*}). \quad (\text{E.19})$$

The user can either allow the method to produce its own random numbers (using the appropriate routines from NumPy [529]), or the user can provide random numbers which follow either the standard normal distribution  $\mathcal{N}(0, 1)$  or the uniform distribution  $\mathcal{U}(0, 1)$ . This allows the user to employ advanced sampling strategies such as Latin hypercube sampling [355], quasi-Monte Carlo sampling [291] or sparse quadrature [338] which may require the problem to be mapped to the unit hypercube for efficient implementation. To compute the matrix square root  $\mathbf{A}$ , the user can choose between using the Cholesky decomposition

$$\text{cov}[z_*, z_* | z] = \mathbf{A}\mathbf{A}^T = \mathbf{L}\mathbf{L}^T \quad (\text{E.20})$$

and the eigendecomposition

$$\text{cov}[z_*, z_* | z] = \mathbf{A}\mathbf{A}^T = \mathbf{Q}\mathbf{\Lambda}^{1/2}(\mathbf{Q}\mathbf{\Lambda}^{1/2})^T. \quad (\text{E.21})$$

When using the eigendecomposition, the number of eigenvalues to retain can be selected. For efficiency, only the requested number of eigenvectors and eigenvalues are computed. Sometimes it is useful to use the (weighted) eigenvectors as basis functions, in which case the values  $\mathbf{u}$  are the weights of each basis function contained in the columns of  $\mathbf{Q}\mathbf{\Lambda}^{1/2}$ . But, the eigenvectors are only defined up to a constant factor; even if normalized eigenvectors are used they are only specified up to a sign. As the sign chosen by the SciPy routine `eigh()` is not defined and can vary from platform to platform and for small changes of the hyperparameters, the `modify_sign` keyword allows the user to control which sign is used. Using this keyword, the sign of the

value, slope or concavity of the eigenvectors (where the derivatives are computed with forward or backwards finite differences, as appropriate) can be set.

If a mean and covariance matrix are not provided, the `predict()` method is used to compute the posterior mean and covariance matrix. If precomputed random variables are not provided and the method is not set to use the eigendecomposition of the posterior covariance matrix, then the NumPy function `multivariate_normal()` is used to draw samples from  $\mathcal{N}(\bar{\mathbf{z}}_*, \text{cov}[\mathbf{z}_*, \mathbf{z}_* | \mathbf{z}])$  directly. If precomputed random variables are not provided and the method is set to use the eigendecomposition of the posterior covariance matrix, then the NumPy function `standard_normal()` is used to draw samples from  $\mathcal{N}(0, 1)$ . If uniform precomputed random variables are provided they are converted to standard normal variables using the inverse CDF of the standard normal,  $u = \Phi^{-1}(U)$  where  $u \sim \mathcal{N}(0, 1)$  and  $U \sim \mathcal{U}(0, 1)$ .

### E.3.6 Using MCMC to sample the hyperparameters

#### E.3.6.1 Generating MCMC samples: the `sample_hyperparameter_posterior()` method

The `sample_hyperparameter_posterior()` method uses an affine-invariant ensemble sampler [353, 354] to draw samples of the hyperparameters from  $f_{\theta|z}(\theta|z)$ . The `EnsembleSampler` instance is returned such that samples can be added as needed to obtain burn-in and then used with the other methods described in this section to evaluate the actual profiles. While it would be in the spirit of object orientation to store the `EnsembleSampler` as an attribute of `GaussianProcess`, this is not possible because `EnsembleSampler` is not pickleable, which means that a `GaussianProcess` instance which stores an `EnsembleSampler` would no longer be pickleable and hence could not be passed to other processors for parallel evaluation.

#### E.3.6.2 Computing profiles from MCMC samples: the `compute_from_MCMC()` method

The `compute_from_MCMC()` method uses the MCMC samples obtained using the `sample_hyperparameter_posterior()` method to evaluate the profile. It uses the `_ComputeGPWrapper` helper object to enable parallel execution on as many processors as are available. What it returns is a collection of samples of  $\mathbb{E}[\mathbf{z}_* | \mathbf{z}, \theta]$  and  $\text{cov}[\mathbf{z}_*, \mathbf{z}_* | \mathbf{z}, \theta]$ .

### E.3.6.3 Combining profiles computed from MCMC samples: the `predict_MCMC()` method

The `predict_MCMC()` method calls the `compute_from_MCMC()` method and post-processes the results as described in [section 2.3.7](#). The `full_MC` keyword overrides the use of the formulas in [section 2.3.7](#). Instead, for each sample of the hyperparameters  $\theta^{(i)}$ , a number of samples are drawn. These can then be filtered through a rejection function the same as was described for the `predict()` method in [section E.3.5.1](#). The mean and covariance matrix returned are then the sample mean and covariance matrix obtained from the surviving samples,  $\mathbb{E}[z_* | z, C]$  and  $\text{cov}[z_*, z_* | z, C]$ . Note, however, that the samples of the hyperparameters were still taken from  $f_{\theta|z}(\theta|z)$  and *not*  $f_{\theta|z,C}(\theta|z, C)$ , so one must be very careful when considering the statistical implications of the result.

## E.4 Representing a covariance kernel computationally: the `Kernel` class and the `gptools.kernel` subpackage

The `kernel` subpackage contains all of the implementations of covariance kernels in `gptools`. A basic diagram of the subpackage is given in [figure E.3](#).

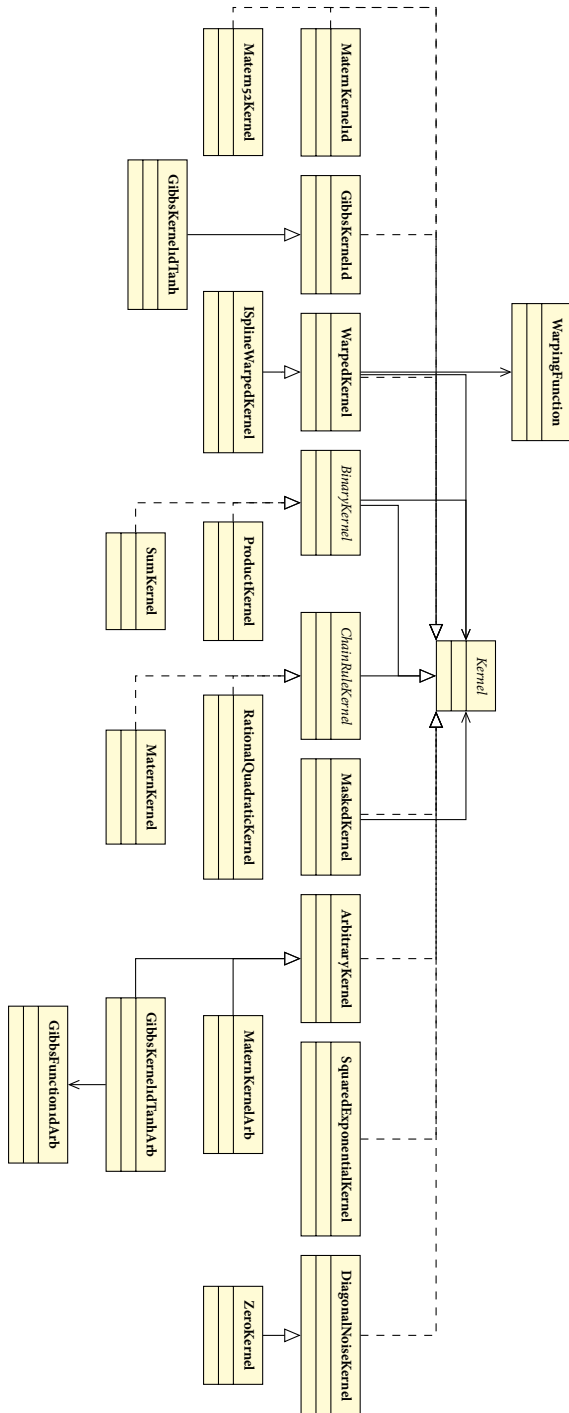
### E.4.1 The core `Kernel` classes

#### E.4.1.1 The `Kernel` abstract class

The `Kernel` class is the other core class of `gptools`, and represents a covariance kernel. A class diagram is given in [figure E.4](#). This class is responsible for keeping track of the current value of the hyperparameters and evaluating the covariance matrix  $\mathbf{K}(\mathbf{X}_i, \mathbf{X}_j)$  for arbitrary pairs of  $\mathbf{X}_i, \mathbf{X}_j$ .

Classes which implement `Kernel` must implement the `__call__()` method, which is often the only thing that needs to be done to create a new covariance kernel. The `__call__()` method evaluates the covariance kernel  $k(x_i, x_j)$  between pairs of points. These must already have been formed into the  $M \times D$  arrays  $X_i$  and  $X_j$  by, for instance, the `compute_Kij()` method of the `GaussianProcess` class. The  $M \times D$  arrays `ni` and `nj` contain the orders of differentiation with respect to each dimension for each point in  $X_i$  and  $X_j$ , respectively. The `hyper_deriv` keyword can be set to an integer, in which case the derivative with respect to the hyperparameter at the index should be returned.

The `__add__()` method overrides the `+` operator such that the sum of two kernels can be created simply by typing `k1 + k2`, which will return an instance of `SumKernel`.



**Figure E.3:** Basic structure of the kernel subpackage. Only a single representative subclass for each of `GibbsKernelId` and `WarpedKernel` is shown. Names of abstract classes are given in *italics*; names of classes which may be instantiated are given in **boldface**. Implementations of an abstract class are given as dashed lines with open arrowheads, inheritance of classes is given as solid lines with open arrowheads and unidirectional associations are indicated as solid lines with “v”-shaped arrowheads.





**Figure E.4:** Class diagram for the `Kernel` abstract class. `__call__()` is a placeholder method and must be implemented in any subclass which implements `Kernel`.

Likewise, the `__mul__()` method overrides the `*` operator such that `k1 * k2` returns an instance of `ProductKernel`.

Many of the kernels require the value of  $r^2/\ell^2$ , as defined in [equation \(B.1\)](#):

$$\frac{r^2}{\ell^2} \rightarrow \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} = \sum_{d=1}^D \frac{(x_{i_d} - x_{j_d})^2}{\ell_d^2}. \quad (\text{E.22})$$

The `_compute_r2l2()` method takes as input the difference matrix  $\boldsymbol{\tau} = \mathbf{X}_i - \mathbf{X}_j$  and evaluates [equation \(E.22\)](#) for each point. Optionally, it can also return the  $M \times D$  matrix of tiled  $\ell$  values for each dimension of each point. This method assumes that the covariance length scales are the last elements in `params`.

The attributes of the `Kernel` class are:

**num\_dim** The number of dimensions the `Kernel` is defined over.

**num\_params** The number of hyperparameters.

**enforce\_bounds** If this is `True`, new values of the hyperparameters which exceed the bounds are set to lie right at the bounds when `set_hyperparams()` is called. This is a way to avoid issues when the finite difference routine in SciPy's optimization package attempts to evaluate a point which is outside of the bounds.

**hyperprior** The prior distribution for the hyperparameters.

**params** The current values of the hyperparameters.

**param\_bounds** The bounds on each of the hyperparameters. This is actually a getter method with a property decorator which returns the `param_bounds` attribute of `hyperprior`.

**param\_names** The names of the hyperparameters.

**fixed\_params** If the entry corresponding to a given hyperparameter is `True` then that hyperparameter will be held fixed during optimization and MCMC sampling.

**num\_free\_params** The number of free hyperparameters. This is actually a getter method with a property decorator.

**free\_param\_idxs** Indices of the free hyperparameters in `params`. This is actually a getter method with a property decorator.

**free\_params** The current values of the free hyperparameters. This is actually a getter method with a property decorator which returns a `MaskedBounds` instance to facilitate convenient setting of the free hyperparameters.

**free\_param\_bounds** The bounds on each of the hyperparameters. This is actually a getter method with a property decorator which returns a `MaskedBounds` instance to facilitate convenient setting of the bounds of the free hyperparameters.

**free\_param\_names** The names of the free hyperparameters. This is actually a getter method with a property decorator which returns a `MaskedBounds` instance to facilitate convenient setting of the names of the free hyperparameters.

### E.4.1.2 Representing binary operations on covariance kernels

The `BinaryKernel` class represents binary operations on covariance kernels, such as addition and multiplication. Given a binary operation such as  $k_1 + k_2$ , the hyperparameters of  $k_1$  will come first and the hyperparameters of  $k_2$  will come second.

The `SumKernel` class represents the sum of two covariance kernels:  $k = k_1 + k_2$ . If derivatives with respect to the hyperparameters are requested, only the covariance kernel to which the hyperparameter belongs is evaluated.

The `ProductKernel` class represents the multiplication of two covariance kernels:  $k = k_1 k_2$ . Equation (B.106) is implemented in the `__call__()` method as follows: first, the derivative orders  $n_i$  and  $n_j$  are joined into the  $M \times 2D$  array  $n_{ij}$  and the unique rows in  $n_{ij}$  are found. Then, equation (B.106) is evaluated for each unique row in  $n_{ij}$ . This is accomplished by forming the “derivative pattern” `deriv_pattern` representing  $n_{ij}$ . For each dimension of  $[\mathbf{x}_i, \mathbf{x}_j]^T$  that is to be differentiated with respect to, `deriv_pattern` contains the index of that dimension repeated a number of times equal to the order of differentiation with respect to that dimension. This corresponds to the set  $\{1, \dots, n\}$  used to index the variables  $x_{[i]}$  in equation (B.106).<sup>3</sup> In other words, if one wishes to find  $\partial^3 k(\mathbf{x}_i, \mathbf{x}_j) / \partial x_{i_1}^2 \partial x_{j_2}$  of the 2D covariance kernel  $k$ , `deriv_pattern` will be `[1, 1, 4]`. Next, the power set  $\mathcal{P}(\{1, \dots, n\})$  is obtained using the `powerset()` function given in [530]. Finally, the method loops over all of the subsets in  $\mathcal{P}(\{1, \dots, n\})$  and evaluates the product of the appropriate derivatives of the two covariance kernels and adds this to the result.

### E.4.1.3 Representing covariance kernels which must be differentiated using the chain rule: the `ChainRuleKernel` abstract class

The `ChainRuleKernel` class represents covariance kernels which must be differentiated through use of the chain rule, including the rational quadratic and Matérn covariance kernels. Implementing subclasses must implement the following methods:

`_compute_k(tau)` Evaluate the covariance kernel  $k(\boldsymbol{\tau})$  at  $\boldsymbol{\tau} = \mathbf{X}_i - \mathbf{X}_j$ . This case with no derivatives is done separately for efficiency.

`_compute_y(tau)` Evaluate the “inner” term  $y(\boldsymbol{\tau})$  of  $k = k(y(\boldsymbol{\tau}))$ .

`_compute_dk_dy(y, n)` Evaluate  $d^n k(y) / dy^n$ .

`_compute_dy_dtau(tau, b, r2l2)` Evaluate  $\partial^n y / \prod_{i=1}^D \partial \tau_i^{n_i}$  where  $n = \sum_{i=1}^D n_i$ .

3. In the actual implementation the indexing of the dimension is zero-based. One-based indexing is used throughout this thesis for clarity.

The `__call__()` method first evaluates the difference matrix  $\boldsymbol{\tau} = \mathbf{X}_i - \mathbf{X}_j$ . Next, it computes  $n_j$ , the total number of times a derivative with respect to any component of  $\mathbf{x}_j$  is taken for each of the points so that the appropriate power of [equation \(B.10\)](#) can be included at the end. After that, the derivative orders  $n_i$  and  $n_j$  are joined into the  $M \times 2D$  array `n_combined` and the unique rows are found. The method then loops over the unique rows and calls the internal method `_compute_dk_dtau()` to evaluate the appropriate derivatives with respect to the dimensions of  $\boldsymbol{\tau}$ . Finally, the result is multiplied by  $\sigma_f^2(-1)^{n_j}$  and returned.

The `_compute_dk_dtau()` method finds  $\partial^n k(\boldsymbol{\tau}) / \prod_{d=1}^D \partial \tau_d^{n_d}$  where  $n = \sum_{d=1}^D n_d$  by carrying out the sum in [equation \(B.30\)](#). It takes as arguments the difference matrix  $\boldsymbol{\tau} = \mathbf{X}_i - \mathbf{X}_j$  and a  $D$  element array giving the order of differentiation with respect to each dimension. It starts out by forming a similar derivative pattern array to what was used in [section E.4.1.2](#), only here the derivatives are only with respect to the elements of  $\boldsymbol{\tau}$  and not the concatenation of  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . Then, if the derivative pattern is empty (i.e., no derivatives), it simply uses the `_compute_k()` method to evaluate  $k(\boldsymbol{\tau})$ . Otherwise, it uses the `generate_set_partitions()` function described in [section E.7.4](#) to generate the partitions  $\Pi$ . For each partition  $\pi \in \Pi$ , it uses the internal method `_compute_dk_dtau_on_partition()` to compute  $k^{(|\pi|)}(y(\boldsymbol{\tau})) \prod_{B \in \pi} \partial^{|B|} y(\boldsymbol{\tau}) / \prod_{j \in B} \partial \tau_{[j]}$  and adds this to the running sum.

The `_compute_dk_dtau_on_partition()` method takes as arguments the difference matrix  $\boldsymbol{\tau} = \mathbf{X}_i - \mathbf{X}_j$  and a list of arrays representing a partition of the derivative pattern from the `_compute_dk_dtau()` method. The method first uses the `_compute_y()` method to find  $y(\boldsymbol{\tau})$ . Then it uses the `_compute_dk_dy()` method to find  $d^{|\pi|} k(y) / dy^{|\pi|}$ . Finally, it loops over all of the blocks  $B$  in the partition  $\pi$  and finds  $\partial^{|B|} y(\boldsymbol{\tau}) / \prod_{j \in B} \partial \tau_j$  using the `_compute_dy_dtau()` method and multiplies this into the running product.

#### E.4.1.4 Using numerical methods to handle arbitrary covariance kernels: the `ArbitraryKernel` class

The `ArbitraryKernel` class represents a covariance kernel for which  $k(\mathbf{x}_i, \mathbf{x}_j)$  has been implemented as a standalone Python function but none of the work to compute derivatives or implement a subclass of `Kernel` has been done yet. Derivatives of the covariance kernel are instead found using the numerical differentiation capabilities of `mpmath` [531]. The `_mask_cov_func()` helper method puts the covariance function into a form with the correct fingerprint for `mpmath` to use.

#### E.4.1.5 Applying covariance kernels only to certain dimensions: the `MaskedKernel` class

It is often desirable to form a tensor product of covariance kernels. For instance, to have a Gibbs covariance kernel with tanh covariance length scale function in the spatial dimension and an SE covariance kernel in the temporal dimension. The `ProductKernel` class multiplies kernels operating on the *same* dimensions, and so is not directly applicable to this case. The `MaskedKernel` class provides a way of specifying which dimension(s) a `Kernel` applies to, such that covariance kernels masked to only operate on specific dimensions can be constructed before forming a `ProductKernel`.

#### E.4.2 The `gptools.kernel.noise` submodule

The `DiagonalNoiseKernel` class represents uncorrelated, homoscedastic noise. It has a single hyperparameter,  $\sigma_n$ . Where  $\mathbf{x}_i = \mathbf{x}_j$  it returns  $\sigma_n^2$ , zero otherwise. Note that this scheme causes a problem when there are multiple data points with the same  $\mathbf{x}$ : this will result in off-diagonal elements being inserted into the covariance matrix. Therefore, the `GaussianProcess` class detects if the noise covariance kernel is an instance of `DiagonalNoiseKernel` and, if so, handles it specially instead of relying on the `compute_Kij()` method. Note that this approach will break when trying to combine an instance of `DiagonalNoiseKernel` with another covariance kernel through either addition or multiplication.

The `ZeroKernel` class is a subclass of `DiagonalNoiseKernel` with  $\sigma_n = 0$  held fixed. This is used to represent when there is no additional homoscedastic noise to be inferred. It can also be used to set the covariance kernel itself to  $k(\mathbf{x}_i, \mathbf{x}_j) = 0$  such that `gptools` can be used to perform Bayesian nonlinear regression.

#### E.4.3 The `SquaredExponentialKernel` class

The `SquaredExponentialKernel` class implements the multidimensional SE covariance kernel described in [section B.1](#), including the full expression for the derivatives of the covariance kernel given in [equation \(B.14\)](#) and derivatives with respect to the hyperparameters. This is carried out in practice by first finding

$$k = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2}\right). \quad (\text{E.23})$$

Then, the array of Hermite polynomial factors

$$\left(\frac{-1}{\sqrt{2\ell_d}}\right)^{n_d} \mathbf{H}_{n_d}\left(\frac{\tau_d}{\sqrt{2\ell_d}}\right) \quad (\text{E.24})$$

is formed. If the derivative with respect to one of the covariance length scales  $\ell_e$  is requested, the relevant entry in the Hermite polynomial factors is multiplied by

$$\frac{\tau_e^2}{\ell_e^3} - \mathbb{I}(n_e > 0) \frac{n_e}{\ell_e} - \mathbb{I}(n_e > 0) \frac{\sqrt{2n_e}\tau_e}{\ell_e^2} \frac{\mathbf{H}_{n_e-1}(\tau_e/(\sqrt{2\ell_e}))}{\mathbf{H}_{n_e}(\tau_e/(\sqrt{2\ell_e}))}. \quad (\text{E.25})$$

Finally, the product over the Hermite factors is multiplied into  $k$  along with  $(-1)^{n_j}$  where  $n_j$  is the total order of differentiation with respect to any of the dimensions of  $\mathbf{x}_j$ . Using the broadcasting capabilities of SciPy, this is carried out without needing to write an explicit loop.

