



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Bayesian Inference of the 1D Electron Density Profile within the WEST Tokamak using Interferometry

by

Daniel Jordan

A masters thesis presented to the University of Padova
in fulfillment of the thesis requirement for the degree of
Physics of Data

The research was carried out at Ghent University with onsite supervision from
Geert Verdoolaege - Associate Professor
Hao Wu - PhD Student

Remote supervision from the university of Padova was provided by
Lidia Piron - Assistant Professor

2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The abstract should briefly highlight the importance of the research and present its key findings.

The electron density profile is a key parameter that affects the performance and stability of tokamak plasmas.

Acknowledgements

A big thank you to Hao Wu, Jeffry de Rycke, and Yangyang Zhang for their enlightening conversation and mentorship. I thank the rest of the team at infusion UGent, Geert Verdoolaege, Sven Van Loo, Jerome Alhage, Leonardo Caputo, and Joseph Hall for accepting me into their research group and including me in their research discussions. An extra thank you to Geert Verdoolaege for giving me this enriching opportunity.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
List of Figures	vii
1 Introduction	1
2 Background Theory of Bayesian Techniques and WEST Interferometry	2
2.1 The Tokamak	3
2.2 NICE	5
2.3 Bayesian Inference and the Simple Regression Problem	8
2.4 Interferometry and Polarimetry	16
2.5 Bayesian Inference for Interferometry	17
2.6 Chapter Summary	22
3 Methodology and Results	24
4 Conclusion	25
5 Future Investigation	26

References	27
APPENDICES	30
A Deriving the Closed Form Posterior Expressions	31
B Deriving the Marginal Likelihood and Loss Function Expression	34
C Complete Set of Distributions and Expressions for Reference	39
C.1 Gaussian Process Regression for Interferometry, Discluding Artificial Observations	39
C.2 Gaussian Process Regression for Interferometry, Including Artificial Observations	41
Glossary	44

List of Figures

2.1	A tokamak and relevant magnetic fields that create the helical particle trajectory [8].	3
2.2	Magnetic flux surfaces [12].	5
2.3	Electron density profile inferred by NICE for an instance in time within the WEST tokamak.	7
2.4	Illustrating how many Gaussians can model a curved line and its uncertainty.	9
2.5	A visualisation of the simple regression problem and the various distributions involved in the Bayesian inference solution.	12
2.6	The geometry of interfero-polarimetry lasers at WEST [6].	16
2.7	An example mesh grid to aid visualisation of the triangular mesh grid interpolation used in the response matrix construction.	18
2.8	Hyperbolic tangent smooth step function for length scale, equation 2.26. Used to capture the drop at the edge of H-mode plasmas [2].	21

Chapter 1

Introduction

Chapter 2

Background Theory of Bayesian Techniques and WEST Interferometry

This chapter aims to equip the reader with the necessary background theory required to reproduce this work and to understand the origin of the inferred electron density profiles presented in the results. It first describes a tokamak fusion device and some relevant physics concepts behind its function. It then describes in a high level manner the inference carried out by Blaise Faugeras and team with their code known as [Newton direct and Inverse Computation for Equilibrium \(NICE\)](#) [4]. After, the chapter outlines Bayesian inference and how a specific implementation can be used to solve a simple regression problem. Interferometry is introduced in enough detail to understand how the electron density profile could be inferred from its data. The Bayesian inference method introduced for the simple regression problem is then altered to allow this inference. Various options for advanced alterations are also explained here and explored in the results section.

2.1 The Tokamak

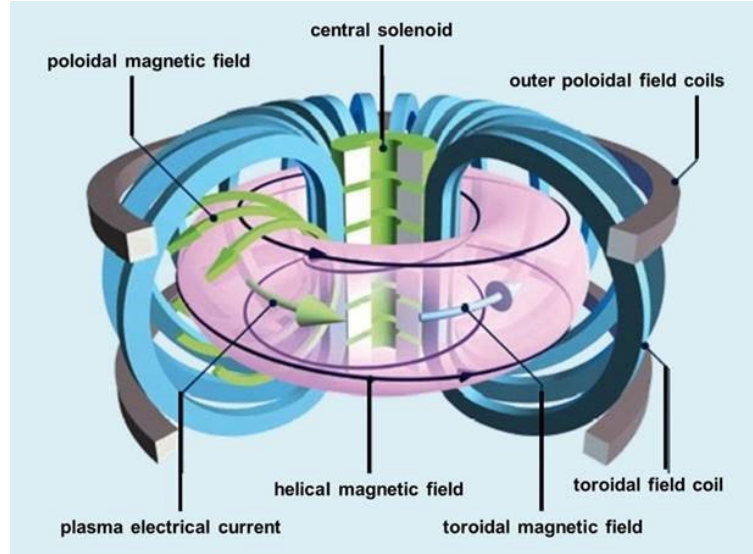


Figure 2.1: A tokamak and relevant magnetic fields that create the helical particle trajectory [8].

Tokamak is a class of fusion devices whose name comes from the abbreviation of a Russian phrase which means “toroidal chamber with magnetic coils”. It consists of a doughnut shaped vacuum chamber surrounded by powerful magnets that aim to confine a high temperature plasma that would otherwise vaporise the chamber. The plasma pressure and temperature are fundamental parameters in the context of nuclear fusion because they dictate the conditions required to overcome the electrostatic repulsion between positively charged atomic nuclei and bring them close enough for the strong nuclear force to initiate fusion reactions. In the core of stars like our Sun, the immense pressure and temperature generated by the gravitational collapse create the conditions where hydrogen nuclei (protons) can overcome their natural repulsion and fuse into helium, releasing a tremendous amount of energy in the process. To initiate fusion, hydrogen must be heated to temperatures in the range of tens of millions of degrees Celsius. In a tokamak, this is mainly accomplished with ohmic heating via a driving plasma current and neutral gas injection. This involves accelerating hydrogen ions to high speeds with electric fields and neutralising them the instant before they enter the chamber. The resulting plasma attains the required temperature, allowing nuclei to collide with sufficient energy for fusion reactions

to occur. Figure 2.1 shows the position of various magnetic field coils within the tokamak. The toroidal magnetic field exerts an inward force on the plasma thus raising its pressure. High pressure is required to increase the frequency of collisions so that the energy output can exceed the large heating energy input. The central solenoid induces a current in the plasma which produces the majority of the poloidal magnetic field. This field is essential for confinement but it also plays a key role in plasma stability. The outer poloidal field coils can be controlled in real time to help mitigate instabilities. A real time inference of the electron density profile would assist in identifying instabilities and informing the algorithm that drives the control coils to mitigate them. In addition to high temperature and pressure, the tokamak design seeks to maximize the confinement time of the plasma. This is essential to allow a sufficient number of fusion reactions to occur before the plasma cools down or loses its stability. The magnetic fields in a tokamak are carefully optimized to prevent rapid plasma loss and minimize heat loss through various mechanisms, including turbulent transport. The shape of the density profile has a large effect on the confinement time.

The combination of the toroidal and poloidal fields shown in figure 2.1 creates a helical magnetic field within the plasma. Electrons and ions are accelerated in opposite toroidal directions by the central solenoid yet both follow a trajectory along the magnetic field lines. This is because a charged particle moving across a magnetic field succumbs to a force perpendicular to its motion. This causes them to gyrate around the magnetic field lines and confines them to follow the magnetic field lines. This is an oversimplification and in reality there are drift forces that cause the particles to deviate from following the magnetic field lines exactly. Collisions also cause deviations. A detailed description of particle motion within a magnetic field is not needed for this thesis. It is enough to know that the particles in general follow the helical path of the magnetic field lines with a small gyration around the field line. In many models used for data analysis the assumption that particles follow the magnetic field lines is used, including within this thesis.

The magnetic field lines are confined to magnetic flux surfaces, figure 2.2. The toroidal and poloidal flux is constant on magnetic flux surfaces, there is 0 flux across magnetic flux surfaces. Since we assume that the particles follow the magnetic field lines which are strictly bound to these surfaces, we also assume that the density is constant on these surfaces. This allows the density of the entire cross-section to be expressed with a 1D profile as a function of normalised radius ρ for example, see NICE's profile, figure 2.3. Where ρ is 0 at the magnetic axis and 1 at the plasma boundary. The magnetic axis is very center point of the core and defined as where the poloidal magnetic flux is minimum and the plasma boundary is the last closed flux surface. Particles past the plasma boundary are no longer bound and may interact with the plasma wall. The existence of nested magnetic flux

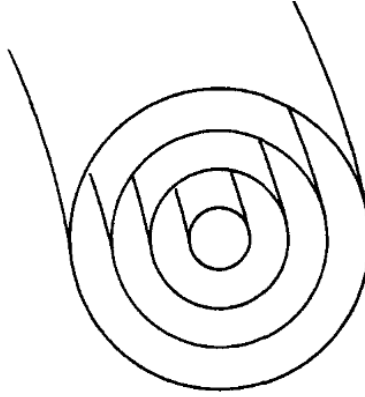


Figure 2.2: Magnetic flux surfaces [12].

surfaces shown in figure 2.2 rely on the ideal [magnetohydrodynamics \(MHD\)](#) assumptions. Experiments frequently discover magnetic islands which discredits the assumption of nested flux surfaces. The electron density profiles inferred by [NICE](#) and this work make the nested flux surface assumption, although for many applications such as real time control, a highly accurate inference is often not required.

2.2 NICE

[NICE](#) is an equilibrium reconstruction code that is routinely deployed for the [Tungsten \(W\) Environment in Steady-state Tokamak \(WEST\)](#) tokamak. It is relevant because it computes an inference of the electron density profile that is available for comparison to the profile inferred in this work, although [NICE's](#) main objective is to infer the shape and position of the magnetic flux surfaces. [NICE](#) uses magnetic diagnostics. At [WEST](#) these include 421 pickup coils, 36 flux loops and 12 Rogowski coils [10]. Magnetic diagnostics provide the majority of the information. [NICE](#) also uses interferometry, polarimetry, motional stark effect and pressure measurements. Equation 2.14 and 2.15 further in the chapter, show how interferometry and polarimetry together can provide information about the poloidal magnetic field, which directly affects the magnetic flux and thus magnetic flux surfaces. [NICE](#) performs the inference by minimising a cost function. The cost function determines how well a physical state of the system matches the data received. A state is a specific position and shape of the magnetic flux surfaces and electron density profile. This requires a forward model. The forward model takes a state of the system and attempts to compute the signals that would be received by error free diagnostics if that state was the

ground truth. The forward model is a simplified mathematical representation of the measurement process and can never be 100% accurate. This introduces errors in the inference that need to be accounted for. The signals from the forward model can be compared to the actual signals received by the diagnostics to compute the cost function. By minimising the cost function the state that best matches the data is found. [NICE](#) uses [Sequential Quadratic Programming \(SQP\)](#) as the minimisation algorithm. The optimal state of the system is then stored in the [Integrated Modeling and Analysis Suite \(IMAS\)](#) database for [WEST](#). This includes the 1D electron density profile used as a comparison for the profile inferred in this work. [NICE](#) also imposes regularisation terms on their cost function. These penalise the cost function when state properties have features that disagree with prior knowledge. This includes smoothness. We expect the magnetic flux surfaces and electron density profile, to be continuous and smooth. A state inputted into the cost function that is not smooth triggers the regularisation term which causes the cost function to be larger. Minimising the cost function now also leads to smooth magnetic flux surfaces and electron density profile. This leads to a difficult question, how smooth should it be? They also have a regularisation term to penalise the cost function if the electron density profile is far from 0 at the last closed flux surface or plasma boundary. It is prior knowledge that the electron density is near 0 at the plasma boundary. How close to 0, and how strong should the regularisation be is still an open question. This work's approach has direct analogues to these regularisation terms. As explained later in more detail the length scale controls smoothness and an artificial observation ensures the density is close to zero at the plasma boundary. [Figure 2.3](#) shows an example of a [NICE](#) inferred electron density profile. It is modelled with a cubic spline function. It is the parameters of the cubic spline that are inputted into the cost function. The errors are calculated using a sensitivity method. In short, the error is deemed larger for the electron density of a particular normalised radius if a large change in the density leads to a small change in the cost function. In this case, we cannot be certain what density is better because many lead to a similarly low cost function and thus match the data similarly well. To include some more details, the [SQP](#) minimisation algorithm computes the hessian of the cost function for minimisation, but this hessian can also be used to measure the sensitivity and thus the errors. The diagonal of the hessian contains the second differential of the cost function for each input parameter. This describes the curvature of the cost function in the direction of each parameter. A smaller curvature means a smaller sensitivity and thus a larger error.

