Modelling Step Discontinuous Functions using Bayesian emulation.

James Mark Anderson

A thesis submitted to Auckland University of Technology in partial fulfilment of the requirements for the degree of Master of Science (MSc)

2017

School of Engineering, Computer and Mathematical Sciences

Abstract

Bayesian emulation is used in modelling complex simulators and is seen as an efficient and powerful statistical tool. The simulations can be very time-consuming to run, resulting in only being able to run the simulator a limited number of times. From the literature, it is has been suggested that the emulator is best suited for continuous functions; however, it is very common to find physical problems containing discontinuities. These discontinuities' positions may also be unknown and therefore, for this research, information on where the discontinuities will be withheld from the emulator. This thesis focuses on emulating the Heaviside function as one simple function containing step-discontinuities and then progresses to slightly more complex functions with step-discontinuities. Specific goodness-of-fit measure have been designed to highlight and measure the emulator when applied to theses step-discontinuous, such as mean squared error and a simple design of Jaccard index. The numerical calculations of the goodness-of-fit techniques are carried out in the R statistical programming language, with the BACCO package for the emulator. It is found that the emulator is able to model discontinuous functions to some degree but, unless there is certainty about the discontinuities' locations, care should be taken when using Bayesian emulation to model discontinuous functions.

Contents

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signed: _____ Date: _____

Acknowledgement

Firstly, I would like to thank my supervisors, Dr Robin Hankin (primary supervisor) for the guidance throughout my thesis. I would also like to thank the Student Learning Centre at AUT, particularly Dr David Parker, and Dr Pedro Silva, for their support and guidance in the writing of my thesis. I would also like to thank Jessica Parsons for proofreading my thesis, she was contacted from AUT's current list of proofreaders.

An open source programming language called R was used to process the figures and for Bayesian emulation, using the BACCO package. Matlab R2014a was also used for some 3D plots. This thesis is compiled to a PDF format by LaTeX using the integrated writing environment TeXstudio.

Chapter 1

Introduction

1.1 Scene Setting and Motivation

Many physical problems are modelled by computer simulations. These simulations can be very time-consuming to run, resulting in only being able to run the simulator a limited number of times. However, by using observations from the simulator; a statistical modelling tool called Bayesian emulation can be used as statistical representation of the simulator (O'Hagan, 2006). Bayesian emulation has been found to be statistically rigorous, and has been found to be highly efficient when compared to using the computer simulator alone and requires fewer observations (data points) from the simulation than a simple Monte Carlo method (Oakley, 2002, p. 71).

This thesis will develop an understanding of Bayesian emulation by researching how the Bayesian emulator handles step discontinuous functions and how the emulator might improve the ability to model discontinuous functions when the location of the discontinuous is unknown by providing "goodness of fit" techniques to measure the emulator's performance.

1.1.1 Emulators

A suitable definition of deterministic simulators is that they do not contain random components. However until the simulator has finished, the observer maintains a subjective uncertainty about the results. For the Bayesian paradigm, the simulator is treated as a random function (Currin, Mitchell, Morris, & Ylvisaker, 1991) of a Gaussian process (Oakley, 1999). Before the results from the simulator are observed, any assumptions of the simulator is modelled as prior probability distribution and is used to influence the Bayesian emulator. A Gaussian process is a random function $\eta\colon \mathbb{R}^k \to \mathbb{R}$ which, for any set of points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the domain $\mathcal{D}$, the random vector $\{\eta(\mathbf{x}_1), \ldots, \eta(\mathbf{x}_n)\}$ has a joint distribution of a multivariate Gaussian (Hankin, 2014).

1.1.2 Discontinuity

Figure 1.1 shows a simple piecewise function that contains a discontinuity known as the Heaviside step function[1] (Weisstein, 2002). A definition of discontinuity can be found in most first- or second-year undergraduate calculus courses and calculus textbooks such as Stewart (2010) which follows:

A function f if **continuous at a number** a if

$$\lim_{x \to a} f(x) = f(a)$$

This requires two things if f is continuous at a:

1. $f(a)$ is defined (that is a is in the domain of f)
2. $\lim_{x \to a} f(x) = f(a)$

(Stewart, 2010, p. 113)

For the Heaviside step function illustrated in Figure 1.1, the function is not continuous as the $\lim_{x \to 0} f(x)$ **does not** exist (because the left and right limits are different).

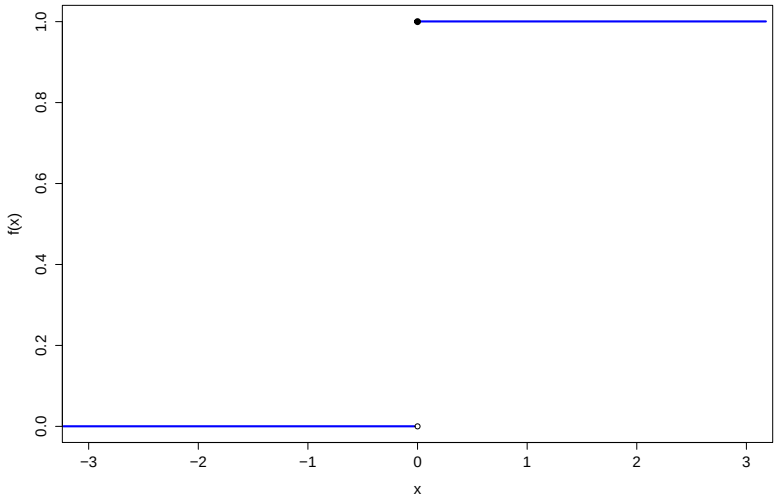

Figure 1.1: Heaviside step function

There are many scientific fields where functions containing discontinuities occur, such as economics, climate science and heavy gas dispersion. These examples will be discussed in Chapter 2: Literature and Mathematics Review

[1] This Heaviside step function will be used for the start of the thesis investigation in Section 5.2

1.1.3 Bayesian Emulators and Discontinuity

The University (AUT) online library has access to multiple science and mathematics databases, including: AccessEngineering, ACM Digital Library, IEEE Xplore, MathSciNet, ScienceDirect, Statistics New Zealand, and Web of Science. Google Scholar was also included as an additional search engine.

The aim of the search of these databases was to find relevant research on modelling functions containing discontinuities using Bayesian emulation. Words used to associate with Bayesian emulation were:

- Bayesian emulation
- Bayesian emulator
- Bayesian inference
- Gaussian process
- Gaussian emulator

From this search, only a few papers were found with Chang, Strong, and Clayton (2015), and Caiado and Goldstein (2015) being the most relevant in terms of modeling discontinuities using Bayesian emulation.

The known literature, has shown that there is a lack if research in how the emulator models simulators with discontinuities. This thesis will address this research gap by answering the research questions (stated below).

Bayesian emulation has been used in a wide range of fields, such as climate science (Conti, Gosling, & O'Hagan, 2009; Caiado & Goldstein, 2015) and economics (Oakley, 2009). From the literature, Bayesian emulation has been most commonly applied to continuous functions; however, it is very common to find physical problems containing discontinuities where the location of the discontinuities are unknown. Because the location of the discontinuity can be unknown, a Bayesian paradigm may be appropriate for modelling these simulators. Current research on Bayesian emulators is mainly focused on continuous functions but not on discontinuous functions with the discontinuity location unknown. From present literature, researchers would state that their function/simulator "is assumed to be a smooth function of the inputs without discontinuities" (Chang et al., 2015, p. 17). Caiado and Goldstein (2015) found the boundary of the regions of the discontinuities, stating "this will be an expensive operation" (Caiado & Goldstein, 2015, p. 131). Once the discontinuities were found, they ran separate emulators for each region, removing the problem of having discontinuities within the emulator's calculations.

The disadvantage of Caiado and Goldstein's (2015) approach is that by running two separate emulators could lead to losing information about the simulator as a whole; however if Caiado and Goldstein (2015) knew where the discontinuities were

it might be possible to use a different prior taking the discontinuities into account. This thesis will look at how the emulator is able to handle functions with one or more discontinuities with an unknown position, where current researchers such as Caiado and Goldstein (2015, p. 131) have avoided this question by calculating where the discontinuity (or discontinuities) are, which tends to be expensive.

Other researchers have also taken the same approach (Becker, Worden, & Rowson, 2013; Gramacy & Lee, 2008; Kim, Mallick, & Holmes, 2005) partition the input space and then and then fitting a Gaussian process (the emulator) to each part of the space. However, one can only partition the input space if the discontinuity's location is certain. (This is shown in Chapter 3)

This thesis will focus on some of the simplest discontinuous functions to understand how Bayesian emulator models functions with discontinuities.

1.2 Research Questions

The three research questions of this study are:

1. How well does Bayesian emulation model simulators containing discontinuities, with the discontinuities at unknown positions?

2. What can be learnt about Bayesian emulation though modelling discontinuous functions?

3. What meaningful goodness-of-fit techniques can be used to measure how well the emulator performs at modelling a simulator with discontinuities?

1.3 Chapter Summaries

Chapter 2: Literature and Mathematics Review

This chapter discusses the literature and an overview of: deterministic simulators, discontinuity and current research on modelling discontinuity, and the Bayesian emulation with a layout of the notations used for this thesis.

Chapter 3: An Introduction to the Emulator Modelling Step Discontinuous Functions

This chapter provides a short demonstration on how the emulator models the Heaviside function, using a simple approach on selecting the data points for the emulator. This chapter then looks at introducing the a discontinuous function as information given to the emulator, however this shows that there must be the assumption that the location of the discontinuity is known.

Chapter 4: Research Method for Quantitative Analysis

This chapter discusses the research objectives and approach, including what software was used for the experiments, and the structure of the experiment.

Chapter 5: Numerical Findings from the Emulator Experiments

This chapter documents the findings of the goodness-of-fit when applied to step-discontinuous functions. This thesis started with a Heaviside function containing one discontinuity and progresses to more complex step-discontinuous functions. For each experiment a number of goodness-of-fit techniques were applied to measure how well (or how badly) the emulator was able to model the function. By running the emulator multiple times with different data points a probability density function (PDF) was made for each goodness-of-fit technique. Changing the number of data points per emulator run was also performed, showing how the number of data points affects the accuracy.

Chapter 6: Discussion and Conclusion

This chapter discusses the key findings of this research, a summary and conclusion of the findings, and recommendations when using the emulator to model discontinuous functions. This chapter also proposes some suggestions as further research.

Appendix

The appendix contains

- Properties and proofs, demonstrating why the experiments were done numerically rather than analytically.
- An investigation on optimising B in the correlation function.
- Minor observations found during the experiments that may be less relevant to the focus in the findings.
- The computer software used for the experiments, including:
    - ▷ Adjustments made to the software or code; and
    - ▷ Example of the code used in the experiments.

Chapter 2

Literature and Mathematics Review

2.1 Introduction

This chapter will discuss some of the key concepts of Bayesian emulation and discontinuity, discuss some applications where step discontinuous functions are found and the current research using Bayesian emulation to model simulators that contain discontinuities. This chapter will also introduce key notations for Bayesian emulation. Most notations come from Oakley (1999) and Hankin (2012).

2.2 Deterministic Simulators

Simulators are widely used in all fields of science and technology to describe and understand complex systems.[1] These simulators can reduce the need for real-world experiments, as a result reducing cost. Many of these simulators are complexly coded computer models designed with the aim of high accuracy. Because of the demand for high accuracy, the simulator will tend to be very time-consuming to run, and impractical to run after a certain amount of runs. These simulators are deterministic in the sense that they contain no randomness components, and running the simulator twice with identical inputs will result in identical outputs(Hankin, 2005).

2.3 Step Discontinuity

A discontinuity exists when a function contains at least one point or interval where the function is not continuous (Stover & Weisstein, 2013).

The Heaviside piecewise function is step discontinuity function where, at a certain point(s), there is a sudden increase or decrease in the function and the function is

[1]Complex: in this terminology the system contains many parts and/or a very large amount of code

continuous everywhere before and after the discontinuity. Figure 1.1 is an example of the Heaviside step function (see page 2). Using the Heaviside function for the first part of the investigation will give a clear demonstration of how the Bayesian emulator models a function containing a discontinuity.

2.3.1 Example of Step Discontinuities

Step discontinuities are found in a wide range of applications, for example:

Heavy Gas Dispersion

A gas with a higher density than air is called a heavy gas; given the same volume, the heavy gas will weigh more than the air. When heavy gases such as chlorine are mass-produced and stored in large quantities, occasionally these gases are accidentally released (for example, due to failure of the tank) and the gas will spread (due to its own density and wind) across the terrain as low, flat clouds (Blackmore, Herman, & Woodward, 1982).

For industrial risk assessment, it is important to be able to understand and model the flow of heavy gas dispersion, as when the gas is released, it has the potential to cause death as it spreads across the terrain. Figure 2.1 shows how, at the boundary of the gas, there exists a discontinuity where there is a sudden drop in the amount of gas. During the dispersion of the gas, there is a region where the gas is present (this changes as gas spreads). Within this region, the concentration level of the gas is almost constant; however, the concentration rapidly decreases around the edge of the cloud. This rapid decrease can be mathematically represented as a discontinuity, and knowing the location of this discontinuity provides an estimate on how far the gas could spread. This can be mathematically represented as a function with one or more discontinuities at the boundary of the gas (Hankin & Britter, 1999).

Figure 2.2 is an example of an aerial photo after a release of ammonia from a tanker in Houston in 1976. The orange region is where the vegetation has been destroyed by the ammonia. As can be seen from the figure (on page 8) there is a sharp region where, on one side, vegetation is unaffected and, on the other side, vegetation has been destroyed; this indicates a discontinuity.

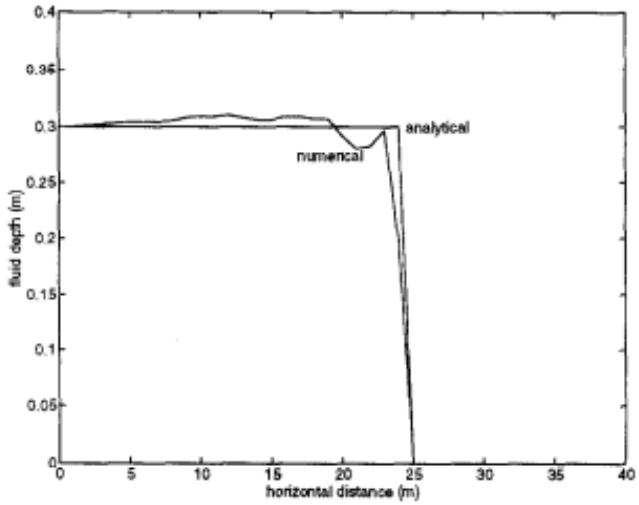

Figure 2.1: Example of a heavy gas dispersion showing a discontinuity around 23m (Hankin & Britter, 1999, p. 233)

Figure 2.2: Aerial photo of ammonia released in Houston; the brown-orange region is where the vegetation was destroyed from the ammonia. Note the sharp edge along the region.

Shock Waves

An example where piecewise step discontinuity functions exist in climate science is in shock waves. The discontinuities exist within the boundary of the shock wave region. Villarreal (2006) studied the existence and non-existence of shock wave solutions for the Burger equations (a partial differential equation) by using a Heaviside generalized function. Villarreal (2012) extended his own research, looking at the Heaviside generalized function with Colombeaus theory context with the study of shock wave as the motivation. Figure 2.3 is an illustration of another shock wave example from Hankin (2001).

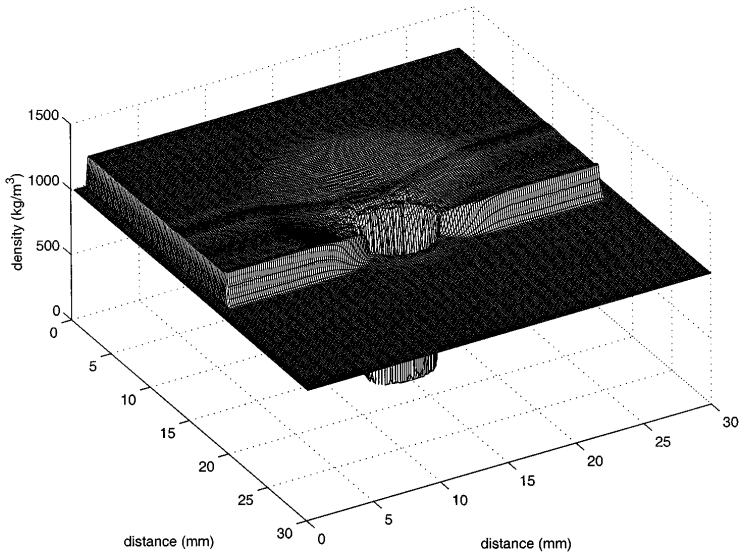

Figure 2.3

2.4 Current Research on Modelling Discontinuities

One current method in modelling discontinuous function is called Regression Discontinuity Design, however RDD is typically used to measure the difference at the discontinuity. Regression Discontinuity Design (RDD) has been found to be used in economics, (Imbens & Lemieux, 2008; Porter & Yu, 2015; Lee & Lemieux, 2010), pharmacology (Moscoe, Bor, & Brnighausen, 2015) and education (Thistlethwaite & Campbell, 1960).

The equation of the RDD follows:

$$y = Y_0(x)(1 - D) + Y_1(x)D + \epsilon \tag{2.1}$$

(Porter & Yu, 2015)

where Y_0 and Y_1 are usually linear equations with respect to x and D is a binary variable equal to one if $x \geq d$ where d is the position of the discontinuity, and ϵ is some residual error.

At first RDD was designed with the assumption that the location of the **discontinuity is known**, however; Porter and Yu (2015) noted this major restriction and extended RDD for unknown discontinuity points. Figure 2.4 is a example of a setup for RDD.

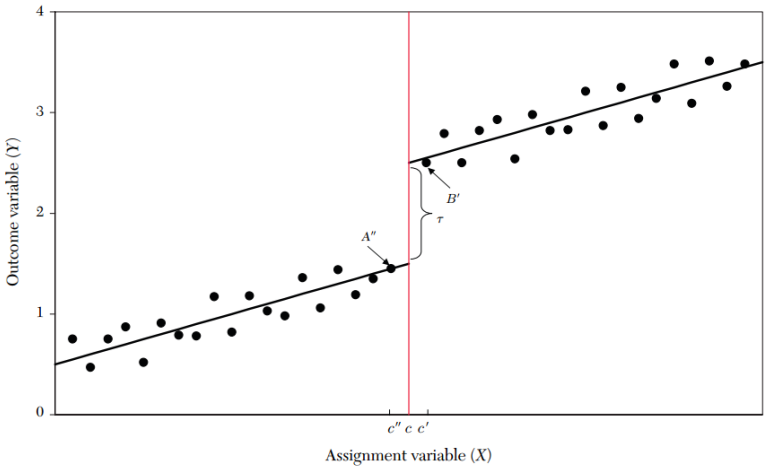

Figure 2.4: Simple Linear RD Setup

(Lee & Lemieux, 2010, p. 287)

RDD was first introduced by Thistlethwaite and Campbell (1960) as a replacement to the *ex post facto* design and was applied in education research. A study was done looking at how receiving a merit awards affected students' future academic outcomes (career aspirations, enrolment in postgraduate programs, etc.) (Lee &

Lemieux, 2010). Thistlethwaite and Campbell (1960) used RDD compared the student's test scores to discover how much the students were benefited by receiving the scholarship (similar to Figure 2.4).