#### E.4.4 The `RationalQuadraticKernel` class

The `RationalQuadraticKernel` class implements the multidimensional RQ covariance kernel described in [section B.2](#), with  $\alpha$  treated as a free hyperparameter. This is accomplished by implementing `ChainRuleKernel`. The `_compute_k()` method returns  $y^{-\alpha}$ , where

$$y = 1 + \frac{1}{2\alpha} \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \quad (\text{E.26})$$

is computed with the `_compute_y()` method. The `_compute_dk_dy()` method computes

$$\frac{d^n k(y)}{dy^n} = (1 - \alpha - n)_n y^{-\alpha-n}. \quad (\text{E.27})$$

where the correct product form of the Pochhammer symbol as described in [section E.7.1](#) is used. Finally, `_compute_dy_dtau()` computes  $\partial^{|\mathbf{B}|} y(\boldsymbol{\tau}) / \prod_{j \in B} \partial \tau_{|j|}$  according to [equation \(B.31\)](#).

#### E.4.5 The `gptools.kernel.matern` submodule

The `gptools.kernel.matern` submodule provides several implementations of the Matérn covariance kernel described in [section B.3](#).

#### E.4.5.1 The full Matérn covariance kernel: the `MaternKernel` class

The `MaternKernel` class implements `ChainRuleKernel` and provides full support for arbitrary order  $\nu$  and arbitrary orders of differentiation as described in [section B.3](#). The `_compute_k()` method computes

$$k = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( 2\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \right)^{\nu/2} K_\nu \left( \left( 2\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2} \right)^{1/2} \right), \quad (\text{E.28})$$

where the `_compute_y()` method is used to compute

$$y = 2\nu \sum_{d=1}^D \frac{\tau_d^2}{\ell_d^2}. \quad (\text{E.29})$$

The `_compute_dk_dy()` method computes

$$\frac{d^n k}{dy^n} = \frac{2^{1-\nu}}{\Gamma(\nu)} \frac{d^n}{dy^n} (y^{\nu/2} K_\nu(y^{1/2})), \quad (\text{E.30})$$

where the derivative term is implemented in the helper function `yn2Kn2Der()` described in [section E.7.3](#). `_compute_dy_dtau()` computes  $\partial^{|B|} y(\boldsymbol{\tau}) / \prod_{j \in B} \partial \tau_{[j]}$  according to [equation \(B.48\)](#). Finally, in order to handle the issues near the origin discussed in [section B.3.1](#), `MaternKernel` overrides the base implementation of the `_compute_dk_dtau_on_partition()` method. When looping over the blocks in the partition, it keeps track of  $n_1$ , the number of times any variable is involved in a block of length one. If at any point there is a block with length longer than two or a mixed derivative in a block of length two, the function breaks out of the loop and returns zero. Finally, it computes  $2\nu - 2n + n_1$ , the exponent on  $\tau$  in [equation \(B.59\)](#). If this is equal to zero, the limit is undefined and the method returns `scipy.nan` (i.e., not-a-number). If this is greater than zero,  $d^{|n|} k / dy^{|n|}$  is set to zero.

#### E.4.5.2 Other Matérn covariance kernel implementations

Because `MaternKernel` was so complicated to derive and implement, several other implementations were developed to permit testing with the Matérn covariance kernel. The `MaternKernelArb` class implements `ArbitraryKernel` and uses numerical differentiation and limit finding tools to handle derivatives. The `MaternKernel1d` class is a basic implementation of the 1D Matérn covariance which treats  $\nu$  as a free hyperparameter but only supports up to first order derivatives. The `Matern52Kernel` class is a fast Cython [532] implementation of the multidimensional Matérn covariance kernel for fixed  $\nu = 5/2$  and only up to first derivatives. The `Matern52Kernel` implementation was contributed by R. McGibbon [533].

### E.4.6 The `gptools.kernel.gibbs` submodule

The `gptools.kernel.gibbs` submodule provides two possible ways of implementing the Gibbs covariance kernel described in [section B.4.1](#). For basic cases (univariate, only up to first derivatives, easy to compute first derivative of covariance length scale function), the `GibbsKernel1d` class may be instantiated directly. The covariance length scale function should take the locations  $x$  as its first argument, the derivative order  $n$  as its second argument and the hyperparameters as the remaining arguments. This formulation is used to implement a wide array of covariance length scale functions, such as the specific class `GibbsKernel1dTanh` which uses the `tanh` covariance length scale function. If multivariate data, higher order derivatives or particularly complicated covariance length scale functions are to be used, the `GibbsKernel1dArb` class implements `ArbitraryKernel` and uses `mpmath` to find derivatives numerically.

### E.4.7 The `gptools.kernel.warping` submodule

The `warping` submodule provides support for the introduction of non-stationarity by means of input warping. This module is a generalization of a contribution from R. McGibbon, who suggested the use of beta-CDF warping [396] and provided a basic implementation [534, 535]. The fully general implementation developed for this thesis is built around two classes: `WarpingFunction` and `WarpedKernel`.

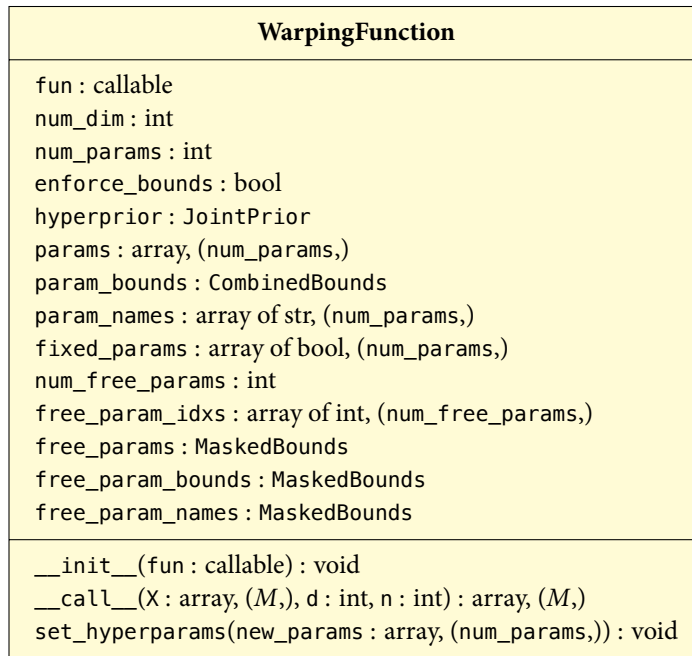
#### E.4.7.1 Representing input warping functions: the `WarpingFunction` class

The `WarpingFunction` class represents an input warping function. It keeps track of its hyperparameters through the same machinery as the `Kernel` class. A class diagram is given in [figure E.5](#). The attributes are similar to those of the `Kernel` class as described in [section E.4.1.1](#), with the addition of the warping function `fun` which should take as arguments an array  $\mathbf{x}_d$  of  $M$  positions corresponding to a given dimension  $d$ , the dimension index  $d$ , the derivative order  $n$  and all of the warping hyperparameters and which should return the warped inputs at the points in  $\mathbf{x}_d$ .

#### E.4.7.2 Representing a covariance kernel with warped inputs: the `WarpedKernel` class

The `WarpedKernel` class subclasses `Kernel` and has a `WarpingFunction` instance stored in its `w` attribute and an associated `Kernel` instance stored in its `k` attribute. Its `__call__()` method first passes the inputs through the warping function before passing them to the `__call__()` method of the covariance kernel to be warped, `k`.





**Figure E.5:** Class diagram for the WarpingFunction class.

At present, only first derivatives are supported. If these are requested, the products in [equation \(B.89\)](#) through [equation \(B.91\)](#) are computed. The `w_func()` method is a helper method for computing the input warping function and its derivatives. If the covariance kernel to be warped is itself a `WarpedKernel`, `w_func()` computes

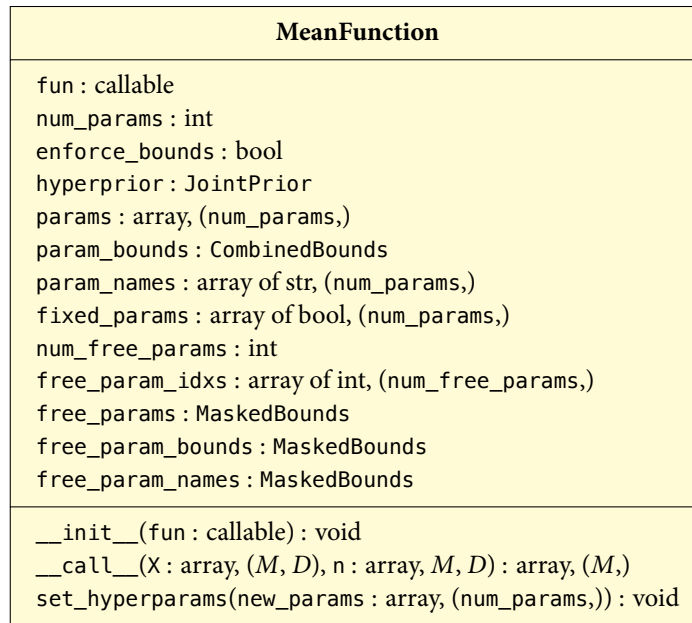
$$w = f(g(x)) \quad (\text{E.31})$$

$$\frac{dw}{dx} = \frac{df}{dg} \frac{dg}{dx}, \quad (\text{E.32})$$

where  $f$  is the input warping function of the covariance kernel to be warped  $k$  and  $g$  is the input warping function of the `WarpedKernel` instance itself.

## E.5 The MeanFunction class and the gptools.mean submodule

The `gptools.mean` submodule provides support for non-zero, parametric mean functions. Specifically, it provides the class `MeanFunction` which wraps a callable object in a way that the `GaussianProcess` class can use. The `MeanFunction` class



**Figure E.6:** Class diagram for the MeanFunction class.

keeps track of its hyperparameters through the same machinery as the Kernel class. A class diagram is given in [figure E.6](#). The attributes are similar to those of Kernel described in [section E.4.1.1](#), with the addition of the mean function `fun` which should take as arguments the  $M \times D$  array of locations  $\mathbf{X}$ , the  $D$  element vector of derivative orders with respect to each dimension and the parameters and return an  $M$  element vector with the mean function evaluated at each point in  $\mathbf{X}$ . In addition, the mean function should support the `hyper_deriv` keyword to compute derivatives with respect to its hyperparameters.

### E.5.1 The mtanh mean function

The `mtanh_profile()` function implements the `mtanh` mean function given in [equation \(2.81\)](#):

$$z = \frac{x_0 - x}{\delta} \quad (\text{E.33})$$

$$\text{mtanh}(\alpha, z) = \frac{(1 + \alpha z)e^z - e^{-z}}{e^z + e^{-z}} \quad (\text{E.34})$$

$$y = \frac{h+b}{2} + \frac{h-b}{2} \text{mtanh}(\alpha, z) = \frac{h+b}{2} + \frac{h-b}{2} \frac{(1 + \alpha z)e^z - e^{-z}}{e^z + e^{-z}} \quad (\text{E.35})$$

$$\frac{dy}{dx} = -\frac{h-b}{2\delta} \frac{1 + \alpha(1 + 2z + e^{2z})/4}{\cosh^2 z}. \quad (\text{E.36})$$

To enable the use of a gradient-based optimizer, first note that

$$\frac{\partial z}{\partial x_0} = \frac{1}{\delta} \quad (\text{E.37})$$

$$\frac{\partial z}{\partial \delta} = -\frac{x_0 - x}{\delta^2} = -\frac{z}{\delta} \quad (\text{E.38})$$

$$\frac{\partial y}{\partial z} = \frac{h-b}{2} \frac{1 + \alpha(1 + 2z + e^{2z})/4}{\cosh^2 z}. \quad (\text{E.39})$$

Then,

$$\frac{\partial y}{\partial x_0} = \frac{\partial y}{\partial z} \frac{\partial z}{\partial x_0} = \frac{h-b}{2\delta} \frac{1 + \alpha(1 + 2z + e^{2z})/4}{\cosh^2 z} \quad (\text{E.40})$$

$$\frac{\partial y}{\partial \delta} = \frac{\partial y}{\partial z} \frac{\partial z}{\partial \delta} = -z \frac{h-b}{2\delta} \frac{1 + \alpha(1 + 2z + e^{2z})/4}{\cosh^2 z} \quad (\text{E.41})$$

$$\frac{\partial y}{\partial \alpha} = z \frac{h-b}{2} \frac{e^z}{e^z + e^{-z}} \quad (\text{E.42})$$

$$\frac{\partial y}{\partial h} = \frac{1}{2} + \frac{1}{2} \frac{(1 + \alpha z)e^z - e^{-z}}{e^z + e^{-z}} \quad (\text{E.43})$$

$$\frac{\partial y}{\partial b} = \frac{1}{2} - \frac{1}{2} \frac{(1 + \alpha z)e^z - e^{-z}}{e^z + e^{-z}}. \quad (\text{E.44})$$

<i>JointPrior</i>
<i>i</i> : float <i>bounds</i> : array, ( <i>n</i> ,)
__call__( <i>theta</i> : array, ( <i>n</i> ,)) : float random_draw() : array, ( <i>n</i> ,) sample_u( <i>q</i> : array, ( <i>n</i> ,)) : array, ( <i>n</i> ,) elementwise_cdf( <i>p</i> : array, ( <i>n</i> ,)) : array, ( <i>n</i> ,) __mul__( <i>other</i> : JointPrior) : ProductJointPrior

**Figure E.7:** Class diagram for the JointPrior abstract class. The number of variables represented is *n*.

Next, note that

$$\frac{\partial}{\partial z} \frac{dy}{dx} = -\frac{h-b}{2\delta} \frac{\alpha - (\alpha z + 2) \tanh z}{\cosh^2 z} \quad (\text{E.45})$$

$$\frac{\partial}{\partial x_0} \frac{dy}{dx} = \frac{\partial}{\partial z} \frac{dy}{dx} \frac{\partial z}{\partial x_0} = -\frac{h-b}{2\delta^2} \frac{\alpha - (\alpha z + 2) \tanh z}{\cosh^2 z} \quad (\text{E.46})$$

$$\frac{\partial}{\partial \delta} \frac{dy}{dx} = \frac{h-b}{2\delta^2} \frac{1 + \alpha(1 + 2z + e^{2z})/4}{\cosh^2 z} + z \frac{h-b}{2\delta^2} \frac{\alpha - (\alpha z + 2) \tanh z}{\cosh^2 z} \quad (\text{E.47})$$

$$\frac{\partial}{\partial \alpha} \frac{dy}{dx} = -\frac{h-b}{8\delta} \frac{1 + 2z + e^{2z}}{\cosh^2 z} \quad (\text{E.48})$$

$$\frac{\partial}{\partial h} \frac{dy}{dx} = -\frac{1 + \alpha(1 + 2z + e^{2z})/4}{2\delta \cosh^2 z} \quad (\text{E.49})$$

$$\frac{\partial}{\partial b} \frac{dy}{dx} = \frac{1 + \alpha(1 + 2z + e^{2z})/4}{2\delta \cosh^2 z}. \quad (\text{E.50})$$

These expressions are all implemented in `mtanh_profile()` such that a gradient-based optimizer may be used when fitting data with the `mtanh` mean function.

## E.6 Representing prior distributions: the JointPrior abstract class and its subclasses

### E.6.1 The JointPrior abstract class

The `JointPrior` class represents a prior distribution for one or more hyperparameters. A class diagram is given in [figure E.7](#).

The attributes of the `JointPrior` class are:

**i** The interval that bounds should represent. The default is one, which means that bounds corresponds to the support of the prior distribution. Sometimes this is not useful for using optimizers and so forth, so `i` can be set to a value less than one to use a smaller central interval.

**bounds** The bounds of each hyperparameter. The interval this represents is modified by `i`. This will often be a getter method with a property decorator.

The methods of the `JointPrior` class are:

**`__call__()`** Evaluate the log-prior PDF for the values `theta`.

**`random_draw()`** Draw a sample from the prior distribution. Multiple samples can also be requested at once.

**`sample_u()`** Use the inverse CDF to generate samples using variables distributed on  $\mathcal{U}(0, 1)$ . This allows the use of techniques such as Latin hypercube sampling [355], quasi-Monte Carlo sampling [291] or sparse quadrature [338] which may require the problem be mapped to the unit hypercube for efficient implementation.

**`elementwise_cdf()`** Use the CDF to map samples from the distribution to samples from  $\mathcal{U}(0, 1)$ . This is the inverse operation of `sample_u()`.

**`__mul__()`** Form the joint distribution for independent variables:  $f_{\theta_1, \theta_2}(\theta_1, \theta_2) = f_{\theta_1}(\theta_1)f_{\theta_2}(\theta_2)$ .

The following sections describe the specific distributions implemented, notably including the derivatives of the PDFs needed to use a gradient-based optimizer.

### E.6.2 Combining `JointPrior` instances: `ProductJointPrior` class

The `ProductJointPrior` class represents the product of two independent distributions:

$$f_{\theta_1, \theta_2}(\theta_1, \theta_2) = f_{\theta_1}(\theta_1)f_{\theta_2}(\theta_2). \quad (\text{E.51})$$

The log-PDF is then simply given by the sum of the log-PDFs of the two distributions:

$$\ln f_{\theta_1, \theta_2}(\theta_1, \theta_2) = \ln f_{\theta_1}(\theta_1) + \ln f_{\theta_2}(\theta_2). \quad (\text{E.52})$$

By taking products of `ProductJointPrior` instances, prior distributions for many independent hyperparameters can be constructed. All operations are simply the

combination of the results from the two component `JointPrior` instances. When using a gradient-based optimizer, the derivative of the log-PDF with respect to a given hyperparameter  $\theta_i$  is found by determining which of the two component `JointPrior` instances  $\theta_i$  belongs to and evaluating only that `JointPrior` with the appropriate keyword set.

### E.6.3 Normal prior distributions: the `NormalJointPrior` class

The `NormalJointPrior` class represents the Normal distribution given in [equation \(1.36\)](#). The derivative of the PDF with respect to  $x$  is

$$\frac{df_X(x)}{dx} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot -\frac{x-\mu}{\sigma^2}, \quad (\text{E.53})$$

which means that the derivative of the log-PDF is

$$\frac{d}{dx} \ln f_X(x) = \frac{df_X(x)/dx}{f_X(x)} = -\frac{x-\mu}{\sigma^2}. \quad (\text{E.54})$$

### E.6.4 Log-normal prior distributions: the `LogNormalJointPrior` class

The `LogNormalJointPrior` class represents the log-normal distribution given in [equation \(1.39\)](#). The derivative of the PDF with respect to  $x$  is

$$\frac{df_X(x)}{dx} = -\frac{1}{x}f_X(x) + f_X(x) \cdot -\frac{\ln(x/M)}{\sigma^2 X}, \quad (\text{E.55})$$

so the derivative of the log-PDF is

$$\frac{d}{dx} \ln f_X(x) = \frac{df_X(x)/dx}{f_X(x)} = -\frac{1}{x} \left( 1 + \frac{\ln(x/M)}{\sigma^2} \right). \quad (\text{E.56})$$

### E.6.5 Gamma prior distributions: the `GammaJointPrior` and `GammaJointPriorAlt` classes

The `GammaJointPrior` class provides a direct implementation of the gamma distribution whose PDF is given in [equation \(1.44\)](#). The derivative of the PDF with respect to  $x$  is

$$\frac{df_X(x)}{dx} = \frac{\alpha-1}{x}f_X(x) + f_X(x) \cdot -\beta, \quad (\text{E.57})$$

so the derivative of the log-PDF is

$$\frac{d}{dx} \ln f_X(x) = \frac{df_X(x)/dx}{f_X(x)} = \frac{\alpha - 1}{x} - \beta. \quad (\text{E.58})$$

The alternate parameterization in terms of mode  $m$  and standard deviation  $\sigma$  given in [equation \(1.48\)](#) and [equation \(1.49\)](#) is used in the `GammaJointPriorAlt` class to provide a more convenient way of specifying gamma distributions with  $\alpha \geq 1$ .

## E.6.6 Prior distributions for ordered values

### E.6.6.1 Prior distribution for sorted, uniformly-distributed variables: the `SortedUniformJointPrior` class

It is often required that values have some ordered relationship: the core covariance length scale is longer than the edge covariance length scale or the knots of a spline are nondecreasing, for instance. The `SortedUniformJointPrior` class accomplishes this. This represents the information that some collection of variables  $\{t_i\}_{i=1}^N$  are sorted  $a \leq t_1 \leq t_2 \leq \dots \leq t_N \leq b$  and are uniformly distributed over some interval  $[a, b]$ . When an instance is created, it is given the number of variables  $N$  and the upper and lower bounds  $a$  and  $b$ , respectively, to use. To construct this distribution, note that what is needed to draw from this distribution is  $N$  draws from  $\mathcal{U}(a, b)$  which are then sorted. This distribution therefore represents the *order statistics* of  $N$  samples from  $\mathcal{U}(a, b)$ , for which it can be shown that

$$f_T(\mathbf{t}) = \begin{cases} N! / (b - a)^N, & a \leq t_1 \leq t_2 \leq \dots \leq t_N \leq b \\ 0, & \text{otherwise} \end{cases}. \quad (\text{E.59})$$

### E.6.6.2 Prior distribution for core and edge covariance length scales: the `CoreEdgeJointPrior` class

The `CoreEdgeJointPrior` class imposes the constraint  $a \leq x_2 \leq x_1 \leq b$  while giving  $x_1$  a uniform marginal density and  $x_2$  a uniform conditional density. Consider the

density

$$f_{X_1}(x_1) = \mathcal{U}(a, b) = \begin{cases} 1/(b-a), & a \leq x_1 \leq b \\ 0, & \text{otherwise} \end{cases} \quad (\text{E.60})$$

$$f_{X_2|X_1}(x_2|x_1) = \mathcal{U}(a, x_1) = \begin{cases} 1/(x_1-a), & a \leq x_2 \leq x_1 \\ 0, & \text{otherwise} \end{cases} \quad (\text{E.61})$$

$$f_{X_1, X_2}(x_1, x_2) = f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1) = \begin{cases} 1/((x_1-a)(b-a)), & a \leq x_2 \leq x_1 \leq b \\ 0, & \text{otherwise} \end{cases}. \quad (\text{E.62})$$

The smaller value  $x_2$  then has marginal PDF

$$f_{X_2}(x_2) = \int_{x_1}^b \frac{1}{(x_1-a)(b-a)} dx_1 \quad (\text{E.63})$$

$$= \frac{1}{b-a} \ln \frac{b-a}{x_2-a} \text{ for } a \leq x_2 \leq b. \quad (\text{E.64})$$

The mean of this distribution is

$$\mathbb{E}[x_2] = \int_a^b \frac{x_2}{b-a} \ln \frac{b-a}{x_2-a} dx_2 = \frac{3a+b}{4}. \quad (\text{E.65})$$

The conditional PDF of  $x_1$  is

$$f_{X_1|X_2}(x_1|x_2) = \frac{f_{X_1, X_2}(x_1, x_2)}{f_{X_2}(x_2)} = \frac{1}{(x_1-a) \ln((b-a)/(x_2-a))} \text{ for } a \leq x_2 \leq x_1 \leq b. \quad (\text{E.66})$$

Early on, this was used for the prior distribution for the core and edge length scales  $\ell_1$  and  $\ell_2$ , respectively, of the Gibbs covariance kernel with tanh covariance length scale function. The data seemed to prefer  $\ell_2$  fairly close to  $\ell_1$ , which tended to drive  $\ell_1$  to higher values. Therefore, while this distribution is still included to allow comparison with the old methods and to reproduce the results of older codes, its use is not recommended – the independent gamma distribution prior distributions used in the bulk of this thesis delivered much better performance.