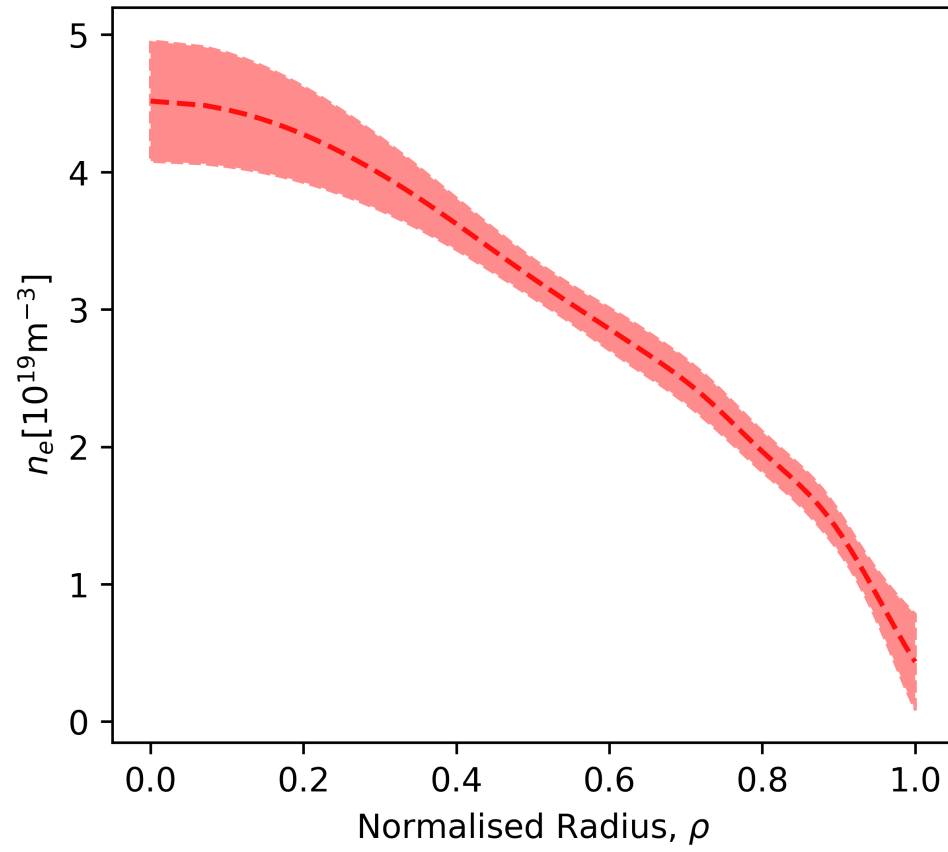


Figure 2.3: Electron density profile inferred by NICE for an instance in time within the WEST tokamak.

2.3 Bayesian Inference and the Simple Regression Problem

This work aims to use Bayesian inference to obtain the electron density profile. Bayesian inference will be introduced generally and then it will be used to define a specific implementation applied to a simple regression problem. The method introduced will later be extended to solve the problem of inferring the electron density profile with interferometry data. The predictive Bayes' theorem for a physical quantity of interest q is expressed as,

$$P(q|D, I) = \frac{P(D|q, I)P(q|I)}{P(D|I)}, \quad (2.1)$$

the posterior $P(q|D, I)$ is the probability density distribution of q given the measured data D and some prior information I . The q that maximises the posterior is the most probable value of q given the data and prior information. The uncertainty of q can also be obtained from the posterior. The likelihood $P(D|q, I)$ is the probability density function that expresses the probability of the measured data given a fixed value of q and the prior information. The likelihood is described by the experimental error for the data collection. The prior $P(q|I)$ contains information assumed about q before the data is taken. The marginal likelihood or evidence $P(D|I)$ is simply the probability of the data given the prior information only. For posterior computation, the marginal likelihood serves as a normalisation factor. Normalisation is often carried out with other means to simplify the posterior computation. Although the marginal likelihood can be used to tune hyperparameters. For example, the degree or strength of prior information is uncertain and by finding the strength that maximises the marginal likelihood we find the prior that matches the data the best. Maximising the marginal likelihood to tune the hyperparameters also aids in avoiding over-fitting, as the trade off between model complexity and data-fit is automatic via Occam's razor principle [9]. The marginal likelihood method is powerful although it is important to remember that it is not perfect and does not guarantee to find the hyperparameters that lead to the most accurate posterior.

The version of Bayes' theorem for a simple regression problem is,

$$P(\vec{y}|\vec{d}, \vec{\epsilon}, \theta) = \frac{P(\vec{d}|\vec{y}, \vec{\epsilon})P(\vec{y}|\theta)}{P(\vec{y}|\vec{\epsilon}, \theta)}, \quad (2.2)$$

where \vec{y} contains the values of a curve at regular x values. The goal is to find the most likely \vec{y} given the data and prior information. \vec{d} contains curve measurements at known

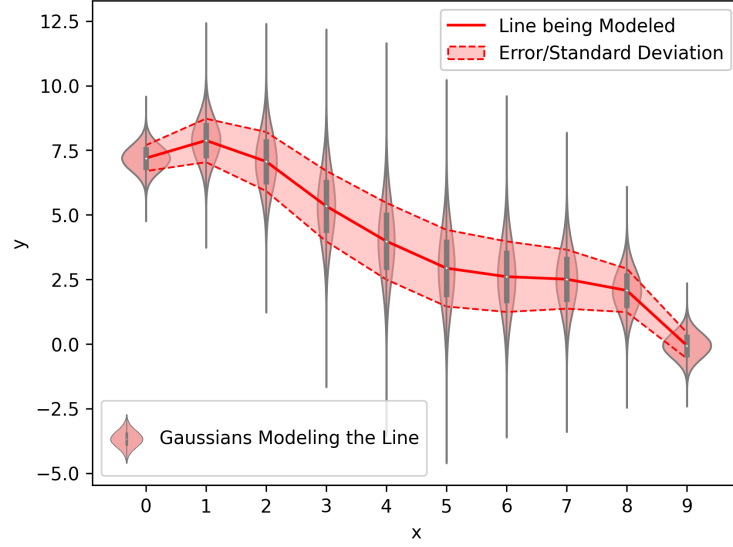


Figure 2.4: Illustrating how many Gaussians can model a curved line and its uncertainty.

x values with some experimental errors $\vec{\epsilon}$. θ is a set of parameters related to the prior form, explained more later. The likelihood and prior are going to be clearly defined as multivariate Gaussians giving a multivariate Gaussian posterior which needs to somehow model the curve \vec{y} . Figure 2.4 illustrates how a multivariate gaussian can model a curve. The functional form of a multivariate Gaussian is,

$$\mathcal{N}(\vec{y}, \vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^{\frac{n}{2}} |\Sigma|}} \exp \left[-\frac{1}{2} (\vec{y} - \vec{\mu})^T \Sigma^{-1} (\vec{y} - \vec{\mu}) \right], \quad (2.3)$$

the mean vector $\vec{\mu}$ holds the y values of the curve at regular intervals along the x axis. The diagonal of the covariance matrix holds the standard deviations of each Gaussian within the multivariate. These represent the errors of the curve. Figure 2.4 shows 10 Gaussians with each mean connected by a straight line. In practice many Gaussians are used in a small space so that even a linear interpolation appears as a smooth curve. In our simple regression problem 101 Gaussians are used, thus \vec{y} has a length of 101. The posterior that models the most likely curve given the data and prior information can be expressed as,

$$\mathcal{N}(\vec{y}, \vec{\mu}_{post}, \Sigma_{post}), \quad (2.4)$$

where μ_{post} has a length of 101, the same as the unknown y values.

In order to compute μ_{post} and Σ_{post} we must define the likelihood and prior. If m measurements are taken the likelihood is defined as,

$$P(\vec{d}|\vec{y}, \vec{\epsilon}) = \mathcal{N}(\vec{d}, \vec{\mu}_{li} = R\vec{y}, \Sigma_{li}), \Sigma_{li} = \vec{\epsilon}I = \begin{bmatrix} \epsilon_1 & 0 & \dots & 0 \\ 0 & \epsilon_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \epsilon_m \end{bmatrix}, \quad (2.5)$$

where R is the response matrix. Given some curve \vec{y} to be true, $R\vec{y}$ is a vector that has the same length as \vec{d} and contains the values of the curve \vec{y} at the same x values that the data was collected at. The response matrix is an error free model of the measurement process. In the likelihood of figure 2.5 the blue line is an example of a given \vec{y} and if this was the ground truth and we took an error free measurement at the same x points as our original data then we would get the points indicated by the mean of each gaussian. These points are computed with $R\vec{y}$ and provide the mean vector of the likelihood. The likelihood represents the probability of getting the black data points given the blue line \vec{y} is the ground truth. Regression is inherently an inverse problem and the response matrix is a forward model.