2.5 Bayesian Emulation

2.5.1 Introduction

Haylock, O'Hagan, Kennedy and Oakley are some active researchers of the Bayesian emulation.

Haylock and O'Hagan (1996) first introduced the Bayesian emulation based on a Gaussian process prior model, deriving the posterior mean and variance, which were extended further by Oakley and O'Hagan (1998). Observations from the simulator is used as a way of training the emulator (Conti et al., 2009). The emulator was found to emulate the simulator's output to a high degree of precision using fewer simulator runs than a Monte Carlo method (Conti et al., 2009).

As an alternative to Bayesian emulation (before the research was conducted on Bayesian emulation), a Monte Carlo method would be used that requires thousands of simulator runs (Oakley, 1999). Using Monte Carlo as an analysis process is simple; however it requires a large number of observations from the simulator and can become impractical if the simulator is costly to run (M. C. Kennedy & O'Hagan, 2001).

2.5.2 The Bayesian Approach

For any outcome, until it has been observed by the individual, the results are subjectively uncertain and can be modelled as a random variable (Hankin, 2012). For example, if one were to predict the USA presidential election outcome, the individual would use the information provided (e.g., previous polls and who has been selected) and beliefs to make a esstimate. This could be mathematically modelled as a random variable. Even though the results are deterministic in the sense that a recount of the votes would make no difference (excluding human factors), until the individual has seen the results, there exists that subjective uncertainty.

Deterministic simulators contain no randomness components; however, until the simulator has been computed, the results are subjectively unknown (or subjectively uncertain). Under the Bayesian paradigm, this subjective uncertainty of the simulator's outcome is treated as a random variable. The true values of the simulator can be drawn from a distribution, conditional on the prior knowledge of the simulator, and statistical inferences of the simulator can be made, allowing predictions of the simulator that have not yet been observed. As stated previously, these simulators may be time-consuming to run, and Bayesian emulation provides computational

cheapness and a statistically rigorous approach (Hankin, 2005).

The Bayesian emulator has a Gaussian process prior for functions in regression(Oakley, 1999). However, in general the simulator is very complex, and a simple regression function is used (Conti et al., 2009) such as a linear model; this allows the data from simulator runs to be the main influence on the emulator.

2.5.3 Bayes' theory

Bayes's theorem follows as:

let A and B be two events

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Given that event B has occurred, the probability that A will occur is equal to the probability that both A and B will occur divided by the probability that B will occur.

In Bayesian inference, B is denoted as the observed data points from the simulator and will be represented by the notation of D, and A is denoted as the set of unknown parameters: X

$$P(X|D) = \frac{P(D|X)\,Pr(X)}{P(D)}$$

Each part of the equation is labelled as:

$P(X|D)$ is the posterior, which is the target distribution we are trying to obtain.

$P(D|X)$ is the likelihood, which is asking: what is the likelihood of obtaining the data given the unknown parameters?

$Pr(X)$ is the prior, which is information about the unknown parameters. The prior distribution can be seen as a calculated guess of X based on general knowledge and research. We should aim to find the best distribution that fit with X.

$P(D)$ is known as the marginal distribution, where $P(D) = \int P(D, X)\,dX = \int P(D|X)\,P(X)\,dX$(Gelman, 2009). $P(D)$ does not depend on X as the data does not change, therefore it can be seen as a constant in which changes the equation to:

$$P(X|D) \propto P(D|X)\,Pr(X) \tag{2.2}$$

2.5.4 Gaussian Process

For Gaussian process: for any set of points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the domain $\mathcal{D}$, the random vector $\{\eta(\mathbf{x}_1), \ldots, \eta(\mathbf{x}_n)\}$ has a joint distribution of a multivariate Gaussian (Hankin, 2014).

Where the expected value of the random function output's vector is:

$$\mu = [\mathbb{E}(\eta(\mathbf{x}_1)), \mathbb{E}(\eta(\mathbf{x}_2)), \ldots, \mathbb{E}(\eta(\mathbf{x}_n))]$$

and a covariance matrix Σ:

$$\Sigma = \begin{bmatrix} \mathrm{Var}(\eta(\mathbf{x}_1)) & \mathrm{COV}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_2)) & \cdots & \mathrm{COV}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_n)) \\ \mathrm{COV}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_2)) & \mathrm{Var}(\eta(\mathbf{x}_2)) & & \mathrm{COV}(\eta(\mathbf{x}_2), \eta(\mathbf{x}_n)) \\ \vdots & & \ddots & \vdots \\ \mathrm{COV}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_n)) & \cdots & & \mathrm{Var}(\eta(\mathbf{x}_n)) \end{bmatrix}$$

There are many fields of applications where the Gaussian process is used to make a statistical inference about a certain problem or function.

2.5.5 Bayesian Emulation and Notations

Let $\eta\colon \mathbb{R}^k \to \mathbb{R}$ now be a deterministic simulator where it is assumed to be complex, very time-consuming to run and impractical to run after so many multiple runs. An example of a simulator that models physical world is TWODEE (Hankin & Britter, 1999),[2] which models the flow of heavy gas dispersion spreading across a terrain.

From Oakley (1999), the simulator is a Gaussian process: an infinite collection of random variables with the property that any subset of these variables will be a multivariate Gaussian distribution (Oakley, 1999). The observations/results directly from the deterministic simulator can also seen as being drawn from a multivariate Gaussian, but with a variance of $\mathbf{0}$, this is because the results have been observed and therefore there is no uncertainty. By using the observations directly from the simulator, Bayesian inferences can then be made of the simulator.

One Bayesian inference technique is called Bayesian emulation (Oakley, 1999), providing statistical inferences about the simulator as a Gaussian process. Using Bayesian emulation has been found to be a more economical approach compared to using the simulator alone (Hankin, 2005). Bayesian emulation is best suited for continuous functions rather than functions containing discontinuities, and although there are many physical problems and simulations that contain discontinuities, very little research has been done applying Bayesian emulation to model these functions.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a set of training runs of the simulator ($\mathbf{x}_i \in \mathbb{R}^k$) in which the simulator observations: $y_1 = \eta(\mathbf{x}_1), y_2 = \eta(\mathbf{x}_2), \ldots, y_n = \eta(\mathbf{x}_n)$, which will be denoted as $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$, and is to be seen as data in which will be used to emulate the simulator. (O'Hagan, 2006)

[2]This thesis will not be using TWODEE within the experiments.

Oakley (1999) begins to model η using a stochastic process approach as:

$$\eta(\mathbf{x}) = \sum_{i=1}^{r} \beta_i h_i(\mathbf{x}) + Z(\mathbf{x}) \quad \text{Eq: 2.1 (Oakley, 1999, p. 9)} \tag{2.3}$$

where for each value of i, $h_i(\mathbf{x})$ is a known regressor function of $\mathbf{x}$ and β_i is an unknown coefficient. $Z(\cdot)$ is a stochastic process with mean zero, and the covariance between $Z(\mathbf{x})$and $Z(\mathbf{x}')$ is given by a covariance function (COV$(\mathbf{x}, \mathbf{x}')$: see below). Matrices are used for the calculations, and both β and the known regressor functions are vectors $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_r(\mathbf{x})]$, rewriting the equation as:

$$\eta(\mathbf{x}) = \mathbf{h}(\mathbf{x})\beta + Z(\mathbf{x})$$

Given β, the expectation of η is $\mathbb{E}[\mathbf{h}(\mathbf{x})\beta + Z(\mathbf{x})] = \mathbb{E}[\mathbf{h}(\mathbf{x})\beta] + \mathbb{E}[Z(\mathbf{x})] = \mathbf{h}(\mathbf{x})\beta + 0$

$$\mathbb{E}[\eta(\mathbf{x})|\beta] = \mathbf{h}(\mathbf{x})\beta \quad \text{Eq: 2.6 (Oakley, 1999, p. 10))}$$

$\mathbf{h}(\mathbf{x})$ is a $1 \times q$ vector from the regression function.

For example, if $\mathbf{h}(\mathbf{x}) = [\mathbf{x}^2, \mathbf{x}, 1]$ then $\mathbf{h}(\mathbf{x}) = [x_1^2, x_2^2, \ldots, x_k^2, x_1, x_2, \ldots, x_k, 1]$, resulting in $q = 2k + 1$.

Within the emulator, a matrix of the regressor is used containing n regressor vectors for each of the unique input parameters $\mathbf{X}$.

$$H^T = [\mathbf{h}(\mathbf{x}_1), \mathbf{h}(\mathbf{x}_2), \ldots, \mathbf{h}(\mathbf{x}_n)]^T \quad \text{Eq: 2.15 (Oakley, 1999, p. 13)}$$

H has the dimensions of $n \times q$ and is denoted as $H_{[n \times q]}$

The correlation between two observations is defined as a function: $\mathrm{c}(\mathbf{x}, \mathbf{x}')$, where: $0 \leq \mathrm{c}(\mathbf{x}, \mathbf{x}') \leq 1$ and as $\|\mathbf{x}' - \mathbf{x}\| \to 0$ then $\mathrm{c}(\mathbf{x}, \mathbf{x}') \to 1$ and as $\|\mathbf{x}' - \mathbf{x}\| \to \infty$ then $\mathrm{c}(\mathbf{x}, \mathbf{x}') \to 0$. In Oakley (1999), the following correlation function was used that has been designed for smooth functions:

$$c(\mathbf{x}, \mathbf{x}') = e^{-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')} \tag{2.4}$$

where $B_{[k \times k]}$ is a matrix of a smoothness parameters and is positive definite and is unknown. This correlation function has been used in several research papers (Caiado & Goldstein, 2015; Chen, Zabaras, & Bilionis, 2015; Conti et al., 2009; Hankin, 2012; M. C. Kennedy & O'Hagan, 2001; Montagna & Tokdar, 2016; Oakley, 2002; Zhang, Konomi, Sang, Karagiannis, & Lin, 2015).

From Bochner's theorem (Hankin, 2012), for $\mathrm{c}(\mathbf{x}, \mathbf{x}')$ to be a correlation function, $\mathrm{c}(\mathbf{x}, \mathbf{x}')$ must be the characteristic function of a random variable and symmetric about the origin.

B is in most cases unknown, and an estimate is calculated, using the information

that B is proportional to an inverse gamma density function (Oakley, 1999) and some trial and error.

For the regressor $\mathbf{h}(\mathbf{x})^T \beta$, in practice β is normally unknown, and an estimate, $\hat{\beta}$, is used with the equation:

$$\hat{\beta} = \left(H^T A^{-1} H\right)^{-1} H^T A^{-1} \mathbf{y} \quad \text{Eq: 2.24 (Oakley, 1999, p. 14)} \tag{2.5}$$

(by using maximum likelihood estimation of $\mathbf{y} = \mathbf{h}(\mathbf{x})\beta + \epsilon$)

The posterior mean of $\eta(x)$ is given by

$$\mathbb{E}\left[\eta(\mathbf{x}) | \hat{\beta}, \mathbf{y}\right] = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \mathbf{t}(\mathbf{x})^T A^{-1} \left(\mathbf{y} - H\hat{\beta}\right) = \hat{\eta}(\mathbf{x}) \tag{2.6}$$

Eq: 2.30 (Oakley, 1999, p. 14)

Where $A_{[n \times n]}$ is a matrix of correlations.

$$A = \begin{pmatrix} 1 & c(\mathbf{x}_1, \mathbf{x}_2) & \cdots & c(\mathbf{x}_1, \mathbf{x}_2) \\ c(\mathbf{x}_2, \mathbf{x}_1) & 1 & & \vdots \\ \vdots & & \ddots & \\ c(\mathbf{x}_n, \mathbf{x}_1) & & & 1 \end{pmatrix} \tag{2.7}$$

The posterior variance of $\eta(x)$ is given by

$$\mathrm{COV}\left(\eta(\mathbf{x}), \eta(\mathbf{x}')\right) = \sigma^2 \mathrm{c}(\mathbf{x}, \mathbf{x}') \tag{2.8}$$

Eq: 2.7 (Oakley, 1999, p. 11)

where σ is the overall variance, and

$\mathrm{c}^{**}(\mathbf{x}, \mathbf{x}') = c(\mathbf{x}, \mathbf{x}') - \mathbf{t}(\mathbf{x})^T A^{-1} \mathbf{t}(\mathbf{x}') +$

$+ \left(\mathbf{h}(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T A^{-1} H\right) \left(H^T A^{-1} H\right)^{-1} \left(\mathbf{h}(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T A^{-1} H\right)^T$

Eq: 2.31 (Oakley, 1999, p. 14)

where $\mathbf{t}(\mathbf{x}) = \left[\mathrm{c}(\mathbf{x}, \mathbf{x}_1), \mathrm{c}(\mathbf{x}, \mathbf{x}_2), \ldots, \mathrm{c}(\mathbf{x}, \mathbf{x}_n)\right]^T$

Oakley then derives the Bayesian emulator as:

$$\eta(\mathbf{x}) | \mathbf{y}, \sigma^2 \sim \mathcal{N}(\hat{\eta}(\mathbf{x}), \sigma^2 \mathrm{COV}^{**}(\mathbf{x}, \mathbf{x})) \tag{2.9}$$

Statistical inferences about the simulator can now be made by using the above equations.

In practice σ^2 is unknown and an estimation of σ^2 denoted as $\hat{\sigma}^2$ is used, resulting in the simulator being a Student-t process with $n - q - 2$ degrees of freedom.

$$\frac{\eta(\mathbf{x}) - \hat{\eta}(\mathbf{x})}{\hat{\sigma}\sqrt{\mathrm{COV}^{**}(\mathbf{x}, \mathbf{x})}} \sim t_{n-q-2} \tag{2.10}$$

where $\hat{\sigma}^2 = \frac{\mathbf{y}^T\left(A^{-1}-A^{-1}H\left(H^TA^{-1}H\right)H^TA^{-1}\right)\mathbf{y}}{n-q-2}$
and $\mathrm{COV}^{**}(\mathbf{x},\mathbf{x}') = c(\mathbf{x},\mathbf{x}') - \mathbf{t}(\mathbf{x})^T A^{-1}\mathbf{t}(\mathbf{x}') +$
$+\left(\mathbf{h}(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T A^{-1}H\right)\left(H^TA^{-1}H\right)^{-1}\left(\mathbf{h}(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T A^{-1}H\right)^T$

2.5.6 BACCO in R

The BACCO suite of R packages (Hankin, 2005) is used here for Bayesian emulation. It was originally demonstrated to model a simulator that could predict sea surface temperatures. The simulator took from 12 to 24 hours to calculate the sea surface temperature, whereas using the emulator took 0.1 seconds to calculate the posterior mean and variance of the sea surface temperature (Hankin, 2005), demonstrating that the use of the emulator results in a significant reduction in computation time. Other fields are mentioned by Hankin (2005), such as Earth systems science and climate science. These real-world applications demonstrate the potential of Bayesian emulation. This thesis will develop Bayesian emulation by investigating simulators with discontinuities.

2.5.7 Current Research on Emulators

Most studies in the literature include a summary of Bayesian emulation and then apply it to for example: modelling a rainfall-runoff simulator (Conti et al., 2009), health economics, (Oakley, 2009), the Sheffield Dynamic Global Vegetation model (Conti & O'Hagan, 2010), and modelling the climate of Atlantic (Caiado & Goldstein, 2015). Rasmussen and Williams (2006) has also applied the the emulator as a method of machine learning.

2.6 Current Research on Modelling Discontinuities with Bayesian Emulation

See section 1.1.3 : Bayesian Emulators and Discontinuity

Chapter 3

An Introduction to the Emulator Modelling Step Discontinuous Functions

This thesis investigates how the emulator models step discontinuous functions. The Heaviside function will be used to start with, as it is a simple function with its main property being a step discontinuity. Equation 3.1 is the Heaviside function for this thesis.

$$\eta(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{3.1}$$

The correlation function used is from equation 2.4: $c(\mathbf{x}, \mathbf{x}') = e^{-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')}$, which is commonly used from other research, B will be equal to 1. Because the Heaviside function has no slope, the regression function will be a constant number $\mathbf{h}(x) = 1$. Both the correlation and regression function will been discussed in more detail in Section 4.2.1 (page 32).

This chapter will provide some visual analysis of the emulator modelling step discontinuous functions using simple approaches. Data points will be chosen with at least one near the discontinuity and "stitched"[1] at the discontinuity. Some of the goodness-of-fit measures will be applied to visually measure the accuracy of the emulator.

[1]The word stitched in this context means to have two points evaluated close to and either side of the discontinuity

The below table is a summary of examples of the figures that will be discussed in this chapter. The purpose of these figures and their discussion is to give an insight on some techniques and its limitations on how the emulator models functions like the Heaviside.

Figure	**description**
Figure 3.1	Using a continuous constant regressor to model the Heaviside function with no stitching
Figure 3.2	The discontinuity's position is not known, however one data point is close to the discontinuity (left side)
Figure 3.3	The discontinuity's position is not known, however one data point is close to the discontinuity (right side)
Figure 3.4	The discontinuity's position is effectively known, a continuous regressor is still used and is now "stitched"; two data points are evaluated close to and either side of the discontinuity
Figure 3.5	Using a discontinuous regressor, the discontinuity is partly known, with one point close to the discontinuity (similar to Figure 3.2)
Figure 3.6	Using the same discontinuous regressor as in Figure 3.5, with two data points "stitched" at the discontinuity
Figure 3.7	Using a discontinuous regressor to model the Heaviside function with no stitching, however the location of the discontinuity is provided from the regressor
Figure 3.8 (a & b)	Using a discontinuous regressor to model the Heaviside function with no stitching, and the location of the discontinuity is inferred from the data points
Figure 3.9	Using a discontinuous regressor to model the Heaviside function with no stitching, and the location of the discontinuity is incorrect

Table 3.1: Summary of the Figures

3.1 Example 1: One data point at the discontinuity

Starting with Figure 3.2 (see page 24), where; $-3 \leq x \leq -0.05$ and $1 \leq x \leq 3$ the emulator is a reasonable representation of the Heaviside function as if the true function is smooth. For this Heaviside function the slope is always positive between the two data points on either side of the discontinuity (even though we don't know where it is) and there is no over/under shooting of the function between these points. Estimating the location of the discontinuity by $\hat{\eta} = 0.5$ gives a location of 0.5.

Figure 3.3 (see page 25) is obtained with the only change from Figure 3.2 (page 24) of the one data point at the discontinuity. Estimating the location of the discontinuity by $\hat{\eta} = 0.5$ gives a location of -0.5. Because the data points are symmetric, the approximated location of the discontinuity is also symmetric to Figure 3.2. Between the domain of 0 to 3 it can be seen that the emulator is a reasonable representation of the Heaviside function. As expected, the emulator will always converge to the prior outside of the data points due to the lack of information (no data points).

3.2 Example 2: Two data point stitched at the discontinuity

For Figure 3.4 (see page 26), there are two data points very close on either side on the discontinuity, giving a simple strategy of using the data points to manipulate the emulator, as if the location of the discontinuity is almost certain. This has made the emulator worse compared to Figure 3.2 (page 24) and 3.3 (page 25), causing the emulator to be very inaccurate in predicting the function with a lot of over/under shooting The slope of the emulator is high at the point of the discontinuity which has caused the extreme over and under shooting and the closer the two points are the larger the slope would be.