## E.7 Miscellaneous classes and functions from the `gptools.utils` submodule

### E.7.1 Correcting SciPy's Pochhammer symbol: the `fixed_poch()` function

As noted in [equation \(B.27\)](#), the Pochhammer symbol is defined as

$$(a)_n = \prod_{k=0}^{n-1} (a + k). \quad (\text{E.67})$$

This is often converted to

$$(a)_n = \frac{\Gamma(a + n)}{\Gamma(a)}, \quad (\text{E.68})$$

which is in fact what SciPy's implementation `scipy.special.poch()` does. But, this is not valid when  $a$  is a negative integer. In order to correct this, `fixed_poch()` implements the product in [equation \(E.67\)](#) directly.

### E.7.2 Finding the derivatives of $K_\nu(y^{1/2})$ : the `Kn2Der()` function

`Kn2Der()` computes the  $n^{\text{th}}$  derivative of  $K_\nu(y^{1/2})$  using [equation \(B.44\)](#). The derivatives of  $K_\nu$  are found using the SciPy routine `scipy.special.kvp()` and the incomplete Bell polynomial is computed with the routine `incomplete_bell_poly()` which implements the recursive form of the incomplete Bell polynomial given in [equation \(B.38\)](#) and [equation \(B.39\)](#).

### E.7.3 Finding the derivatives of $y^{1/2}K_\nu(y^{1/2})$ : the `yn2Kn2Der()` function

`yn2Kn2Der()` computes the  $n^{\text{th}}$  derivative of  $y^{1/2}K_\nu(y^{1/2})$  using [equation \(B.47\)](#) (but without the normalizing constants). The outer sum is implemented using a loop over index  $k$  and the inner sums are contained in `Kn2Der()`. If the value at  $y = 0$  is required, the  $\nu \notin \mathbb{Z}$  case is handled as described in [section B.3.1](#). If  $\nu \in \mathbb{Z}$ , the interpolation given in [equation \(B.72\)](#) is used.

The function also provides the ability to use the series expansion given in [equation \(B.51\)](#) within a specified distance from  $y = 0$ . If  $\nu \in \mathbb{Z}$ , the interpolation described in [equation \(B.72\)](#) is again used.

### E.7.4 Generating all of the partitions of a set: the `generate_set_partitions()` function

In order to evaluate Faà di Bruno’s formula, it is necessary to generate all of the partitions of the set of variables to differentiate with respect to. `gptools` includes the `generate_set_partitions()` function which accomplishes this by using the `generate_set_partition_strings()` function to produce the restricted growth strings for all of the partitions of a set of size  $n$ . This is accomplished using algorithm H of [474]. The `generate_set_partitions()` function then converts the restricted growth strings to the form required by the `ChainRuleKernel` class.

## E.8 The `gptools.splines` submodule

In order to provide a useful set of basis functions for the construction of both length scale functions and input warping functions, `gptools` provides an implementation of B-splines, M-splines and I-splines [321, 322, 489, 490, 497] in `gptools.splines`, namely the function `spev()`.

### E.8.1 Mathematical preliminaries

These splines were introduced in section B.4, but the definitions are given again here for convenience. The  $i^{\text{th}}$  B-spline basis function of degree<sup>4</sup>  $d$  is a piecewise polynomial of degree  $d$  given by the recursion relation

$$B_{i,0}(x|\mathbf{t}) = \begin{cases} 1, & t_i \leq x < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (\text{E.69})$$

$$B_{i,d}(x|\mathbf{t}) = \frac{x - t_i}{t_{i+d} - t_i} B_{i,d-1}(x|\mathbf{t}) + \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(x|\mathbf{t}), \quad (\text{E.70})$$

where  $\mathbf{t}$  is the complete knot set (including any edge knots). Note that given  $n_t$  internal knots with  $d$  repeated boundary knots at each end there will be  $n_t + d - 1$  non-zero basis functions. The M-spline basis functions

$$M_{i,d}(x|\mathbf{t}) = \frac{d+1}{t_{i+d+1} - t_i} B_{i,d}(x|\mathbf{t}) \quad (\text{E.71})$$

are a different normalization of the B-spline basis functions: while the B-spline basis functions are normalized such that  $\sum_i B_{i,d}(x|\mathbf{t}) = 1$  for all  $x$ , the M-splines

4. Note that the literature on splines commonly defines B-splines in terms of “order”  $k = d + 1$ . This is very unfortunate notation as it masks the polynomial degree of the spline, and so the clearer but less conventional notation in terms of the polynomial degree  $d$  is used here.

are normalized such that  $\int_{-\infty}^{\infty} M_{i,d}(x|\mathbf{t}) dx = 1$  for all  $i$ . In `spev()`, however, the recursion relation

$$M_{i,0}(x|\mathbf{t}) = \begin{cases} 1/(t_{i+1} - t_i), & t_i \leq x < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (\text{E.72})$$

$$M_{i,d}(x|\mathbf{t}) = \frac{d+1}{d(t_{i+d+1} - t_i)} \left( (x - t_i)M_{i,d-1}(x|\mathbf{t}) + (t_{i+d+1} - x)M_{i+1,d-1}(x|\mathbf{t}) \right) \quad (\text{E.73})$$

is used directly. The I-spline basis functions are integrals of M-spline basis functions<sup>5</sup>:

$$I_{i,d}(x|\mathbf{t}) = \int_L^x M_{i,d-1}(u|\mathbf{t}) du, \quad (\text{E.74})$$

where  $L = \min \mathbf{t}$  is the start of the domain. In practice, however, the expression

$$I_{i,d}(x|\mathbf{t}) = \begin{cases} 0, & i > j \\ \sum_{m=i}^j (t_{m+d+1} - t_m) M_{m,d}(x|\mathbf{t}) / (d+1), & j - d + 1 \leq i \leq j \\ 1, & i < j - d + 1 \end{cases} \quad (\text{E.75})$$

is used. Here,  $j$  is the index in  $\mathbf{t}$  for which  $t_j \leq x < t_{j+1}$ . Because the M-spline basis functions are always positive, the I-spline basis functions will always be monotonic. Because the I-splines are given as integrals of  $M_{i,d-1}(x|\mathbf{t})$ , there will only be  $n_t + d - 2$  non-zero basis functions for an I-spline of degree  $d$ . The derivatives of the I-splines are obviously

$$\frac{dI_{i,d}(x|\mathbf{t})}{dx} = M_{i,d-1}(x|\mathbf{t}). \quad (\text{E.76})$$

The derivatives of the B-spline basis functions are [321]

$$\frac{dB_{i,d}(x|\mathbf{t})}{dx} = d \left( \frac{B_{i,d-1}(x|\mathbf{t})}{t_{i+d} - t_i} - \frac{B_{i+1,d-1}(x|\mathbf{t})}{t_{i+d+1} - t_{i+1}} \right) = M_{i,d-1}(x|\mathbf{t}) - M_{i+1,d-1}(x|\mathbf{t}), \quad (\text{E.77})$$

which means that the derivatives of the M-spline basis functions are

$$\frac{dM_{i,d}(x|\mathbf{t})}{dx} = \frac{d}{dx} \left( \frac{d+1}{t_{i+d+1} - t_i} B_{i,d}(x|\mathbf{t}) \right) = \frac{d+1}{t_{i+d+1} - t_i} (M_{i,d-1}(x|\mathbf{t}) - M_{i+1,d-1}(x|\mathbf{t})). \quad (\text{E.78})$$

5. The conventional notation [490] is to have the I-spline basis function of “order”  $k$  be  $I_{i,k}(x|\mathbf{t}) = \int_L^x M_{i,k}(u|\mathbf{t}) du$ . This is a piecewise polynomial of degree  $k$ , thus adding further confusion to the difference between spline order and polynomial degree. Here, we refer to  $I_{i,d}(x|\mathbf{t})$  by its polynomial degree  $d$ .

Note that the ultimate goal, once the basis functions have been computed, is to obtain

$$y(x) = \sum_{i=0}^{n_t+d-2} C_i S_{i,d}(x|\mathbf{t}), \quad (\text{E.79})$$

where  $S_{i,d}(x|\mathbf{t})$  is one of  $B_{i,d}(x|\mathbf{t})$ ,  $M_{i,d}(x|\mathbf{t})$  or  $I_{i,d}(x|\mathbf{t})$ .

### E.8.2 Software implementation

The `spev()` function takes as arguments an array containing the *internal* knots  $\mathbf{t}_{\text{int}}$ , an array containing the spline coefficients  $\mathbf{C}$ , the polynomial degree  $d$  and an array containing the grid to evaluate the spline on  $\mathbf{x} \in \mathbb{R}^{n_x}$ . An optional argument allows the derivative order  $n$  to be selected. In addition, the covariance matrix  $\mathbf{\Sigma} = \text{cov}[\mathbf{C}, \mathbf{C}]$  for the coefficients can be specified as a 2D array to produce an uncertainty estimate. If the coefficients are not correlated, then the variances  $\sigma^2 = \text{var}[\mathbf{C}]$  may be given as a 1D array instead. Boolean keywords are provided to select between B-splines (the default), M-splines and I-splines. The inputs must obey the following requirements:

- $\mathbf{t}_{\text{int}}$  has  $n_t$  elements which are in nondecreasing order.
- $\mathbf{C}$  has  $n_t + d - 1$  elements. Note that for I-splines there are only  $n_t + d - 2$  nontrivial basis functions. The first element of  $\mathbf{C}$ ,  $C_1$ , provides a constant offset in this case.
- $d$  is a non-negative integer. The cubic spline with  $d = 3$  is standard in practice, but `spev()` can handle any polynomial degree.
- $\mathbf{x}$  may have any number of elements which can be given in any order. In addition, the extrema of  $\mathbf{x}$  should satisfy  $\min \mathbf{x} \geq \min \mathbf{t}_{\text{int}}$ ,  $\max \mathbf{x} \leq \max \mathbf{t}_{\text{int}}$ . These conditions, however, are not enforced and values outside of the knots are silently set to zero.
- $\mathbf{\Sigma}$ , if provided, should either be a 2D array with shape  $(n_t + d - 1, n_t + d - 1)$  or a 1D array with  $n_t + d - 1$  elements.
- $n$ , if provided, should be a non-negative integer.

To set the behavior at the edges,  $d$  boundary knots are appended to each end of  $\mathbf{t}_{\text{int}}$  to form the complete knot set  $\mathbf{t}$ . In other words, for  $d = 3$ ,

$$\mathbf{t} = [t_1, t_1, t_1, t_1, t_2, \dots, t_{n_t-1}, t_{n_t}, t_{n_t}, t_{n_t}, t_{n_t}]. \quad (\text{E.80})$$

The derivatives are handled recursively using [equation \(E.76\)](#) through [equation \(E.78\)](#). If  $n > d$ , all zeros are returned. For the I-spline basis functions, [equation \(E.76\)](#) is implemented by dropping the constant offset  $C_1$  from  $\mathbf{C}$  and calling `spev()` with the same  $\mathbf{t}_{\text{int}}$ ,  $d_{\text{new}} = d - 1$ ,  $n_{\text{new}} = n - 1$ , the M-spline keyword set to `True` and the I-spline keyword set to `False`. For the B-spline basis functions, note from [equation \(E.77\)](#) that a given M-spline basis function  $M_{i,d-1}(x|\mathbf{t})$  appears in the derivative of [equation \(E.79\)](#) with coefficient  $-C_{i-1} + C_i$ . This is implemented by forming a new coefficient array

$$\mathbf{C}_{\text{new}} = -\mathbf{C}_{1:n_t+d-2} + \mathbf{C}_{2:n_t+d-1} \quad (\text{E.81})$$

and calling `spev()` with the same  $\mathbf{t}_{\text{int}}$ ,  $d_{\text{new}} = d - 1$ ,  $n_{\text{new}} = n - 1$  and the M-spline keyword set to `True`. This means that M-spline basis function  $M_{i,d-1}(x|\mathbf{t})$  will have coefficient  $-C_i + C_{i+1}$ . The indexing looks a little strange, but it accounts for both the fact that there is no  $C_0$  and the fact that there is one fewer non-trivial basis function for  $d_{\text{new}} = d - 1$ , which means that the indices  $i$  for  $d - 1$  are shifted by one. For the M-spline basis functions, note from [equation \(E.78\)](#) that a given basis function  $M_{i,d-1}(x|\mathbf{t})$  appears in the derivative of [equation \(E.79\)](#) with coefficient

$$-\frac{d+1}{t_{i+d} - t_{i-1}} C_{i-1} + \frac{d+1}{t_{i+d+1} - t_i} C_i. \quad (\text{E.82})$$

This is implemented by forming a new coefficient array

$$\mathbf{C}_{\text{new}} = (d+1) \left( -\frac{\mathbf{C}_{1:n_t+d-2}}{t_{d+2:n_t+2d-1} - t_{1:n_t+d-2}} + \frac{\mathbf{C}_{2:n_t+d-1}}{t_{d+3:n_t+2d} - t_{2:n_t+d-1}} \right), \quad (\text{E.83})$$

where the division takes place elementwise, and again calling `spev()` with the same  $\mathbf{t}_{\text{int}}$ ,  $d_{\text{new}} = d - 1$ ,  $n_{\text{new}} = n - 1$  and the M-spline keyword set to `True`. By implementing the derivatives in this manner, derivatives of any order can be computed recursively.

To evaluate the I-spline basis functions, the M-spline basis functions are computed before applying [equation \(E.75\)](#). To compute the B- or M-spline basis functions, a 3D array  $\mathbf{B}$  is created, where  $\mathbf{B}$  has shape  $(d+1, n_t+2d-1, n_x)$ . In other words, the first dimension of  $\mathbf{B}$  runs through the increasing polynomial degrees  $d$  (starting at  $d = 0$ ), the second dimension runs through the basis function indices  $i$  and the third dimension runs through all of the spatial points. The zero degree components are evaluated through direct application of [equation \(E.69\)](#) and [equation \(E.72\)](#) by looping from  $i = d$  (where the first set of repeated knots ends) to  $i = n_t + d - 2$  (where the second set of repeated knots begins). To obtain results which are continuous at the

right hand side of the domain, the zero degree basis function for  $i = n_t + d - 2$  is modified to be nonzero over  $t_i \leq x \leq t_{i+1}$  instead of  $t_i \leq x < t_{i+1}$ . Then the code loops over polynomial degree from  $d_1 = 1$  to  $d$ . For each  $d_1$ ,  $i$  loops from  $d - d_1$  to  $d + n_t - 2$ , adding terms according to either [equation \(E.70\)](#) or [equation \(E.73\)](#). The B- or M-spline basis functions are then given as a function of  $\mathbf{x}$  in  $\mathbf{B}_{d+1,1:n_t+d-1}$ . If an I-spline is requested, [equation \(E.75\)](#) is implemented by looping over  $i$  from 0 to  $n_t + d - 2$ . For each value of  $i$ , a loop from  $m = i$  to  $m = n_t + d - 2$  is performed, adding terms according to [equation \(E.75\)](#). The final result is then found by taking the dot product  $\mathbf{y} = \mathbf{C} \cdot \mathbf{B}_{d+1,1:n_t+d-1}$ . If a covariance matrix for  $\mathbf{C}$  is provided, the covariance matrix for  $\mathbf{y}$  is found from the matrix product  $\text{cov}[\mathbf{y}] = \mathbf{B}_{d+1,1:n_t+d-1} \boldsymbol{\Sigma} \mathbf{B}_{d+1,1:n_t+d-1}^T$ .

# F

## profiletools: an open-source Python package for handling profile data

### F.1 Package overview

`profiletools` is a Python package designed to make working with profile data easier. The source code is available in [308] and the user documentation is available in [309]. The objective when developing `profiletools` was to provide a set of helper functions to load the data from multiple diagnostics, put the data into the same coordinate system, perform time averaging and pass the combined data arrays to `gptools` to be fit with Gaussian process regression. The intent of providing a data structure for multivariate data is similar to the goal of the Python package `pandas` [536, 537], but `profiletools` necessarily uses a more flexible data model in order to be able to fully interface with `gptools`.

The core class of `profiletools` is the `Profile` class, described in [section F.2](#). Extensions to handle plasma data are provided by the `BivariatePlasmaProfile` class, described in [section F.3](#). If a different subclass of `BivariatePlasmaProfile` were provided for each diagnostic, merging data sets would yield an object whose class was not necessarily representative of its contents. Instead, each of the functions for loading data described in [section F.4](#) returns an instance of `BivariatePlasmaProfile` loaded with the requested diagnostic's data. A key component of `profiletools` is the flexibility in selecting how data are averaged, a process described in detail in [section F.5](#). An example of loading the temperature profile and finding the MAP estimate for a Gaussian process with the Gibbs covariance kernel with tanh covariance length scale function is given in [code listing F.1](#).

**Code Listing F.1:** Example showing to use of `profiletools` to fit the  $T_e$  profile. The data are fit as a function of  $r/a$  and are averaged over the window  $1\text{ s} \leq t \leq 1.1\text{ s}$ . The core and edge Thomson scattering data are loaded separately from the two grating polychromator diagnostics because the Thomson scattering data include uncertainty estimates and hence can use weighted averaging. The `create_gp()` method defaults to using a Gibbs covariance kernel with tanh covariance length scale function, so the covariance kernel does not need to be specified.

---

```

1 import profiletools
2 p = profiletools.TeTS(1101014006, t_min=1.0, t_max=1.1, abscissa='r/a')
3 p.time_average(weighted=True)
4 p_2 = profiletools.Te(
5     1101014006, t_min=1.0, t_max=1.1, abscissa='r/a', include=['GPC', 'GPC2']
6 )
7 p_2.time_average(weighted=False)
8 p.add_profile(p_2)
9 p.create_gp()
10 p.find_gp_MAP_estimate()

```

---

## F.2 The `profiletools` data model and the `Profile` class

A class diagram of the `profiletools.core.Profile` class is given in [figure F.1](#). The independent variables  $\mathbf{X}$  are stored in the  $n \times D$  array  $\mathbf{X}$  and the corresponding values of the dependent variable  $\mathbf{y}$  are stored in  $\mathbf{y}$ . This approach is similar to how a sparse matrix is stored:  $\mathbf{X}$  has a row for each data point which holds that point's coordinates. This allows data of arbitrary dimensionality which were not taken on a regular grid to be stored. The uncertainties in  $\mathbf{y}$  are stored in `err_y` and the uncertainties in  $\mathbf{X}$  are stored in `err_X`. The attribute `channels` keeps track of which points should logically be grouped together when performing averaging. When averaging along a given dimension, all points with the same values in the *other* columns of `channels` are grouped together.

A `Profile` instance can also keep track of transformed quantities. In order to keep the data structure for non-transformed quantities as clear and fast as possible, these are stored separately in the `transformed` array. Each channel of a diagnostic which takes transformed measurements is stored as an instance of the `Channel` class, described in [section F.2.3](#).

The remaining attributes of the `Profile` class are:

**`X_dim`** The number of dimensions in  $\mathbf{X}$ ,  $D$ .

**`X_units`** List of strings containing the units of each dimension of  $\mathbf{X}$ .



Profile
y : array, (n, X : array, (n, D) err_y : array, (n, err_X : array, (n, D) channels : array, (n, D) X_dim : int X_units : list of str y_units : str X_labels : list of str y_label : str weightable : bool transformed : array of Channel, (N, gp : GaussianProcess
__init__() : void add_data(X : array, (n, D), y : array, (n,)) : void add_profile(other : Profile) : void drop_axis(axis : int) : void keep_slices(axis : int, vals : array, (M,)) : void average_data() : void plot_data() : AxesSubplot remove_points(conditional : array of bool, (n,)) : tuple of arrays remove_outliers() : tuple of arrays remove_extreme_changes() : tuple of arrays create_gp() : void find_gp_MAP_estimate() : tuple plot_gp() : AxesSubplot or tuple smooth(X : array, (n <sub>*</sub> , D)) : array, (n <sub>*</sub> ,) or tuple or dict write_csv(filename : str) : void

**Figure F.1:** Class diagram for the Profile class. Notice that most of the methods return void: the methods mutate the Profile instance in place.

**y\_units** String containing the units of the dependent variable *y*.

**X\_labels** List of strings containing the names of each dimension of **X**.

**y\_label** String containing the name of the dependent variable *y*.

**weightable** If True, weighted averaging can be used.

**gp** The GaussianProcess instance created to fit the data.

### F.2.1 Automatically removing suspicious data points

In order to enable automated first-cut data analysis, the `Profile` class provides two very basic forms of outlier detection. The `remove_outliers()` method is a wrapper to apply the very basic form of Gaussian process-based outlier detection described in [section E.3.2.4](#). The `remove_extreme_changes()` method implements an even more basic form of outlier detection based on the assumption that the profile is smooth and has been sampled densely: if a point is more than a given number of standard deviations (the default is 10) off from its neighbor, it will be removed. The user can choose whether a point must be off relative to both of its neighbors or just one of its neighbors.