The prior is also a multi variate Gaussian and thus can be said to follow a gaussian process. Regression carried out with Bayesian inference and a Gaussian process prior is often referred to as Gaussian Process Regression. Although, the method being introduced is more general than the typical implimentation of Gaussian Process Regression. This is so that it can easily be extended later to allow regression in situations where the data resides in a different space. To avoid confusion with the typical version of Gaussian Process Regression the term is not used for this implimentation. For an introduction to typical Gaussian Process Regression I suggest the text book Gaussain Processes for Machine Learning [11]. The prior can be defined as,

$$\mathcal{N}(\vec{y}, \vec{\mu}_{pr} = \vec{0}, K), K_{ij} = k(y_i, y_j) = \sigma^2 \exp \left[\frac{(y_i - y_j)^2}{2l^2} \right], \quad (2.6)$$

where $\vec{0}$ is a vector of 101 zeros, the same length as \vec{y} . The zero vector is a commonly used ‘non informative’ prior mean vector. The covariance matrix K is constructed using the kernel $k(x_i, x_j)$. The main role of the amplitude, σ , in the kernel is to set the prior strength. A high amplitude means the inference has a low prior strength and the resultant curve can be far from the prior mean $\mu_{pr} = \vec{0}$. See the prior in figure 2.5, the amplitude σ is the standard deviation of these gaussians shown. For visualisation purposes only 5 prior

Gaussians are shown in figure 2.5, yet in reality there is 101, the same number as there are unknown y values. The length scale, l sets the strength of the correlation between the Gaussians. A low length scale means that only Gaussians close in x are highly correlated. Gaussians further in x would have a low correlation, meaning they can have a very different mean value. A low length scale allows the fitted curve to have more complexity similar to a high order polynomial and can lead to overfitting. A high length scale limits the fit's ability to curve sharply leading to a simple model, similar to a low order polynomial, leading to underfitting. A very high length scale leads to an almost linear fit. This prior is far from perfect. For instance, it is often known that the inferred values must be positive, for example, you cannot have a negative electron density. Since the prior mean vector is set to $\vec{0}$, a negative value is as likely to be inferred as a positive value. Since it is Gaussian, values close to 0 are more likely to be inferred than values far from 0. To mitigate this a high amplitude can be used to lower the prior strength and allow the data in the likelihood to have more influence on the posterior result. The kernel $k(x_i, x_j)$ in equation 2.6 is known as the exponential square kernel. It is a very commonly used kernel but far from the only choice. The single value of the length scale prevents the inference from having long smooth regions with few features followed by regions of high variability. This can be an issue when inferring H-mode tokamak plasmas that have a sharp drop-off in density at the plasma edge. For these situations, a non-stationary kernel can be used that allows the length scale to be a function of x which can then allow for posteriors of varying complexity. Regardless of the kernel used, deciding the optimal values of its parameters for a problem is not obvious. A common solution is to use the marginal likelihood of equation 2.1. The parameters that maximise the marginal likelihood also maximise the probability of the data being measured. The marginal likelihood method is also known for automatically deploying Occam's razor principle which finds a balance between closely fitting the data and having a simple model that accounts for the data's errors to have a more accurate inference [9] [11]. Essentially maximising the marginal likelihood avoids overfitting. The maximisation can be done with gradient based methods. Although this method is powerful it does not guarantee to produce parameters that lead to the most accurate fit. To get a more accurate fit, Bayesian sampling techniques can be used, although this is more computationally expensive.

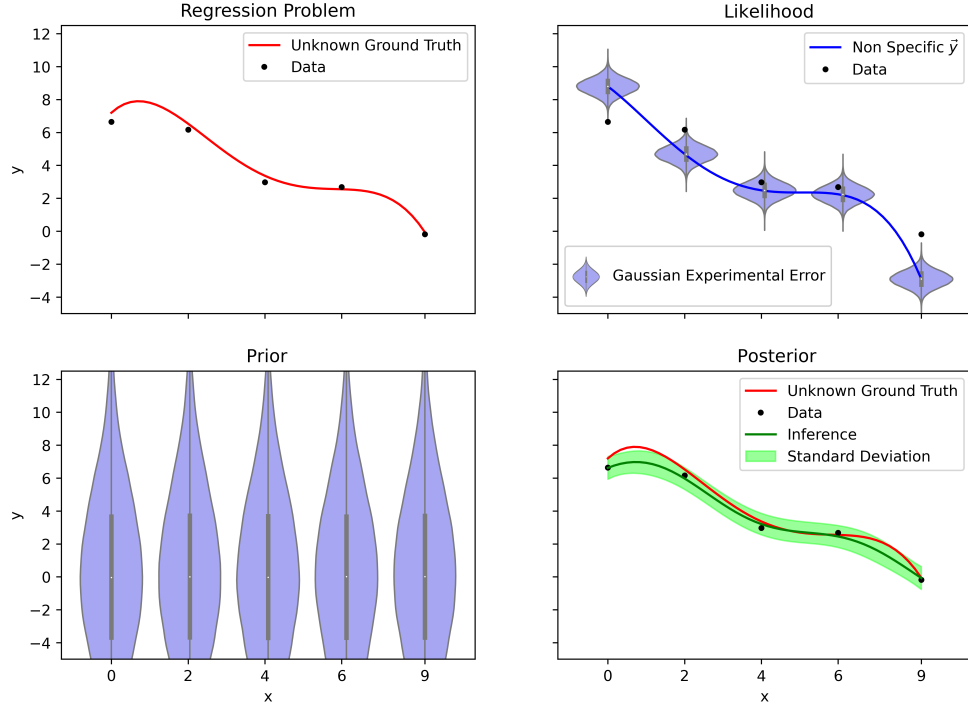


Figure 2.5: A visualisation of the simple regression problem and the various distributions involved in the Bayesian inference solution.

Figure 2.5 shows the simple regression problem, it then tries to show the Gaussians that make up the likelihood and prior although it is not perfect. We assume the observations are independent giving the likelihood a diagonal covariance matrix. So the multiple Gaussians that make up the likelihood are not dependent on each other, thus representing them individually provides all the information in the likelihood. The prior has a more complex covariance matrix K , and figure 2.5 does not have a complete representation of the priors form. Figure 2.5 also shows the 101 gaussians that make up the posterior by plotting the regularly spaced $\vec{\mu}_{post}$ values in green with a green shadow showing the standard deviations of each gaussian that are found in the diagonal of Σ_{post} . The posterior mean vector and covariance matrix can be computed with the closed form expressions,

$$\vec{\mu}_{post} = \vec{\mu}_{pr} + (K^{-1} + R^T \Sigma_{li}^{-1} R)^{-1} R^T \Sigma_{li}^{-1} (\vec{d} - R \vec{\mu}_{pr}), \quad (2.7)$$

$$\Sigma_{post} = (R^T \Sigma_{li}^{-1} R + K^{-1})^{-1}, \quad (2.8)$$

which are derived in appendix A. The main steps include multiplying the functional forms of the prior and likelihood, ignoring all scaling factors, simplifying until they form a single unnormalised multivariate Gaussian and then comparing this with the posterior. There is still the issue of deciding the values of the hyperparameters to place in K and the experimental error to place within Σ_{li} . Often the experimental error is not precisely known and can also be treated as a hyperparameter. In order to select the hyperparameters we can perform a separate hyperparameter Bayesian inference,

$$P(\theta, \vec{\epsilon} | \vec{d}) = \frac{P(\vec{d} | \theta, \vec{\epsilon}) P(\theta, \vec{\epsilon})}{P(\vec{d})}, \quad (2.9)$$

Where the most probable hyperparameters given the same data can be found by maximising the hyperparameter posterior, $P(\theta, \vec{\epsilon} | \vec{d})$. Notice the likelihood $P(\vec{d} | \vec{\epsilon})$ is exactly the marginal likelihood from the predictive Bayesian formula 2.1. The prior $P(\vec{d})$ needs to accurately represent our prior knowledge of the hyperparameters. By maximising the numerator of Bayes theorem the posterior is also maximised. The marginal likelihood from the hyperparameter Bayes theorem $P(\vec{d})$, can safely be ignored as it is a normalisation constant. All further mentions of marginal likelihood refers to the predictive distributions marginal likelihood $P(\vec{d} | \theta, \vec{\epsilon})$; as is the usual terminology for this procedure. It can be computed by integrating the numerator of the predictive Bayesian over the quantity of interest:

$$\begin{aligned} P(\vec{d} | \vec{\epsilon}, \theta) &= \int P(\vec{d} | \vec{y}, \vec{\epsilon}) P(\vec{y} | \theta) d\vec{y} \\ &= \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\Sigma_{li} + RKR^\top|}} \exp \left[-\frac{1}{2} (\vec{d} - R\vec{\mu}_{pr})^\top (\Sigma_{li} + RKR^\top)^{-1} (\vec{d} - R\vec{\mu}_{pr}) \right]. \end{aligned} \quad (2.10)$$

The values of the marginal likelihood can become very large and troublesome to compute with standard 64-bit float precision. For this reason, the logarithm is computed. It is the convention when performing optimisation to define a loss function to be minimised, thus the negative log marginal likelihood is used. Scaling constants do not affect the minimum value and can be ignored. The negative log marginal likelihood used as a loss function for hyper-parameters is then,

$$loss(\vec{\epsilon}, \theta) = \ln(|\Sigma_{li} + RKR^\top|) + (\vec{d} - R\vec{\mu}_{pr})^\top (\Sigma_{li} + RKR^\top)^{-1} (\vec{d} - R\vec{\mu}_{pr}), \quad (2.11)$$

the full derivation of this expression can be found in appendix B. To include a uniform prior for the hyperparameters $P(\theta)$, the loss function can be programmed to return infinity (or a very large number) when the proposed hyperparameters are outside their bounds. Although some minimisation algorithms can avoid proposing values outside of determined bounds.

An alternative solution is to perform a full bayesian analysis where the hyperparameters are marginalised out of the predictive posterior. This allows the predictive posterior to become independant of the hyperparameters and allows for a more robust and flexible inference. Marginalisation of the posterior can be written as,

$$P(\vec{y}|\vec{d}) = \int \int P(\vec{y}, \theta, \vec{\epsilon}|\vec{d}) d\theta d\vec{\epsilon}, \quad (2.12)$$

and with conditional probability rules,

$$P(\vec{y}|\vec{d}, \vec{\epsilon}, \theta) = \int \int P(\vec{y}|\theta, \vec{\epsilon}, \vec{d}) P(\theta, \vec{\epsilon}|\vec{d}) d\theta d\vec{\epsilon}. \quad (2.13)$$

Notice that $P(\vec{y}|\theta, \vec{\epsilon})$ is our original predictive posterior and $P(\theta, \vec{\epsilon}|\vec{d})$ is the hyperparameter posterior from earlier. Sampling y from the joint distribution $P(\vec{y}|\theta, \vec{\epsilon})P(\theta, \vec{\epsilon}|\vec{d})$ is equivalent to sampling from the posterior $P(\vec{y}|\vec{d})$ that does not depend on the hyperparameters. To accomplish this a first step is to use [Markov chain Monte Carlo \(MCMC\)](#) sampling techniques to sample θ and $\vec{\epsilon}$ from the hyperparameter posterior. This only requires the ability to compute the log of the hyperparameter likelihood and prior given a proposed set of hyperparameters. The prior is defined by us and so must be computable and the likelihood is exactly the predictive marginal likelihood and has an analytical form given previously in equation 2.10. This thesis uses the python package EMCEE which involves initialising a group of 'walkers' at random parameter positions. They then explore the space of the provided posterior distribution using a proposal function and the Metropolis Hastings criteria. The proposal function suggests a new position in the hyperparameter space based on the current position. If the new position satisfies the Metropolis Hastings criteria it is accepted and the values of the hyperparameters at the new position are collected as a sample. After some number of iterations the samples can closely represent samples from the true posterior. These samples of θ and $\vec{\epsilon}$ can be used to sample \vec{y} from the predictive posterior $P(\vec{y}|\theta, \vec{\epsilon})$; which is a multivariate Gaussian of mean and covariance given by equations 2.7 and 2.8. Sampling from a known multivariate Gaussian can be done with more simple sampling techniques that have zero possibility of

sample rejection: [MCMC](#) sampling is not required. Remember that each sampled \vec{y} is a curve where the vector items correspond to y values of the curve at regularly spaced x intervals; as shown in figure 2.5. Thus the most likely value of the curve is the mean or median of the posterior samples at each x value. For uncertainty, the standard deviation or certain quantiles can be computed.