Note that the extreme over/under shooting could be reduced by changing B in the correlation function, however because the Heaviside function has no length scale B is kept at 1

3.3 Example 3: Using a discontinuous function as the regressor, with one data point at the discontinuity

If we were to know the location of the discontinuity, then everything changes; our current setup with the regressor and correlation function might have to change (partic-

ularly the regressor function as seen in Section 3.5). To demonstrate this, let $\eta(x) = \begin{cases} 2x+5 & x \leq 0 \\ \frac{x}{2}-3 & x > 0 \end{cases}$, and the regressor function to be $h(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$, and using the same data points as Figure 3.2.

From Figure 3.5, the region outside of the data points of the discontinuity ($-3 \leq x \leq 0$ and $1 \leq x \leq 3$) the emulator is a reasonable representation of η, however the emulator overshoots between 0 and 1.

The variance is also discontinuous due to equation 2.8 (page 15), where $c^{**}(\mathbf{x}, \mathbf{x}')$ contains the discontinuous regression function.

3.4 Example 4: Using a discontinuous function as the regressor, with two data point stitched at the discontinuity

Knowing that the discontinuity is at 0 we obtain Figure 3.6. Because we the data points either side of the discontinuity are close together there is very little over/under shoot. This shows that, if the data points are "stitched" at the discontinuity; it could be suitable to use a regressor that contains the discontinuity at the stitched location. (as compared to Figure 3.4). As a side note of the emulator; because it is "stitched", the prior is restricted to be the same distance above each of the two stitched data points (in this case 3).

3.5 Example 4: Using a discontinuous function as the regressor to model the Heaviside Function

Returning back to the Heaviside function, let the regressor now be:
$h(x) = \begin{cases} \frac{x}{2}-3 & x \leq 0 \\ 2x+5 & x > 0 \end{cases}$. From this, Figure 3.7 is obtained. It is observed that with this prior, the emulator has shown less of an over/under shooting at the discontinuity, compared to when the constant regressor was used (Figure 3.1). This is because information about the discontinuity has been provided to the emulator.

However, if the location of the discontinuity is unknown, then the only information available is the data points. Figure 3.8 demonstrates two extreme ends, assuming the discontinuity would be between the two data points in which "jumps" from 0 to 1. Comparing Figure 3.1 to Figure 3.8 it could be suggested that using a discontinuous regressor when the location of the discontinuity is unknown, does not significantly improve the emulator in modelling the Heaviside function.

When the data points are ignored to estimate where the discontinuity might be the emulator could result in something similar to Figure 3.9 that shows that, any

information conflicting with the prior results in a high level of errors for the emulator causing over/under shooting.

3.6 Conclusion

The software of the emulator works as expected; however, Bayesian emulation and modelling of discontinuous functions can be a complex process alone and more so together. The emulator is able to provide a reasonable representation of the Heaviside function when using a constant regression function but does show a decrease in accuracy near the discontinuity. This inaccuracy at the discontinuity is due to multiple factors, particularly the regression function and correlation, but more so on the assumption that the location of the discontinuity is unknown.

From the literature, it has been suggested to run separate emulators for each region, removing the problem of having discontinuities (Caiado & Goldstein, 2015). This chapter has demonstrated that it is possible to use one emulator by using a discontinuous regressor. However in both cases the discontinuity's location is known.

The emulator can be improved at modelling a discontinuous function by specifying the regressor function to include the discontinuity when the location of the discontinuity is known; in which Figure 3.6 has demonstrated. When the location of the discontinuity is unknown, the emulator only maintains acceptable accuracy outside the domain between the data points either side of the discontinuity.

This has shown how strongly dependent the emulator is on the regressor function, the correlation function, and the data points. It has been demonstrated that when using a discontinuous regressor, the emulator relies heavily on the assumption that the location of the discontinuity is known; however, there are times when it may be impractical (or effectivity impossible) to run the function/simulator multiple times to narrow down where the discontinuity is.

For this reason; this thesis will be focusing on the emulator with unknown position of the discontinuity, by not providing the location of the discontinuity or that the function even has a discontinuity to the emulator. Until now, this thesis has looked at informal measures of the emulator performance. However, this thesis will discuss formal goodness-of-fit measures (as discuss in Chapter 4); goodness-of-fit techniques will be designed to measure the emulator's accuracy measuring upon the Heaviside function and other similar functions conducted in this research. These goodness-of-fit techniques are difficult to quantify objectively on what makes a "good" emulation in terms of being able to fit to a discontinuous function like the Heaviside. The goodness-of-fit techniques should measure certain properties from the discontinuous function such as the gradient at the point of the discontinuity.

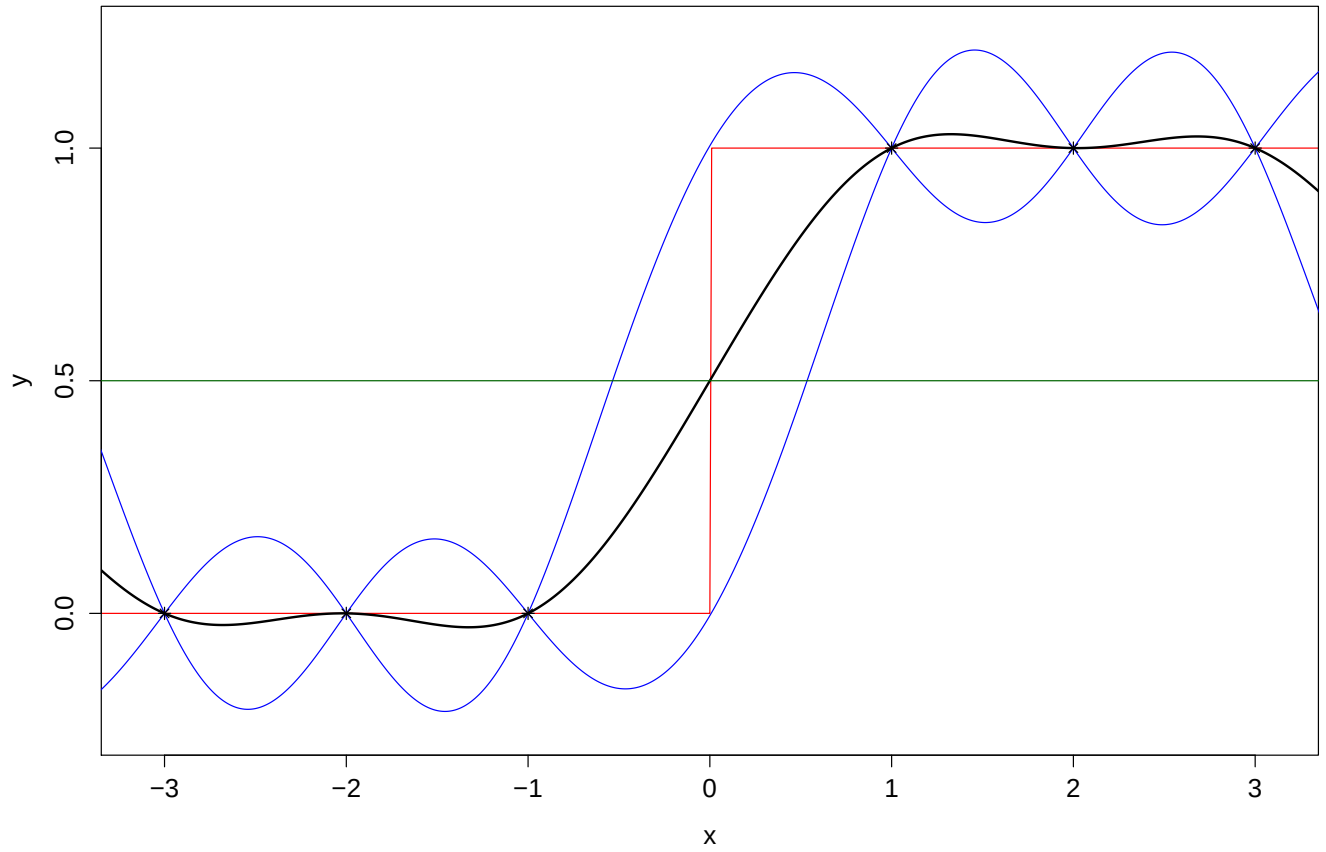

Figure 3.1: A simple example of the emulator modelling the Heaviside function with the data points symmetric.

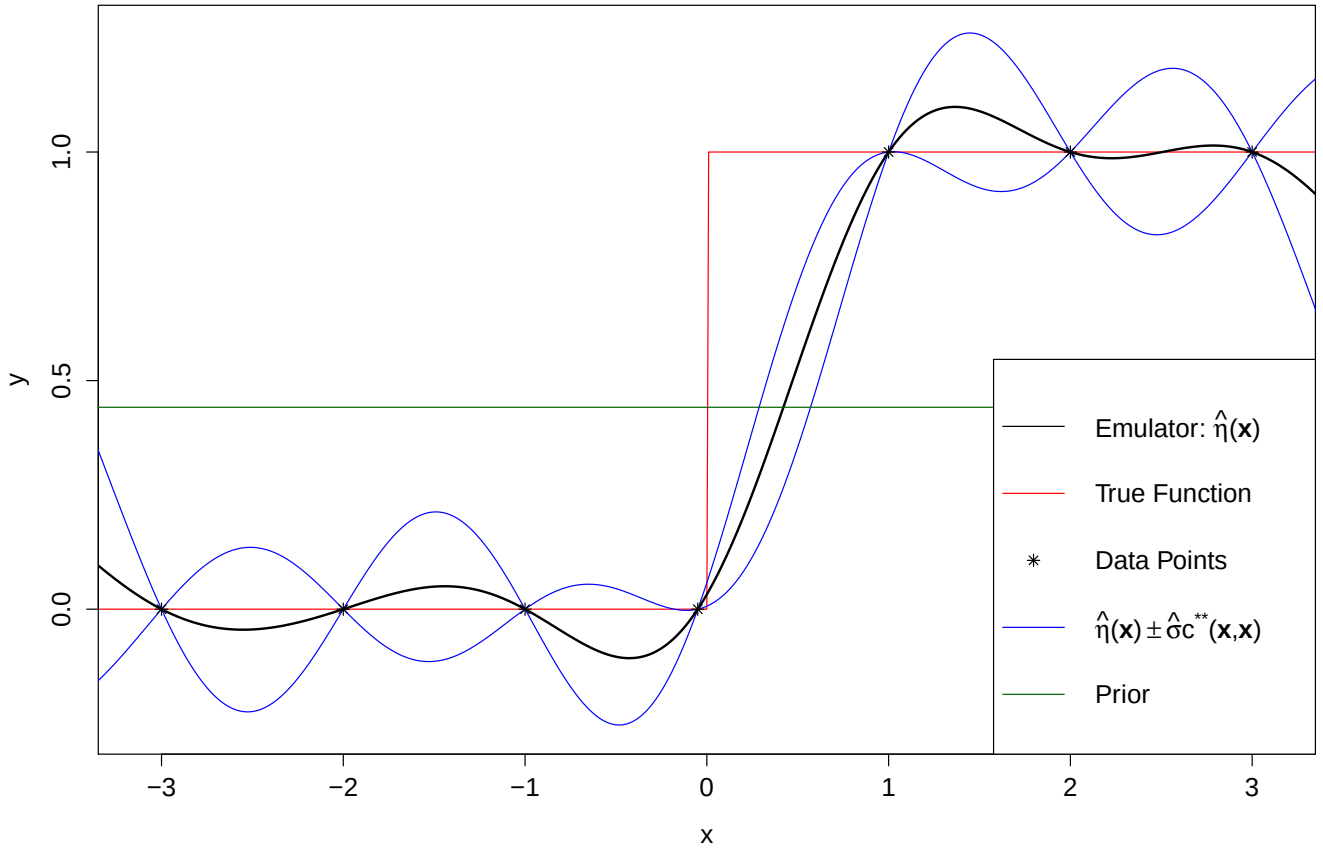

Figure 3.2: An example of the emulator modelling the Heaviside function with one data point close to the discontinuity:
Where; $-3 \leq x \leq -0^{-}$ and $1 \leq x \leq 3$ the emulator is a reasonable representation of the Heaviside function as if the true function is smooth. For this Heaviside function the slope is always positive between the two data points on either side of the discontinuity (even though we don't know where it is) and there is no over/under shooting of the function between these points.

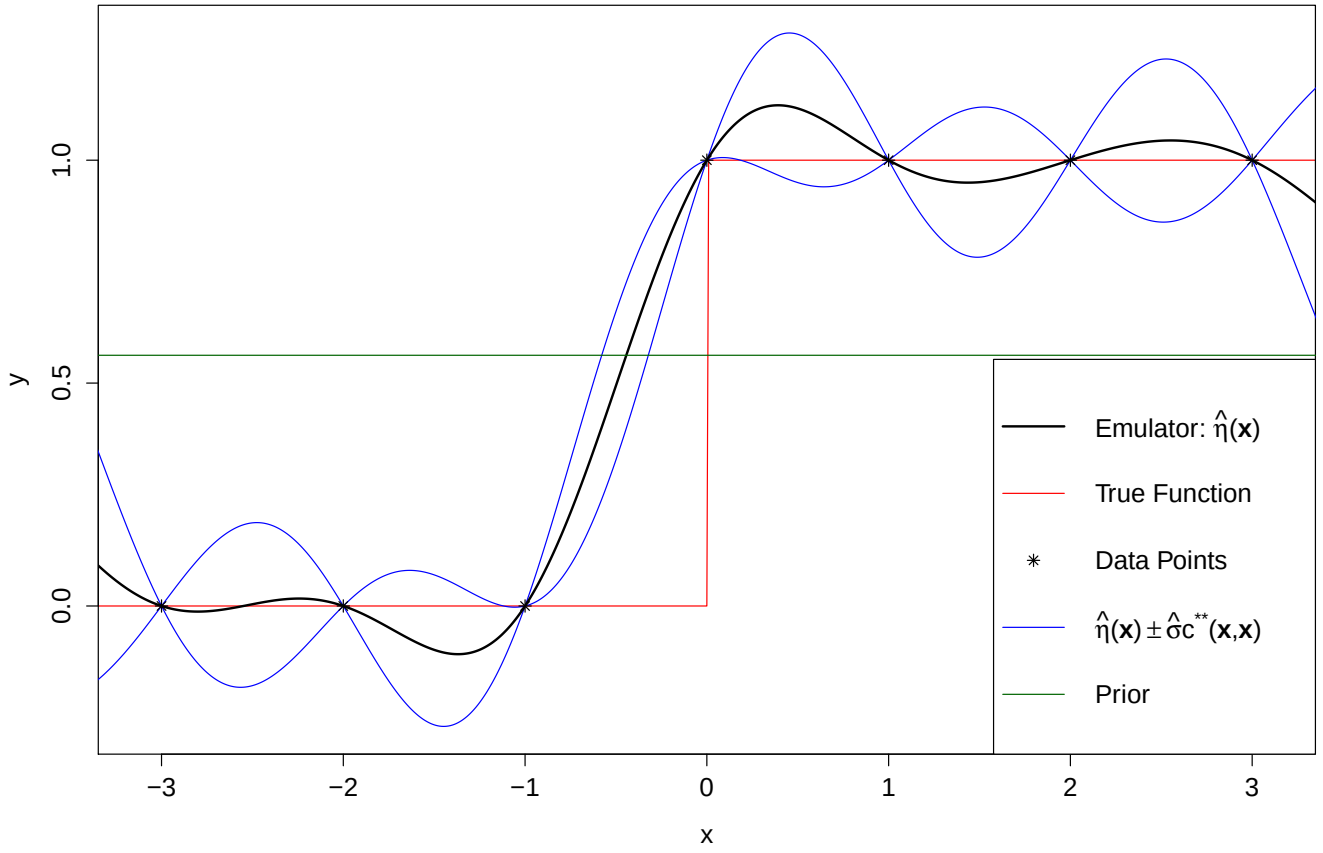

Figure 3.3: An example of the emulator modelling the Heaviside function with one data point close to but on the other side of the discontinuity: The only change from Figure 3.2 is the one data point at the discontinuity. Between the domain of 0 to 3 it can be seen that the emulator is a reasonable representation of the Heaviside function. As expected, the emulator will always converge to the prior outside of the data points due to the lack of information (no data points).

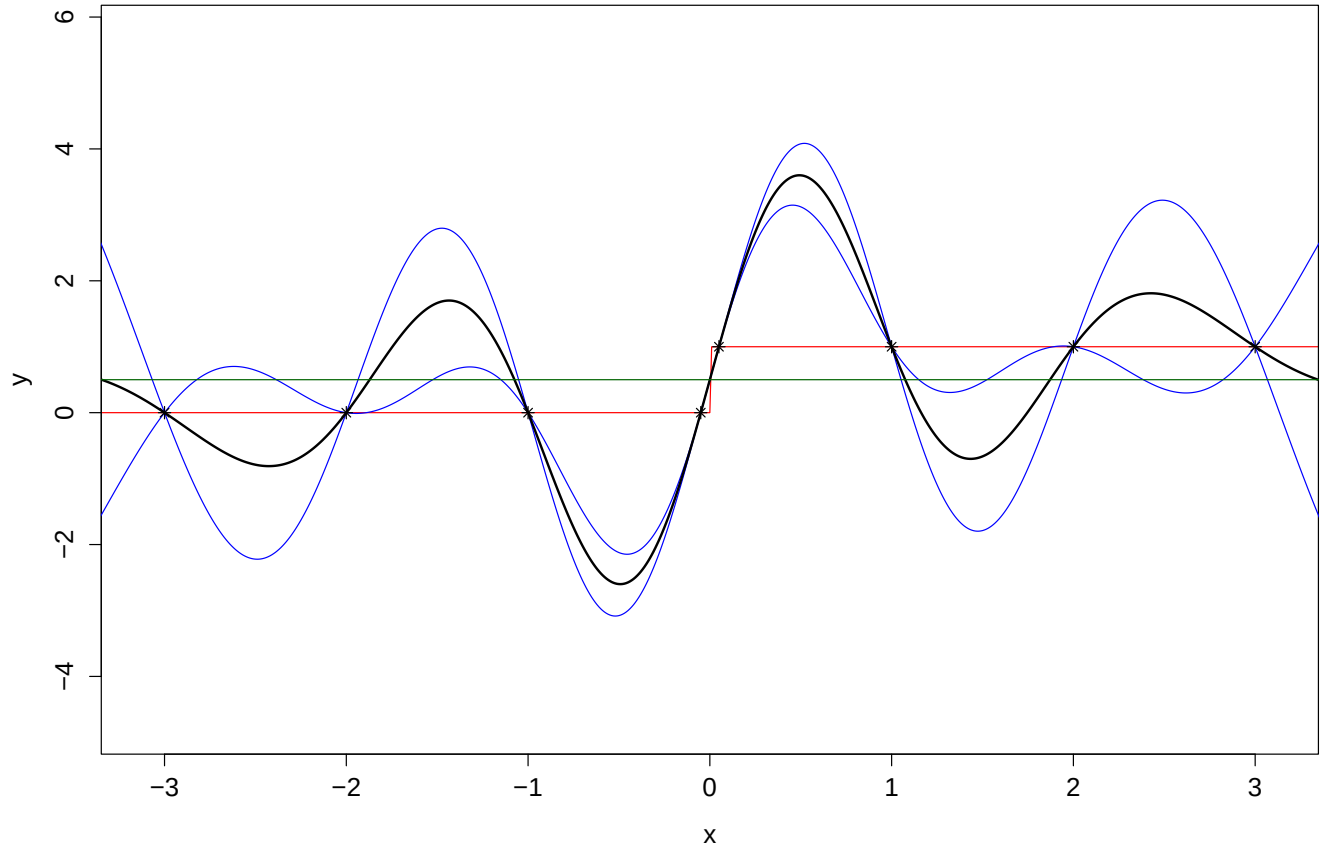

Figure 3.4: An Example when using two data points close on either side the discontinuity (note the y-axis and the extreme over/under shooting, legend is same as Figure 3.1):
The two data points are very close on either side to the discontinuity, giving a simple strategy of using the data points to manipulate the emulator,as if the location of the discontinuity is almost certain. This has made the emulator worse compared to Figure 3.2 and 3.3, causing the emulator to be very inaccurate in predicting the function with a lot of over/under shooting The slope of the emulator is high at the point of the discontinuity which has caused the extreme over and under shooting and the closer the two points are the larger the slope would be.