### F.2.2 Creating a Gaussian process

The `create_gp()` method creates a `GaussianProcess` instance containing the data from the `Profile` instance and stores it in the `gp` attribute. The user may specify a covariance kernel explicitly by passing a `Kernel` instance or with a string. By default the `SE` covariance kernel is used and reasonable bounds for each of the hyperparameters are chosen by looking at the data:  $\sigma_f$  is taken to lie within the interval  $[0, 5 \max y]$ , covariance length scales for each dimension  $d$  are taken to lie within the interval  $[0, 5(\max x_d - \min x_d)]$  where  $x_d$  is the  $d^{\text{th}}$  column of  $\mathbf{X}$ .

### F.2.3 Keeping track of transformed quantities: the `Channel` class

`profiletools` uses the `Channel` class to keep track of transformed quantities. A class diagram is given in [figure F.2](#). The `Channel` class stores  $n$  measurements of the transformed quantity  $y$  in the attribute `y`. The  $n \times n_Q$  transformation matrix  $\mathbf{T}$  is in the attribute `T` and the quadrature points  $\mathbf{X}$  are in the 3D array `X`. By storing the data this way, each measurement  $y$  in `y` has associated with it a  $1 \times n_Q$  transformation matrix  $\mathbf{T}$  and a  $n_Q \times D$  matrix of  $D$ -dimensional quadrature points  $\mathbf{X}$ . This is in contrast to the block diagonal approach used for combining transformation matrices in `gptools`, and was done to make it easier to average the transformation matrix when averaging the data along a given dimension.

## F.3 Extensions to handle plasma profile data: the `BivariatePlasmaProfile` class

The `BivariatePlasmaProfile` class represents data from a plasma physics profile measurement. The word “bivariate” in the name refers to data which are a function

<b>Channel</b>
y_label : str y_units : str X : array, (n, n <sub>Q</sub> , D) y : array, (n, err_X : array, (n, n <sub>Q</sub> , D) err_y : array, (n, T : array, (n, n <sub>Q</sub> )
__init__(X : array, (n, n <sub>Q</sub> , D), y : array, (n,)) : void keep_slices(axis : int, vals : array, (M,)) : bool average_data() : void remove_points(conditional : array of bool, (n)) : tuple

**Figure F.2:** Class diagram for the Channel class.

of space and time; the class still supports data of arbitrary dimensionality. By convention, time should always be the first column in **X** (referred to as “axis zero” because Python uses zero-based indexing). Then, an arbitrary number of other (typically one to three spatial) dimensions may follow. A class diagram is given in [figure F.3](#).

The new attributes of the BivariatePlasmaProfile class are:

**shot** The shot number. If the `add_profile()` method is used to combine two instances from different shots, a warning will be produced.

**efit\_tree** The Equilibrium instance to use for coordinate mapping.

**abscissa** The spatial coordinate system to use. `profiletools` understands all of the coordinate systems which `eqtools` supports.

**times** The specific time points used when only keeping specific times.

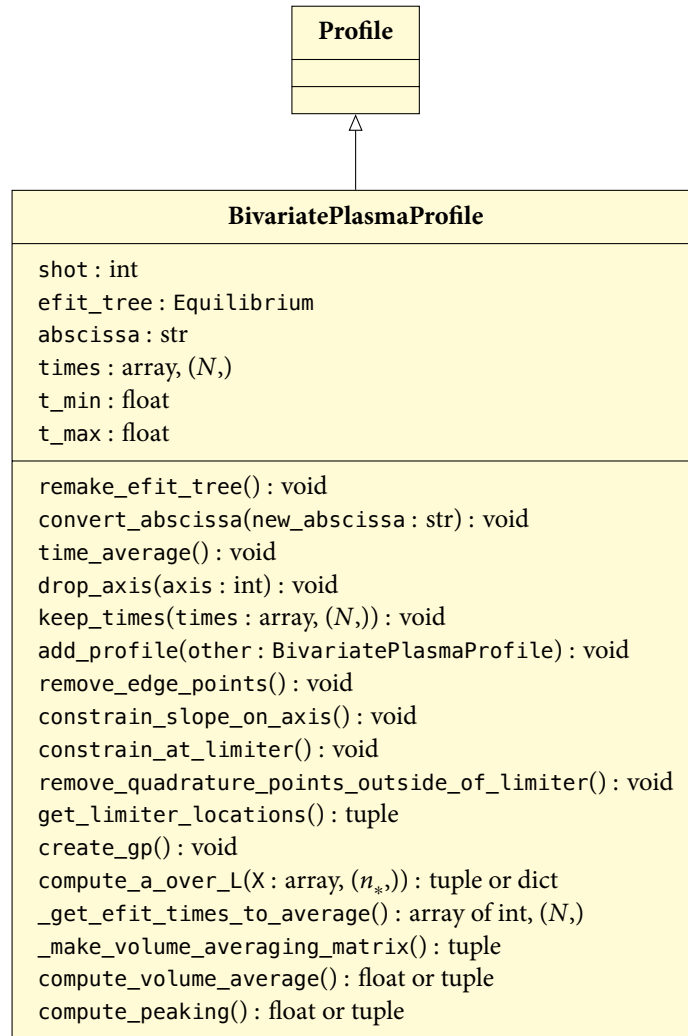
**t\_min** Lower bound of time window which was averaged over.

**t\_max** Upper bound of time window which was averaged over.

The following methods are modified relative to how they are implemented in the base Profile class:

**drop\_axis()** Stores `t_min` and `t_max` if the axis to be dropped corresponds to time.

**add\_profile()** Warns the user if the shot number of the two instances to be combined do not match. Ensures the abscissa of the two profiles matches.



**Figure F.3:** Class diagram for the `BivariatePlasmaProfile` class. Only the methods and attributes which are new or modified in the subclass are shown, see [figure F.1](#) for the full details of the superclass, `Profile`.

**create\_gp()** Defaults to using the Gibbs covariance kernel with tanh covariance length scale function. Will also call the `constrain_slope_on_axis()` and `constrain_at_limiter()` methods automatically unless the user tells the method to not do so.

When the data have been reduced through time-averaging, one of either `times` or the pair `t_min`, `t_max` is stored such that quantities from the magnetic equilibrium reconstruction can be computed over the appropriate time window when generating limiter constraints and computing  $a/L_y$ . The `_get_efit_times_to_average()` method gets the indices to use when performing a time average. If the `times` attribute is present, it simply finds the indices corresponding to the times included in the profile. If the `times` attribute is not present, it first checks to see if `t_min` is equal to `t_max`. If so, it simply finds the point in the timebase of `efit_tree` nearest to that single time point. Otherwise, it finds the indices in the timebase of `efit_tree` which correspond to times  $t_{\min} \leq t \leq t_{\max}$ .

### F.3.1 Converting coordinates: the `convert_abcissa()` method

The `convert_abcissa()` method converts the spatial part of the independent variables  $\mathbf{X}$  into a new coordinate system. If the profile has not yet been time-averaged, the conversion is done directly by using the appropriate conversion method of `efit_tree`. If the profile has already been time-averaged, the conversion is performed at each of the times returned by the `_get_efit_times_to_average()` method. The average is put into `X` and the standard deviation is put into `err_X`.

### F.3.2 Imposing constraints: the `constrain_slope_on_axis()` and `constrain_at_limiter()` methods

Both of the constraint methods assume that a `GaussianProcess` instance has already been created with the `create_gp()` method. The `constrain_slope_on_axis()` method imposes a zero-slope constraint at the magnetic axis. If the abscissa is `R_mid` this is accomplished by fetching the location of the magnetic axis from the associated `Equilibrium` instance. For normalized coordinates, the location is simply  $\rho = 0$ . The `constrain_slope_at_limiter()` method imposes an approximate zero slope and value constraint outside of the location of the limiter. It uses the `get_limiter_locations()` method to fetch the location of the GH limiter from the tree. Because this location is not stored in the tree for some older shots,  $R_{\text{GH}} = 0.91$  m is used when the data are not available. The method then adds to the associated `GaussianProcess` instance several linearly-spaced points at and just outside the location of the limiter for which the slope and value should be approximately zero.

### F.3.3 Computing gradient scale lengths: the `compute_a_over_L()` method

The `compute_a_over_L()` method uses the `GaussianProcess` instance `gp` to compute the gradient scale length according to the analysis described in [section 2.3.7](#). The method assumes that the data have already been time-averaged. The method first uses the `GaussianProcess` instance's `predict()` method to compute the mean values of  $y$ ,  $y' = dy/d\rho$  and possibly  $y'' = d^2y/d\rho^2$  along with their covariance matrix, where  $\rho$  is whatever coordinate is specified by `abscissa`. The variances in these quantities,  $\sigma_y^2$ ,  $\sigma_{y'}^2$  and  $\sigma_{y''}^2$ , are extracted from the diagonal of the covariance matrix and the covariances  $\sigma_{y,y'}$ ,  $\sigma_{y,y''}$  and  $\sigma_{y',y''}$  are extracted from the relevant off-diagonal elements. The gradient scale length is then computed from

$$\frac{a}{L_y} = -a \frac{dy/dr}{y}, \quad (\text{F.1})$$

where the minus sign is used in place of the absolute value under the assumption that profiles will typically be monotonically decreasing. But  $y$  may have been fit as a function of a coordinate  $\rho$  other than  $R_{\text{mid}}$ ,  $r$  or  $r/a$ . So, what is used instead is

$$= -\frac{1}{y} \frac{dy}{d\rho} \frac{d\rho}{d(r/a)} = -\frac{y'\rho'}{y}, \quad (\text{F.2})$$

where  $y' = dy/d\rho$  and  $\rho' = d\rho/d(r/a)$ . Applying the uncertainty propagation equation to this yields an expression for the variance

$$\text{var} \left[ \frac{a}{L_y} \right] = \sigma_y^2 \left( \frac{y'\rho'}{y^2} \right)^2 + \sigma_{y'}^2 \left( -\frac{\rho'}{y} \right)^2 + 2\sigma_{y,y'} \left( \frac{y'\rho'}{y^2} \right) \left( -\frac{\rho'}{y} \right) + \sigma_{\rho'}^2 \left( -\frac{y'}{y} \right)^2 \quad (\text{F.3})$$

$$= \sigma_y^2 \left( \frac{y'\rho'}{y^2} \right)^2 + \sigma_{y'}^2 \left( \frac{\rho'}{y} \right)^2 - 2\sigma_{y,y'} \frac{y'\rho'^2}{y^3} + \sigma_{\rho'}^2 \left( \frac{y'}{y} \right)^2. \quad (\text{F.4})$$

While in general one might expect there to be correlations between  $\rho'$  and the other quantities, these are not computed at present. When normalized second derivatives are required, they are given by [equation \(2.73\)](#) and [equation \(2.74\)](#):

$$\frac{a}{L_{\nabla y}} = \frac{ad^2y/dr^2}{dy/dr} = \frac{d^2y/d(r/a)^2}{dy/d(r/a)} \quad (\text{F.5})$$

$$= \left( \frac{d^2y}{d\rho^2} \left( \frac{d\rho}{d(r/a)} \right)^2 + \frac{dy}{d\rho} \frac{d^2\rho}{d(r/a)^2} \right) \frac{1}{dy/d\rho \cdot d\rho/d(r/a)} \quad (\text{F.6})$$

$$= \frac{y''\rho'}{y'} + \frac{\rho''}{\rho'} \quad (\text{F.7})$$

$$\begin{aligned} \text{var} \left[ \frac{a}{L\sqrt{y}} \right] &= \sigma_{y'}^2 \left( -\frac{y''\rho'}{y'^2} \right)^2 + \sigma_{y''}^2 \left( \frac{\rho'}{y'} \right)^2 + 2\sigma_{y',y''} \left( -\frac{y''\rho'}{y'^2} \right) \left( \frac{\rho'}{y'} \right) \\ &\quad + \sigma_{\rho'}^2 \left( \frac{y''}{y'} - \frac{\rho''}{\rho'^2} \right)^2 + \sigma_{\rho''}^2 \left( \frac{1}{\rho'} \right)^2 + 2\sigma_{\rho',\rho''} \left( \frac{y''}{y'} - \frac{\rho''}{\rho'^2} \right) \left( \frac{1}{\rho'} \right) \end{aligned} \quad (\text{F.8})$$

$$\begin{aligned} &= \sigma_{y'}^2 \left( \frac{y''\rho'}{y'^2} \right)^2 + \sigma_{y''}^2 \left( \frac{\rho'}{y'} \right)^2 - 2\sigma_{y',y''} \frac{y''\rho'^2}{y'^3} \\ &\quad + \sigma_{\rho'}^2 \left( \frac{y''}{y'} - \frac{\rho''}{\rho'^2} \right)^2 + \sigma_{\rho''}^2 \frac{1}{\rho'^2} + 2\sigma_{\rho',\rho''} \frac{1}{\rho'} \left( \frac{y''}{y'} - \frac{\rho''}{\rho'^2} \right) \end{aligned} \quad (\text{F.9})$$

$$\frac{a^2 d^2 y / dr^2}{y} = \frac{d^2 y / d(r/a)^2}{y} = \frac{1}{y} \frac{d^2 y}{d\rho^2} \left( \frac{d\rho}{d(r/a)} \right)^2 + \frac{1}{y} \frac{dy}{d\rho} \frac{d^2 \rho}{d(r/a)^2} \quad (\text{F.10})$$

$$= \frac{y''\rho'^2 + y'\rho''}{y} \quad (\text{F.11})$$

$$\begin{aligned} \text{var} \left[ \frac{d^2 y / d(r/a)^2}{y} \right] &= \sigma_y^2 \left( -\frac{y''\rho'^2}{y^2} - \frac{y'\rho''}{y^2} \right)^2 + \sigma_{y''}^2 \left( \frac{\rho'^2}{y} \right)^2 \\ &\quad + 2\sigma_{y,y''} \left( -\frac{y''\rho'^2}{y^2} - \frac{y'\rho''}{y^2} \right) \left( \frac{\rho'^2}{y} \right) \\ &\quad + \sigma_{\rho'}^2 \left( \frac{2y''\rho'}{y} \right)^2 + \sigma_{\rho''}^2 \left( \frac{y'}{y} \right)^2 \\ &\quad + 2\sigma_{\rho',\rho''} \left( \frac{2y''\rho'}{y} \right) \left( \frac{y'}{y} \right) \end{aligned} \quad (\text{F.12})$$

$$\begin{aligned} &= \sigma_y^2 \frac{1}{y^4} (y''\rho'^2 + y'\rho'')^2 + \sigma_{y''}^2 \left( \frac{\rho'^2}{y} \right)^2 \\ &\quad - 2\sigma_{y,y''} \frac{\rho'^2}{y^3} (y''\rho'^2 + y'\rho'') \\ &\quad + 4\sigma_{\rho'}^2 \left( \frac{y''\rho'}{y} \right)^2 + \sigma_{\rho''}^2 \left( \frac{y'}{y} \right)^2 + 4\sigma_{\rho',\rho''} \frac{y'y''\rho'}{y^2}, \end{aligned} \quad (\text{F.13})$$

where  $y' = dy/d\rho$ ,  $y'' = d^2y/d\rho^2$ ,  $\rho' = d\rho/d(r/a)$  and  $\rho'' = d^2\rho/d(r/a)^2$ . In general, correlations between  $\rho'$ ,  $\rho''$ ,  $y'$  and  $y''$  would be expected, but these are not computed at present.

Note that for the special case that  $\rho = r/a$  the coordinate transformation terms simplify to  $\rho' = 1$ ,  $\rho'' = 0$  with variance and covariance zero. For the special case  $\rho = R_{\text{mid}} = a(r/a) + R_0$ , the terms simplify to  $\rho' = a$ ,  $\sigma_{\rho'}^2 = \text{var}[a]$ ,  $\rho'' = 0$ ,

$\sigma_{\rho''}^2 = 0$ ,  $\sigma_{\rho', \rho''} = 0$ . In order to account for the uncertainty corresponding to the time-variation of the quantities  $\rho'$  and  $\rho''$ , the `_get_efit_times_to_average()` method is used to determine which `EFIT` time points are included in the time average. For the  $\rho = R_{\text{mid}}$  case the sample mean and variance of  $a$  are used for  $\rho'$  and  $\sigma_{\rho'}^2$ . For coordinates other than  $r/a$  and  $R_{\text{mid}}$ , the method loops over the times to include. For each time, it forms a grid of  $r/a$  from zero to two with three times as many points as `EFIT` used for its radial grid. It then converts this grid to  $\rho$  using the `rho2rho()` method of the associated `Equilibrium` instance. A cubic interpolating spline is then fit to  $\rho = f(r/a)$ . This spline can then be evaluated at the  $r/a$  points  $a/L_y$  is to be computed at in order to obtain the derivatives

$$\rho' = \frac{d\rho}{d(r/a)}, \quad \rho'' = \frac{d^2\rho}{d(r/a)^2}. \quad (\text{F.14})$$

The means, variances and covariances of  $\rho'$  and  $\rho''$  are then computed from the sample statistics.

Use of the expressions above to compute the variances can be overridden using the `full_MC` keyword, which is also passed to the `predict()` method of the `GaussianProcess` instance. When this is in effect, the `predict()` method returns a number of Monte Carlo samples which have potentially been filtered through a rejection function. Equation (F.2) (and possibly equation (F.7) and equation (F.11)) is then evaluated for each sample and the sample mean and standard deviation are computed. In this case the uncertainties from the time-variation of  $\rho'$  and  $\rho''$  are not taken into account, but these are usually found to be quite small in practice.

### F.3.4 Computing volume averages: the `compute_volume_average()` method

The `compute_volume_average()` method computes the volume average  $\langle y \rangle_{\text{vol}}$  of the profile and its uncertainty using the steps described in section 2.3.9.3. The quadrature weights are computed using the `_make_volume_averaging_matrix()` method. This method can operate in one of two ways. If no quadrature points are supplied, an  $n_Q$  point grid which is uniformly-spaced in  $V_n$  is generated and converted to the appropriate coordinate system defined by the `abscissa` attribute. The weights are then simply computed using the trapezoid rule. Otherwise, explicit quadrature points may be provided. These are converted to  $V_n$  and the weights are again computed from the trapezoid rule, but this time accounting for the nonuniform spacing. These quadrature points and quadrature weights can then be passed to the `predict()` method of the associated `GaussianProcess` instance.



### F.3.5 Computing peaking factors: the `compute_peaking_factor()` method

The `compute_peaking_factor()` method computes the peaking factor defined in [equation \(2.68\)](#). The method first uses the `_make_volume_averaging_matrix()` method to produce the quadrature points and weights needed to compute the volume average  $\langle y \rangle_{\text{vol}}$ , then combines this with a single point at  $\psi_n = 0.2$  (converted to the appropriate coordinate system) to obtain the covariance necessary to evaluate the uncertainty in [equation \(2.69\)](#).

## F.4 Functions for retrieving Alcator C-Mod profiles

The `profiletools.CMod` submodule contains a variety of functions to load profile data from the Alcator C-Mod MDSplus tree. These act as functions which return `BivariatePlasmaProfile` instances so that all of the profiles for a given quantity end up having the same class so that the object classes still make sense after a method such as `add_profile()` has been used. Each of the functions include basic logic to reject points which have been flagged as bad.

### F.4.1 The electron density profile, $n_e$

#### F.4.1.1 The core Thomson scattering system: `neCTS()`

The `neCTS()` function loads the data from the core Thomson scattering system. A point is removed if any of the following conditions are met, obtained from the rules the Thomson scattering system uses to flag bad points:

- The uncertainty is `scipy.nan` (not-a-number) or  $\pm\text{scipy.inf}$  (infinite).
- The uncertainty is exactly 0.0,  $1.0 \times 10^{20} \text{ m}^{-3}$  or  $2.0 \times 10^{20} \text{ m}^{-3}$ .
- The value is exactly 0.0 and the `remove_zeros` keyword is `True`.

#### F.4.1.2 The edge Thomson scattering system: `neETS()`

The `neETS()` function loads the data from the edge Thomson scattering system. A point is removed if any of the following conditions are met:

- The value in the “pointmask” node in the edge Thomson scattering tree is 0.
- The uncertainty is `scipy.nan` (not-a-number) or  $\pm\text{scipy.inf}$  (infinite).

- The uncertainty is exactly 0.0,  $1.0 \times 10^{20} \text{ m}^{-3}$  or  $2.0 \times 10^{20} \text{ m}^{-3}$ .
- The value is exactly 0.0 and the `remove_zeros` keyword is `True`.

#### F.4.1.3 The two-color interferometer: `neTCI()` and `neTCI_old()`

The `neTCI()` function loads the data from the two-color interferometer and was implemented as part of an undergraduate research project. This function uses a fixed set of quadrature points shared between all of the TCI chords to reduce the computational overhead of evaluating the Gaussian process. The quadrature weights are computed using the TRIPPY code [362]. Unfortunately, the function is not robust to changes in the TCI chord locations and so is not reliable for older data.

The `neTCI_old()` function is the initial implementation of TCI data in `profi-letools`. Each chord uses a set of quadrature points uniformly spaced along the chord, and the quadrature weights are computed using the trapezoid rule. While this technique is more computationally expensive than that used for `neTCI()`, it is robust to changes in the TCI chord locations.

In either case, points are only retained for which the line-integrated density is greater than  $10^{17} \text{ m}^{-2}$ .

#### F.4.1.4 The scrape-off layer reflectometer: `neReflect()`

The `neReflect()` function loads the data from the scrape-off layer reflectometer attached to the lower hybrid launcher. Points for which the value is exactly 0.0 are removed. Care needs to be taken when using data from this diagnostic as most of the scatter in a given channel over a given time window tends to be in the independent variable rather than the dependent variable.

### F.4.2 The electron temperature profile, $T_e$

#### F.4.2.1 The core Thomson scattering system: `TeCTS()`

The `TeCTS()` function loads the data from the core Thomson scattering system. A point is removed if any of the following conditions are met:

- The uncertainty is `scipy.nan` (not-a-number) or `±scipy.inf` (infinite).
- The uncertainty is exactly 0.0 or 1.0 keV.
- The value is exactly 0.0 and the `remove_zeros` keyword is `True`.