An additional caveat of [MCMC](#) sampling is that a chain of samples from a walker are autocorrelated, yet a true sample from the posterior would not depend on the current position of a walker. To reduce autocorrelation the chains are often thinned. Thinning by a degree of three means to remove every third sample. This reduces autocorrelation and the samples more accurately represent samples from the posterior. Over thinning limits the number of samples obtainable with the given computation power and time limits, leading to a loss of precision of the inference. To aid in deciding the thinning degree one can compute the [Integrated Autocorrelation Time \(IAT\)](#). This is a measure of how long it takes for a Markov chain to forget its initial state and become uncorrelated. Thinning by a degree of the [IAT](#) will ensure there is almost no correlation between the samples but this is extreme and often means very long run times for a few samples that produce much less precise results than if more samples are kept.

Another way to reduce autocorrelation is to tune the [MCMC](#) algorithms hyperparameters.

For [emcee](#) the main hyperparameter is the 'moves', which is their term for proposal function. It is also possible to pass multiple moves and weights when sampling. It will then randomly select a move in proportion to the weights. In theory this should make the next sample more random and less correlated to the previous. Each move also has a single parameter. These can be trialed with smaller sample numbers in an attempt to minimise the autocorrelation. Optuna is a hyperparameter tuning framework for python that proposes trials in an attempt to minimise the objective function. By default it uses the tree-structured Parzen estimator algorithm which is also a Bayesian method. In this thesis four of the moves and their parameters are trialed with Optuna in an attempt to reduce the autocorrelation. The lowest trialed autocorrelation time for each move is used to determine parameter value of each move. It also determines the weights, allowing for the best performing move to be used more frequently.

2.4 Interferometry and Polarimetry

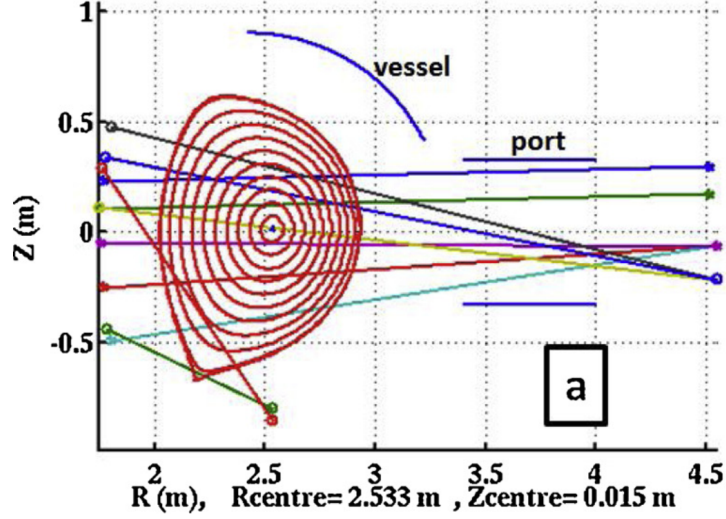


Figure 2.6: The geometry of interfero-polarimetry lasers at WEST [6].

Interferometry and Polarimetry are techniques that use the interference of electromagnetic waves to measure the properties of a medium. Interfero-polarimetry lasers within a tokamak penetrate the plasma at various angles. The geometry at WEST is shown in figure 2.6. Each laser is split into two beams: one that passes through the plasma and one that bypasses it. The two beams are then recombined and detected by a receiver. The phase difference between the two beams depends on the difference in the optical path length, which is affected by the electron density along the line of sight. Interferometry measures the phase difference; allowing one to calculate the line integrated electron density of the plasma $\int n_e dl$,

$$\Delta\phi = \frac{\lambda e^2}{4\pi\epsilon_0 m_e c^2} \int n_e dl [7]. \quad (2.14)$$

The laser wavelength λ , is combined with other common physical constants to ascertain the constant of proportionality. WEST has stored the line integrated electron density as raw interferometry data in the IMAS database. This is the data that will be used for this work. Although there is not enough information to completely and accurately reconstruct the electron density profile a best guess given the data can be inferred.

Polarimetry measures the Faraday rotation angle of the lasers. The linearly polarised lasers experience a rotation as the circularly polarised components travel through the plasma at different speeds. This is due to the small gyration of the electrons around the magnetic field. The Faraday rotation angle is proportional to the line integrated density of $n_e B_{||}$ along the line of sight of the lasers,

$$\theta_F = \frac{\lambda^2 e^3}{8\pi^2 c^3 \epsilon_0 m_e^2} \int n_e B_{||} dl \text{ [7]}, \quad (2.15)$$

where $B_{||}$ is the magnetic field strength parallel to the line of sight. Polarimetry has information about electron density and this work could be extended to become a Bayesian integrated analysis which includes this information in the inference. Only interferometry information is used in this thesis. Polarimetry can be used in combination with interferometry to gain information about the poloidal magnetic field and this is why [NICE](#) uses it to determine the position of the magnetic flux surfaces.

2.5 Bayesian Inference for Interferometry

To infer the electron density profile with interferometry, the previously defined regression process is altered. \vec{y} becomes \vec{n}_e , the $\vec{0}$ prior mean can remain the same. The amplitude σ and length-scale l can be re-optimised by maximising the marginal likelihood. The data is now in a different space and thus is the likelihood. The response matrix R must be created so that it will transform a profile \vec{n}_e into what would be measured by an error free version of the [WEST](#) interferometry system given \vec{n}_e is the true profile. The result of $R\vec{n}_e$ is a vector the same length as the data \vec{d} where each element corresponds to a different interferometry laser or channel.

The response matrix computation can be summarised in a few steps. [NICE](#) provides the magnetic flux at a set of grid points on the tokamak poloidal cross-section. It also provides the flux at a set of flux surfaces. The normalised radius ρ of each flux surface is known. A simple 1D interpolation can be used to determine the normalised radius at each grid point. Then using \vec{n}_e another 1D interpolation can be done to determine the electron density at each grid point. After the density at any point along a laser's line of sight $n_e(l_i)$, can be computed using triangular mesh interpolation. The density at the golden cross in figure 2.7 can be computed as a weighted sum of the density at the three nearest grid points $\{g_1, g_2, g_3\}$ that form the golden triangle,

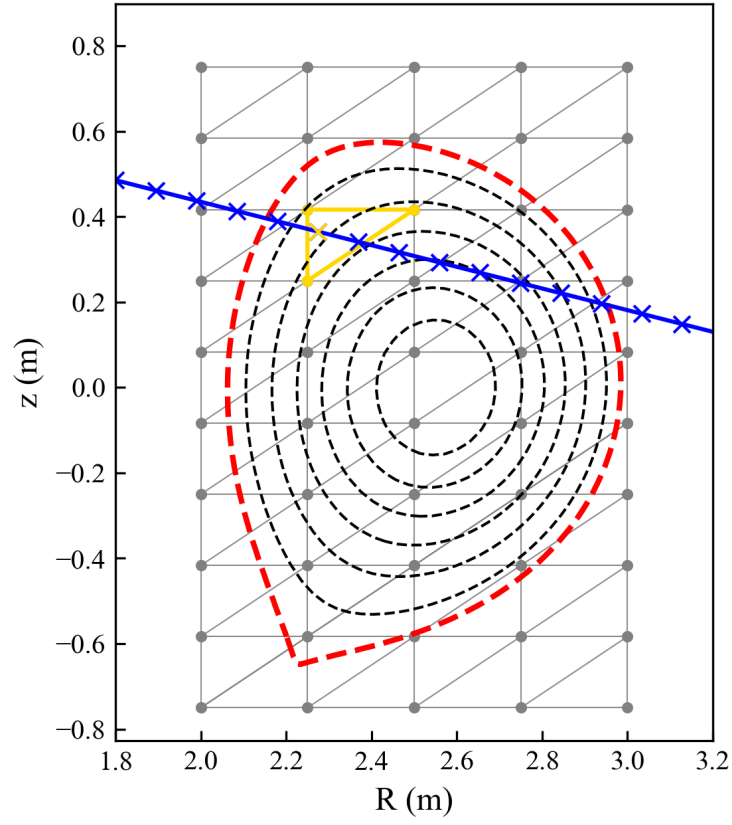


Figure 2.7: An example mesh grid to aid visualisation of the triangular mesh grid interpolation used in the response matrix construction.

$$n_e(l_i) = \lambda_1 n_e(g_1) + \lambda_2 n_e(g_2) + \lambda_3 n_e(g_3), \quad (2.16)$$

where λ values can be computed using the $(R_1, z_1), (R_2, z_2), (R_3, z_3)$ coordinates of the 3 known density points and the point of interest (R, z) ,

$$\lambda_1 = \frac{(z_2 - z_3)(R - R_3) + (R_3 - R_2)(z - z_3)}{(z_2 - z_3)(R_1 - R_3) + (R_3 - R_2)(z_1 - z_3)}, \quad (2.17)$$

$$\lambda_2 = \frac{(z_3 - z_1)(R - R_3) + (R_1 - R_3)(z - z_3)}{(z_2 - z_3)(R_1 - R_3) + (R_3 - R_2)(z_1 - z_3)}, \quad (2.18)$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2. \quad (2.19)$$

These λ values are known as the barycentric coordinates of the point of interest. The line integrated density can be approximated as a sum of electron densities at many points along the line of sight, l_i , times the width of their separation Δl ,

$$\int n_e dl \approx \sum_i n_e(l_i) \Delta l. \quad (2.20)$$

The contribution $w(g_i)$ of each grid point g_i is a sum of all the mesh interpolation coefficients λ_j used on that point,

$$\int n_e dl \approx \Delta l \sum_i w(g_i) n_e(g_i), \quad w(g_i) = \sum_j \lambda_j. \quad (2.21)$$

Each point can be associated with the nearest flux surface f_i equally spaced in ρ . This way the contribution $w(f_i)$ of each flux surface is a sum of the contribution at each of its associated grid points g_j ,

$$\int n_e dl \approx \Delta l \sum_i w(f_i) n_e(f_i), \quad f = \sum_j g_j. \quad (2.22)$$

All of these steps equate to a simple re-ordering of the original summation [2.20](#) to extract the contribution of each flux surface on the final integrated density value. Equation [2.22](#) can be computed using a vector product,

$$\int n_e dl \approx \Delta l \vec{w}^\top \vec{n}_e. \quad (2.23)$$

The contribution vector applies to one line of sight. The computation for all lines of sight can be performed by placing the $\Delta l \vec{w}$ vector for each line of sight as a row in the response matrix R . Thus, a vector of line integrated densities for the likelihood can be created,

$$\vec{\mu}_{li} = R \vec{n}_e. \quad (2.24)$$

This response matrix R can then be used in the closed form expressions 2.7 and 2.8, to perform a 1D electron density profile inference.