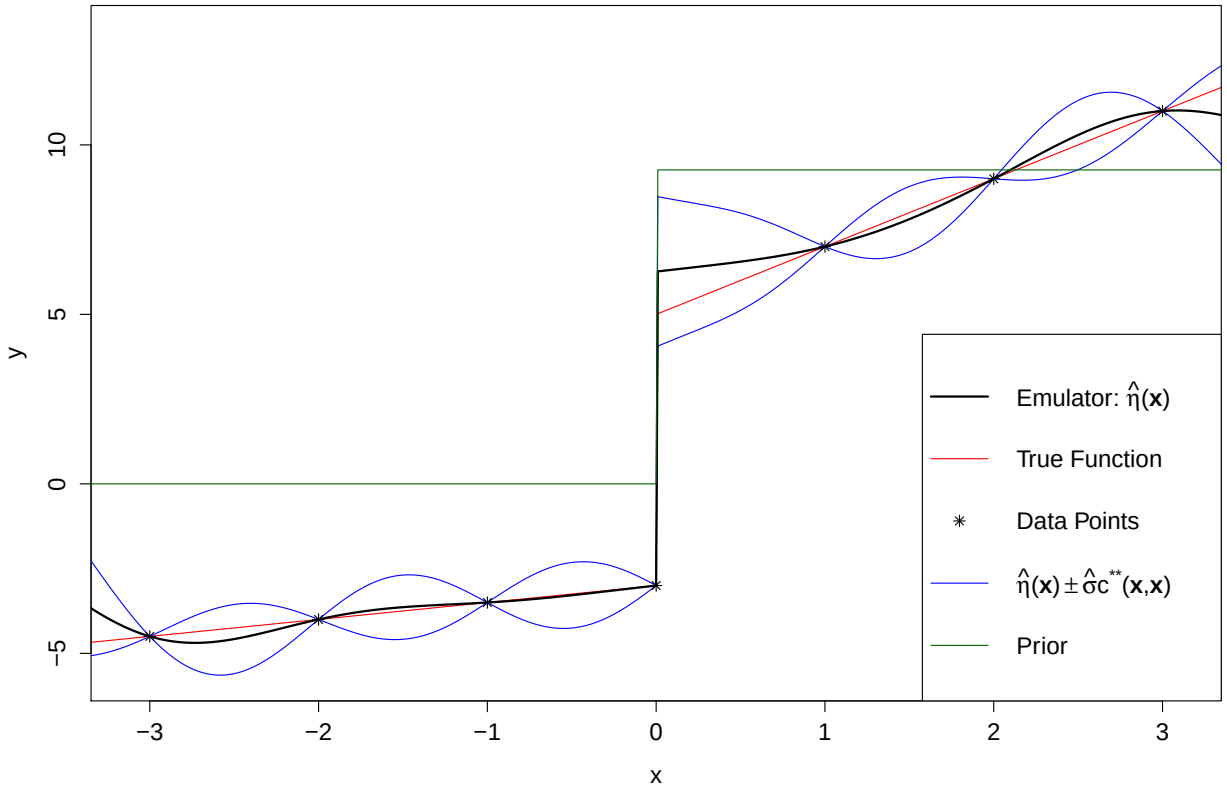

Figure 3.5: An Example of the emulator with a discontinuous prior. Note that the position of the discontinuity has to be known in order to define the regressor function.

If we were to know the location of the discontinuity, then everything changes; the current setup from Figures 3.2 to 3.4 with the regressor and correlation function might have to change. To demonstrate this, let $\eta(x) = \begin{cases} \frac{x}{2} - 3 & x \leq 0 \\ 2x + 5 & x > 0 \end{cases}$, and the regressor function to be $h(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$ (Note that at the discontinuity; η jumps from -3 to 5)

As we can see from the graph, the domains outside of the data points of the discontinuity ($-3 \leq x \leq 0$ and $1 \leq x \leq 3$) the emulator is a reasonable representation of η, however the emulator overshoots between 0 and 1.

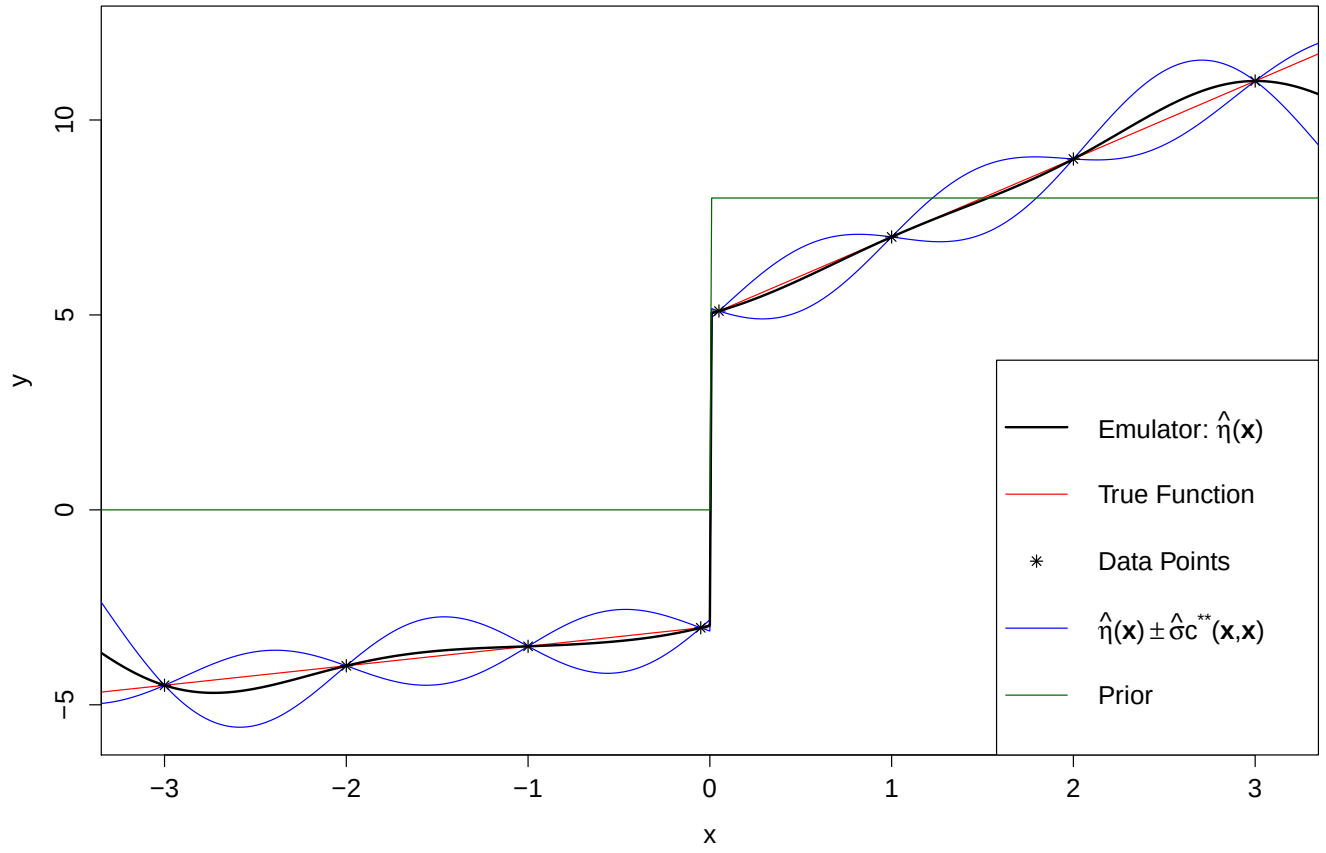

Figure 3.6: Knowing that the discontinuity is at 0 we obtain the above figure. Because the data points either side of the discontinuity are close together there is very little over/under shoot. This shows that, if the data points are "stitched" at the discontinuity; it could be suitable to use a regressor that contains the discontinuity at the stitched location. (as compared to Figure 3.4).

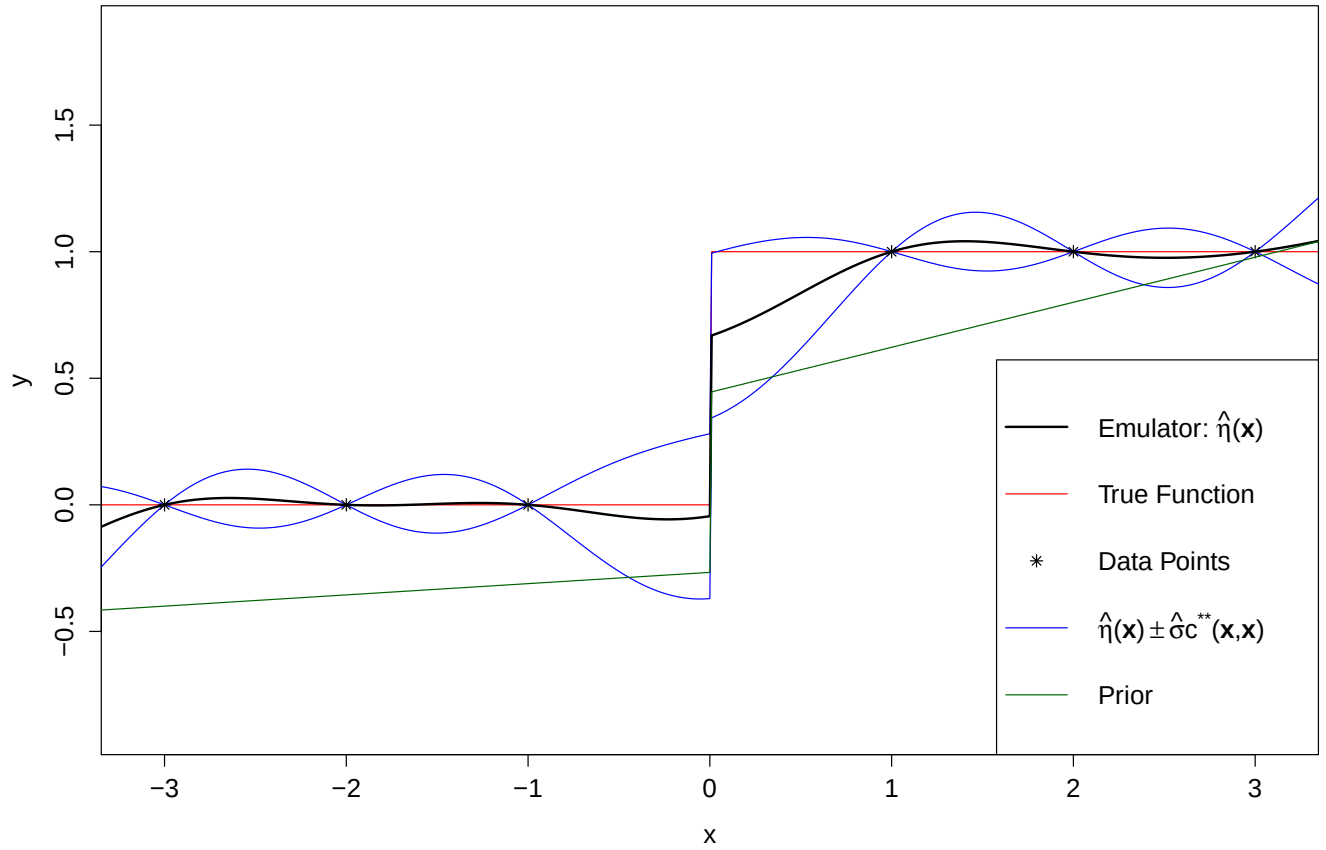

Figure 3.7: An example of using a discontinuous regressor to model the Heaviside function:
It is observed that with this discontinuous regressor, the emulator is a better representation of the Heaviside function compared to the constant regressor. This is because information of the discontinuity has been provided to the emulator via the prior.

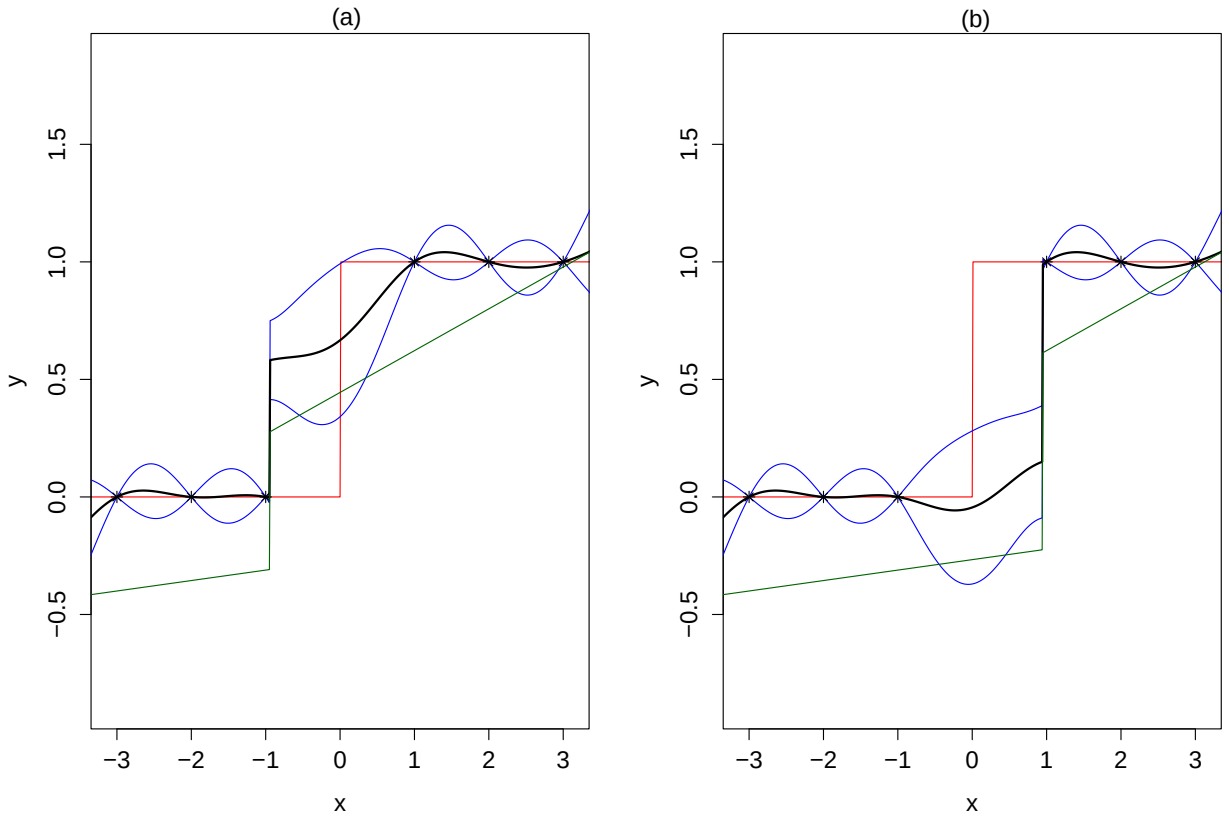

Figure 3.8: An example of using a discontinuous regressor to model the Heaviside function, assuming that the location of the discontinuity is unknown (legend is same as Figure 3.1):
If the location of the discontinuity is unknown, then the only information available is the data points. This figure demonstrates two extreme ends, assuming the discontinuity would be between the two data points in which "jumps" from 0 to 1. Comparing this figure to Figure 3.1 it could be suggested that using a discontinuous regressor when the location of the discontinuity is unknown, does not significantly improve the emulator as modelling the Heaviside function.
The relationship between Figures 3.8 and 3.8b is more complicated than between Figures 3.2 and 3.3.

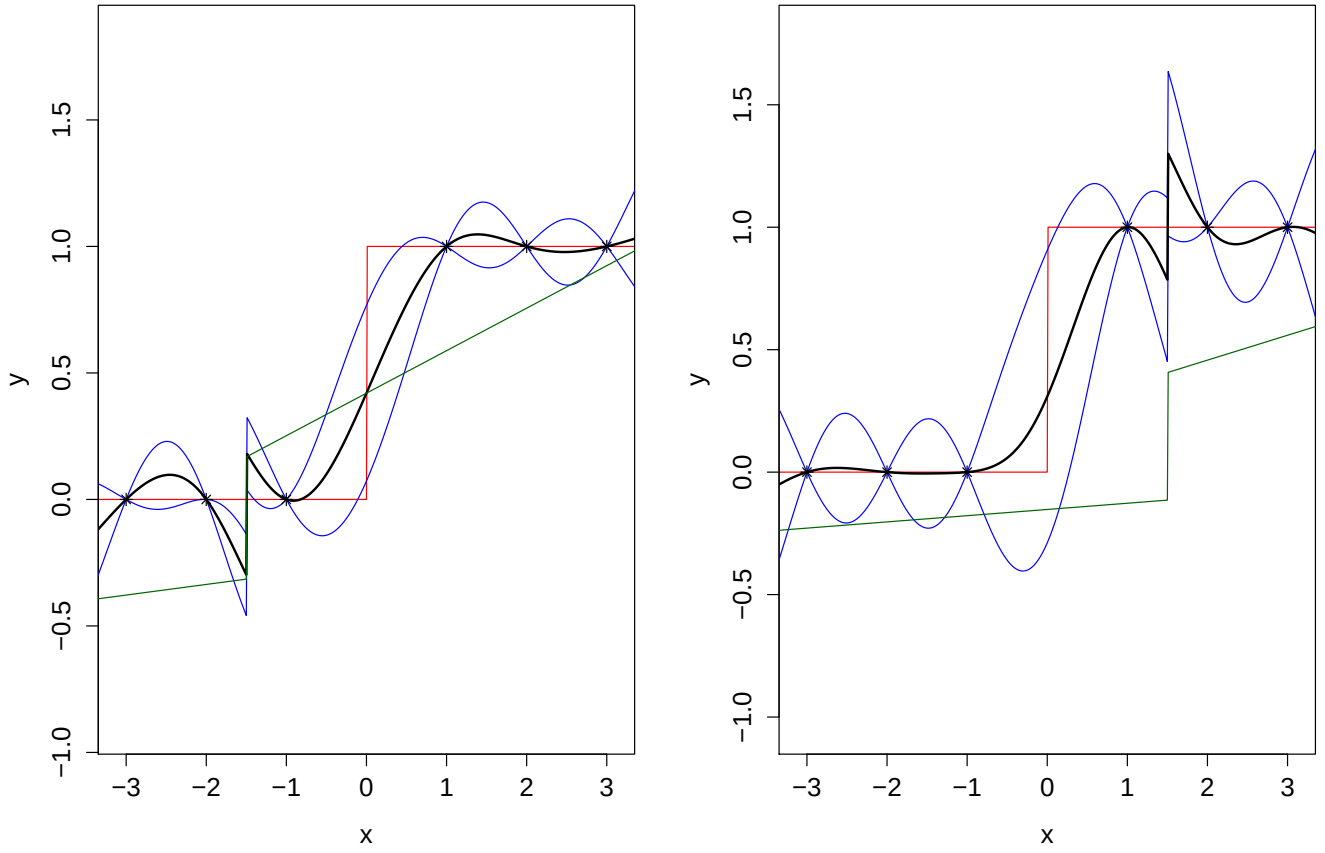

Figure 3.9: An example of using a discontinuous regressor that conflicts the data points to model the Heaviside function (legend is same as Figure 3.1):
When the data points are ignored to estimate where the discontinuity might be the emulator could result in something similar to the above figure, that shows that, any information conflicting with the prior will result in a high level of errors for the emulator causing over/under shooting.