**F.4.2.2 The edge Thomson scattering system: TeETS()**

The TeETS() function loads the data from the edge Thomson scattering system. A point is removed if any of the following conditions are met:

- The value in the “pointmask” node in the edge Thomson scattering tree is 0.
- The uncertainty is `scipy.nan` (not-a-number) or `±scipy.inf` (infinite).
- The uncertainty is exactly 0.0, 0.5 keV or 1.0 keV.
- The value is exactly 0.0 and the `remove_zeros` keyword is `True`.
- The value is exactly 0.0 and the uncertainty is exactly 0.029 999 999 329 447 746 keV. This flag appears to have been in use for older data.

**F.4.2.3 The electron cyclotron emission diagnostics: TeGPC(), TeGPC2(), TeFRCECE() and TeMic()**

The TeGPC() function loads the data from the first grating polychromator electron cyclotron emission diagnostic, the TeGPC2() function loads the data from the second grating polychromator electron cyclotron emission diagnostic, the TeFRCECE() function loads the data from the Fusion Research Center electron cyclotron emission diagnostic and the TeMic() function loads the data from the Michelson interferometer diagnostic. In all cases, points for which the value is less than 0.15 keV are rejected.

**F.5 Averaging schemes and their interpretations**

Often profile data will be measured at multiple times. The user must then decide how to utilize this information in a statistically reasonable way. The most basic approach would be to fit a profile to each time slice, in which case the `keep_times()` method may be used to select the specific time point to retain. Assuming the uncertainties on the input data are correct, the 1D fits and their uncertainty envelopes then represent an estimate of the temporal evolution of the profile, albeit without having accounted for the correlation between nearby time points. If a summary of the behavior over a steady-state period is required, the user must then consider the same questions about averaging the individual fits together as are asked about the data themselves below.

Another option is to use `drop_axis(0)` to drop the temporal information (recall that axis number zero is time in a `BivariatePlasmaProfile` which has not yet

been time-averaged) and include *all* of the points in the fit with their individual uncertainty estimates. This is equivalent to the assumption that each time slice represents an independent sample from the same underlying distribution, such that each point in a given channel represents a sample  $y_i \sim \mathcal{N}(\mu, \sigma_i^2)$ . The Gaussian process itself then combines all of the points to arrive at a well-constrained estimate of the single, underlying mean curve  $\mu(\mathbf{x})$  which the data represent noisy samples of. As more points are added, the uncertainty in the Gaussian process will tend towards zero. This would not be a valid approach if the profile is varying from time-to-time in a manner which needs to be represented in the uncertainty envelope of the mean curve, such as undergoing sawtooth oscillations or if the mean of the profile itself is exhibiting a stochastic variation which is not included in the diagnostic uncertainty estimates. Furthermore, this approach is very computationally expensive: computing the Cholesky decomposition of the covariance matrix has asymptotic complexity  $\mathcal{O}(n^3)$  where  $n$  is the number of observations, so incorporating data from many time points will make the Gaussian process much more computationally expensive to work with.

The typical use case for `profiletools` therefore requires averaging the data over a given time window: all of the samples in a given channel are combined into a single point with a single uncertainty estimate. `profiletools` provides a number of ways to perform this averaging; the user is left to make a reasonable choice in line with the statistical interpretations outlined in this section. In order to provide a consistent interface for averaging both local and transformed quantities, `profiletools` carries out all of its averaging with the `profiletools.core.average_points()` function. The `average_data()` method of the `Profile` class loops over all of the unique channels and applies `average_points()` to them one-by-one. When calling the `average_points()` method, the user provides the independent variables  $\mathbf{X}$  as either an  $n \times D$  or  $n \times n_Q \times D$  array, the dependent variables  $\mathbf{y}$  as an  $n$  element array, the uncertainties in the independent variables  $\boldsymbol{\sigma}_x$  in an array the same shape as  $\mathbf{X}$ , the uncertainties in the dependent variables  $\boldsymbol{\sigma}_y$  in an array with the same shape as  $\mathbf{y}$  and possibly the transformation matrix  $\mathbf{T}$  as an  $n \times n_Q$  array. The result is a single estimate of  $\bar{y} \pm \sigma_y$ ,  $\bar{\mathbf{x}} \pm \boldsymbol{\sigma}_x$  and possibly  $\bar{\mathbf{T}}$ , where  $\bar{y}, \sigma_y \in \mathbb{R}$ ;  $\bar{\mathbf{x}}, \boldsymbol{\sigma}_x \in \mathbb{R}^D$  or  $\mathbb{R}^{n_Q \times D}$  and  $\bar{\mathbf{T}} \in \mathbb{R}^{n_Q}$ . The user can then make the following choices about how the averaging proceeds, which are described in detail in the following sections:

- Whether or not to use robust estimators (median, interquartile range) instead of standard estimators (mean, standard deviation).
- Whether or not to use weighted estimators.
- What method to use for computing the uncertainty in  $\mathbf{y}$ .

- What method to use for computing the uncertainty in  $\mathbf{X}$ .

In all cases the resultant summary is in terms of the mean and standard deviation of a normal distribution which best fits (in the sense to be described below) the observed distribution of the data. The assumption that the noise follows the normal distribution can be checked by making a probability (“Q-Q”) plot of the quantiles of the data against the quantiles of the normal distribution [401].

### F.5.1 Robust versus standard estimators

As described in section 1.4.3.2, the mean and standard deviation are both non-robust estimators: a single bad point may drag the estimate arbitrarily far from the rest of the data. In order to account for the possibility of outliers in the data, the user can opt to use the sample median as an estimate of the center of the data and the sample interquartile range as an estimate of the width (i.e., uncertainty) of the distribution. If the interquartile range is used, the uncertainty is expressed as the standard deviation of the normal distribution having the observed interquartile range:  $\sigma \approx \text{IQR}/1.349$  (see equation (1.37)). This has the implication that the underlying distribution of the data is in fact normal, but that there are one or more outliers which would have corrupted the calculation of the sample mean and standard deviation.

### F.5.2 Weighted versus non-weighted estimators

When diagnostic uncertainty estimates are provided, this information can be used to obtain a better estimate of the sample statistics by letting points for which there is a smaller uncertainty estimate have more influence on the estimates of the mean and uncertainty.

#### F.5.2.1 One view of weighted and unweighted estimators: the empirical distribution and its properties

In order to provide an intuitive understanding of the types of averaging supported by `profiletools`, this section obtains expressions for the unweighted and weighted estimators using arguments based on the empirical distribution of the data.

**F.5.2.1.1 Unweighted estimators obtained from the empirical distribution** A sample  $y_i$  of  $y$  in a given channel can be thought of as being obtained from the probability distribution described by the empirical CDF

$$\hat{F}_Y(y) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(y_i \leq y), \quad (\text{F.15})$$

where  $\mathbb{I}(y_i \leq y)$  is an indicator function. This CDF jumps up by  $1/n$  at each measurement. The sample median  $M$  is simply the point at which  $\hat{F}_Y(y) = 1/2$ . If there is an even number of points, the median is taken to be the midpoint between the points which bracket  $\hat{F}_Y(y) = 1/2$ . The first and third quartiles of the same are obtained in the same way to allow an estimate of the sample interquartile range. The derivative of the empirical CDF yields the empirical PDF, which has the form

$$\hat{f}_Y(y) = \frac{1}{n} \sum_{i=1}^n \delta(y_i - y), \quad (\text{F.16})$$

where

$$\delta(x) = \begin{cases} \infty, & x = 0 \\ 0, & \text{otherwise} \end{cases} \quad (\text{F.17})$$

is the Dirac delta function. The mean of this distribution is

$$\mathbb{E}[y] = \int_{-\infty}^{\infty} y \hat{f}_Y(y) dy \quad (\text{F.18})$$

$$= \int_{-\infty}^{\infty} y \frac{1}{n} \sum_{i=1}^n \delta(y_i - y) dy \quad (\text{F.19})$$

$$= \frac{1}{n} \sum_{i=1}^n \int_{-\infty}^{\infty} y \delta(y_i - y) dy \quad (\text{F.20})$$

$$= \frac{1}{n} \sum_{i=1}^n y_i = \bar{y}, \quad (\text{F.21})$$

which is the sample mean as given in [equation \(1.28\)](#). The variance of this distribution is

$$\text{var}[y] = \int_{-\infty}^{\infty} (y - \mathbb{E}[y])^2 \hat{f}_Y(y) dy \quad (\text{F.22})$$

$$= \int_{-\infty}^{\infty} (y - \mathbb{E}[y])^2 \frac{1}{n} \sum_{i=1}^n \delta(y_i - y) dy \quad (\text{F.23})$$

$$= \frac{1}{n} \sum_{i=1}^n \int_{-\infty}^{\infty} (y - \mathbb{E}[y])^2 \delta(y_i - y) dy \quad (\text{F.24})$$

$$= \frac{1}{n} \sum_{i=1}^n (y_i - \mathbb{E}[y])^2, \quad (\text{F.25})$$

which is similar to the sample variance given in [equation \(1.29\)](#), but lacks the  $n/(n-1)$  factor which corrects for the bias which is introduced by using the data to estimate  $\mathbb{E}[y]$ . Therefore, what is used in practice is

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2. \quad (\text{F.26})$$

**F.5.2.1.2 Weighted estimators obtained from the empirical distribution** The sample percentiles described in [section 1.4.3.1](#) as well as the empirical distribution of the previous section treat all of the data as being equally reliable. When some data in a given channel are more reliable than others, however, it is useful to construct a *weighted* empirical CDF which accounts for this. Instead of jumping by  $1/n$  at each point, the weighted empirical CDF jumps by  $1/(\sigma_i^2 \sum_{j=1}^n 1/\sigma_j^2)$  at each point:

$$\hat{F}_{Y,w}(y) = \frac{1}{\sum_{i=1}^n 1/\sigma_i^2} \sum_{i=1}^n \frac{\mathbb{I}(y_i \leq y)}{\sigma_i^2}. \quad (\text{F.27})$$

The weighted quartiles are then obtained from this CDF as before, an operation which is implemented by the `scoreatpercentilew()` function in `profiletools`. The weighted empirical PDF is

$$\hat{f}_{Y,w}(y) = \frac{1}{\sum_{i=1}^n 1/\sigma_i^2} \sum_{i=1}^n \frac{\delta(y_i - y)}{\sigma_i^2}. \quad (\text{F.28})$$

The mean of this distribution is

$$\mathbb{E}[y] = \int_{-\infty}^{\infty} y \hat{f}_{Y,w}(y) dy \quad (\text{F.29})$$

$$= \int_{-\infty}^{\infty} y \frac{1}{\sum_{i=1}^n 1/\sigma_i^2} \sum_{i=1}^n \frac{\delta(y_i - y)}{\sigma_i^2} dy \quad (\text{F.30})$$

$$= \frac{1}{\sum_{i=1}^n 1/\sigma_i^2} \sum_{i=1}^n \int_{-\infty}^{\infty} \frac{y}{\sigma_i^2} \delta(y_i - y) dy \quad (\text{F.31})$$

$$= \frac{\sum_{i=1}^n y_i / \sigma_i^2}{\sum_{i=1}^n 1/\sigma_i^2} = \bar{y}_w, \quad (\text{F.32})$$

which is known as the *weighted sample mean*. The weighted sample mean is implemented in the `meanw()` function provided by `profiletools`. The variance of the

weighted empirical distribution is

$$\text{var}[y] = \int_{-\infty}^{\infty} (y - \mathbb{E}[y])^2 \hat{f}_{Y,w}(y) dy \quad (\text{F.33})$$

$$= \int_{-\infty}^{\infty} (y - \mathbb{E}[y])^2 \frac{1}{\sum_{i=1}^n 1/\sigma_i^2} \sum_{i=1}^n \frac{\delta(y_i - y)}{\sigma_i^2} dy \quad (\text{F.34})$$

$$= \frac{1}{\sum_{i=1}^n 1/\sigma_i^2} \sum_{i=1}^n \int_{-\infty}^{\infty} \frac{(y - \mathbb{E}[y])^2}{\sigma_i^2} \delta(y_i - y) dy \quad (\text{F.35})$$

$$= \frac{\sum_{i=1}^n (y - \mathbb{E}[y])^2 / \sigma_i^2}{\sum_{i=1}^n 1/\sigma_i^2}. \quad (\text{F.36})$$

Similar to the unweighted sample variance, this estimator is biased when the data are used to estimate the mean. It can be shown that

$$s_w^2 = \frac{\sum_{i=1}^n (y_i - \bar{y}_w)^2 / \sigma_i^2}{\sum_{i=1}^n 1/\sigma_i^2 - \frac{\sum_{i=1}^n 1/\sigma_i^4}{\sum_{i=1}^n 1/\sigma_i^2}} \quad (\text{F.37})$$

is a consistent, unbiased estimator for the *weighted sample variance*. This is implemented in the `varw()` function in `profiletools`.

### F.5.2.2 A Bayesian view of weighted and unweighted estimators

This section attempts to use a purely Bayesian framework to make it clear what the probabilistic assumptions underlying the estimators discussed above are. This section closely follows the development section 3.3 of [283].

#### F.5.2.2.1 Unweighted, non-robust estimators as Bayesian parameter estimates

Consider a set of  $n$  samples  $\mathbf{y}$  where each element  $y_i \sim \mathcal{N}(\mu, \sigma^2)$ . The likelihood is

$$f_{Y|M,\Sigma}(\mathbf{y}|\mu, \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \quad (\text{F.38})$$

$$= \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right). \quad (\text{F.39})$$



Assume the following (improper<sup>1</sup>) prior density for  $\mu$  and  $\sigma$ :

$$f_{M,\Sigma}(\mu, \sigma) \propto \begin{cases} 1, & \sigma \geq 0 \\ 0, & \text{otherwise} \end{cases}. \quad (\text{F.40})$$

The posterior PDF for  $\mu$  and  $\sigma \geq 0$  given the data is then

$$f_{M,\Sigma|Y}(\mu, \sigma|y) \propto \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right). \quad (\text{F.41})$$

First, find the marginal posterior density for  $\mu$ :

$$f_{M|Y}(\mu|y) \propto \int_0^\infty \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right) d\sigma. \quad (\text{F.42})$$

Make the substitution  $t = 1/\sigma$ ,  $dt = -1/\sigma^2$ :

$$= \int_0^\infty \frac{t^{n-2}}{(2\pi)^{n/2}} \exp\left(-\frac{t^2}{2} \sum_{i=1}^n (y_i - \mu)^2\right) dt. \quad (\text{F.43})$$

Now make the substitution  $u = t \sqrt{\sum_{i=1}^n (y_i - \mu)^2/2}$ ,  $du = dt \sqrt{\sum_{i=1}^n (y_i - \mu)^2/2}$ :

$$= \frac{1}{(2\pi)^{n/2}} \left( \sum_{i=1}^n (y_i - \mu)^2/2 \right)^{-(n-1)/2} \int_0^\infty u^{n-2} e^{-u^2} dt. \quad (\text{F.44})$$

Finally, drop the constant factors to obtain

$$\propto \left( \sum_{i=1}^n (y_i - \mu)^2 \right)^{-(n-1)/2}. \quad (\text{F.45})$$

Therefore, the marginal posterior log-PDF for  $\mu$  is

$$\ln f_{M|Y}(\mu|y) = -\frac{n-1}{2} \ln \left( \sum_{i=1}^n (y_i - \mu)^2 \right). \quad (\text{F.46})$$

1. An *improper prior distribution* is a prior distribution which cannot be normalized. The use of such a prior distribution is contingent upon having sufficient data to yield a posterior distribution which is normalizable.

To find the MAP estimate for  $\mu$ , set the first derivative of this equal to zero:

$$\left. \frac{d}{d\mu} \ln f_{M|Y}(\mu|\mathbf{y}) \right|_{\mu=\mu_0} = \frac{(n-1) \sum_{i=1}^n (y_i - \mu_0)}{\sum_{i=1}^n (y_i - \mu_0)^2} = 0 \quad (\text{F.47})$$

$$\sum_{i=1}^n (y_i - \mu_0) = 0 \quad (\text{F.48})$$

$$\mu_0 = \frac{1}{n} \sum_{i=1}^n y_i = \bar{y}. \quad (\text{F.49})$$

Therefore, the unweighted sample mean as given in [equation \(1.28\)](#) and [equation \(F.21\)](#) is the MAP estimate for the mean of the distribution which generated the  $n$  independent and identically distributed samples  $\mathbf{y}$ . The uncertainty in the mean can be estimated by finding the second derivative of the marginal posterior log-PDF:

$$\frac{d^2}{d\mu^2} \ln f_{M|Y}(\mu|\mathbf{y}) = (n-1) \frac{-n \sum_{i=1}^n (y_i - \mu)^2 + 2 \left( \sum_{i=1}^n (y_i - \mu) \right)^2}{\left( \sum_{i=1}^n (y_i - \mu)^2 \right)^2}. \quad (\text{F.50})$$

But, at  $\mu = \mu_0$ ,  $\sum_{i=1}^n (y_i - \mu) = 0$ , so

$$\left. \frac{d^2}{d\mu^2} \ln f_{M|Y}(\mu|\mathbf{y}) \right|_{\mu=\mu_0} = -\frac{n(n-1)}{\sum_{i=1}^n (y_i - \mu_0)^2}. \quad (\text{F.51})$$

The variance in the estimate  $\mu_0$  is then

$$\text{var}[\mu_0] = -\frac{1}{d^2 \ln f_{M|Y}(\mu|\mathbf{y})/d\mu^2|_{\mu=\mu_0}} = \frac{1}{n} \frac{1}{n-1} \sum_{i=1}^n (y_i - \mu_0)^2 = \frac{s^2}{n}, \quad (\text{F.52})$$

where  $s^2$  is the unbiased estimate of the sample variance given in [equation \(1.29\)](#) which means that  $s/\sqrt{n}$  is the MAP estimate's uncertainty in the mean.

Now consider the marginal posterior density for  $\sigma$ . First, note that

$$\sum_{i=1}^n (y_i - \mu)^2 = n(\mu_0 - \mu)^2 + \sum_{i=1}^n (y_i - \mu_0)^2. \quad (\text{F.53})$$

Then,

$$f_{\Sigma|Y}(\sigma|\mathbf{y}) \propto \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right) d\mu \quad (\text{F.54})$$

$$= \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu_0)^2\right) \int_{-\infty}^{\infty} \exp\left(-\frac{n(\mu_0 - \mu)^2}{2\sigma^2}\right) d\mu \quad (\text{F.55})$$

$$\propto \sigma^{1-n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu_0)^2\right). \quad (\text{F.56})$$

Therefore, the marginal posterior log-PDF for  $\sigma$  is

$$\ln f_{\Sigma|Y}(\sigma|\mathbf{y}) = (1-n) \ln \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu_0)^2. \quad (\text{F.57})$$

Taking the first derivative and setting it to zero yields

$$\frac{d}{d\sigma} \ln f_{\Sigma|Y}(\sigma|\mathbf{y}) \Big|_{\sigma=\sigma_0} = \frac{1-n}{\sigma_0} + \frac{1}{\sigma_0^3} \sum_{i=1}^n (y_i - \mu_0)^2 = 0 \quad (\text{F.58})$$

$$\sigma_0^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \mu_0)^2 = s^2. \quad (\text{F.59})$$

Therefore, the unbiased estimate of the sample variance as given in [equation \(1.29\)](#) is the MAP estimate for the variance of the distribution which generated the  $n$  independent and identically distributed samples  $\mathbf{y}$ .

**F.5.2.2.2 Weighted sample mean as a Bayesian parameter estimate** Now consider the case where each measurement has its own uncertainty, but the means are all the same:  $y_i \sim \mathcal{N}(\mu, \sigma_i^2)$ . The likelihood is then

$$f_{Y|M}(\mathbf{y}|\mu) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma_i^2}\right) \quad (\text{F.60})$$

$$= \frac{1}{(2\pi)^{n/2} \prod_{i=1}^n \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^n \frac{(y_i - \mu)^2}{\sigma_i^2}\right). \quad (\text{F.61})$$

Assuming the same improper prior distribution for  $\mu$  used above, the posterior PDF for  $\mu$  is simply

$$f_{M|Y}(\mu|\mathbf{y}) \propto \frac{1}{(2\pi)^{n/2} \prod_{i=1}^n \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^n \frac{(y_i - \mu)^2}{\sigma_i^2}\right), \quad (\text{F.62})$$

which means the log-PDF is

$$\ln f_{M|Y}(\mu|\mathbf{y}) \propto -\frac{1}{2} \sum_{i=1}^n \frac{(y_i - \mu)^2}{\sigma_i^2}. \quad (\text{F.63})$$

Differentiating this and setting it to zero gives

$$\left. \frac{d}{d\mu} \ln f_{M|Y}(\mu|\mathbf{y}) \right|_{\mu=\mu_o} = \sum_{i=1}^n \frac{y_i - \mu_o}{\sigma_i^2} = 0 \quad (\text{F.64})$$

$$\mu_o = \frac{\sum_{i=1}^n y_i / \sigma_i^2}{\sum_{i=1}^n 1 / \sigma_i^2} = \bar{y}_w, \quad (\text{F.65})$$

which is exactly the weighted sample mean obtained from the weighted empirical CDF in [equation \(F.32\)](#). To find the uncertainty in the mean, take the second derivative of the log-PDF:

$$\left. \frac{d^2}{d\mu^2} \ln f_{M|Y}(\mu|\mathbf{y}) \right|_{\mu=\mu_o} = - \sum_{i=1}^n \frac{1}{\sigma_i^2} \quad (\text{F.66})$$

$$\text{var}[\mu_o] = \frac{1}{\sum_{i=1}^n 1 / \sigma_i^2}. \quad (\text{F.67})$$

Unfortunately, there does not appear to be a simple Bayesian interpretation of the weighted sample variance given in [equation \(F.37\)](#).