Some further alterations to the inference method can be made to further increase reliability. These include altering the kernel and adding artificial observations to include prior knowledge. The kernel can be changed to a non-stationary kernel,

$$K_{ij} = k(\rho_i, \rho_j) = \sigma^2 \left(\frac{2l(\rho_i)l(\rho_j)}{l(\rho_i)^2 + l(\rho_j)^2} \right)^{1/2} \exp \left(\frac{(\rho_i - \rho_j)^2}{l(\rho_i)^2 + l(\rho_j)^2} \right), \quad (2.25)$$

this allows the length scale to change as a function of ρ . The length scale controls smoothness, model complexity and curvature. If these are free to change for different regions of the plasma then there is a greater range of possibilities for the final inference. Chilenski used a hyperbolic tangent function,

$$l(\rho) = \frac{l_{core} + l_{edge}}{2} + \frac{l_{core} - l_{edge}}{2} \tanh \left(\frac{\rho - \rho_{stepcenter}}{\rho_{stepwidth}} \right) [2], \quad (2.26)$$

to form a smooth step down from a high length scale at the core to low at the edge, see figure 2.8. The extra freedom at the edge allows the inference to accommodate for a large sudden drop in electron density, which is a common feature for H-mode plasmas. H-mode plasmas are known to have a longer confinement time and thus better fusion performance. WEST does not operate in H-mode, although this method is tested with synthetic data from a simple H-mode simulation.

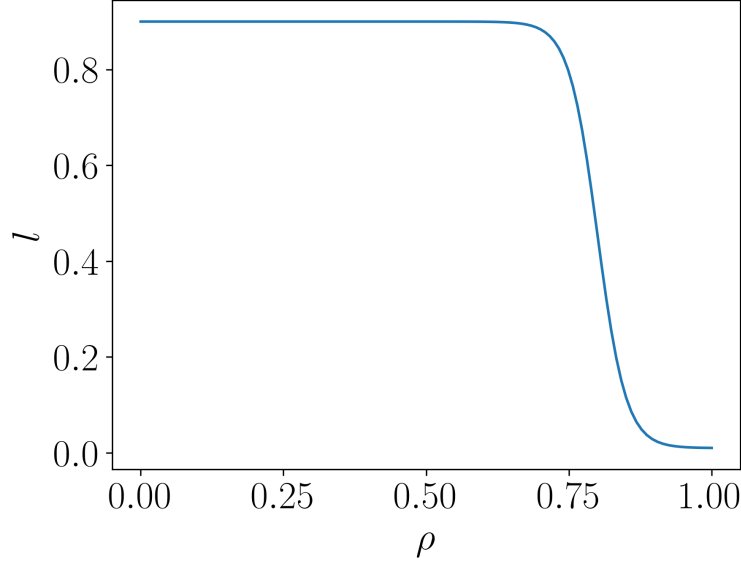


Figure 2.8: Hyperbolic tangent smooth step function for length scale, equation 2.26. Used to capture the drop at the edge of H-mode plasmas [2].

Traditionally prior information should be included in the prior distribution. However, in practice precision errors can make this difficult. The closed form expressions for $\vec{\mu}_{post}$ and Σ_{post} involve inversions of both the likelihood and prior covariance matrices, although the likelihood covariance matrix Σ_{li} is diagonal and so is certainly positive definite and inversion does not suffer from precision errors. This is not true for the prior covariance matrix K , it is difficult to define a prior covariance matrix that includes all the prior information, remains non positive definite and does not suffer from precision errors. For these reasons it is often more convenient to place prior information into the likelihood in the form of artificial observations. This method was also adopted by Chilenski [2]. The density is known to be close to 0 at the plasma boundary, ($\rho = 1$). It is also known that the density profile is smooth and symmetric meaning the gradient of the profile on the magnetic axis must be close to 0. This information can be included in the data, \vec{d} , with an artificial experimental error determining the strength of the information included in \vec{e} . Other parts of the method need to be altered to accommodate the new information. The vector to be inferred \vec{a} is not only n_e but also includes $n_e(\rho = 1)$ and $n_e'(\rho = 0)$ concatenated onto the end. This allows the response matrix alteration to be simple,

$$R^{alt} = \begin{bmatrix} R_{m \times n} & O_{m \times 2} \\ O_{2 \times n} & I_{2 \times 2} \end{bmatrix} = \begin{bmatrix} & & 0 & 0 \\ & R_{m \times n} & \vdots & \vdots \\ & & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}, \quad (2.27)$$

where n is the number of unknown electron density values in \vec{n}_e and m is the number of interferometry lasers. The prior covariance matrix must also be altered. The covariance between a gradient and non-gradient data point is simply the differential of the covariance over the gradient data point. For two gradient data points, it is a differential over each point.

$$K'_{ij} = k'(\rho_i, \rho_j) = \frac{\partial k'(\rho'_i, \rho_j)}{\partial \rho'_i} [2], \quad (2.28)$$

$$K''_{ij} = k''(\rho'_i, \rho'_j) = \frac{\partial k''(\rho'_i, \rho'_j)}{\partial \rho'_i \partial \rho'_j} [2]. \quad (2.29)$$

In this notation ρ' indicates the position of a gradient data point. The alternate kernel is then,

$$K^{alt} = \begin{bmatrix} K & K' \\ K'^\top & K'' \end{bmatrix}. \quad (2.30)$$

The necessary adaptations to our defined Bayesian regression method to accomodate interferometry data have been described. For reference, the various final distributions and expressions after the adaptations are fully shown in appendix C.

2.6 Chapter Summary

The electron density profile is important as it plays a key role in determining the energy confinement time and informing real time control systems. With the assumption of magnetic flux surfaces, one can express it as a 1D profile. [NICE](#) is an equilibrium reconstruction code that also infers the electron density profile that can be used as a comparison in this thesis. Bayesian inference with multivariate gaussians describing the various distributions

can be applied to interferometry data to infer the electron density profile. A non-stationary kernel can be used to allow the inference to have a model complexity that varies with ρ . Hyperparameters can be tuned by minimising the negative log marginal likelihood. Prior information can be easily included in the likelihood with artificial observations. In the results section, these methods will be deployed on synthetic data. The inference performance can be determined by how closely it fits the ground truth profile. They will also be deployed on real [WEST](#) data and the results will be compared to that obtained by [NICE](#).

Chapter 3

Methodology and Results

Chapter 4

Conclusion

The conclusion should have a short summary of each chapter highlighting the main parts of a story from data to inference to insights. Lead into future investigations.

Chapter 5

Future Investigation

References

- [1] Jiahong Chen, Zhibin Wang, Tianbo Wang, Yonghao Yang, Hao Wu, Yonggao Li, Guoliang Xiao, Geert Verdoolaege, Didier Mazon, Zengchen Yang, Min Jiang, Zhongbing Shi, Jian Chen, Yi Yv, Wulyu Zhong, and Min Xu. Integrated data analysis on the electron density profile of hl-2a with the bayesian probability inference method. *Plasma physics and controlled fusion*, 65(5):55027–, 2023.
- [2] 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. *Nuclear fusion*, 55(2):23012–20, 2015.
- [3] A. J. Creely, M. J. Greenwald, S. B. Ballinger, D. Brunner, J. Canik, J. Doody, T. Fülöp, D. T. Garnier, R. Granetz, T. K. Gray, C. Holland, N. T. Howard, J. W. Hughes, J. H. Irby, V. A. Izzo, G. J. Kramer, A. Q. Kuang, B. LaBombard, Y. Lin, B. Lipschultz, N. C. Logan, J. D. Lore, E. S. Marmar, K. Montes, R. T. Mumgaard, C. Paz-Soldan, C. Rea, M. L. Reinke, P. Rodriguez-Fernandez, K. Särkimäki, F. Sciortino, S. D. Scott, A. Snicker, P. B. Snyder, B. N. Sorbom, R. Sweeney, R. A. Tinguely, E. A. Tolman, M. Umansky, O. Vallhagen, J. Varje, D. G. Whyte, J. C. Wright, S. J. Wukitch, and J. Zhu. Overview of the sparcs tokamak. *Journal of plasma physics*, 86(5), 2020.
- [4] Blaise Faugeras. An overview of the numerical methods for tokamak plasma equilibrium computation implemented in the nice code. *Fusion engineering and design*, 160:112020–, 2020.
- [5] R. Fischer and A. Dinklage. Integrated data analysis of fusion diagnostics by means of the bayesian probability theory. *Review of Scientific Instruments*, 75(10):4237–4239, 2004.

- [6] C. Gil, G. Colledani, M. Domenes, D. Volpe, A. Berne, F. Faisse, C. Guillon, J. Morales, P. Moreau, and B. Santraine. Renewal of the interfero-polarimeter diagnostic for west. *Fusion engineering and design*, 140:81–91, 2019.
- [7] Ian H. Hutchinson and Ian H. Hutchinson. *Principles of plasma diagnostics / I. H. Hutchinson*. Cambridge University press, Cambridge etc, c1987.
- [8] S. Li, H. Jiang, Z. Ren, and C. Xu. Optimal tracking for a divergent-type parabolic pde system in current profile control. *Abstract and applied analysis*, 2014:1–8, 2014.
- [9] David J.C. MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.
- [10] P. Moreau, A. Le-Luyer, P. Spuig, P. Malard, F. Saint-Laurent, J. F. Artaud, J. Morales, B. Faugeras, H. Heumann, B. Cantone, M. Moreau, C. Brun, R. Nouailletas, E. Nardon, B. Santraine, A. Berne, P. Kumari, and S. Belsare. The new magnetic diagnostics in the west tokamak. *Review of scientific instruments*, 89(10):10J109–10J109, 2018.
- [11] Carl Edward. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006.
- [12] John Wesson and John Wesson. *Tokamaks / John Wesson ; with contributions from D.J. Campbell et al.* The Oxford engineering science series. Clarendon Press, Oxford, 2. ed edition, 1997.
- [13] A. B. Zylstra, A. L. Kritcher, O. A. Hurricane, D. A. Callahan, J. E. Ralph, D. T. Casey, A. Pak, O. L. Landen, B. Bachmann, K. L. Baker, L. Berzak Hopkins, S. D. Bhandarkar, J. Biener, R. M. Bionta, N. W. Birge, T. Braun, T. M. Briggs, P. M. Celliers, H. Chen, C. Choate, D. S. Clark, L. Divol, T. Döppner, D. Fittinghoff, M. J. Edwards, M. Gatu Johnson, N. Gharibyan, S. Haan, K. D. Hahn, E. Hartouni, D. E. Hinkel, D. D. Ho, M. Hohenberger, J. P. Holder, H. Huang, N. Izumi, J. Jeet, O. Jones, S. M. Kerr, S. F. Khan, H. Geppert Kleinrath, V. Geppert Kleinrath, C. Kong, K. M. Lamb, S. Le Pape, N. C. Lemos, J. D. Lindl, B. J. MacGowan, A. J. Mackinnon, A. G. MacPhee, E. V. Marley, K. Meaney, M. Millot, A. S. Moore, K. Newman, J.-M. G. Di Nicola, A. Nikroo, R. Nora, P. K. Patel, N. G. Rice, M. S. Rubery, J. Sater, D. J. Schlossberg, S. M. Sepke, K. Sequoia, S. J. Shin, M. Stadermann, S. Stoupin, D. J. Strozzi, C. A. Thomas, R. Tommasini, C. Trosseille, E. R. Tubman, P. L. Volegov, C. R. Weber, C. Wild, D. T. Woods, S. T. Yang, and C. V. Young. Experimental

achievement and signatures of ignition at the national ignition facility. *Physical review. E*, 106(2):025202–025202, 2022.