Chapter 4

Research Method for Quantitative Analysis

4.1 Introduction

This chapter will discuss the approach to answering the thesis questions (Section 1.2) by introducing the assumptions made for the setup of the experiment and an outline of the experiments reported in Chapter 5. Throughout the experiments, goodness-of-fit techniques were used to measure how well the emulator handled simulators with step discontinuities.

For the experiments R (A programming language commonly used in statistical computing) was used in conjunction with the BACCO package, which contains the Bayesian emulation formulas discussed in Section 2.5.5.

4.2 Research Approach

4.2.1 Experiment Setup and Assumptions

The following sub-sections describe the components and assumptions required to set up the experiments for this thesis.

Regressor Function

Figure 4.1 is an example of why a linear prior ($H[X] = [1, X]$) will not be used for the findings because beyond the data points, the prior has the main influence on the emulator due to the lack of information. Figure 4.2 is an example of the emulator with a constant as its prior. When there is a lack of information from the simulator's data points, the emulator's result will come from the prior. In this thesis a constant prior is used because of the lack of trend from the discontinuous functions.

Because the step discontinuity functions have no trend, it could be suggested that

for the Heaviside function, a regressor function of $H[X] = [1]$ would be appropriate[1]. The 1 is an arbitrary non-zero constant number as $\hat{\beta}$ (see equation 2.5) will be adjusted to best fit the data ($H[X] = [1, X]$ would also be equivalent to $H[X] = [100, 2X]$).

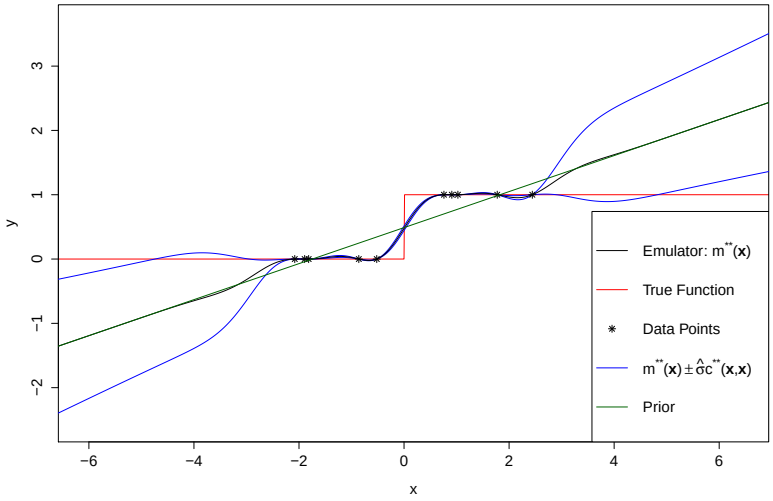

Figure 4.1: Example of the emulator using a linear prior

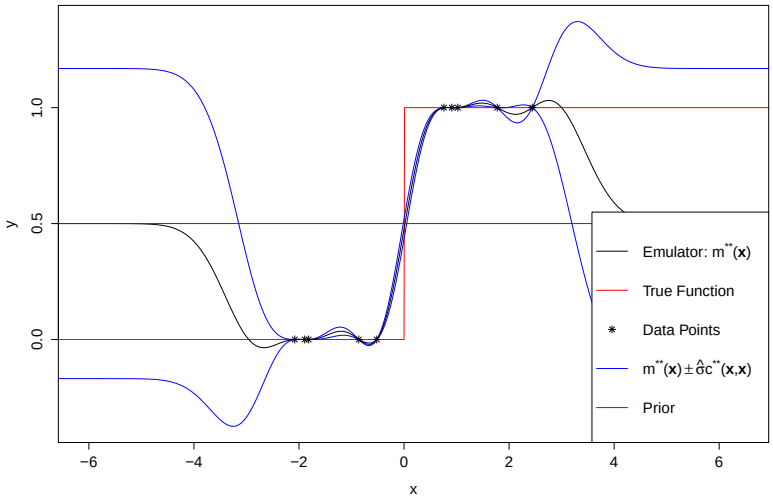

Figure 4.2: Example of the emulator using a constant prior

The emulator has no prior knowledge of the location of the discontinuity (see Section 3.5 on page 21 for more details); this would mean that using a function with a discontinuity in the regressor would be unsuitable, because the prior $(\mathbf{h}(\mathbf{x})^T \hat{\beta})$ is linear (see Section 3.5 for a detailed demonstration). The simulator itself is able to better approximate this discontinuity by running the simulation multiple times. However, this would take a considerable amount of time, whereas the emulator

[1]The conclusion to use a no trend prior was made during the finding of optimising B in Appendix B.

is quicker to run and should be able to give an estimation of the discontinuity's location.

Sample Space and Distribution for the Data Points

During the experiments, data points for the simulator were randomly drawn from a uniform distribution. This is because it is assumed that there is no (or very little) prior information on the location of the discontinuities, and therefore it is assumed the position of the discontinuities could be anywhere. However, it is assumed that there does exist some prior knowledge of the position of the discontinuities, this is to minimise the number of runs of the emulators where the location of discontinuity does not exist, therefore a suitable range (typically between -3 and 3) is chosen where the probability of encountering the discontinuity is high.

Sampling Data Points vs Sampling Discontinuity Positions

For this thesis, because the locations of the discontinuities are unknown, for the experiments the discontinuity was fixed; however, this information was not given to the emulator but was used in testing how well the emulator performed. If the data points were fixed and the position of the discontinuity were to change, there would be times when the emulator has repeating results. When there has been a change in the simulator but not in the data points output from the simulator, there would be no change in the emulator, as no new information has been given to it. Figure 4.3 demonstrates this, showing that there would be only a limit of six different emulators (or the number of data points plus one). Sampling data points provides more information about how well the emulator models discontinuities than sampling discontinuity positions.

Correlation Function

The correlation function used throughout the research is:

$$c\left(x, x'\right) = e^{(x'-x)B(x'-x)} \tag{4.1}$$

This correlation function may be better suited for continuous functions and a more suitable correlation for discontinuity functions may be possible. However, due to the lack of research in modelling discontinuity functions with Bayesian emulation, this thesis will be focusing more upon the emulator and its well-researched methods, rather than adjusting the correlation function. This correlation function has been used in several research papers (Caiado & Goldstein, 2015; Chen et al., 2015; Conti et al., 2009; Hankin, 2012; M. C. Kennedy & O'Hagan, 2001; Montagna & Tokdar,

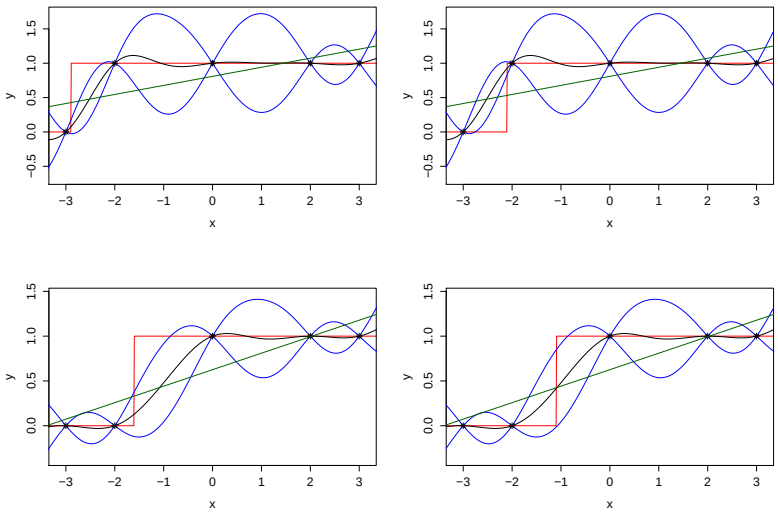

Figure 4.3: Examples of emulators if data points were fixed and Discontinuity Positions were sampled

2016; Oakley, 2002; Zhang et al., 2015) and will be used in this thesis to demonstrate the advantages and limitations of the Bayesian emulator.

B from the correlation function is a diagonal matrix and describes how "rough" the function is. According to Oakley (1999) there is no analytical way of calculating B.

In this thesis, because the Heaviside has no length scale, B will not be optimised, as the research focus is on modelling discontinuity functions with Bayesian emulation. Therefore B in most cases will be equal to 1 or an identity matrix in multiple dimensions.

A short experiment is reported in Appendix B concluding that, to avoid an added complexity, for this research B is best kept at 1. If B is too high, the correlation function will pull the emulator to the prior (similar to Figure 4.7) and if B is too low, the prior has little effect on the emulator, causing a high error for $\hat{\eta}$ outside the data point range and between the data points (as shown in Figure 4.6).

It should also be noted that the Heaviside function has no length scale, therefore having the data points scaled using the equation $X\sqrt{B}$ would provide equivalent emulators. For example, Figure 4.4 is the output when the data points are $(-3, -2, 0, 2, 3)$ and B is 1, and Figure 4.5 is the equivalent emulator, yet only the x-axis has been scaled, B has been changed to 20 and the data points are scaled as $\left(\frac{-3}{\sqrt{20}}, \frac{-2}{\sqrt{20}}, 0, \frac{2}{\sqrt{20}}, \frac{3}{\sqrt{20}}\right)$. Because the data points for the experiments in this thesis can be scaled it was found it be best to let B to be equal to 1.

4.2.2 Order of Experiment with the Emulator

This thesis will mainly be focusing using numerical experiments of the Bayesian emulator. The first part of the experiment will run the emulator modelling a simple

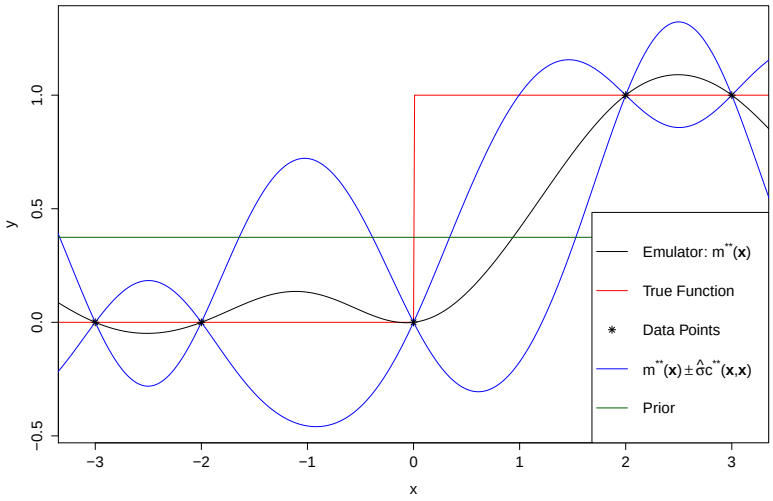

Figure 4.4: Example of emulator with contain prior and $B = 1$

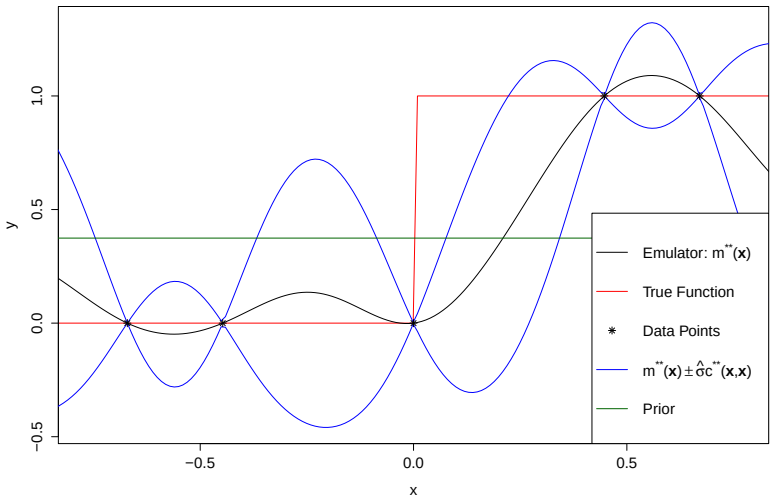

Figure 4.5: Example of emulator with containt prior and $B = 20$ and the data points scaled

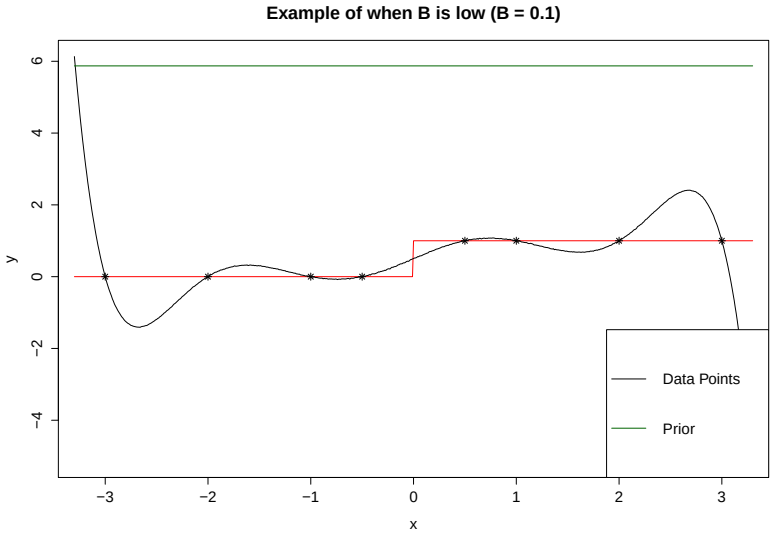

Figure 4.6: Example of when B from the correlation function is low

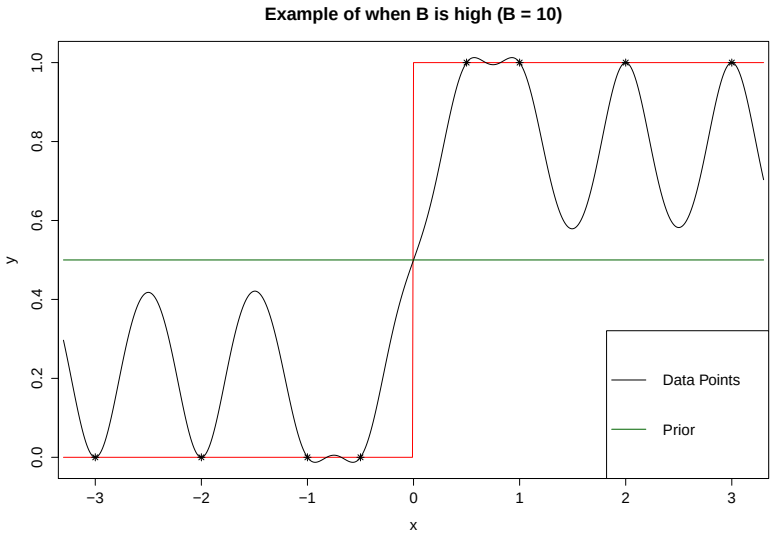

Figure 4.7: Example of when B from the correlation function is high

discontinuity function, the Heaviside function, similar to Figure 1.1.

The Heaviside function will be the first function which the emulator will model. Testing the emulator with the Heaviside function allows a basic understanding of the emulator's behaviour when modelling discontinuities and how the emulator is able to approximate where the discontinuity is. If the emulator is unable to handle this simple function, then it can be expected that it will not be able to handle more complex functions with discontinuities.

Next the dimensions of the function will be increased, containing a region where it is discontinuous. The findings from the one-dimensional case will be used to guide the research when applied to this simple two-dimensional case. A circle was chosen because it could be seen as a simple representation of the flow of a heavy gas at a fixed point in time.

Once an understanding of how the emulator deals with the two dimensional case the experiments will then extend this to three-dimensions. The third dimension will be time, using the same function as before, but having the region of the circle increase. It will be assumed that the gas is released from a source (e.g., a tank) continuously (not instantaneously), and because the tank's internal pressure will decrease as it releases the gas, the rate at which the gas escapes is proportional to the amount of gas remaining in the tank. Using this assumption will create a simple function that could be represented as a heavy gas spreading outwards from a source.

4.2.3 Goodness of Fit

This thesis has presented examples of goodness-of-fit techniques and will be used to measure the emulator's estimation to the discontinuous function. These results will be used to compare and evaluate how the emulator has been able to model the functions.

Given $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is the set of training runs of the simulator containing n observation points (or data points) from the simulator and $\mathbf{x}_m$ is m unique test points[2] that will be used to test the emulator and are ideally evenly spaced[3].

$$\mathbf{x} = \{x_1, x_2, \ldots, s_m\} \text{ where } x_{i+1} = x_i + \Delta, \Delta = \frac{B-A}{m}, x_1 = A \text{ and } x_m = B$$

Below are examples of goodness-of-fit techniques that will be used in this thesis. A probability density function (PDF) and a cumulative distribution function (CDF) will be produced for each technique. In Appendix A it has been shown that an analytical solution would be difficult to obtain, and therefore these solutions were calculated numerically.

- Mean Square Error (MSE):
  To find the MSE, the following formula is used:

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} \left[\hat{\eta}\left(x_i\right) - \eta\left(x_i\right)\right]^2 \tag{4.2}$$

  where s_i is the i^{th} element of the test points that has not been observed by the simulator.

  As m gets large the summation approaches into an integration:

$$\lim_{m\to\infty} \sum_{i=1}^{m} \left[\hat{\eta}\left(x_i\right) - \eta\left(x_i\right)\right]^2 = \int_{\mathcal{D}} \left[\hat{\eta}\left(x\right) - \eta\left(x\right)\right]^2 dx \tag{4.3}$$

  Because the emulator: $\hat{\eta}$ will become more inaccurate outside the data points range. The MSE will only be measured between the most upper and lower boundary of the data points value. For example, if the data points are $-5, 2.3, 0.1, 2.8, 7$ then the MSE will be calculated between -5 and 7, from the equation $\mathcal{D}$ is this range. By measuring the MSE along this region, the research will suggest how accurate the emulator is when modelling a discontinuous function. Ideally, a suitable accurate emulator will have a low error value.