### F.5.3 Methods of averaging

`profiletools` provides five methods of averaging the data  $\mathbf{y}$  and  $\mathbf{X}$ . For each method, the estimators used can either be robust or non-robust and weighted or unweighted. When using a weighted estimator, the weights obtained from  $\mathbf{y}$  are used for weighting the  $\mathbf{X}$  data such that the estimate  $\bar{x} \pm \sigma_x$  is consistent with the weighting applied to  $\mathbf{y}$ . In the following sections only the unweighted, non-robust case will typically be shown. Replacing the means, standard deviations and variances with the appropriate weighted and/or robust versions will yield the other types of estimates. The following sections explain each of the five methods and their statistical implications. In all cases, the sample mean or median is used as the estimate for the center of the samples  $\bar{y}$ ; the question is how to determine the uncertainty of the samples, indicated by  $\sigma_y$ , in the following.

### F.5.3.1 Sample statistics

The simplest approach is to use the sample standard deviation (or the sample interquartile range):  $\sigma_y = s$ . This corresponds to the case where the points in a given channel may not exhibit purely statistical variation and this variation should be represented in the uncertainty of the fit. As more points are added, the uncertainty for a given channel will converge to a finite value: this method represents an uncertainty which does not get averaged away. This is often the best choice for plasma data, as the profile is rarely truly steady over a given period of time, so the uncertainty of the fit should communicate the temporal variation of the data.

### F.5.3.2 Root-mean-square uncertainties

When the sample size is too small to obtain a reasonable estimate of the sample variance, an average uncertainty may be approximated by the root-mean-square (RMS) standard deviation:

$$\sigma_y = \sqrt{\mathbb{E}[\sigma^2]} = \sqrt{\frac{1}{n} \sum_{i=1}^n \sigma_i^2}. \quad (\text{F.68})$$

The statistical implication is that the variances of the samples  $y_i \sim \mathcal{N}(\mu, \sigma_i^2)$  have some distribution  $\sigma_i^2 \sim f_{\Sigma^2}(\sigma^2)$ . This method then estimates the mean of this distribution,  $\mathbb{E}[\sigma^2]$ , when there are too few samples to estimate the sample variance itself.

### F.5.3.3 Total variance

Sometimes the means and uncertainties in a channel represent the output of a procedure for which the measurements represent the expected value and its uncertainty under different conditions,  $t$ . In this case the variance in  $y$  is given by the law of total variance:

$$\text{var}[y] = \text{var}_t [\mathbb{E}[y|t]] + \mathbb{E}_t [\text{var}[y|t]]. \quad (\text{F.69})$$

This is essentially the sum of the variances corresponding to the sample statistics and root-mean-square uncertainty cases. It represents the fact that there is variance in the expectation between different conditions  $t$  as well as an expected level of variance which would not be captured by sample statistics because  $y$  already represents an average.

#### F.5.3.4 The uncertainty in the mean

When it is the case that the samples in a given channel do in fact represent independent samples of a given quantity, the uncertainty in their mean will tend to zero like  $1/\sqrt{n}$  as the number of samples increases. The variance in the unweighted sample mean is obtained from the uncertainty propagation equation:

$$\text{var}[\bar{y}] = \frac{1}{n^2} \sum_{i=1}^n \sigma_i^2. \quad (\text{F.70})$$

For equal uncertainties  $\sigma_i = \sigma$ , this reduces to the standard result

$$\text{var}[\bar{y}] = \frac{\sigma^2}{n} \quad (\text{F.71})$$

which was obtained in [equation \(F.52\)](#). The variance in the weighted mean was obtained in [equation \(F.67\)](#):

$$\text{var}[\bar{y}_w] = \frac{1}{\sum_{i=1}^n 1/\sigma_i^2}. \quad (\text{F.72})$$

If performing a weighted average of the  $\mathbf{X}$  values, the weights are  $1/\sigma_i^2$ , where  $\sigma_i$  is the uncertainty in  $y_i$ . Therefore, [equation \(F.72\)](#) does not hold. Instead, use the uncertainty propagation equation to obtain

$$\text{var}[\bar{X}_w] = \frac{\sum_{i=1}^n \text{var}[X_i]/\sigma_i^4}{(\sum_{i=1}^n 1/\sigma_i^2)^2}. \quad (\text{F.73})$$

#### F.5.3.5 The uncertainty in the mean using the sample variance

If reliable uncertainty estimates are not available, the expressions in the previous section cannot be used. To obtain an uncertainty estimate with an equivalent statistical interpretation that can be used in this case, consider each point to have a variance equal to the sample variance. The variance in the unweighted mean is then

$$\text{var}[\bar{y}] = \frac{s^2}{n}. \quad (\text{F.74})$$

The use of this approach with the weighted mean of  $\mathbf{y}$  does not make sense as the uncertainties will have been used to form the weighted mean but would then not be used to estimate the uncertainty in the weighted mean. But, this is useful for the weighted mean of  $\mathbf{X}$ , where the uncertainties of the samples contained in  $\mathbf{X}$  may not

be known but the weights are obtained from the uncertainties in  $\mathbf{y}$ . Applying the uncertainty propagation equation, the variance in the weighted mean of  $\mathbf{X}$  is

$$\text{var}[\bar{X}_w] = s_{\mathbf{x},w}^2 \frac{\sum_{i=1}^n 1/\sigma_i^4}{(\sum_{i=1}^n 1/\sigma_i^2)^2}, \quad (\text{F.75})$$

where  $s_{\mathbf{x},w}^2$  is the weighted sample variance of  $\mathbf{x}$ .

### F.5.3.6 Illustration of the various averaging schemes

To illustrate the preceding discussion,  $T_e$  data from channel 14 of the edge Thomson scattering system<sup>2</sup> on Alcator C-Mod shot 1110329013 (the sawtooth-suppressed shot used to illustrate the different averaging schemes in section 2.5.4) are shown in figure F.4, the empirical distributions are shown in figure F.5 and the means and uncertainties computed with the various techniques are listed in table F.1.

In the first row of figure F.5, the empirical CDF and weighted empirical CDF were constructed with the formulas given in section F.5.2.1. The other CDFs shown are for normal distributions having each of the means and variances computed using all of the techniques described in the previous sections. The CDF of the distribution for the mean (cyan and yellow curves) is much steeper than the others because of the  $1/\sqrt{n}$  scaling of the uncertainty in the mean. Because of this difference in interpretation, the distribution for the mean is not shown on the histograms or Q-Q plots. Instead, a vertical line at the location of the mean and a vertical  $\pm 1\sigma$  uncertainty band is shown.

The histograms are shown with eight bins in accordance with the Freedman-Diaconis rule that the bin width be  $2 \cdot \text{IQR}/n^{1/3}$  [538]. The weighted histogram in the middle row of figure F.5 is clearly biased towards lower values, and also appears to be somewhat skew. This is likely because there cannot be a negative temperature, so the assumption of a normal distribution breaks down at small values. The key aspect of the weighted versus unweighted histogram is the influence of the presumed outlier with  $T_e \approx 1.2$  keV: on the unweighted histogram, this is the single blob in the tail of the distribution which draws all of the unweighted, non-robust estimators towards it. The point has such a large uncertainty, however, that it effectively disappears in the weighted histogram, thereby bringing the weighted mean and median into agreement.

The Q-Q plots in the bottom row were generated by plotting  $\Phi^{-1}(\hat{F}_Y(y))$  as a function of  $y$ , where  $\Phi^{-1}(x)$  is the inverse CDF of the standard normal,  $\mathcal{N}(0, 1)$ . If the data lie on a straight line then it is reasonable to believe that they can be described

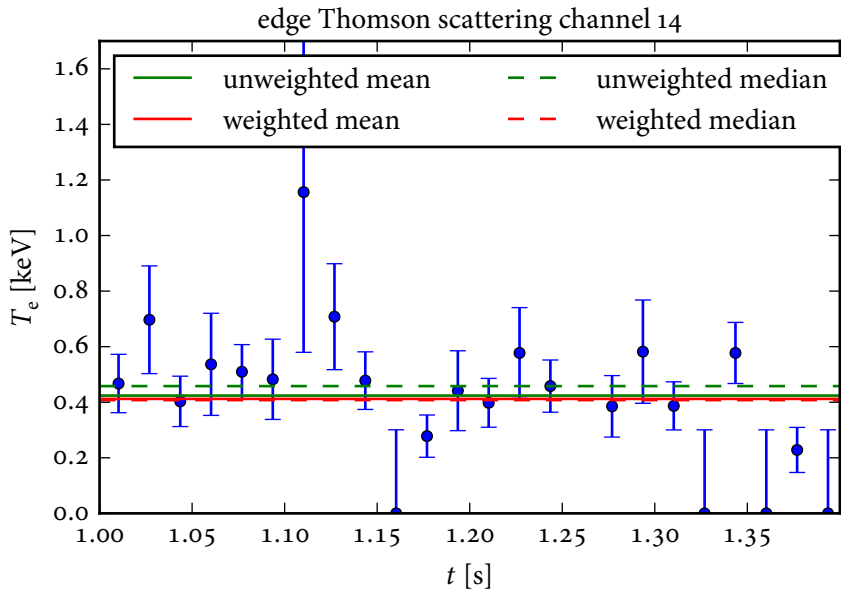
2. Where the innermost channel is 0 and the outermost channel is 17.

by the normal distribution. Despite the fact that the data are not expected to be truly normal because there cannot be a negative temperature, the normal distribution still does appear to be a reasonable description of the data.

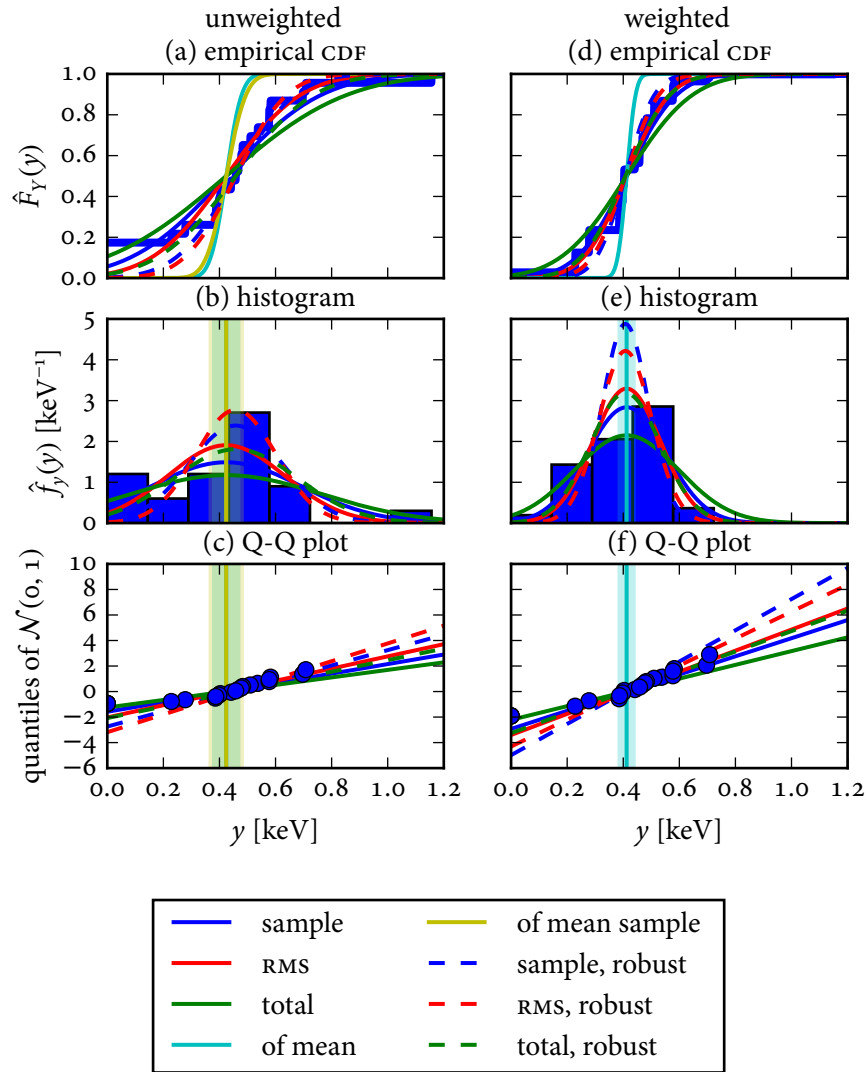
Given the quantiles of  $\mathcal{N}(0, 1)$  on the vertical axis, any other normal distribution  $\mathcal{N}(\mu, \sigma^2)$  will be a straight line with  $x$ -intercept  $\mu$  and slope  $1/\sigma$ . This formulation was used to draw the straight lines in the Q-Q plots. Points at the edges which lie above or below the trend extrapolated from the points in the middle of the distribution may be outliers. This may be the case with the lowest two points and the three highest points (including the one with  $T_e \approx 1.2$  keV which is not shown on the Q-Q plot because  $\Phi^{-1}(1) = \infty$ ), as the robust sample estimator (blue dashed line) has a clearly different slope (i.e., uncertainty) than the non-robust estimators on both the weighted and unweighted plots. In other words, the robust estimate is able to ignore possibly non-Gaussian outliers even when their uncertainties are not big enough for weighting to eliminate them. That being said, this may in fact yield overly optimistic results, given that so many points in the tails of the distribution are effectively being ignored. That is why this thesis used the weighted sample mean and standard deviation whenever diagnostic uncertainty estimates were available.

From figure F.5, it is clear that  $\mathcal{N}(\bar{y}, s^2)$  describes the data very well with robust, unweighted estimators and both robust and non-robust weighted estimators for  $\bar{y}$  and  $s^2$ . The RMS variance is reasonable, but does not fit as well: for this case, there are 19 data points, which is more than adequate to estimate the sample standard deviation, so this method is not recommended here anyways. The total variance does not describe the empirical distribution well, which is expected given that it represents an assumption that each data point represents an average and hence the scatter in the data is not representative of the actual scatter in what is being measured.





**Figure F.4:** Data from channel 14 of the edge Thomson scattering system from Alcator C-Mod shot 1110329013 used to illustrate the different averaging schemes. The time window is the same as was used in [section 2.5.4](#). Recall that this portion of the discharge is sawtooth-suppressed and relatively constant, so the variation from time point to time point is expected to be almost purely statistical. The points are shown with the  $\pm 1\sigma$  error bars provided by the Thomson scattering analysis itself. The [green solid line](#) is the unweighted sample mean, the [green dashed line](#) is the weighted sample median, the [red solid line](#) is the weighted sample mean and the [red dashed line](#) is the weighted sample median. The choice between robust and non-robust estimators (dashed versus solid) has little effect, but the choice between weighted and unweighted estimators ([green](#) versus [red](#)) has a larger effect: the data points with higher values have higher uncertainties, so the weighted estimators are lower than the unweighted ones. Note the one point with  $T_e \approx 1.2$  keV and the two points with  $T_e \approx 0.25$  keV: one might expect these to be outliers given their distance from the sample mean and median. This is confirmed by the Q-Q plots in [figure F.5](#). The weighted, robust estimators are the least influenced by these three points. This figure was produced using `make_averaging_demo.py`.



**Figure F.5:** Empirical distributions for the data shown in [figure F.4](#). The left column gives the unweighted data and the right column gives the weighted data. The top row is the cumulative distribution function (CDF), the middle row is the histogram and probability density function (PDF), the bottom row is the Q-Q plot. Non-robust estimators are shown as solid lines, robust estimators as dashed lines. In the bottom two rows the distribution of the mean is not shown. Instead, a vertical line at the location of the mean is shown with its  $\pm 1\sigma$  envelope. For the unweighted estimators, both the uncertainty in the mean from uncertainty propagation ([equation \(F.72\)](#), [cyan curve](#)) and from sample statistics ([equation \(F.74\)](#), [yellow curve](#)) are shown, but they overlap completely to yield the [green band](#). Note that Q-Q plots are traditionally shown with the theoretical quantiles on the horizontal axis, but they are shown on the vertical axis here so the plots can all have the same horizontal axis. The maximum value of the data set (which happens to be the presumed outlier) has  $\Phi^{-1}(1) = \infty$  and is not shown. This figure was produced using [make\\_averaging\\_demo.py](#).

**Table F.1:** Results of the different averaging schemes applied to the data from channel 14 of the edge Thomson scattering system from Alcator C-Mod shot 1110329013.

Method		unweighted		weighted	
		$\bar{y}$	$\sigma_y$	$\bar{y}$	$\sigma_y$
Non-robust	sample	0.424	0.27	0.412	0.14
	RMS		0.21		0.12
	total		0.34		0.19
	of mean		0.044		0.025
	of mean, sample		0.056		—
Robust	sample	0.458	0.17	0.408	0.082
	RMS		0.14		0.095
	total		0.22		0.13

# G

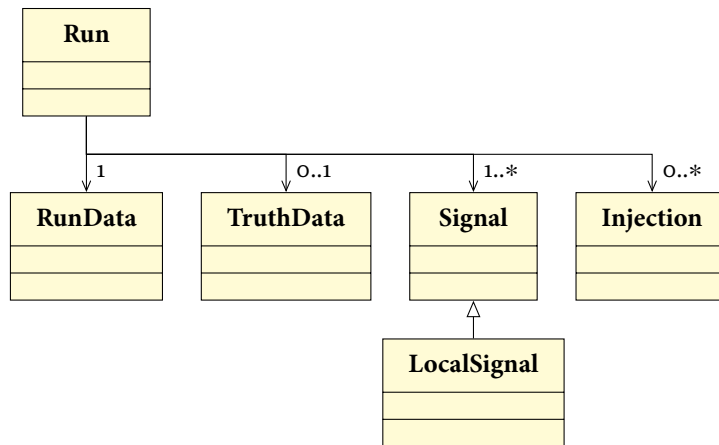
## bayesimp: an open-source Python package for Bayesian impurity transport analysis

### G.1 Package overview

`bayesimp` is a Python package which implements the Bayesian analysis of impurity transport data described in [chapter 3](#). The source code is available in [\[539\]](#), and documentation is provided directly in the source files. The code consists of two modules: `bayesimp` implements the actual analysis, including retrieval and processing of the experimental data, wrapping the STRAHL impurity transport code [\[276\]](#), and performing the actual inference using some combination of MULTINEST [\[452–455\]](#), `scipy.optimize` [\[293\]](#) and NLOpt [\[540\]](#). `lines` implements the simplified spectral model described in [section 3.5.2](#), and is based on the IDL script `lines.pro` [\[45\]](#). To function, the code requires a master directory be provided containing the actual STRAHL executable and its associated files, including the ionization, recombination and photon emission rate coefficients in the appropriate ADAS file formats. For each version of each analysis of a given shot, this directory is cloned and the relevant data are stored in pickle files to support efficient re-analysis of a given case.

### G.2 The `bayesimp` module

The basic structure of the `bayesimp` module is given in [figure G.1](#). The core class is `Run`, which contains the background  $n_e$  and  $T_e$  profile data in an instance of `RunData` and the diagnostic signals in one of more instances of `Signal` and `LocalSignal`. For real data, the individual injections which were combined into a composite time



**Figure G.1:** Basic structure of the bayesimp package. The main class is Run, which contains one instance of RunData, zero or one instances of TruthData (depending on whether or not it contains synthetic data), one or more instances of Signal or its subclass LocalSignal and zero or more instances of Injection (depending on whether or not it contains synthetic data).

series are stored as one or more Injection instances. For synthetic data, the true (i.e., noise-free) values are stored in an instance of TruthData.

### G.2.1 The Run class

The Run class is the main class of bayesimp: the first step to analyzing a shot is to create a Python program which invokes the Run class' constructor, and all subsequent analysis is conducted by calling the various methods of the Run instance. The following sections explain the methods of the Run class roughly in the order in which they are invoked.

#### G.2.1.1 Creating and restoring a Run: the `__init__()` method and associated helper methods

The constructor of Run takes a large number of arguments to specify the properties of the data to load/synthesize and the basis functions to use when inferring the transport coefficient profiles. When first analyzing a shot, the `setup_files()` method is called. This method clones the master STRAHL directory into a new directory with name "strahl\_SHOT\_VERSION" then calls the IDL script `setup_strahl_run.pro` to pull the HIREX-SR data from the tree and generate the grid file required by STRAHL using routines provided by M. Reinke and T. Pütterich. The raw data are stored

in an IDL save file called “run\_data.sav” for future use. Once the directory is in place, the constructor looks to see if the data file called “run\_data.pkl” containing the background  $n_e$  and  $T_e$  profile data exists. If not, a RunData instance is created. The RunData class’ constructor launches the `gpfitt` GUI from `profiletools` to allow the user to fit the  $n_e$  and  $T_e$  profiles interactively using GPR. The RunData instance containing the fitted profiles is then stored in a pickle file called “run\_data.pkl” for future use. Next, the constructor checks to see if the pickle files containing the processed diagnostic signals exist. Line-integrated data are stored in a pickle file called “signals.pkl” and local data are stored in a pickle file called “local\_signals.pkl.” For line-integrated datasets, the argon data are also loaded and are stored in a pickle file called “ar\_signal.pkl.” For synthetic datasets, the true (i.e., noise-free) signals are stored in a pickle file called “truth\_data.pkl.” If the relevant files do not exist, the data are loaded: if the `params_true` keyword is not provided, real experimental data are loaded from the MDSplus tree, otherwise a synthetic dataset is created. When loading experimental data, the `HirexData`, `VUVDData` and `XTOMODData` helper classes are used to preprocess the data into the standardized form required by the `Signal` class. Once all of the data have been loaded and stored in the list `self.signals`, the `compute_view_data()` method is called. This method uses `TRIPPY` to compute the matrices used to implement the line integrals. If synthetic data are to be created, empty placeholders are first created for all of the local and line-integrated signals to be synthesized. Then, the appropriate functions (described below) are used to synthesize the diagnostic signals given the true parameters. Finally, the `apply_noise()` method is used to add synthetic noise. Once this process is complete, the data are stored in the relevant pickle files for future use and the Run instance is ready.

#### G.2.1.2 The model parameters and the `split_params()` and `get_prior()` methods

When evaluating the forward model, there are eight types of parameters. In the order they are given when calling the relevant methods, these are:

1. Coefficients for the  $D$  profile. If the basis function method (stored in the method attribute) is “linterp” (i.e., linear interpolation) these are simply the values at each knot except the one at  $r/a = 0$ , which is determined by the boundary conditions. If the basis function method is “spline” these are the actual B-spline coefficients. The number of coefficients is determined by the `num_eig_D` attribute and the order of spline to use is stored in the `spline_k_D` attribute.