APPENDICES

/

Appendix A

Deriving the Closed Form Posterior Expressions

The inference begins with Bayes theorem,

$$P(\vec{y}|\vec{d}, \vec{\epsilon}, \theta) = \frac{P(\vec{d}|\vec{y}, \vec{\epsilon})P(\vec{y}|\theta)}{P(\vec{d}|\vec{\epsilon}, \theta)}, \quad (\text{A.1})$$

where the likelihood can be written as,

$$P(\vec{d}|\vec{y}, \vec{\epsilon}) = \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\Sigma_{li}|}} \exp \left[-\frac{1}{2} (\vec{d} - R\vec{y})^\top \Sigma_{li}^{-1} (\vec{d} - R\vec{y}) \right], \quad \Sigma_{li} = \vec{\epsilon}I, \quad (\text{A.2})$$

the prior as,

$$P(\vec{y}|\theta) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|K|}} \exp \left[-\frac{1}{2} (\vec{y} - \vec{\mu}_{pr})^\top K^{-1} (\vec{y} - \vec{\mu}_{pr}) \right], \quad (\text{A.3})$$
$$\theta \rightarrow \{\sigma, l\}, \quad K_{ij} = k(\rho_i, \rho_j) = \sigma^2 \exp \left[\frac{(\rho_i - \rho_j)^2}{2l^2} \right],$$

and the posterior as,

$$P(\vec{y}|\vec{d}, \vec{\epsilon}, \theta) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|\Sigma_{post}|}} \exp \left[-\frac{1}{2} (\vec{y} - \mu_{post})^\top \Sigma_{post}^{-1} (\vec{y} - \vec{\mu}_{post}) \right]. \quad (\text{A.4})$$

To derive $\vec{\mu}_{post}$ and Σ_{post} the likelihood and prior are multiplied together and re-arranged. Only first and second order \vec{y} terms are kept as the constants do not affect the shape of the multivariate Gaussian and thus do not affect $\vec{\mu}_{post}$ or Σ_{post} . Then using the completing the square formula for matrices they can be combined into a single multivariate Gaussian. By comparing with the posterior we find the closed form expressions for $\vec{\mu}_{post}$ and Σ_{post} . When the distributions are multiplied together the exponential powers are summed,

$$-\frac{1}{2} \left[(\vec{d} - R\vec{y})^\top \Sigma_{li}^{-1} (\vec{d} - R\vec{y}) + (\vec{y} - \vec{\mu}_{pr})^\top K^{-1} (\vec{y} - \vec{\mu}_{pr}) \right],$$

ignoring the $-\frac{1}{2}$ for now and multiplying it out gets,

$$\begin{aligned} & \left(\vec{d}^\top \Sigma_{li}^{-1} \vec{d} - \vec{d}^\top \Sigma_{li}^{-1} R\vec{y} - (R\vec{y})^\top \Sigma_{li}^{-1} \vec{d} + (R\vec{y})^\top \Sigma_{li}^{-1} R\vec{y} \right), \\ & + \left(\vec{y}^\top K^{-1} \vec{y} - \vec{y}^\top K^{-1} \vec{\mu}_{pr} - \vec{\mu}_{pr}^\top K^{-1} \vec{y} + \vec{\mu}_{pr}^\top K^{-1} \vec{\mu}_{pr} \right), \end{aligned}$$

focusing on the 1st order terms and remembering that the transpose of a scalar is itself and the transpose of a symmetric matrix (e.g. Σ_{li}) is itself, it can be shown that the first order terms equate to

$$-\vec{d}^\top \Sigma_{li}^{-1} R\vec{y} - (R\vec{y})^\top \Sigma_{li}^{-1} \vec{d} - \vec{y}^\top K^{-1} \vec{\mu}_{pr} - \vec{\mu}_{pr}^\top K^{-1} \vec{y} = -2\vec{y}^\top (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr}) = -2\vec{y}^\top \vec{b}$$

in which a substitution was made to ease the use of the competing square formula,

$$\vec{b} = R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr}$$

switching the focus to the 2nd order terms,

$$(R\vec{y})^\top \Sigma_{li}^{-1} R\vec{y} + \vec{y}^\top K^{-1} \vec{y} = \vec{y}^\top (R^\top \Sigma_{li}^{-1} R + K^{-1}) \vec{y} = \vec{y}^\top M \vec{y},$$

in which a substitution was made to ease the use of the completing square formula,

$$M = (R^\top \Sigma_{li}^{-1} R + K^{-1})$$

ignoring 0 order terms that do not affect the shape, the original exponential power takes the form,

$$-\frac{1}{2} \left[\vec{y}^\top M \vec{y} - \vec{y}^\top \vec{b} \right],$$

by completing the squares we obtain

$$\vec{y}^\top M \vec{y} - \vec{y}^\top \vec{b} = (\vec{y} - M^{-1} \vec{b})^\top M (\vec{y} - M^{-1} \vec{b}) - \vec{b}^\top M^{-1} \vec{b}.$$

We can ignore $\vec{b}^\top M^{-1} \vec{b}$ as it doesn't affect the shape of the Gaussian. Finally, for the posterior we have

$$P(\vec{y} | \vec{d}, \vec{\epsilon}, \theta) \propto \exp \left[-\frac{1}{2} (\vec{y} - \vec{\mu}_{post})^\top \Sigma_{post}^{-1} (\vec{y} - \vec{\mu}_{post}) \right] \propto \exp \left[-\frac{1}{2} (\vec{y} - M^{-1} \vec{b})^\top M (\vec{y} - M^{-1} \vec{b}) \right],$$

from comparison, it can be seen that,

$$\vec{\mu}_{post} = M^{-1} \vec{b} = (R^\top \Sigma_{li}^{-1} R + K^{-1})^{-1} (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr}), \quad \Sigma_{post} = M^{-1} = (R^\top \Sigma_{li}^{-1} R + K^{-1})^{-1}. \quad (\text{A.5})$$

The posterior mean is often written in another form. This form can be found with the following steps,

$$\begin{aligned} \vec{\mu}_{post} &= (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr}) \\ &= (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} R^\top \Sigma_{li}^{-1} \vec{d} + (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} (K^{-1} + R^\top \Sigma_{li}^{-1} R - R^\top \Sigma_{li}^{-1} R) \vec{\mu}_{pr} \\ &= \vec{\mu}_{pr} + (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} R^\top \Sigma_{li}^{-1} \vec{d} - (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} R^\top \Sigma_{li}^{-1} R \vec{\mu}_{pr} \\ &= \vec{\mu}_{pr} + (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} R^\top \Sigma_{li}^{-1} (\vec{d} - R \vec{\mu}_{pr}). \end{aligned}$$

The final closed form expression of the posterior mean and covariance is

$$\vec{\mu}_{post} = \vec{\mu}_{pr} + (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} R^\top \Sigma_{li}^{-1} (\vec{d} - R \vec{\mu}_{pr}) \quad (\text{A.6})$$

$$\Sigma_{post} = (R^\top \Sigma_{li}^{-1} R + K^{-1})^{-1}. \quad (\text{A.7})$$

The error of each value in $\vec{\mu}_{post}$ can be found on the diagonal of Σ_{post} .

Appendix B

Deriving the Marginal Likelihood and Loss Function Expression

The marginal likelihood is the denominator in Bayes theorem for the inference

$$P(\vec{y}|\vec{d}, \vec{\epsilon}, \theta) = \frac{P(\vec{d}|\vec{y}, \vec{\epsilon})P(\vec{y}|\theta)}{P(\vec{d}|\vec{\epsilon}, \theta)}, \quad (\text{B.1})$$

since the marginal likelihood is a normalizing constant it can be expressed as

$$P(\vec{d}|\vec{\epsilon}, \theta) = \int P(\vec{d}|\vec{y}, \vec{\epsilon})P(\vec{y}|\theta) d\vec{y}, \quad (\text{B.2})$$

the likelihood is,

$$P(\vec{d}|\vec{y}, \vec{\epsilon}) = \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\Sigma_{li}|}} \exp \left[-\frac{1}{2}(\vec{d} - R\vec{y})^\top \Sigma_{li}^{-1}(\vec{d} - R\vec{y}) \right], \quad \Sigma_{li} = \vec{\epsilon}I, \quad (\text{B.3})$$

and the prior is,

$$\begin{aligned} P(\vec{y}|\theta) &= \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|K|}} \exp \left[-\frac{1}{2}(\vec{y} - \vec{\mu}_{pr})^\top K^{-1}(\vec{y} - \vec{\mu}_{pr}) \right], \\ \theta &\rightarrow \{\sigma, l\}, \quad K_{ij} = k(\rho_i, \rho_j) = \sigma^2 \exp \left[\frac{(\rho_i - \rho_j)^2}{2l^2} \right], \end{aligned} \quad (\text{B.4})$$

when multiplied together the exponential powers become

$$\begin{aligned} & \left(\vec{d}^\top \Sigma_{li}^{-1} \vec{d} - \vec{d}^\top \Sigma_{li}^{-1} R \vec{y} - (R \vec{y})^\top \Sigma_{li}^{-1} \vec{d} + (R \vec{y})^\top \Sigma_{li}^{-1} R \vec{y} \right) \\ & + \left(\vec{y}^\top K^{-1} \vec{y} - \vec{y}^\top K^{-1} \vec{\mu}_{pr} - \vec{\mu}_{pr}^\top K^{-1} \vec{y} + \vec{\mu}_{pr}^\top K^{-1} \vec{\mu}_{pr} \right), \end{aligned}$$

the first order terms of \vec{y} can be simplified,

$$-\vec{d}^\top \Sigma_{li}^{-1} R \vec{y} - (R \vec{y})^\top \Sigma_{li}^{-1} \vec{d} - \vec{y}^\top K^{-1} \vec{\mu}_{pr} - \vec{\mu}_{pr}^\top K^{-1} \vec{y} = -2\vec{y}^\top (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr}) = -2\vec{y}^\top \vec{b},$$