[2]These test points typically have not yet been observed by the simulator but will be used within the emulator.

[3]If the test points are evenly spaced it might provide a more accurate result.

- Comparing $\hat{\eta}(d)$ to $\eta(d)$:
  For a Heaviside function; given the discontinuity is at d, it would be expected that $\hat{\eta}(d) = \eta(d) = 0.5$. This technique will also provide information of any under/overshooting.

- Approximating and comparing the locations of the discontinuities from the emulator by solving $\hat{\eta}(x) = \eta(d)$ for x where d is the discontinuity position.

  It will be expected that when an accurate emulator will have $\hat{\eta}(d) = 0.5$, for the Heaviside function (and the other functions that are to follow in this thesis). This is because, at the discontinuity, the result from true function may not exist (or at least an extreme difference on each side of the discontinuity) therefore it is assumed that the position of the discontinuity is at the midway point near the discontinuity such that: $\frac{\lim_{x\to a^+}\eta(x) + \lim_{x\to a^-}\eta(x)}{2} = \frac{1+0}{2} = 0.5$. Therefore for this research, $\hat{\eta}(x) = 0.5$ will be solved for x, and x will be compared to the location of the discontinuity d. Ideally, a "good" emulator will have x at the discontinuity or near the discontinuity with a low variance.

- Jaccard index

  To measure the accuracy of the emulator, equation 4.4 will used to post-process the emulator. This will provide a percentage of overlapping from the simulator and the post-process emulator (equation 4.4). Region D is not used in the equation, as by increasing the sample space the accuracy of the emulator can be easily influenced. (Real & Vargas, 1996)

$$g(\mathbf{x}) = \begin{matrix} 1 & \text{if } \hat{\eta}(\mathbf{x}) \geq 0.5 \\ 0 & \text{if } \hat{\eta}(\mathbf{x}) < 0.5 \end{matrix} \tag{4.4}$$

  This can also be represented as in table 4.1.

	$\hat{\eta}(\mathbf{x}) \geq 0.5$	$\hat{\eta}(\mathbf{x}) < 0.5$
$\eta(\mathbf{x}) = 1$	B	A
$\eta(\mathbf{x}) = 0$	C	D

Table 4.1

  $g(\mathbf{x})$ (from equation 4.4) will be compared to the simulator[4], and as a result it is assumed that the discontinuities exist the point $g(\mathbf{x})$ has a discontinuity (i.e. when $\hat{\eta}(\mathbf{x}) = 0.5$). When solving $\mathbf{x}$ in $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d})$, a Jaccard index approach will also be used, by dividing the results from the emulator into four different regions[5]:

[4]Figures 5.22 (page 57) and 5.33 (page 64) are two examples used in this thesis
[5]A visual representation of this can be found in Figures 4.8 and 5.39 (page 42 and 68).

- Region **A**: where the simulator would output a 1 and the emulator has **incorrectly** estimated it to be 0
- Region **B**: where the simulator would output a 1 and the emulator has **correctly** estimated it to be 1
- Region **C**: where the simulator would output a 0 and the emulator has **incorrectly** estimated it to be 1
- Region **D**: where the simulator would output a 0 and the emulator has **correctly** estimated it to be 0

The equation $\frac{B}{A+B+C}$ will be used to measure how accurate the emulator.

- Investigating the steepness of $\hat{\eta}^{'}(0)$

With the step-discontinuous functions, at the position of the discontinuity there is an extreme change in the function. Given the discontinuity is at 0 it would be expected that steepness of $\hat{\eta}^{'}(0)$ (i.e. $\frac{d\hat{\eta}(\mathbf{x})}{d\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{0}}$) are extreme numbers. In conclusion, this thesis will answer the above questions by using the developed goodness-of-fit methods on the described functions such as the Heaviside function, with some assumption which focuses on step-discontinuities.

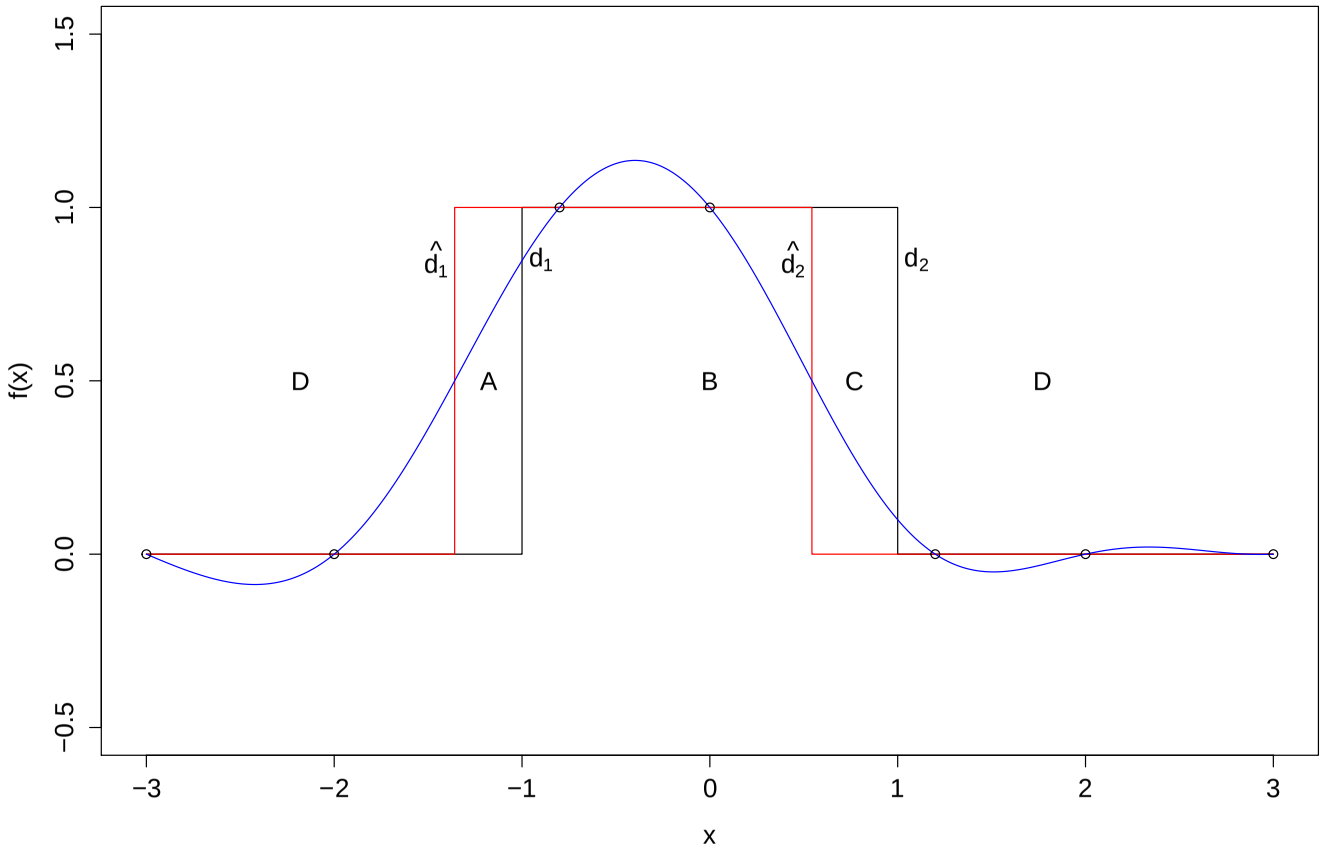

Figure 4.8: Example of Jaccard regions for one dimension with two discontinuities. note that the red line represents the approximation of where the discontinuous is based on the emulator (the black curved line) $\hat{\eta} = 0.5$

Chapter 5

Numerical Findings from the Emulator Experiments

5.1 Introduction

This chapter contains the findings from the experiments in using the Bayesian emulator to model simulators with one or more discontinuities.

Each experiment will be a slightly different function (yet not too complex) and will be treated as a deterministic simulator which the emulator is to model. A number of goodness-of-fit techniques (see section 4.2.3) will be applied to observe the performance of the emulator. The number of data points will also be increased to investigate how more data points from the simulator affect the emulator.

The Heaviside function, which is a simple function containing one discontinuity, will be used for the start of the investigation. If the emulator has a poor goodness-of-fit measure score when modelling a discontinuity with this simple function, then at least the same number of problems would also be expected when modelling a more complex simulator. This investigation with the Heaviside function will help progresses the research by suggesting adjustments to the emulator as the research progress to more complex simulators containing discontinuities.

5.2 One Dimension with One Discontinuity

Let $\eta(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$ be the Heaviside function. For the Bayesian emulator, the regression function will be: $H(x) = [1]$ and a correlation function: $c(\mathbf{x}, \mathbf{x}') = \exp\left(-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')\right)$, with $B = 1$. (see Experiment Setup and Assumptions in Section 4.2.1 on page 32 for reasoning).

Figure 5.1 shows an example when there is a large difference between the emulator's estimation and the true function.

The following data points were selected (not randomly chosen) to show how the

emulator is able to model the discontinuous function, a full discussion of this was covered in Chapter 3.

$$[(-3,0),(-2,0),(0,0),(2,1),(3,1)]$$

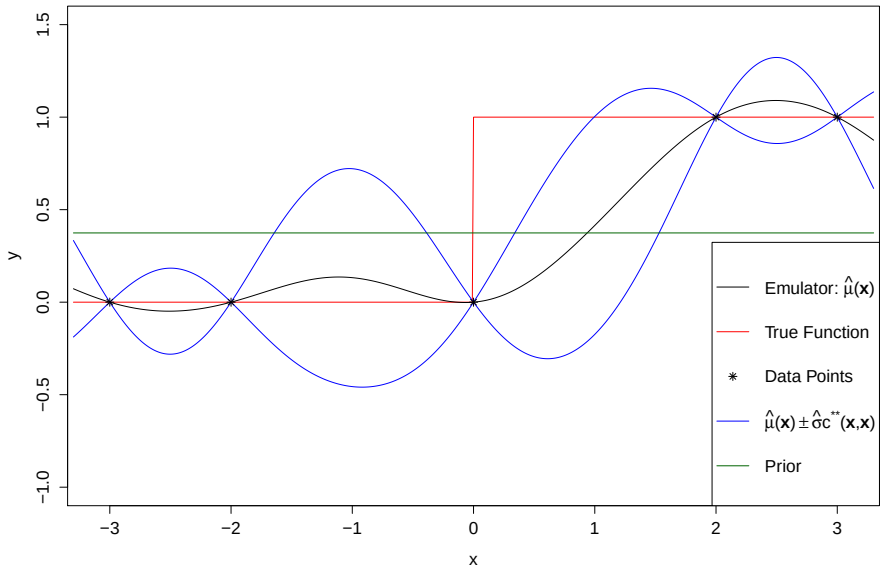

Figure 5.1: Bayesian emulator modelling Heaviside function with a constant as a prior: $H(x) = 0.5$

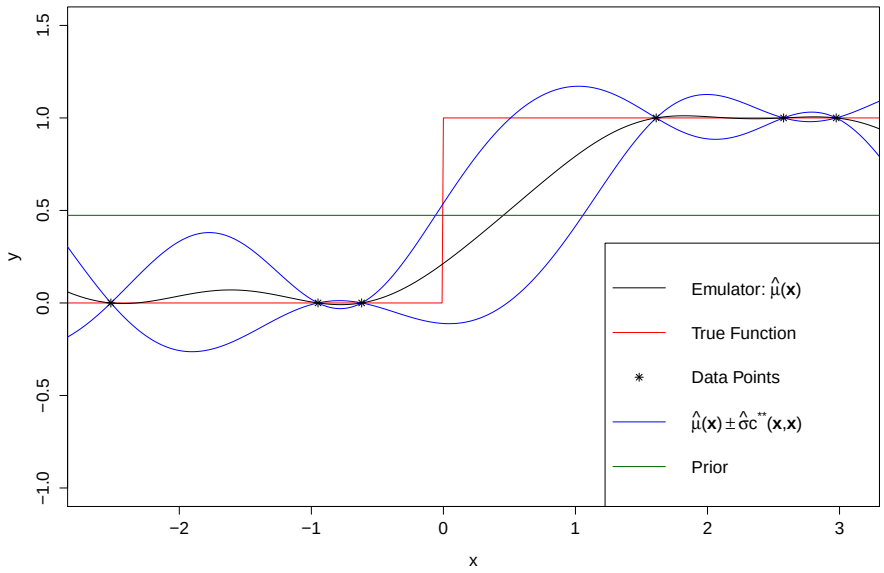

Figure 5.2: Another example of the Bayesian emulator modelling Heaviside function with a constant as a prior: $H(x) = 0.5$

An empirical PDF and a CDF will be used in the goodness-of-fit technique ap-

plied to the data obtained from 30,000[1] runs. To start with, each run will contain four data points from the true function; two data points on each end of the discontinuity. The data points are drawn from a uniform distribution within a domain (two between -3 and 0 and two between 0 and 3) so that every run has information that there may exist two discontinuities yet the location and the knowledge that they are discontinuities is withheld from the emulator. The goodness-of-fit technique will then be compared to the number of data points from the true function, and it should be expected that, on the average, more information will result in a more accurate emulator.

5.2.1 Mean Square Error (MSE)

The MSE was calculated within the region of the data sample space. This is because the outside the sample space the emulator will converge to the prior causing the MSE to be much higher if the experiments were to include too much of the region outside the data points causing the MSE to be dominated by the prior.

Figure 5.3 starts to indicate what the MSE's PDF might look like. When the number of data points is increased from four to eight, figure 5.4 is obtained.

One problem found when calculating the MSE was that sometimes the MSE would be very high and could be seen as an outlier. In one case where the data points were $[-2.19, -0.02, 0.02, 0.445]$, the MSE was 204. Figure 5.7 is an example demonstrating the emulator as these data points. This shows that when sampling it is important to try to keep the data points as evenly spread as possible.

Figure 5.5 shows a PDF (4 data points per run), and Figure 5.6 shows the CDF of the MSE.

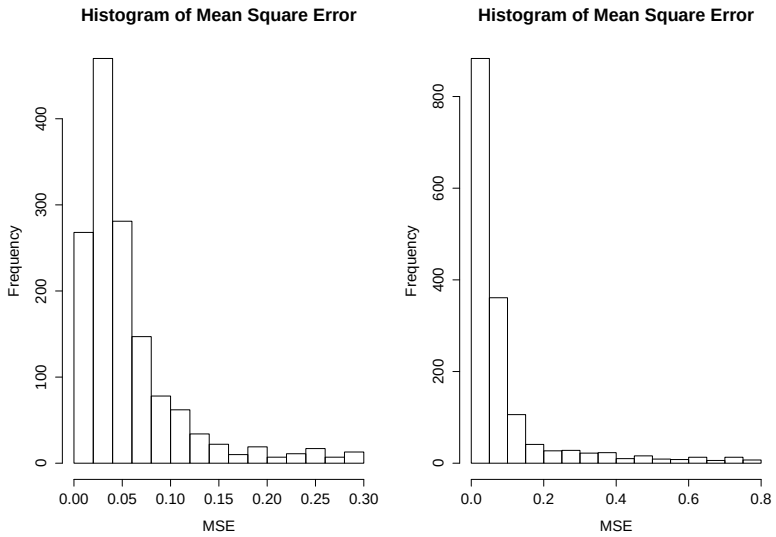

Figure 5.3: Histogram of MSE: right figure is zoomed out to show the skewness

[1]30,000 was chosen, as it is a representation of a large sample size that is a reasonable compromised based on the computer resources available. There are cases where this number was reduce 20,000.

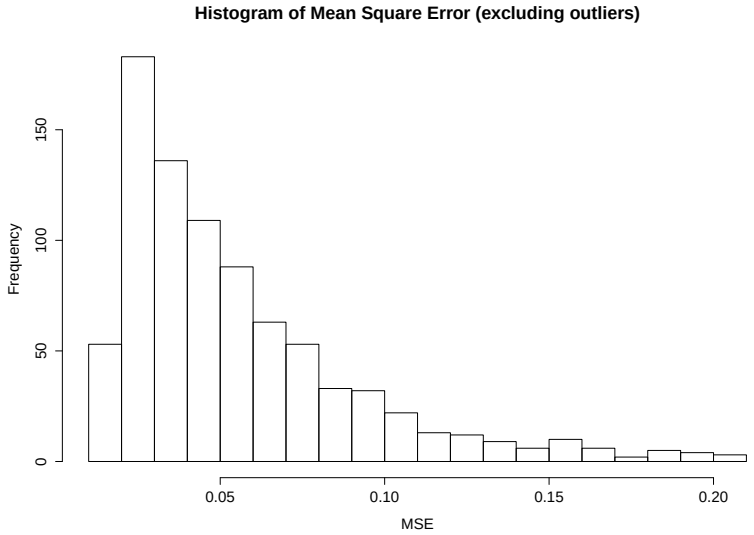

Figure 5.4: Histogram of MSE with eight data points

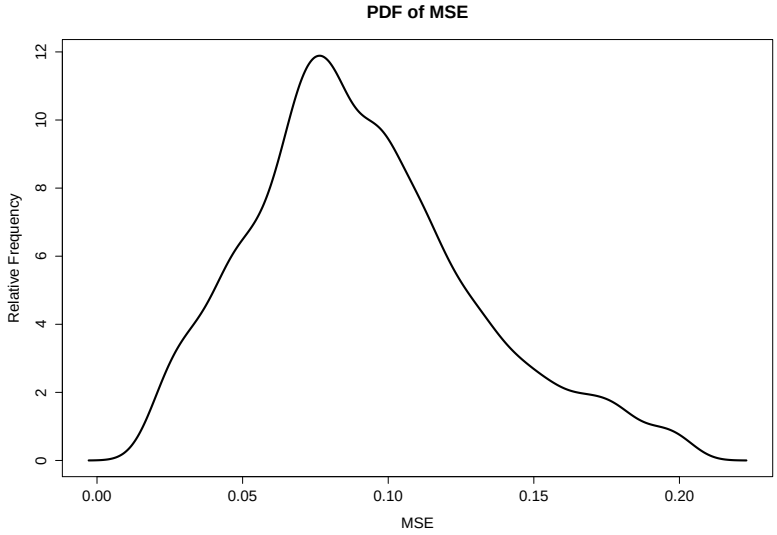

Figure 5.5: PDF of the MSE with four data points

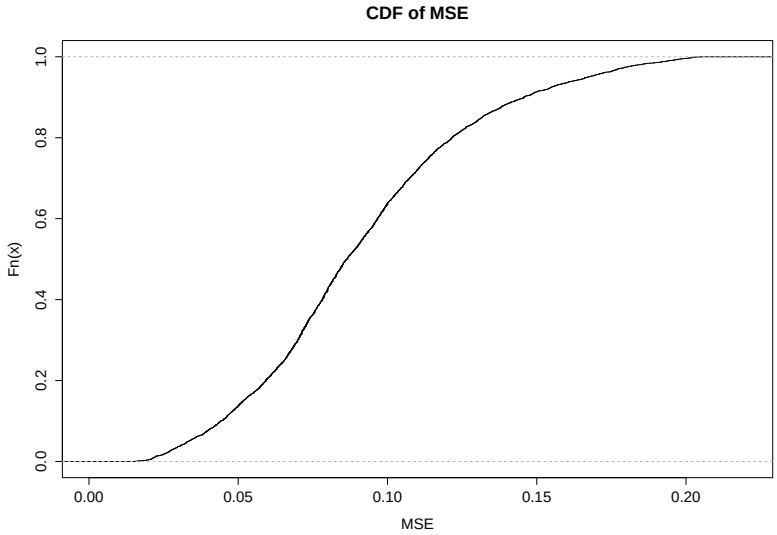

Figure 5.6: CDF of the MSE with four data points

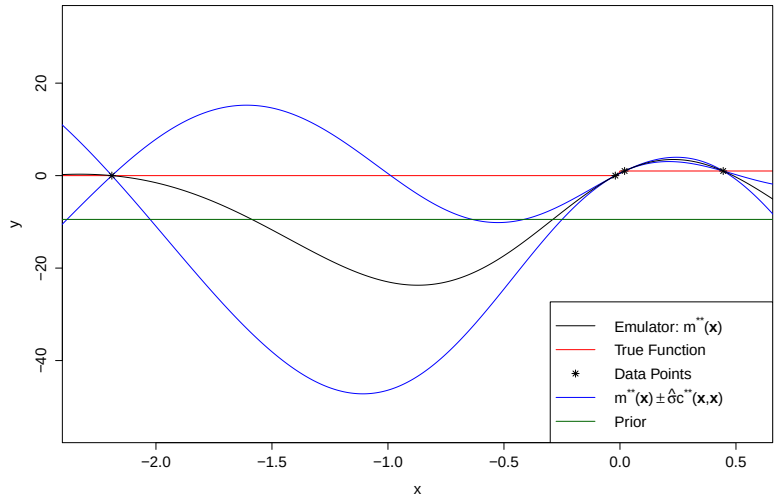

Figure 5.7: Example of the emulator when MSE is very high. Note the vertical axis