2. Coefficients for the  $V$  profile. The number of coefficients is determined by the `num_eig_V` attribute and the order of spline to use is stored in the `spline_k_V` attribute.
3. Internal knot locations for the  $D$  profile. The number of knots is determined by the number of coefficients and the spline order.
4. Internal knot locations for the  $V$  profile.
5. Scaling parameters for the signals. When the signals are normalized, the uncertainty in the peak value can be accounted for by including nuisance parameters which scale the normalized signals for each diagnostic.
6. Time shifts for the signals. It has been observed that the timebases of HIREX-SR and the impurity injector do not agree for some shots. This can be accounted for by introducing nuisance parameters which shift each diagnostic's timebase.
7. Weights for the  $n_e$  profile. Used to propagate the uncertainty in the  $n_e$  profile using a truncated eigendecomposition of the Gaussian process fit.
8. Weights for the  $T_e$  profile.

The `split_params()` method splits a parameter vector into these components. The `get_prior()` method constructs a `gptools.JointPrior` instance which represents the prior distribution for the parameters.

### G.2.1.3 Calling STRAHL and evaluating the log-posterior

The `DV2cs_den()` method is the interface between Python and STRAHL, which is a compiled Fortran program. Unfortunately, a direct code interface to STRAHL is not available, so control of the code is accomplished by writing various input files to disk. Several files do not change and are only written when `Run.__init__()` executes:

- `Ca.atomdat` tells STRAHL which spectral lines to compute emissivity profiles for, and is only used if STRAHL's radiation calculations are enabled.
- `nete/CaflxSHOT.dat` contains the particle source function and is written using the `write_source()` method.
- `nete/grid_SHOT.0` is the grid file which describes the magnetic geometry and is written using the IDL routine `make_cmod_grid.pro`.

Given an input parameter vector, `DV2cs_den()` works by first forming the  $D$ ,  $V$ ,  $n_e$  and  $T_e$  profiles. Then, the `write_control()` method is used to write the file `strahl.control` which tells STRAHL how long to run the simulation for and to execute without waiting for manual intervention. Next, the `write_pp()` method is used to write the plasma parameter file `nete/ppSHOT.0` which contains the  $n_e$  and  $T_e$  profiles. Then, the `write_param()` file is used to write the parameter file `param_files/run_SHOT.0` which contains the  $D$  and  $V$  profiles. With these files in place, STRAHL is called to evaluate the spatiotemporal evolution of the charge state density profiles. These results are written to the file `result/strahl_result.dat`, which `DV2cs_den()` reads and returns the contents of.

**G.2.1.3.1 Method chain for line-integrated data** The `cs_den2dlines()` method converts the charge state density profiles obtained from `DV2cs_den()` to the necessary local emissivity profiles. For isolated spectral lines the `compute_emiss()` function is used to evaluate the emissivity according to the procedure described in [section 3.5.2.1](#). For the broadband soft x-ray emission the `lines.compute_SXR()` function is used to evaluate the emissivity according to the procedure described in [section 3.5.2.2](#).

The `dlines2sig()` method converts the local emissivity profiles obtained from `cs_den2dlines()` into simulated diagnostic signals. The emissivity profiles are interpolated onto the diagnostic timebase using cubic splines and then the line integrals are performed using the transformation matrices computed using TRIPPy.

The `sig2diffs()` method computes the difference between the simulated diagnostic signals obtained from `dlines2sig()` and the observed data.

The `diffs2ln_prob()` method converts the signal differences obtained from `sig2diffs()` into either the log-likelihood or the log-prior, using the method described in [section 3.5.3.1](#).

**G.2.1.3.2 Method chain for local data** The `cs_den2local_sigs()` method converts the charge state density profiles obtained from `DV2cs_den()` to simulated local diagnostic signals. The local impurity densities are interpolated onto the diagnostic grid in space and time using bivariate cubic splines.

The `local_sig2local_diffs()` computes the difference between the simulated local diagnostic signals obtained from `cs_den2local_sigs()` and the observed data.

The `local_diffs2ln_prob()` method converts the signal differences obtained from `local_sig2local_diffs()` into either the log-likelihood or the log-prior, using the method described in [section 3.5.3.1](#).



**G.2.1.3.3 End-to-end methods for log-probabilities** The `DV2ln_prob()` method carries out all of the steps above to convert a parameter vector into a log-posterior, using either local or line-integrated data.

The `u2ln_prob()` method uses the inverse CDF of the prior distribution to map a parameter vector in the unit hypercube to a parameter vector in the parameter space defined by the prior distribution before calling `DV2ln_prob()`. It also provides the ability to compute the gradient with respect to the parameters using finite differences. A forward difference is used unless a parameter is too close to its bounds, in which case a backwards difference is used. This is an improvement over the finite difference routine provided by `scipy.optimize` which always uses forward differences and hence can request the model be evaluated at a point outside of the allowed parameter space.

**G.2.1.3.4 Methods for gradients** The `DV2jac()` method uses `numdifftools` [541] to compute the gradient with respect to the parameters.

The `DV2d_ln_prob()` method computes the derivative with respect to a single parameter using finite differences. By default, a high-order centered difference [542] is used unless the parameter is too close to its bounds, in which case either a backwards or forwards difference is used, as appropriate.

The `DV2d2_ln_prob()` method computes the second derivative with respect to a pair of parameters using finite differences applied to repeated calls of `DV2d_ln_prob()`.

The `DV2hessian()` method computes the Hessian matrix with respect to the parameters using repeated calls to `DV2d2_ln_prob()`.

The `DV2hessian_2()` method computes the Hessian matrix with respect to the parameters by fitting a second-order polynomial to repeated evaluations of `DV2ln_prob()` over a dense grid.

#### **G.2.1.4 Efficient handling of nuisance parameters: pseudo-marginal MCMC and the `dlines2ln_prob_marg()` method**

Notice that there are two types of parameters: the ones which are actually of interest,  $\theta$ , and the ones which are uninteresting nuisance parameters such as time shifts and scaling factors,  $z$ . In our case the part of the forward model which depends on  $\theta$  is very expensive to evaluate (i.e., the STRAHL run and the spectral model), but the part which depends on  $z$  is fairly cheap (interpolating onto the diagnostic timebase and finding the sum of squares residual). This structure can be exploited using pseudo-marginal sampling [543]: instead of sampling the full posterior distribution, the MCMC sampling takes place only for the interesting parameters  $\theta$ . At each step of MCMC sampling, the posterior density is marginalized over the nuisance parameters.

The marginalization is of low enough dimension that the integral can be performed with (quasi) Monte Carlo sampling or Gaussian quadrature. This process is computationally cheap because it does not require any STRAHL runs. *bayesimp* provides experimental support for this procedure using the `dlines2ln_prob_marg()` method and the `use_PMMCMC` keyword.

**G.2.1.4.1 Implementation of the log-posterior** The full posterior distribution is

$$f_{\theta, z|D}(\theta, z|D) = \frac{f_{D|\theta, z}(D|\theta, z)f_{\theta}(\theta)f_z(z)}{f_D(D)}, \quad (\text{G.1})$$

where  $D$  is the experimental data and independent prior distributions for  $\theta$  and  $z$  have been assumed. What we actually care about is the marginal posterior distribution for  $\theta$ :

$$f_{\theta|D}(\theta|D) = \int f_{\theta, z|D}(\theta, z|D) dz \propto \int f_{D|\theta, z}(D|\theta, z)f_{\theta}(\theta)f_z(z) dz. \quad (\text{G.2})$$

This is equivalent to finding

$$f_{\theta|D}(\theta|D) \propto \mathbb{E}_z [f_{D|\theta, z}(D|\theta, z)f_{\theta}(\theta)] = f_{\theta}(\theta) \mathbb{E}_z [f_{D|\theta, z}(D|\theta, z)]. \quad (\text{G.3})$$

To find the expectation, use the Monte Carlo estimate

$$\mathbb{E}_z [f_{D|\theta, z}(D|\theta, z)] \approx \frac{1}{N} \sum_{i=1}^N f_{D|\theta, z_i}(D|\theta, z_i) \quad (\text{G.4})$$

$$z_i \sim f_z(z). \quad (\text{G.5})$$

For the next complication, note that what we actually need is the marginal *log*-posterior:

$$\ln f_{\theta|D}(\theta|D) \propto \ln f_{\theta}(\theta) + \ln \mathbb{E}_z [f_{D|\theta, z}(D|\theta, z)]. \quad (\text{G.6})$$

To avoid underflow when implementing this expression, care is needed when handling the last term:

$$\ln \mathbb{E}_z [f_{D|\theta,z}(D|\theta, z)] \approx \ln \left( \frac{1}{N} \sum_{i=1}^N f_{D|\theta,z_i}(D|\theta, z_i) \right) \quad (\text{G.7})$$

$$= \ln \left( \sum_{i=1}^N \exp \left( \ln f_{D|\theta,z_i}(D|\theta, z_i) \right) \right) - \ln N \quad (\text{G.8})$$

$$= \ln \left( \sum_{i=1}^N \exp \left( \ln f_{D|\theta,z_i}(D|\theta, z_i) - A \right) \exp(A) \right) - \ln N \quad (\text{G.9})$$

$$= \ln \left( \sum_{i=1}^N \exp \left( \ln f_{D|\theta,z_i}(D|\theta, z_i) - A \right) \right) + A - \ln N, \quad (\text{G.10})$$

where

$$A = \max_i \ln f_{D|\theta,z_i}(D|\theta, z_i). \quad (\text{G.11})$$

**G.2.1.4.2 Implementation of the marginal log-likelihood** The full likelihood is

$$f_{D|\theta,z}(D|\theta, z). \quad (\text{G.12})$$

For nested sampling we want the marginal likelihood

$$f_{D|\theta}(D|\theta) = \int f_{D|\theta,z}(D|\theta, z) f_z(z) dz = \mathbb{E} [f_{D|\theta,z}(D|\theta, z)]. \quad (\text{G.13})$$

Therefore, the exact same framework as above can be used, but the prior terms can be omitted.

**G.2.1.4.3 Exploiting the structure to apply Gaussian quadrature** Note that equation (G.2) involves an integral of the form

$$\int h(z) f_z(z) dz. \quad (\text{G.14})$$

If  $f_z(z)$  is taken to be Gaussian (a convenient choice in any case, as the time shift is typically taken to be known to lie within some interval around some value), this integral can be efficiently computed using Gauss-Hermite quadrature. In general,

there will be multiple nuisance parameters and sparse grids will be necessary to efficiently compute the multidimensional integral. In the particular case of parameters which only affect one diagnostic at a time, however, the problem permits the efficient use of a dense tensor product grid. Specifically, note that the likelihood can be written as a product over the signals  $\mathbf{s}_i$  from the individual diagnostics (each diagnostic having associated parameter(s)  $z_i$ ):

$$f_{D|\theta,z}(D|\theta, \mathbf{z}) = \prod_{i=1}^{N_D} f_{\mathbf{s}_i|\theta,z_i}(\mathbf{s}_i|\theta, z_i) \quad (\text{G.15})$$

$$\ln f_{D|\theta,z}(D|\theta, \mathbf{z}) = \sum_{i=1}^{N_D} \ln f_{\mathbf{s}_i|\theta,z_i}(\mathbf{s}_i|\theta, z_i). \quad (\text{G.16})$$

Therefore, each  $z_i$  can be evaluated on a 1D grid of quadrature points and the results can be combined to form the result on a dense tensor product grid with little additional overhead.

Consider the case of two diagnostics, each with an associated time shift/scaling parameter  $z_i$ . The likelihood is

$$f_{D|\theta,z}(D|\theta, \mathbf{z}) = f_{\mathbf{s}_1|\theta,z_1}(\mathbf{s}_1|\theta, z_1) f_{\mathbf{s}_2|\theta,z_2}(\mathbf{s}_2|\theta, z_2), \quad (\text{G.17})$$

and the marginalized likelihood is

$$f_{D|\theta}(D|\theta) = \iint f_{\mathbf{s}_1|\theta,z_1}(\mathbf{s}_1|\theta, z_1) f_{\mathbf{s}_2|\theta,z_2}(\mathbf{s}_2|\theta, z_2) f_{z_1}(z_1) f_{z_2}(z_2) dz_1 dz_2 \quad (\text{G.18})$$

$$= \int f_{\mathbf{s}_1|\theta,z_1}(\mathbf{s}_1|\theta, z_1) f_{z_1}(z_1) dz_1 \int f_{\mathbf{s}_2|\theta,z_2}(\mathbf{s}_2|\theta, z_2) f_{z_2}(z_2) dz_2. \quad (\text{G.19})$$

Applying Gauss-Hermite quadrature with  $n_Q$  points for each<sup>1</sup> time shift gives

$$\ln f_{D|\theta}(D|\theta) \approx \ln \left( \sum_{i=1}^{n_Q} f_{s_1|\theta, z_{1,i}}(s_1|\theta, z_{1,i}) w_i \sum_{i=1}^{n_Q} f_{s_2|\theta, z_{2,i}}(s_2|\theta, z_{2,i}) w_i \right) \quad (\text{G.20})$$

$$= \ln \left( \sum_{i=1}^{n_Q} \exp \left( \ln f_{s_1|\theta, z_{1,i}}(s_1|\theta, z_{1,i}) + \ln w_i \right) \right) \\ + \ln \left( \sum_{i=1}^{n_Q} \exp \left( \ln f_{s_2|\theta, z_{2,i}}(s_2|\theta, z_{2,i}) + \ln w_i \right) \right) \quad (\text{G.21})$$

$$= \ln \left( \sum_{i=1}^{n_Q} \exp \left( \ln f_{s_1|\theta, z_{1,i}}(s_1|\theta, z_{1,i}) + \ln w_i - A_1 \right) \right) + A_1 \\ + \ln \left( \sum_{i=1}^{n_Q} \exp \left( \ln f_{s_2|\theta, z_{2,i}}(s_2|\theta, z_{2,i}) + \ln w_i - A_2 \right) \right) + A_2, \quad (\text{G.22})$$

where  $w_i$  is the  $i^{\text{th}}$  quadrature weight and

$$A_1 = \max_i \ln f_{s_1|\theta, z_{1,i}}(s_1|\theta, z_{1,i}) + \ln w_i \quad (\text{G.23})$$

(and likewise for  $A_2$ ).

### G.2.1.5 Parameter estimation using optimizers

The `find_MAP_estimate()` method uses optimizers from either `NLOpt` or `scipy.optimize` to find the parameters which best describe the data. A global search is conducted by starting the optimizer at many points distributed throughout the parameter space. These points can either be distributed (pseudo)randomly, or a more systematic sampling can be obtained by using a quasi-random sequence such as the Sobol sequence [544]. The multiple starting points are evaluated in parallel with the help of the `_OptimizeEval` helper class. All of the optimizers available in `scipy.optimize` [293] and `NLOpt` [540] were tested, and it was found that the best convergence (both in terms of number of function evaluations needed for convergence and quality of the final optimum) was obtained using the `Sbplx` algorithm in `NLOpt`, which is based on the Subplex algorithm described in [545]. Furthermore, the

1. In general different numbers of quadrature points could be used for each diagnostic, but  $\max_i n_{Q,i}$  function calls are needed in the current implementation, so it is best to use the same number of points for each diagnostic.

performance of all of the optimizers was improved by mapping all of the parameters to the unit hypercube using the CDF of the prior distribution and performing the optimization in this transformed space.

#### **G.2.1.6 Parameter estimation using affine-invariant ensemble samplers**

The affine-invariant ensemble sampler [353] implemented in the `emcee` code [354] can be used for parameter estimation, though it was found that convergence was prohibitively slow. The `sample_posterior()` method creates the `emcee.Sampler` instance, using the `_ComputeLnProbWrapper` helper class to support parallelization. This method enables the use of the basic affine-invariant ensemble sampler [353], parallel tempering [450] or adaptive parallel tempering [451]. The `add_samples()` method adds samples to an existing `emcee.Sampler` instance.

#### **G.2.1.7 Parameter estimation and model selection using MULTINEST**

The MULTINEST [452–454] algorithm can be used for parameter estimation and model selection using the `PyMultiNest` [455] interface. The `run_multinest()` method calls `pymultinest.run()` to run the algorithm. The parameters are mapped from the unit hypercube sampling space to the parameter space defined by the prior distribution using the `multinest_prior()` method. When using local measurements, the log-likelihood is evaluated using the `multinest_ll_local()` method. When using line-integrated measurements, the log-likelihood is evaluated using the `multinest_ll_lineintegral()` method. The results can be visualized with the `process_multinest()` method.

#### **G.2.1.8 Other useful methods**

The `parallel_compute_cs_den()` method enables parallelized brute-force evaluation of the log-posterior over an arbitrary grid in parameter space.

The `compute_marginalized_DV()` method evaluates the posterior mean and standard deviation of the  $D$  and  $V$  profiles given samples of the parameters.

### **G.2.2 Enabling parallel execution: the `acquire_working_dir()` and `release_working_dir()` functions**

Running STRAHL in parallel is difficult because all of its input and output is accomplished using files. To solve this issue, `bayesimp` provides several methods to create one temporary working directory containing a copy of the STRAHL executable per thread. When using the native Python `multiprocessing` module for parallelization,

the `make_pool()` function creates a pool where each worker has its own temporary working directory. The `finalize_pool()` function removes these directories when the pool is no longer needed.

The `acquire_working_dir()` and `release_working_dir()` methods are used to control which thread uses which working directory when using MPI for parallelization. A list of available working directories is kept in the text file `working_dirs.txt`. When a thread requests a working directory, the directory's path is removed from `working_dirs.txt`. If the file is empty, a new working directory is cloned. When a working directory is released, its path is written back to `working_dirs.txt`.

### G.3 The lines module

The `lines` module implements the spectral model for SXR radiation described in [section 3.5.2.2](#). The main function provided for this purpose is `compute_SXR()`. This computes the strength of all of the lines above a given energy threshold, weights the emission using the filter transmission curve, and adds the contributions up to yield the total power reaching the detector. The strength of the lines themselves is computed with the `compute_lines()` function, which is a Python port of the original IDL `lines.pro` routines. For the He-like charge state, the program can either use the transitions available in `lines.pro` or the more complete set of collisionally-excited lines obtained from ADAS.

### G.4 Tutorial

Before running any of the following examples, set up a directory to hold all of your STRAHL runs. In that directory, place a copy of the master STRAHL directory. On the Alcator C-Mod server, copy `/home/markchil/codes/bayesimp_demo/strahl`. Each STRAHL analysis has an associated shot number (the actual shot number for experimental data, or the shot number the background profiles are pulled from for synthetic data) and a version number. The version number allows you to try multiple approaches on a given shot, and to generate multiple synthetic datasets from a given shot. For each shot/version pair, you should create a settings file. The settings file is simply a Python script which creates an instance of the `Run` class with the appropriate settings. The file can also contain the code needed to carry out various steps in the analysis, such as running MULTINEST. This section demonstrates three use cases: [section G.4.1](#) shows how to set up to analyze real experimental data, [section G.4.2](#) shows how to create a synthetic dataset with local measurements and [section G.4.3](#) shows how to create a synthetic dataset with line-

integrated measurements. Finally, [section G.4.4](#) shows how to run the analysis itself. All three examples use shot 1101014006, the main shot used for the impurity transport work in this thesis. The actual scripts used for this thesis are on GitHub at <https://github.com/markchil/thesiscode>, which give many examples of how to use the more advanced code options.

### G.4.1 Using real experimental data

**Code Listing G.1:** Settings file to analyze real experimental data.

---

```

1 #!/usr/bin/env python2.7
2 from __future__ import division
3
4 import bayesimp
5 import gptools
6
7 r = bayesimp.Run(
8     shot=1101014006,
9     version=1,
10    time_1=1.165,
11    time_2=1.265,
12    injections=[
13        bayesimp.Injection(0.965, 0.95, 1.02),
14        bayesimp.Injection(1.165, 1.155, 1.213),
15        bayesimp.Injection(1.365, 1.35, 1.440)
16    ],
17    Te_args=['--system', 'TS', 'GPC', 'GPC2'],
18    ne_args=['--system', 'TS'],
19    num_eig_D=3,
20    num_eig_V=3,
21    shift_prior=gptools.NormalJointPrior([-10e-3, 0.0, 0.0], [5e-3, 2e-3, 2e-3]),
22    source_file='/home/markchil/codes/bayesimp_demo/CaLx1101014006.dat'
23 )

```

---

1. Put the contents of [code listing G.1](#) into a Python script. For this example, the name used is “settings\_1101014006\_1.py.” This is the settings file which is used to control the program. All of the options are documented in the *bayesimp* source code, but the parameters used in this example are as follows:

**shot** The shot number.

**version** The version number of the analysis.

**time\_1** The start time for the STRAHL simulation.



**time\_2** The end time for the STRAHL simulation. The conditions halfway between `time_1` and `time_2` are used for the background profiles.

**injections** List of objects describing each of the injections to include. For each injection, the parameters are the time of the injection itself, the beginning of the time window to use data from and the end of the time window to use data from.