the second order terms of \vec{y} can be simplified,

$$(R \vec{y})^\top \Sigma_{li}^{-1} R \vec{y} + \vec{y}^\top K^{-1} \vec{y} = \vec{y}^\top (R^\top \Sigma_{li}^{-1} R + K^{-1}) \vec{y} = \vec{y}^\top M \vec{y},$$

all together, for the marginal likelihood we have

$$\begin{aligned} P(\vec{d}|\vec{\epsilon}, \theta) &= \int P(\vec{d}|\vec{y}, \vec{\epsilon}) P(\vec{y}|\theta) d\vec{y} \\ &= \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\Sigma_{li}|}} \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|K|}} \exp \left[-\frac{1}{2} (\vec{d}^\top \Sigma_{li}^{-1} \vec{d} + \vec{\mu}_{pr}^\top K^{-1} \vec{\mu}_{pr}) \right] \int \exp \left[-\frac{1}{2} \vec{y}^\top M \vec{y} + \vec{y}^\top \vec{b} \right] d\vec{y}, \end{aligned} \tag{B.5}$$

performing a standard Gaussian integral we get that

$$\int \exp \left[-\frac{1}{2} \vec{y}^\top M \vec{y} + \vec{y}^\top \vec{b} \right] d\vec{y} = \frac{(2\pi)^{\frac{n}{2}}}{\sqrt{|M|}} \exp \left[\frac{1}{2} \vec{b}^\top M^{-1} \vec{b} \right],$$

all together, for the marginal likelihood we have

$$\begin{aligned} P(\vec{d}|\vec{\epsilon}, \theta) &= \int P(\vec{d}|\vec{y}, \vec{\epsilon}) P(\vec{y}|\theta) d\vec{y} \\ &= \frac{(2\pi)^{\frac{n}{2}}}{(2\pi)^{\frac{m}{2}} (2\pi)^{\frac{n}{2}} \sqrt{|\Sigma_{li}|} |K| |M|}} \exp \left[-\frac{1}{2} (\vec{d}^\top \Sigma_{li}^{-1} \vec{d} + \vec{\mu}_{pr}^\top K^{-1} \vec{\mu}_{pr} - \vec{b}^\top M^{-1} \vec{b}) \right], \end{aligned}$$

where \vec{b} and M are substitutions made earlier

$$\begin{aligned}\vec{b} &= R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr} \\ M &= (R^\top \Sigma_{li}^{-1} R + K^{-1}),\end{aligned}$$

ignoring the $-\frac{1}{2}$ for now and reverting \vec{b} and M to their original form the exponential power becomes

$$\vec{\mu}_{pr}^\top K^{-1} \vec{\mu}_{pr} + \vec{d}^\top \Sigma_{li}^{-1} \vec{d} - (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr})^\top (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr}),$$

the next step requires the Woodbury identity [11],

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}, \quad (\text{B.6})$$

the exponential power can thus be expanded to be

$$\vec{\mu}_{pr}^\top K^{-1} \vec{\mu}_{pr} + \vec{d}^\top \Sigma_{li}^{-1} \vec{d} - (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr})^\top \left[K - KR^\top (\Sigma_{li} + RKR^\top)^{-1} RK \right] (R^\top \Sigma_{li}^{-1} \vec{d} + K^{-1} \vec{\mu}_{pr}),$$

this can then be rearranged to be

$$\begin{aligned}\vec{d}^\top &\left\{ \Sigma_{li}^{-1} - \Sigma_{li}^{-1} R \left[K - KR^\top (\Sigma_{li} + RKR^\top)^{-1} RK \right] R^\top \Sigma_{li}^{-1} \right\} \vec{d} \\ &- 2\vec{\mu}^\top K^{-1} \left[K - KR^\top (\Sigma_{li} + RKR^\top)^{-1} RK \right] R^\top \Sigma_{li}^{-1} \vec{d} \\ &+ \vec{\mu}^\top \left\{ K^{-1} - K^{-1} \left[K - KR^\top (\Sigma_{li} + RKR^\top)^{-1} RK \right] K^{-1} \right\} \vec{\mu},\end{aligned}$$

the second order term in \vec{d} can be reduced

$$\begin{aligned}\Sigma_{li}^{-1} - \Sigma_{li}^{-1} R \left[K - KR^\top (\Sigma_{li} + RKR^\top)^{-1} RK \right] R^\top \Sigma_{li}^{-1} \\ &= \Sigma_{li}^{-1} - \Sigma_{li}^{-1} RKR^\top \Sigma_{li}^{-1} + \Sigma_{li}^{-1} RKR^\top (\Sigma_{li} + RKR^\top)^{-1} RKR^\top \Sigma_{li}^{-1} \\ &= \Sigma_{li}^{-1} - \Sigma_{li}^{-1} RKR^\top \Sigma_{li}^{-1} + \Sigma_{li}^{-1} (\Sigma_{li} + RKR^\top - \Sigma_{li}) (\Sigma_{li} + RKR^\top)^{-1} RKR^\top \Sigma_{li}^{-1} \\ &= \Sigma_{li}^{-1} - (\Sigma_{li} + RKR^\top)^{-1} RKR^\top \Sigma_{li}^{-1} \\ &= \Sigma_{li}^{-1} - (\Sigma_{li} + RKR^\top)^{-1} (\Sigma_{li} + RKR^\top - \Sigma_{li}) \Sigma_{li}^{-1} \\ &= (\Sigma_{li} + RKR^\top)^{-1},\end{aligned}$$

the first order term in \vec{d} can be reduced

$$\begin{aligned}
& -2\vec{\mu}^\top K^{-1} \left[K - KR^\top (\Sigma_{li} + RKR^\top)^{-1} RK \right] R^\top \Sigma_{li}^{-1} \\
& = -2\vec{\mu}^\top R^\top \Sigma_{li}^{-1} + 2\vec{\mu}^\top R^\top (\Sigma_{li} + RKR^\top)^{-1} RKR^\top \Sigma_{li}^{-1} \\
& = -2\vec{\mu}^\top R^\top \Sigma_{li}^{-1} + 2\vec{\mu}^\top R^\top (\Sigma_{li} + RKR^\top)^{-1} (\Sigma_{li} + RKR^\top - \Sigma_{li}) \Sigma_{li}^{-1} \\
& = -2\vec{\mu}^\top R^\top \Sigma_{li}^{-1} + 2\vec{\mu}^\top R^\top \Sigma_{li}^{-1} - 2\vec{\mu}^\top R^\top (\Sigma_{li} + RKR^\top)^{-1} \\
& = -2\vec{\mu}^\top R^\top (\Sigma_{li} + RKR^\top)^{-1},
\end{aligned}$$

the zero order term in \vec{d} can be reduced

$$K^{-1} - K^{-1} \left[K - KR^\top (\Sigma_{li} + RKR^\top)^{-1} RK \right] K^{-1} = R^\top (\Sigma_{li} + RKR^\top)^{-1} R,$$

now the exponential is

$$\begin{aligned}
& \vec{d}^\top \Sigma_{li}^{-1} \vec{d} + \vec{\mu}_{pr}^\top K^{-1} \vec{\mu}_{pr} - \vec{b}^\top M^{-1} \vec{b} \\
& = \vec{d}^\top (\Sigma_{li} + RKR^\top)^{-1} \vec{d} - 2\vec{\mu}^\top R^\top (\Sigma_{li} + RKR^\top)^{-1} + \vec{\mu}^\top R^\top (\Sigma_{li} + RKR^\top)^{-1} R \vec{\mu} \\
& = (\vec{d} - R \vec{\mu}_{pr})^\top (\Sigma_{li} + RKR^\top)^{-1} (\vec{d} - R \vec{\mu}_{pr}),
\end{aligned}$$

the scaling constant can be simplified using the matrix determinant lemma [11],

$$|A + UCV| = |A| |C| |C^{-1} + VA^{-1}U|, \quad (\text{B.7})$$

$$|\Sigma_{li}| |K| |M| = |\Sigma_{li}| |K| |R^\top \Sigma_{li}^{-1} R + K^{-1}| = |\Sigma_{li} + RKR^\top|,$$

this also helps avoid precision errors as there are fewer matrix inversions and determinants to compute. The marginal likelihood becomes

$$\begin{aligned}
P(\vec{d} | \vec{\epsilon}, \theta) &= \int P(\vec{d} | \vec{y}, \vec{\epsilon}) P(\vec{y} | \theta) d\vec{y} \\
&= \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\Sigma_{li} + RKR^\top|}} \exp \left[-\frac{1}{2} (\vec{d} - R \vec{\mu}_{pr})^\top (\Sigma_{li} + RKR^\top)^{-1} (\vec{d} - R \vec{\mu}_{pr}) \right].
\end{aligned} \quad (\text{B.8})$$

The values of the marginal likelihood can become very large and troublesome to compute with standard 64-bit float precision. For this reason, the logarithm is computed,

$$\ln(P(\vec{d}|\vec{\epsilon}, \theta)) = -\frac{1}{2} \left[m \ln(2\pi) + \ln(|\Sigma_{li} + RK R^\top|) + (\vec{d} - R\vec{\mu}_{pr})^\top (\Sigma_{li} + RK R^\top)^{-1} (\vec{d} - R\vec{\mu}_{pr}) \right]. \quad (\text{B.9})$$

It is convention for loss functions to be minimized so the negative log marginal likelihood is used as the loss function for optimizing the hyper-parameters. When minimizing, the constants do not play a major role, thus the loss function for the hyperparameters is expressed as

$$loss(\epsilon, \theta) = \ln(|\Sigma_{li} + RK R^\top|) + (\vec{d} - R\vec{\mu}_{pr})^\top (\Sigma_{li} + RK R^\top)^{-1} (\vec{d} - R\vec{\mu}_{pr}) \quad (\text{B.10})$$

Appendix C

Complete Set of Distributions and Expressions for Reference

C.1 Gaussian Process Regression for Interferometry, Discluding Artificial Observations