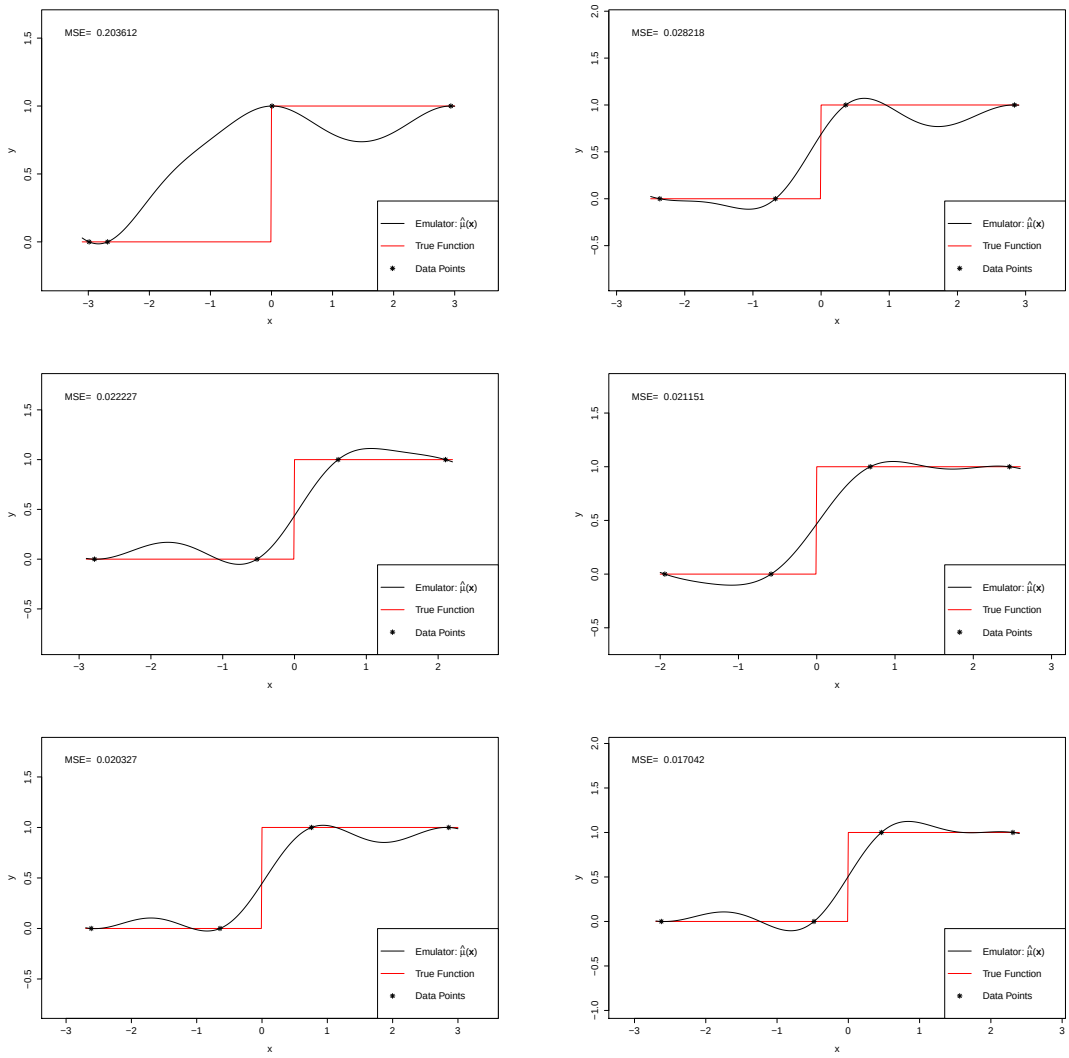

Figure 5.8: Example of the emulator when MSE is low. Figures show a progression of a better (lower) MSE

Figure 5.7 is an example of what happens when there are at least two data points that are widely spread apart causing the MSE to be high. Note from Figure 5.7 that the vertical axis shows the range between -50 to 20 when the true function is between 0 and 1. Figure 5.8 are examples of the emulator progressing to a low MSE. It can be seen that the datapoints are much more evenly spread compared to figure 5.7. The PDF of the MSE is skewed to the right with a few extreme numbers due to the large gap in the data (as shown in Figure 5.7). Because of the outliers, the median of the MSE will be taken. It was found that the MSE does decrease as expected when more data points are used in the emulator (see Figure 5.10 and 5.11).

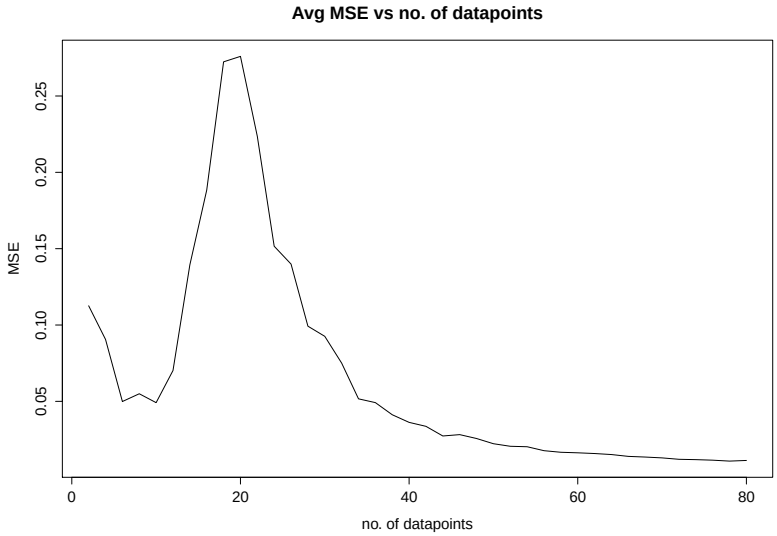

Figure 5.9: Average MSE vs number of data points

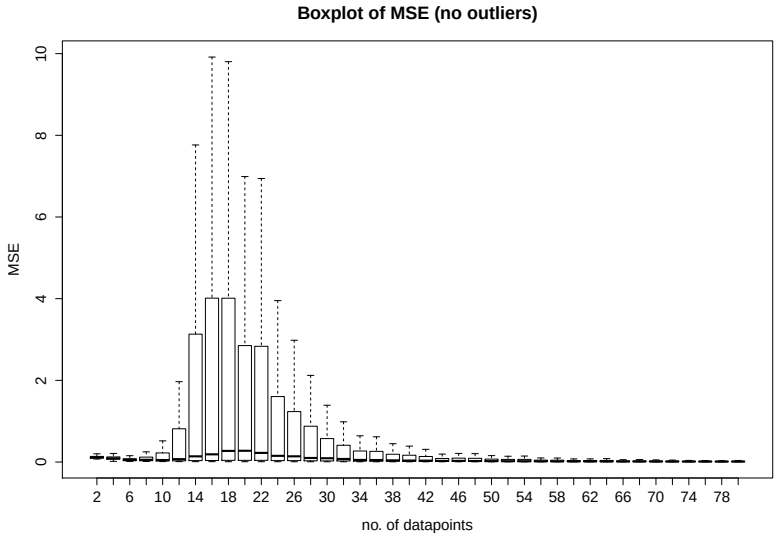

Figure 5.10: Boxplot of MSE vs number of data points

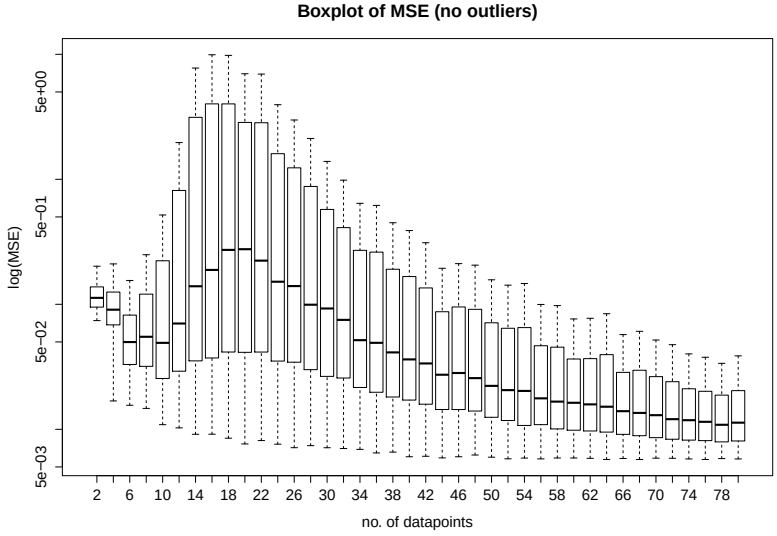

Figure 5.11: Boxplot of MSE vs number of data points - logged scaled

5.2.2 Approximating the Location of the Discontinuities from the Emulator

For this experiment using the Heaviside function, the position of the discontinuity d is at 0, and it is expected that $\hat{\eta}(d)$ will equal 0.5, the midway point of the emulator near the discontinuity. (For more information regarding this goodness-of-fit technique see Section 4.2.3 on page 40.)

As in the experiments described in Section 5.2.1, eight data points were drawn from a uniform distribution (four between -3 and 0 and four between 0 and 3). When solving $\hat{\eta}(\mathbf{x}) = 0.5$ numerically, a histogram (see Figure 5.12) was obtained with a mean of 0 and a variance of 0.11 (for this example). A Q-Q plot (see Figure 5.13) indicated that it was not from a normal distribution; however, it could be suggested that it is from a Laplace distribution $f(x|\mu, b) = \frac{1}{2b}e^{\left(-\frac{|x-\mu|}{b}\right)}$ with $\mu = 0$ and $b = \sqrt{\frac{\text{Var}(\mathbf{x})}{2}}$ which is better fitted to $\mathbf{x}$, as Figures 5.14 and 5.15 illustrate, showing an almost perfect match. A Kolmogorov-Smirnov test results in a p-value of 0.69 when comparing with a randomly sampled set from a Laplace Distribution with the same mean and variances. The variance of $\mathbf{x}$ will decrease if the number of data points is increased because the sample range of the data points is decreased.

This would indicate that $d \sim Laplace(0, b)$. One way of approximating b is to use a leave-one-out bootstrap method. From n-many data points, $n-1$ is sampled (without replacement) and $\mathbf{x}$ is solved from $\mathbf{x}$ for $\hat{\eta}(\mathbf{x}) = 0.5$. After repeating this multiple times μ is calculated by the median of x and $b = \sqrt{\frac{\text{Var}(\mathbf{x})}{2}}$.

When a simple method was used, such as $\hat{d} \approx \frac{(x_1+x_2)}{2}$ where x_1 is a known data point closest to the left side of the discontinuity ($\eta(x_1) = 0$) and x_2 is a known data point closest to the right side of the discontinuity ($\eta(x_1) = 1$), then as expected a mean of 0 was obtained with a variance of 0.12, a difference of only 0.01. The emulator was closer to the discontinuity 54.1% of the time. When increasing the range of the data points, this simple method is better at approximating $\mathbf{x}$ so the results are inconclusive at present.

As expected, the variance decreases the more data points there are as the gap between each data point, on average, would also decrease, Figure 5.16 shows this, showing the variance decreasing towards 0 as the number of points increases.

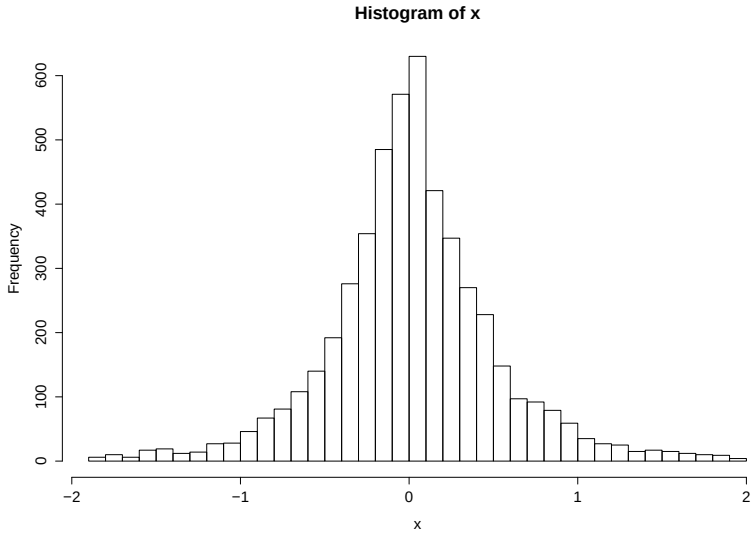

Figure 5.12: PDF of x from $\hat{\eta}\left(x\right) = \eta(d)$ over a Laplace PDF

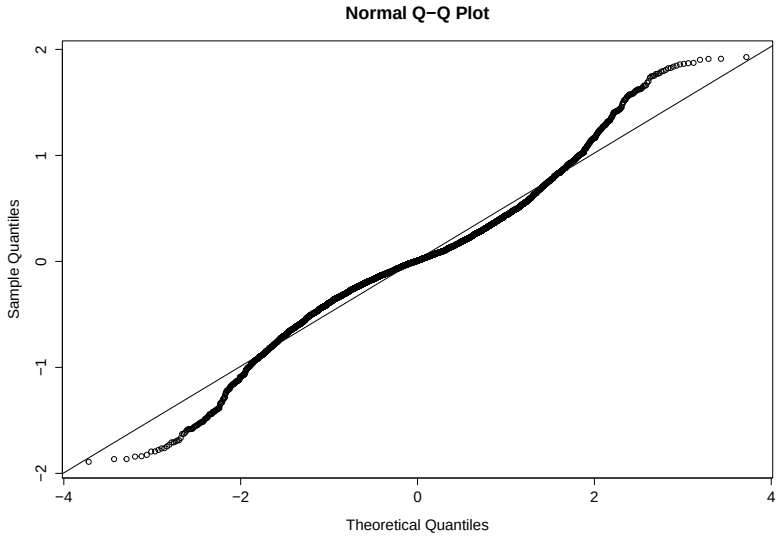

Figure 5.13: Normal qqplot of x from $\hat{\eta}\left(\mathbf{x}\right) = \eta(\mathbf{d})$

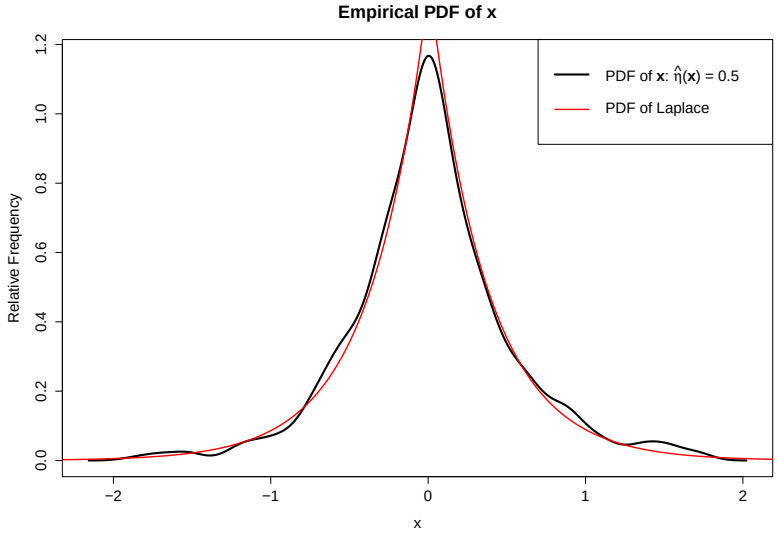

Figure 5.14: PDF of x from $\hat{\eta}\left(\mathbf{x}\right) = \eta(\mathbf{d})$ over a Laplace PDF

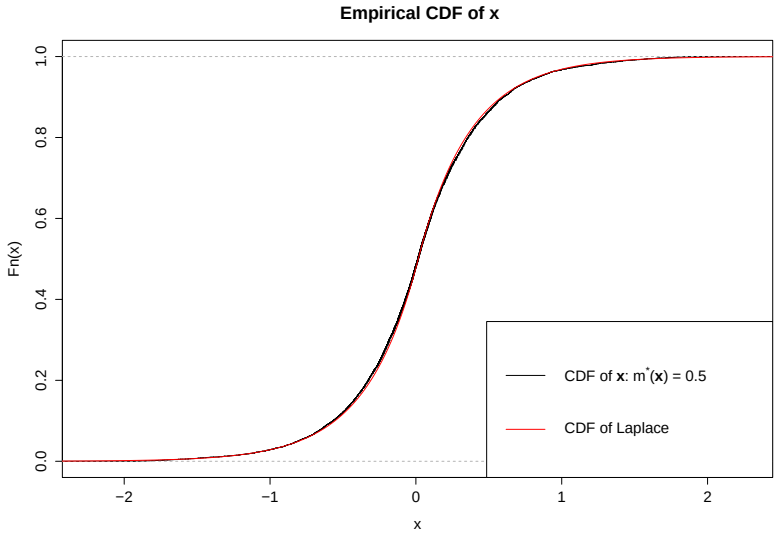

Figure 5.15: CDF of x from $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d})$ over a Laplace CDF