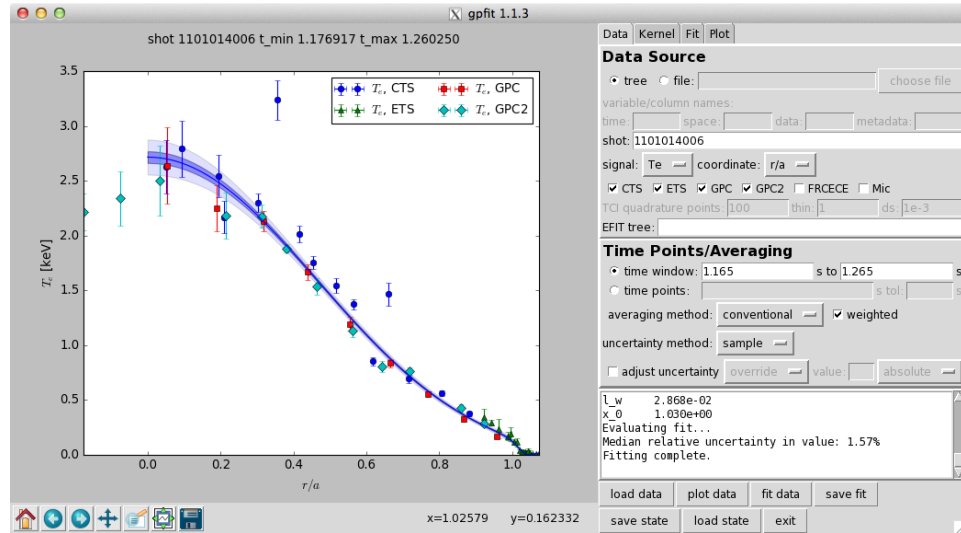
**Te\_args, ne\_args** Arguments passed to `gpfit` when fitting the  $T_e$  and  $n_e$  profiles. In the example, the arguments specify to use the data from Thomson scattering, GPC and GPC2 for  $T_e$  and just Thomson scattering for  $n_e$ . You can get documentation on all of the options by running “`gpfit -h`” from the shell.

**num\_eig\_D, num\_eig\_V** The number of free coefficients for the  $D$  and  $V$  profiles.

**shift\_prior** The prior distribution for the time shift nuisance parameters. The first number is for HIREX-SR, the second for the VUV spectrometers XEUS and LOWEUS, the third for XTOMO. The values used reflect the observation that the HIREX-SR timebase is off by about 10 ms while the other timebases are synchronized with the impurity injector timebase.

**source\_file** The source file to use. This analysis uses a source file which was already created.

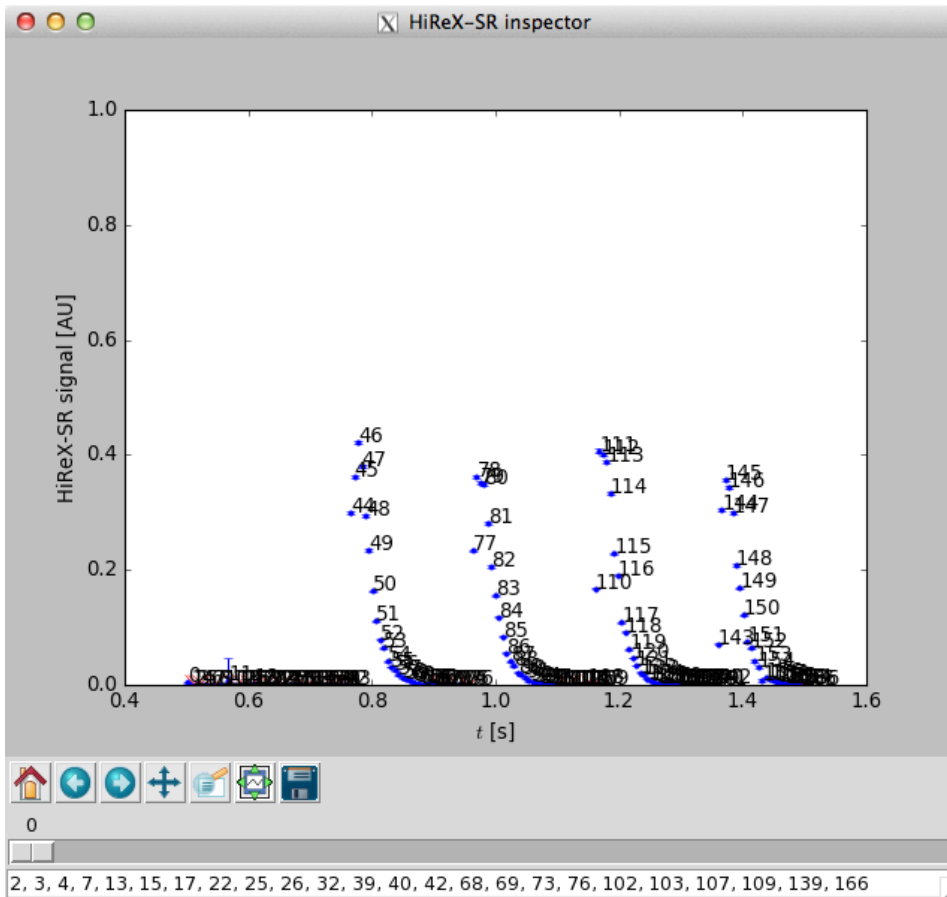
2. Launch IPython (type “`ipython`” in the shell), then run the settings file by typing “`%run settings_1101014006_1.py`”
3. A `gpfit` window will open to fit the  $T_e$  profile, as shown in [figure G.2](#). The time window and evaluation grid have been set automatically. Press the “load data” button to load the data. Then press the “plot data” button to apply your averaging settings and plot the data. Next, press the “fit data” button to fit the profile. Adjust the parameters, re-plot and re-fit the data as needed. When the fit is acceptable, press the “exit” button to continue on to the next step.
4. Another `gpfit` window will open to fit the  $n_e$  profile. As before, when the fit is acceptable, press the “exit” button to continue on to the next step.
5. A “HIREX-SR inspector” window will open to let you flag outlying data points in the HIREX-SR calcium signal, as shown in [figure G.3](#). Use the left and right arrow keys to move between channels, and type the indices of the points to drop into the text box as a comma-separated list. Press the return key to update the plot. Some points will have already been flagged by a simple set of



**Figure G.2:** gpfit window to fit the temperature profile, showing the completed fit.

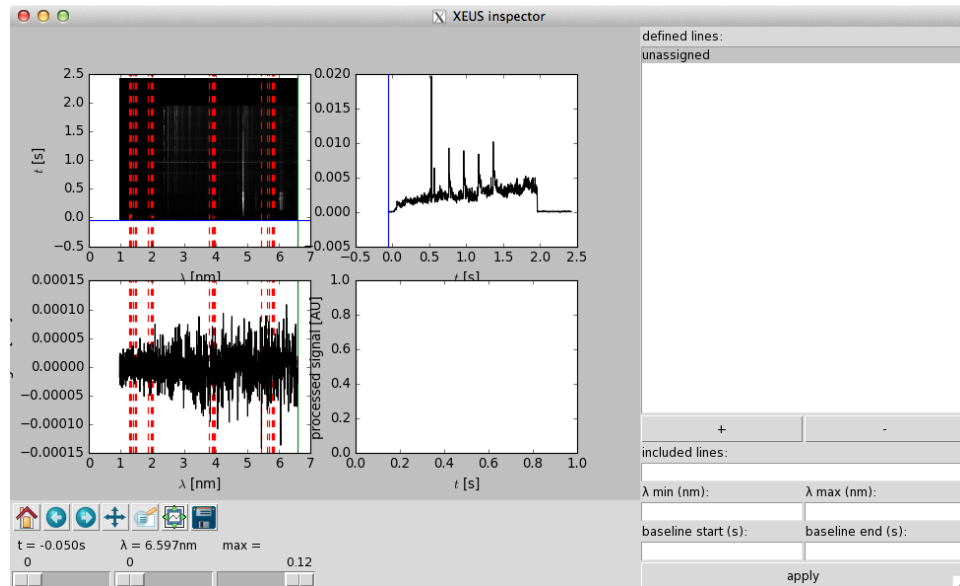
rules, you may put these back in by deleting their index from the text box if the automatic flagging was too aggressive. When this process is done, simply close the window to continue on to the next step.

6. The next step fits the time-series data to compute normalization factors, and will take a while.
7. When the HIREX-SR normalization factors have been computed, a “XEUS inspector” window will open to let you select which lines to use from the XEUS spectrometer, as shown in [figure G.4](#). The upper left plot shows the complete spectrum as a function of wavelength and time, the upper right plot shows the signal on a selected pixel as a function of time and the lower left plot shows the spectrum at a selected time. Clicking in any of the three figures will move the blue and green cursors and update what is shown; fine adjustments may be made using the sliders at the bottom of the window. The vertical lines indicate lines for which photon emission coefficients are available, but not all of these lines may be resolvable on a given shot. Most of the distinguishable Li-like calcium lines are in the 1.85 nm to 2.05 nm range. Each line has an associated index, shown at the bottom of the lower left plot. To include a line in the analysis, type its index in the “included lines” box and indicate the starting and ending wavelengths to include in the appropriate boxes. Multiple lines may be blurred together: simply indicate these by putting all of the indices



**Figure G.3:** HIREX-SR inspector window. The left and right arrow keys are used to switch between channels, and the text box allows outlying points to be flagged. Usually the points which are flagged automatically are in the low-signal regions before and after the injections which are not included in the normalized and combined data anyways. The objective is to filter out any points which are clearly the result of a bad fit to the underlying HIREX-SR spectrum.

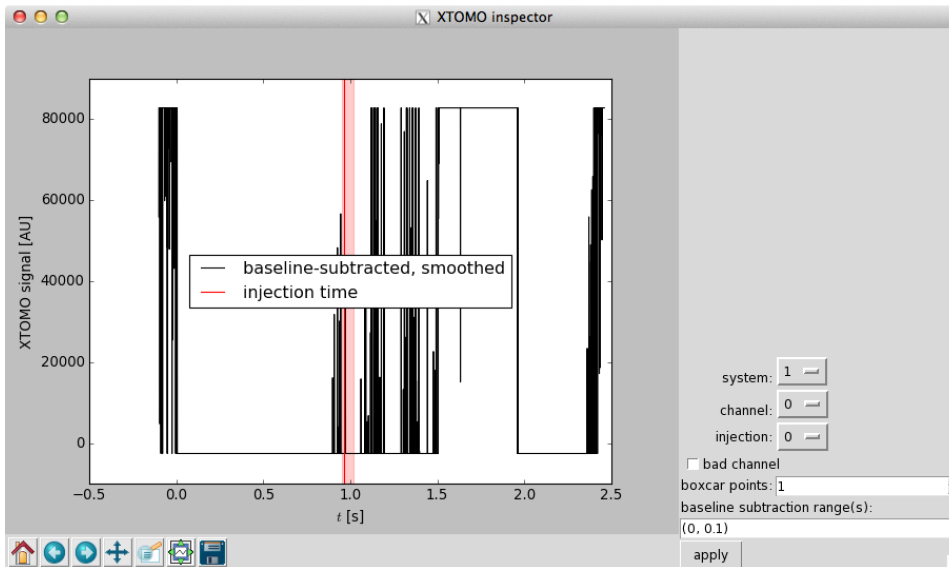
into the “included lines” box as a comma-separated list. In addition to the wavelength range, a time range unaffected by the impurity injections to use for baseline subtraction must be provided. To add another group of lines, simply press the “+” button. To remove a group of lines, select that line in the “defined lines” list and click the “-” button. To update the settings for a given group of lines, press the “apply” button. You can edit existing lines by selecting the relevant line in the list. When this process is done, simply close the window



**Figure G.4:** XEUS inspector window. The upper left plot shows the complete spectrum as a function of wavelength and time. The horizontal blue line is the time cursor, the vertical green line is the wavelength cursor. Clicking in any of the plots will update one or both of these cursors. The cursor locations can be fine-tuned with the sliders at the bottom. The red dashed lines indicate the wavelengths for which photon emission coefficients are available. The upper right plot shows the intensity as a function of time at the selected pixel. The lower left plot shows the spectrum at the selected time. When a spectral range has been selected, the time history of the intensity summed over that range will be shown in the lower right plot. The list on the right shows the combinations of lines which have been defined, and the boxes below are used to specify the line indices to include, the spectral range to use and the time window to use for baseline subtraction.

to continue on to the next step.

8. If LOWEUS data are available for the shot you are analyzing, a LOWEUS inspector window will open. The interface is the same as was used for XEUS.
9. An “xtomo inspector” window will open to let you pre-process the xtomo data, as shown in figure G.5. The “boxcar points” box lets you apply boxcar averaging to the figure to make it easier to see what is happening; note that this has no effect on the data used in the actual analysis, however. Cycle through channels with the left and right arrow keys, and flag any bad channels. Use the up and down arrow keys to cycle through the xtomo systems. Because the



**Figure G.5:** XTOMO inspector window. The channel the window opened to is bad, and needs to be flagged using the “bad channel” checkbox.

XTOMO diagnostics are so strongly affected by other impurities, you will usually need to select a different background subtraction range for each injection. You can select multiple ranges by defining a comma-separated list of ordered pairs. When you are done, simply close the window to continue on to the next step.

10. Another “HIREX-SR inspector” window will open to let you flag bad points in the argon data. As before, close the window when you are done.
11. The loading and pre-processing of the data is now complete. Now, running the settings file will simply restore the data from the data files created during this process.

### G.4.2 Creating local synthetic data

1. Put the contents of [code listing G.2](#) into a Python script. For this example, the name used is “settings\_1101014006\_2.py.” This is the settings file which is used to control the program. All of the options are documented in the `bayesimp` source code, but the parameters used in this example are as follows:

**shot** The shot number to get the magnetic equilibrium and background profiles from.

**Code Listing G.2:** Settings file to create local synthetic data.

---

```

1 #!/usr/bin/env python2.7
2 from __future__ import division
3
4 import bayesimp
5 import scipy
6
7 r = bayesimp.Run(
8     shot=1101014006,
9     version=2,
10    time_1=1.165,
11    time_2=1.265,
12    Te_args=['--system', 'TS', 'GPC', 'GPC2'],
13    ne_args=['--system', 'TS'],
14    num_eig_D=1,
15    num_eig_V=1,
16    source_file='/home/markchil/codes/bayesimp_demo/Caflx1101014006.dat',
17    params_true=[1.0, -10.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
18    num_eig_ne=3,
19    num_eig_Te=3,
20    local_cs=[18, None],
21    local_time_res=6e-3,
22    num_local_space=5,
23    local_synth_noise=5e-2,
24    use_line_integral=False,
25    use_local=True
26 )

```

---

**version** The version number of the analysis.

**time\_1** The start time for the STRAHL simulation.

**time\_2** The end time for the STRAHL simulation. The conditions halfway between `time_1` and `time_2` are used for the background profiles.

**Te\_args, ne\_args** Arguments passed to `gpfit` when fitting the  $T_e$  and  $n_e$  profiles. In the example, the arguments specify to use the data from Thomson scattering, GPC and GPC2 for  $T_e$  and just Thomson scattering for  $n_e$ . You can get documentation on all of the options by running “`gpfit -h`” from the shell.

**num\_eig\_D, num\_eig\_V** The number of free coefficients for the  $D$  and  $V$  profiles.

**source\_file** The source file to use. This analysis uses a source file which was already created.

**params\_true** The parameters to construct the synthetic data with. The order of the parameters is:

- (a) Coefficients for the  $D$  profile.
- (b) Coefficients for the  $V$  profile.
- (c) For profiles with more than one coefficient, there would then be internal knot locations for the  $D$  and  $V$  profiles. This example only has one coefficient per profile, however, so there are no internal knots.
- (d) Weights for the  $n_e$  profile.
- (e) Weights for the  $T_e$  profile.

Note that the local synthetic data do not use time shifts or scaling factors.

**num\_eig\_ne, num\_eig\_Te** The number of eigenvalues to keep for the  $n_e$  and  $T_e$  profiles.

**local\_cs** Charge states to construct local observations of. In this example, “18” means the He-like calcium density profile and “None” means the *total* calcium density profile (summed over all charge states).

**local\_time\_res** Time resolution for local data.

**num\_local\_space** Number of equally-spaced points to produce local data at.

**local\_synth\_noise** Relative noise level for the local data.

**use\_line\_integral** Flag to create line-integrated data.

**use\_local** Flag to create local data.

2. Launch IPython (type “ipython” in the shell), then run the settings file by typing “%run settings\_1101014006\_2.py.”
3. A `gpf` window will open to fit the  $T_e$  profile, as shown in [figure G.2](#). The time window and evaluation grid have been set automatically. Press the “load data” button to load the data. Then press the “plot data” button to apply your averaging settings and plot the data. Next, press the “fit data” button to fit the profile. Adjust the parameters, re-plot and re-fit the data as needed. When the fit is acceptable, press the “exit” button to continue on to the next step.
4. Another `gpf` window will open to fit the  $n_e$  profile. As before, when the fit is acceptable, press the “exit” button to continue on to the next step.
5. The program will now generate the synthetic data. Once the command prompt reappears, the generation of the synthetic data is now complete. In the future, running the settings file will simply restore the data from the data files created during this process.

### G.4.3 Creating line-integrated synthetic data

**Code Listing G.3:** Settings file to create line-integrated synthetic data.

---

```

1 #!/usr/bin/env python2.7
2 from __future__ import division
3
4 import bayesimp
5 import scipy
6
7 r = bayesimp.Run(
8     shot=1101014006,
9     version=3,
10    time_1=1.165,
11    time_2=1.265,
12    Te_args=['--system', 'TS', 'GPC', 'GPC2'],
13    ne_args=['--system', 'TS'],
14    num_eig_D=1,
15    num_eig_V=1,
16    source_file='/home/markchil/codes/bayesimp_demo/Caflx1101014006.dat',
17    params_true=scipy.concatenate((
18        [1.0, -10.0],
19        [1.0,] * 11,
20        [0.0,] * 3,
21        [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
22    )),
23    num_eig_ne=3,
24    num_eig_Te=3,
25    use_line_integral=True,
26    use_local=False,
27    hirex_time_res=6e-3,
28    vuv_time_res=2e-3,
29    synth_noises=[5e-2, 5e-2, 5e-2],
30 )

```

---

1. Put the contents of [code listing G.3](#) into a Python script. For this example, the name used is “settings\_1101014006\_3.py.” This is the settings file which is used to control the program. All of the options are documented in the bayesimp source code, but the parameters used in this example are as follows:

**shot** The shot number to get the magnetic equilibrium and background profiles from.

**version** The version number of the analysis.

**time\_1** The start time for the STRAHL simulation.



**time\_2** The end time for the STRAHL simulation. The conditions halfway between `time_1` and `time_2` are used for the background profiles.

**Te\_args, ne\_args** Arguments passed to `gpfit` when fitting the  $T_e$  and  $n_e$  profiles. In the example, the arguments specify to use the data from Thomson scattering, GPC and GPC2 for  $T_e$  and just Thomson scattering for  $n_e$ . You can get documentation on all of the options by running “`gpfit -h`” from the shell.

**num\_eig\_D, num\_eig\_V** The number of free coefficients for the  $D$  and  $V$  profiles.

**source\_file** The source file to use. This analysis uses a source file which was already created.

**params\_true** The parameters to construct the synthetic data with. The order of the parameters is:

- (a) Coefficients for the  $D$  profile.
- (b) Coefficients for the  $V$  profile.
- (c) For profiles with more than one coefficient, there would then be internal knot locations for the  $D$  and  $V$  profiles. This example only has one coefficient per profile, however, so there are no internal knots.
- (d) Scaling factors for each signal (one for HIREX-SR, one for each VUV line, one for each XTOMO system).
- (e) Time shifts for each diagnostic (one for HIREX-SR, one for the VUV spectrometers, one for the XTOMO systems).
- (f) Weights for the  $n_e$  profile.
- (g) Weights for the  $T_e$  profile.

**num\_eig\_ne, num\_eig\_Te** The number of eigenvalues to keep for the  $n_e$  and  $T_e$  profiles.

**use\_line\_integral** Flag to create line-integrated data.

**use\_local** Flag to create local data.

**hirex\_time\_res, vuv\_time\_res** Time resolution for HIREX-SR and the VUV spectrometers.

**synth\_noises** Relative noise levels for HIREX-SR, the VUV spectrometers and the SXR arrays.

2. Launch IPython (type “`ipython`” in the shell), then run the settings file by typing “`%run settings_1101014006_3.py`.”

3. A `gpfit` window will open to fit the  $T_e$  profile, as shown in [figure G.2](#). The time window and evaluation grid have been set automatically. Press the “load data” button to load the data. Then press the “plot data” button to apply your averaging settings and plot the data. Next, press the “fit data” button to fit the profile. Adjust the parameters, re-plot and re-fit the data as needed. When the fit is acceptable, press the “exit” button to continue on to the next step.
4. Another `gpfit` window will open to fit the  $n_e$  profile. As before, when the fit is acceptable, press the “exit” button to continue on to the next step.
5. The program will now generate the synthetic data. Once the command prompt reappears, the generation of the synthetic data is now complete. In the future, running the settings file will simply restore the data from the data files created during this process.

## G.4.4 Performing the inference

### G.4.4.1 Finding the MAP estimate

1. Add the contents of [code listing G.4](#) to the end of your settings file.

---

**Code Listing G.4:** Code to find the MAP estimate.

---

```

1 pool = bayesimp.make_pool()
2 out = r.find_MAP_estimate(pool=pool)
3 bayesimp.finalize_pool(pool)

```

---

2. Run the settings file as before: type “`%run settings_1101014006_1.py`” in IPython.
3. The code will now make a pool with as many processors as are available, then will run the optimizer starting from random locations. By default, two random starting locations are tried for each worker in the pool.

### G.4.4.2 Sampling with MULTINEST

1. Add the contents of [code listing G.5](#) to end of your settings file. The keyword “`local=False`” tells it to use line-integrated data.

---

**Code Listing G.5:** Code to run MULTINEST.

---

```

1 r.run_multinest(local=False)

```

---

2. To run this on a single processor (very slowly), run the settings file as before: type “%run settings\_1101014006\_1.py” in IPython.
3. To run this in parallel, from the shell type “mpiexec -np 24 python2.7 settings\_1101014006\_1.py” (but replace 24 with the number of threads to use).

# H

## Software configuration

**Table H.1:** Software configuration in use at the time this thesis was written.

Name	Reference(s)	Version
Python	[546, 547]	2.7.11
IPython	[548]	4.1.1
NumPy	[529]	1.10.4
SciPy	[293]	0.17.0
matplotlib	[549]	1.5.1
emcee	[354]	2.1.0
mpmath	[531]	0.19
SymPy	[478]	1.0
pandas	[536, 537]	0.18.0
Cython	[532]	0.23.4
MULTINEST	[452–454]	3.10
PyMultiNest	[455]	1.6
scikit-image	[435]	0.12.3
NLOpt	[540]	2.4.2
numdifftools	[541]	0.9.16
sobol	[550]	0.9