In section 2.3, Bayesian inference was introduced for a simple regression problem. In section 2.5 it was explained how to alter the method so that it could be applied to interferometry data to infer the electron density profile. Here are the mentioned distributions fully described for reference. The likelihood is,

$$\mathcal{N}(\vec{d}, \vec{\mu}_{li} = R\vec{n}_e, \Sigma_{li}) = \frac{1}{\sqrt{(2\pi)^{\frac{n}{2}} |\Sigma_{li}|}} \exp \left[-\frac{1}{2} (\vec{d} - R\vec{n}_e)^\top \Sigma_{li}^{-1} (\vec{d} - R\vec{n}_e) \right],$$
$$\Sigma_{li} = \vec{\epsilon} I = \begin{bmatrix} \epsilon_1 & 0 & \cdots & 0 \\ 0 & \epsilon_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \epsilon_m \end{bmatrix}, \quad (\text{C.1})$$

where R is a matrix composed of flux surface contribution row vectors, where each row vector corresponds to a different line of sight and when multiplied with \vec{n}_e produces the line integrated density over that line of sight, see section 2.5 for more details. The prior is,

$$\begin{aligned}\mathcal{N}(\vec{n}_e, \vec{\mu}_{pr} = \vec{0}, K) &= \frac{1}{\sqrt{(2\pi)^{\frac{n}{2}} |K|}} \exp \left[-\frac{1}{2} \vec{n}_e^\top K^{-1} \vec{n}_e \right], \\ K_{ij} = k(\rho_i, \rho_j) &= \sigma^2 \left(\frac{2l(\rho_i)l(\rho_j)}{l(\rho_i)^2 + l(\rho_j)^2} \right)^{1/2} \exp \left(\frac{(\rho_i - \rho_j)^2}{l(\rho_i)^2 + l(\rho_j)^2} \right),\end{aligned}\tag{C.2}$$

where $l(\rho)$ can be a hyperbolic tangent function or otherwise. If l is not a function but a constant, $l(\rho) = l$, then the kernel reverts back to the stationary kernel,

$$K_{ij} = k(\rho_i, \rho_j) = \sigma^2 \exp \left[\frac{(\rho_i - \rho_j)^2}{2l^2} \right],\tag{C.3}$$

The goal is to compute the posterior,

$$\mathcal{N}(\vec{n}_e, \vec{\mu}_{post}, \Sigma_{post}) = \frac{1}{\sqrt{(2\pi)^{\frac{n}{2}} |\Sigma_{post}|}} \exp \left[-\frac{1}{2} (\vec{n}_e - \vec{\mu}_{post})^\top \Sigma_{post}^{-1} (\vec{n}_e - \vec{\mu}_{post}) \right],\tag{C.4}$$

which can be done with the closed form expressions,

$$\vec{\mu}_{post} = \vec{\mu}_{pr} + (K^{-1} + R^\top \Sigma_{li}^{-1} R)^{-1} R^\top \Sigma_{li}^{-1} (\vec{d} - R \vec{\mu}_{pr})\tag{C.5}$$

$$\Sigma_{post} = (R^\top \Sigma_{li}^{-1} R + K^{-1})^{-1},\tag{C.6}$$

as derived in appendix A. Once known the density profile can be plotted with the $\vec{\mu}_{post}$ values at the same ρ values used in the kernel. The errors are the standard deviations held in the diagonal of Σ_{post} . This calculation is unlikely to be accurate until the hyperparameters are optimised. The parameters in the length scale function $l(\rho)$ are hyperparameters. The experimental errors ϵ can also be hyperparameters if unknown. The optimal hyperparameters can be found by minimising the negative log marginal likelihood. It is derived in appendix B to be,

$$loss(\vec{\epsilon}, \theta) = \ln(|\Sigma_{li} + RKR^\top|) + (\vec{d} - R\vec{\mu}_{pr})^\top (\Sigma_{li} + RKR^\top)^{-1} (\vec{d} - R\vec{\mu}_{pr}).\tag{C.7}$$

There is no change in its form from the simple regression problem. The values of the various matrices and vectors have changed.

C.2 Gaussian Process Regression for Interferometry, Including Artificial Observations

Artificial observations can be placed in the likelihood to include prior knowledge. This circumvents precision issues when including this information in the prior. The process was explained in section 2.5. Here are the full expressions for reference. The likelihood is,

$$\begin{aligned}
\mathcal{N}(\vec{d}^{alt}, \vec{\mu}_{li} = R^{alt} \vec{a}, \Sigma_{li}) &= \frac{1}{\sqrt{(2\pi)^{\frac{n}{2}} |\Sigma_{li}^{alt}|}} \exp \left[-\frac{1}{2} (\vec{d}^{alt} - R^{alt} \vec{a})^\top (\Sigma_{li}^{alt})^{-1} (\vec{d}^{alt} - R^{alt} \vec{a}) \right], \\
\vec{d}^{alt} = \begin{bmatrix} \vec{d} \\ n_e(\rho = 1) = 0 \\ n'_e(\rho = 0) = 0 \end{bmatrix} &= \begin{bmatrix} lid_1 \\ lid_2 \\ \vdots \\ lid_m \\ n_e(\rho = 1) = 0 \\ n'_e(\rho = 0) = 0 \end{bmatrix}, \\
\vec{a} = \begin{bmatrix} \vec{n}_e \\ n_e(\rho = 1) \\ n'_e(\rho = 0) \end{bmatrix} &= \begin{bmatrix} n_e(\rho_1) \\ n_e(\rho_2) \\ \vdots \\ n_e(\rho_n) \\ n_e(\rho = 1) \\ n'_e(\rho = 0) \end{bmatrix}, \\
\Sigma_{li}^{alt} = I \begin{bmatrix} \vec{\epsilon} \\ \epsilon_{edge} \\ \epsilon'_{core} \end{bmatrix} &= I \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \\ \epsilon_{edge} \\ \epsilon'_{core} \end{bmatrix} = \begin{bmatrix} \epsilon_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \epsilon_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \epsilon_m & 0 & 0 \\ 0 & 0 & 0 & 0 & \epsilon_{edge} & 0 \\ 0 & 0 & 0 & 0 & 0 & \epsilon'_{core} \end{bmatrix}, \\
R^{alt} = \begin{bmatrix} R_{m \times n} & O_{m \times 2} \\ O_{2 \times n} & I_{2 \times 2} \end{bmatrix} &= \begin{bmatrix} & & & 0 & 0 \\ & R_{m \times n} & & \vdots & \vdots \\ & & & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix},
\end{aligned} \tag{C.8}$$

where \vec{d} has been altered to include the data from the artificial observations, lid_1 is the line integrated density from the 1st laser of m lasers. \vec{a} is the vector to be inferred and is the original electron density profile \vec{n}_e with the additional artificial observations, $\vec{\epsilon}$ contains the experimental errors of the interferometry for each line of sight and ϵ_{edge} is the error of our artificial observation for the electron density at the edge, it represents the strength of our prior assumption. ϵ'_{core} represents the error of the artificial observation that the density gradient is 0 at the core, it also represents the strength of this prior assumption. R is the original response matrix explained previously and R^{alt} is a small alteration to return the artificial observations when applied to some \vec{a} . The prior is,

$$\begin{aligned}\mathcal{N}(\vec{a}, \vec{\mu}_{pr} = \vec{0}, K^{alt}) &= \frac{1}{\sqrt{(2\pi)^{\frac{n}{2}} |K^{alt}|}} \exp \left[-\frac{1}{2} \vec{a}^\top (K^{alt})^{-1} \vec{a} \right], \\ K^{alt} &= \begin{bmatrix} K & K' \\ K'^\top & K'' \end{bmatrix}, \\ K_{ij} = k(\rho_i, \rho_j) &= \sigma^2 \left(\frac{2l(\rho_i)l(\rho_j)}{l(\rho_i)^2 + l(\rho_j)^2} \right)^{1/2} \exp \left(\frac{(\rho_i - \rho_j)^2}{l(\rho_i)^2 + l(\rho_j)^2} \right), \\ K'_{ij} = k'(\rho'_i, \rho_j) &= \frac{\partial k(\rho'_i, \rho_j)}{\partial \rho'_i}, \\ K''_{ij} = k''(\rho'_i, \rho'_j) &= \frac{\partial k(\rho'_i, \rho'_j)}{\partial \rho'_i \partial \rho'_j},\end{aligned}\tag{C.9}$$

where $l(\rho)$ can be a hyperbolic tangent function or otherwise. If $l(\rho) = l$ then this reverts to the stationary kernel,

$$K_{ij} = k(\rho_i, \rho_j) = \sigma^2 \exp \left[\frac{(\rho_i - \rho_j)^2}{2l^2} \right].\tag{C.10}$$

The K' and K'' are required to account for the fact that now there is gradient information and the covariance for positions of gradient information ρ' requires a differential of the original covariance kernel k . The goal is to compute the posterior,

$$\mathcal{N}(\vec{a}, \vec{\mu}_{post}, \Sigma_{post}) = \frac{1}{\sqrt{(2\pi)^{\frac{n}{2}} |\Sigma_{post}|}} \exp \left[-\frac{1}{2} (\vec{a} - \vec{\mu}_{post})^\top \Sigma_{post}^{-1} (\vec{a} - \vec{\mu}_{post}) \right],\tag{C.11}$$

where since \vec{n}_e has been extended to \vec{a} the $\vec{\mu}_{post}$ and Σ_{post} have also been extended. The careful choice of alterations allows us to use the same closed form expressions as before the

artificial observations simply by inserting the alternate forms of the various matrices and vectors. The marginal likelihood for optimization also holds its form. To get the density profile one must remove the end terms of $\vec{\mu}_{post}$ associated with the artificial observations before plotting. The same applies to the diagonal of Σ_{post} to obtain the errors.

Glossary

emcee emcee is a Python package that allows you to perform Markov chain Monte Carlo simulations using an ensemble of walkers that explore the parameter space in an affine-invariant way. This means that the shape of the posterior distribution does not affect the efficiency of the sampling. [15](#)

IAT Integrated Autocorrelation Time (IAT) is a measure of how many steps it takes for a Markov chain to forget its initial state and become uncorrelated. It is defined as the sum of the normalized autocorrelations for all possible lags. It can be used to estimate the effective sample size and the Monte Carlo error of a chain. [15](#)

IMAS Integrated Modeling and Analysis Suite, is a framework and data management system. It is designed to store, manage, and analyze experimental and simulation data. IMAS provides a standardized platform for sharing and exchanging data among researchers from different institutions and countries. IMAS supports the integration of various fusion modeling codes and allows researchers to compare experimental data with simulation results. [6](#), [16](#)

MCMC Markov chain Monte Carlo (MCMC) is a technique for sampling from complex and high-dimensional probability distributions that are difficult to sample from directly. [14](#), [15](#)

MHD MHD stands for magnetohydrodynamics. It is a model of electrically conducting fluids that treats all interpenetrating particle species together as a single continuous medium. It is primarily concerned with the low-frequency, large-scale, magnetic behavior in plasmas and liquid metals. [5](#)

NICE Newton direct and Inverse Computation for Equilibrium. A code developed by Blaise Faugeras at the Centre national de la recherche scientifique (CNRS) to numerically solve several plasma free-boundary equilibrium problems within a tokamak

including the position of magnetic flux surfaces and the electron density profile. [2](#), [4–6](#), [17](#), [22](#), [23](#)

SQP Sequential Quadratic Programming is a numerical optimization technique used to solve nonlinear constrained optimization problems. It is an iterative method that seeks to find the optimal solution to a problem by iteratively approximating it with a quadratic model and then solving this quadratic subproblem. The key idea is to successively update the solution in a way that moves closer to the optimal solution while satisfying the constraints. [6](#)

WEST Tungsten (W) Environment in Steady-state Tokamak (WEST) is a French tokamak that aims to test and validate the ITER tungsten divertor components and prepare their safe operation [5](#), [6](#), [16](#), [17](#), [20](#), [23](#)