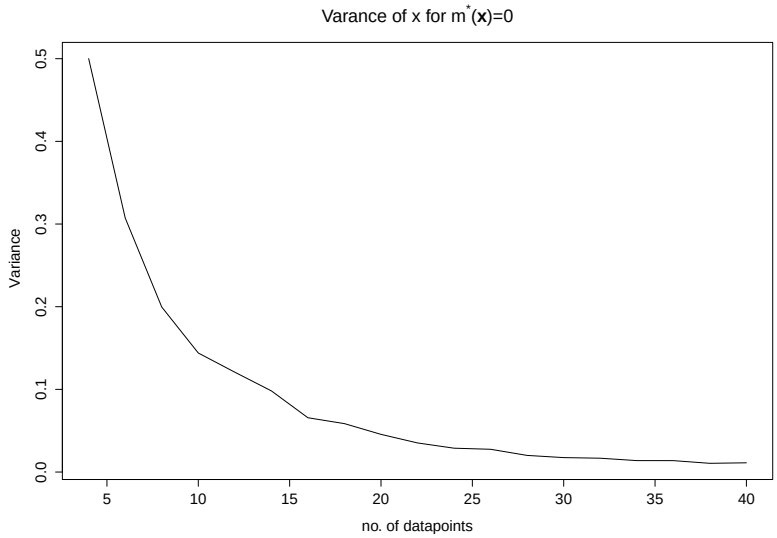

Figure 5.16: Variance of x from $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d})$ vs no. of data points

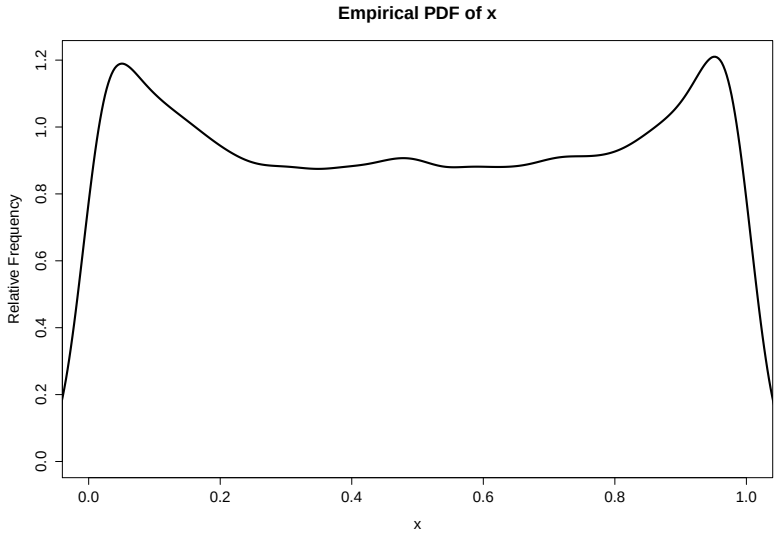

Figure 5.17: PDF of $\hat{\eta}(0)$

5.2.3 Comparing $\hat{\eta}(0)$ to $\eta(0)$

As stated previously, given the discontinuity is at 0 it would be reasonable to expect that $\hat{\eta}(0) = \eta(0) = 0.5$. Eight data points were drawn from a uniform distribution (four between -3 and 0 and four between 0 and 3). Figure 5.17 is the PDF and and Figure 5.18 is the CDF of the function $\hat{\eta}(0)$. The range of $\hat{\eta}(0)$ is between 0 and 1, and it does not go above these values (even after 50,000 runs), indicating that when there is only one discontinuity, there was no over/undershooting.

Given the data points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where X is sorted such that: $\mathbf{x}_1 \leq \mathbf{x}_2 \leq \ldots \leq \mathbf{x}_{n-1} \leq \mathbf{x}_n$ and $Y = \{\eta(\mathbf{x}_1), \eta(\mathbf{x}_2), \ldots, \eta(\mathbf{x}_n)\} = \{0, 0, \ldots 0, 1, 1, \ldots, 1, 1\}$, then a mathematical representation defining no overshooting is: $0 \leq \hat{\eta}(\mathbf{x}) \leq 1$ or $\eta(\mathbf{x}_j) \leq \mathbf{h}(\mathbf{x})^T \beta + \mathbf{t}(\mathbf{x})^T \Sigma(\mathbf{y} - H\beta) \leq \eta(\mathbf{x}_{j+1})$ where $\mathbf{x}_j \leq \mathbf{x} \leq \mathbf{x}_{j+1}$, meaning that $\mathbf{h}(\mathbf{x})^T \beta - 1 \leq \mathbf{t}(\mathbf{x})^T \Sigma(\mathbf{y} - H\beta) \leq \mathbf{h}(\mathbf{x})^T \beta$. This result has not been proven in this thesis.

The PDF is symmetric, based on a Kolmogorov-Smirnov Test that was performed, which returned a P-value of 0.72 (which fails to reject the null hypothesis that the PDF is symmetric). Other symmetric tests performed were: the Cabilio-Masaro test, the Mira test and the MGG test using the symmetry.test (Gastwirth et al., 2015) in R, obtaining a P-value above 0.7. However, to ensure that the reason for the symmetric was not because the data points were drawn evenly on both sides of the discontinuity's location, a different type of sampling process will be used. When eight data points are drawn between -3 to 3 from a uniform distribution but only accepting the data points when there is at least one data point on each side of the discontinuity, then a PDF as shown in Figure 5.19 was obtained and is symmetric, in conclusion. It can be suggested that $\hat{\eta}(0)$ is symmetric for the Heaviside function.

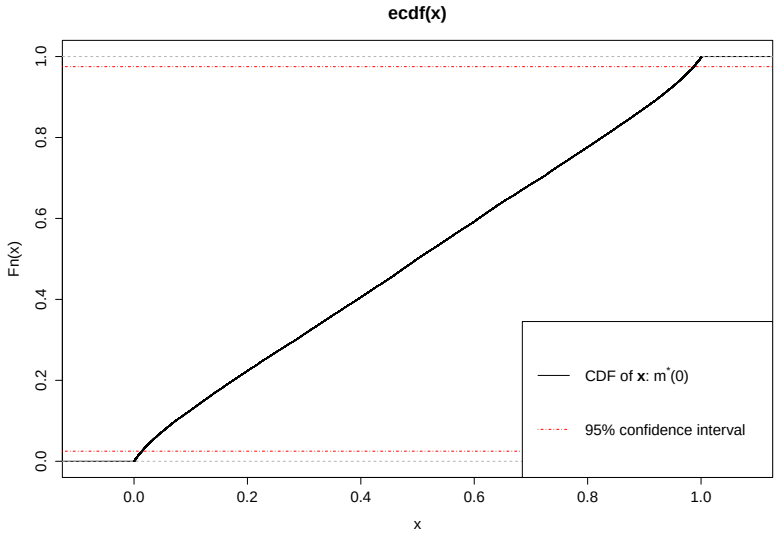

Figure 5.18: CDF of $\hat{\eta}\,(0)$

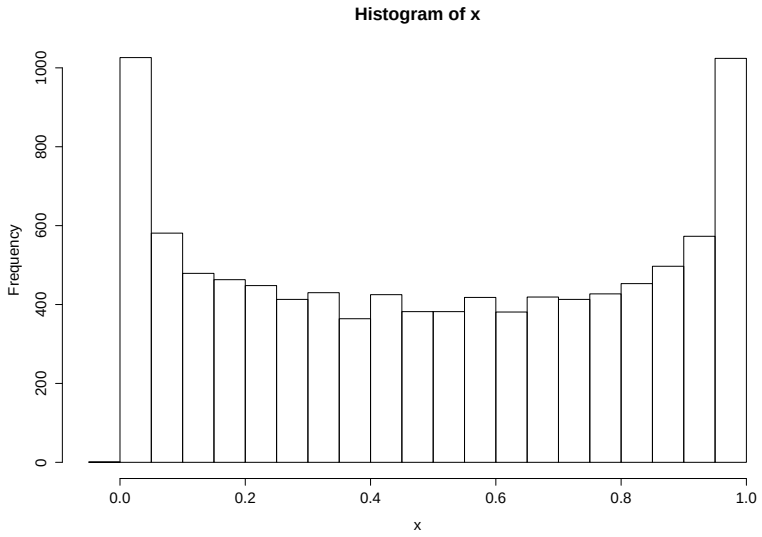

Figure 5.19: PDF of $\hat{\eta}\,(0)$ with different type of sampling process

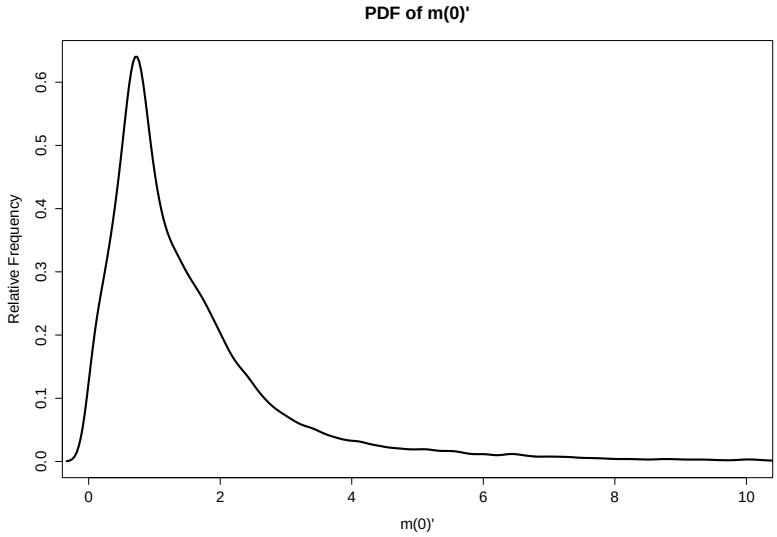

Figure 5.20: PDF of $\frac{d\,\hat{\eta}(0)}{d\mathbf{x}}$

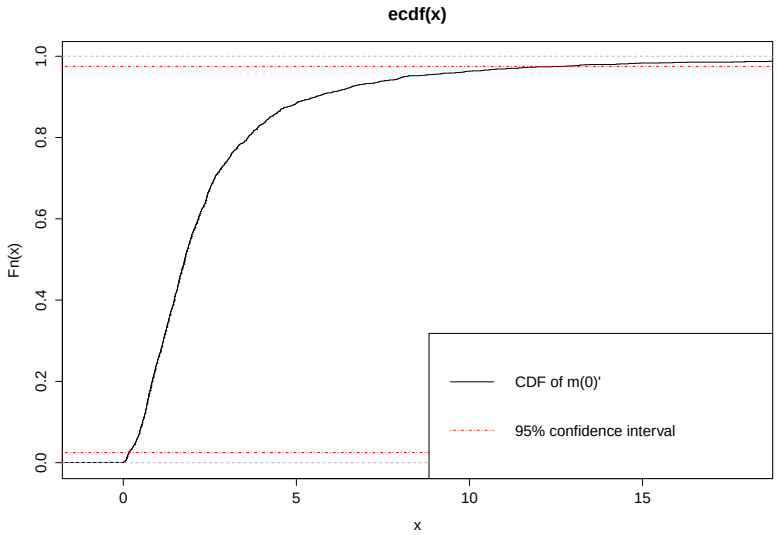

Figure 5.21: CDF of $\frac{d\,\hat{\eta}(0)}{d\mathbf{x}}$

5.2.4 Investigating the steepness of $\hat{\eta}\,(\mathbf{0})$

From the Heaviside function, it would be expected that the steepness at the discontinuity (i.e. $\frac{d\hat{\eta}(\mathbf{x})}{d\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{0}}$) should be high, and because it was found previously (see Section 5.2.3) that $\hat{\eta}\,(0)$ is always positive, one might expect that the gradient of emulator will always be positive at this point also. From 30,000 runs this assumption that steepness of $\hat{\eta}\,(0)$ is positive has been suggested to be correct (see Figures 5.20 and 5.21); however the gradient averages about 1.78. A high gradient is only obtained when two sample points on either side of the discontinuity are close together. When the sample size increases or the sample space decreases, then there will be more events when the data points are close together near the discontinuity, and therefore the average of the gradient will increase.

5.2.5 Summary

The findings from the MSE have shown how important it is to have the data points evenly spread rather than having them focused in one area or unevenly spread as in figure 5.7 (see p. 47), where there is a high error in the emulator between -2 to 0 due to the lack of information between these points. As the gap between the data points increases, the emulator gains more uncertainty, which leads to the emulator converging to the prior as it has the majority influence over the emulator rather than the closest data points. Figure 5.11 (see p. 49) shows that as the number of data points increases, the average of the MSE decreases, as well as the variances of the MSE.

When solving $\mathbf{x}$ for $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d})$, the PDF looked similar to a Laplace distribution. Figure 5.14 (see p. 51) is an example where the discontinuity was at 0 ($d = 0$) and $\eta(\mathbf{0}) = 0.5$. It was inconclusive on whether using the emulator was better than picking the halfway point between the data points on each side of the discontinuity; however, one could not say it was worse either. Figure 5.16 (see p. 52) shows that, as expected, the variance of the estimation decreases the more data points there are.

When comparing $\hat{\eta}(0)$ to $\eta(0)$, it was discovered that after many numerical runs that there is strong evidence for under/overshooting to be rare between the two data points on either side of the discontinuity.

From the previous findings with $\hat{\eta}(0)$, it was expected that steepness of $\hat{\eta}(0)$ will always be positive, and it can be said the higher the steepness of $\hat{\eta}(0)$ is, the better the emulator has fitted the data at the discontinuity.

5.3 One Dimension with Two Discontinuities

Let $\eta(x) = \begin{cases} 1 & -1 < x < 1 \\ 0 & \text{otherwise} \end{cases}$ be a simple function with two discontinuities. For the Bayesian emulator, the regression function will be: $H(x) = [1]$ and a correlation function: $c(\mathbf{x}, \mathbf{x}') = \exp\left(-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')\right)$, with $B = 1$. (see Experiment Setup and Assumptions in Section 4.2.1 on page 32 for reasoning). Figure 5.22 is a representation of this function. An empirical PDF and a CDF will be used in the goodness-of-fit technique applied to the data obtained from 30,000 runs. To start with, each run will contain six[2] data points from the true function; two data points between the two discontinuities and two on each end of each discontinuity. The data points are drawn from a uniform distribution within a domain (two between -3 and -1, two between -1 and 1, and two between 1 and 3) so that every run has information that there may exist two discontinuities yet the location and the knowledge that they are discontinuities is withheld from the emulator. The goodness-of-fit technique will then be compared to the number of data points from the true function, and it should be expected that, on the average, more information will result in a more accurate emulator.

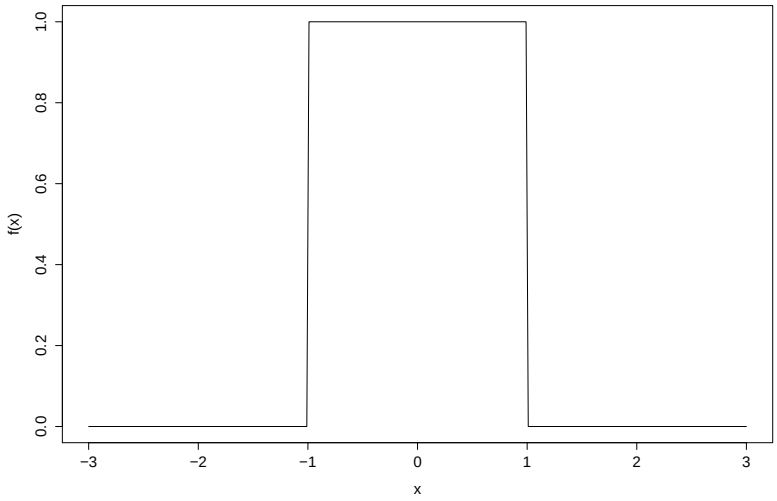

Figure 5.22: One dimension with two discontinuities function

5.3.1 Mean Square Error (MSE)

Figure 5.23 shows a PDF (six data points per run), and Figure 5.24 shows the CDF of the MSE. The results are very similar to what was found with one discontinuity (in section 5.2.1) where the MSE would occasionally be very high, resulting in the PDF being skewed to the right.

[2]six instead of four as in the previous section due to the added complexity of the function

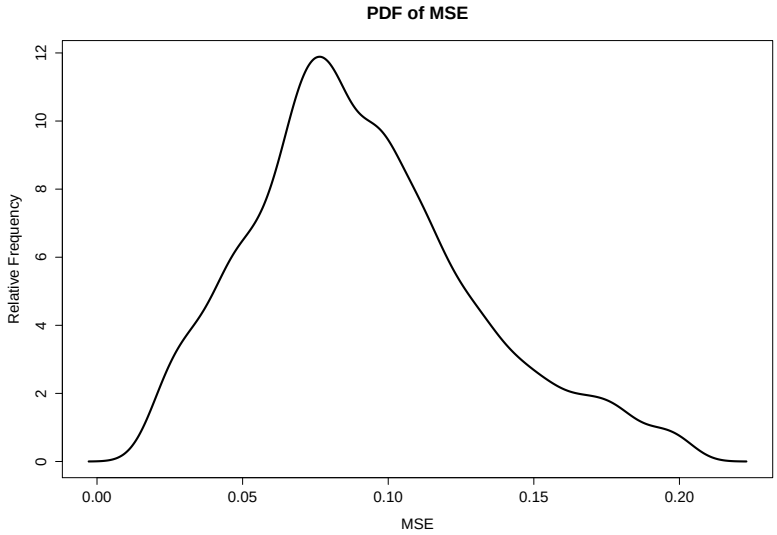

Figure 5.23: PDF of the MSE with six data points

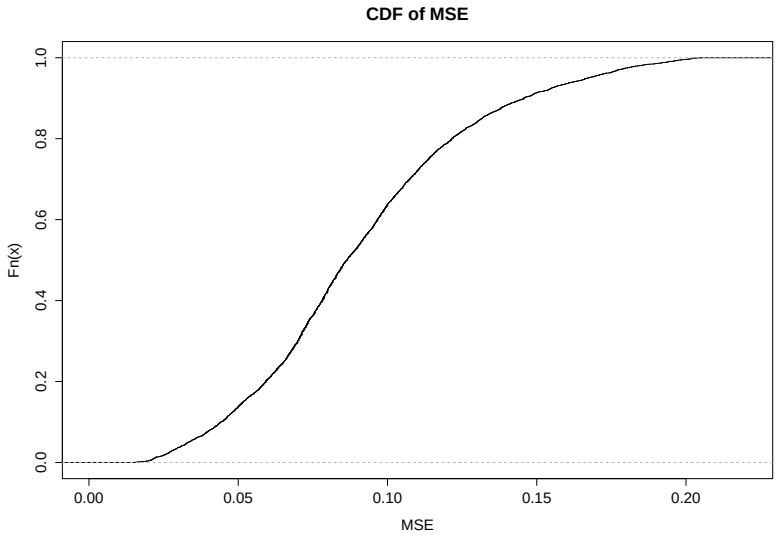

Figure 5.24: CDF of the MSE with six data points

5.3.2 Approximating and Comparing the Location of the Discontinuities from the Emulator

As previously, for each run, six data points were drawn from a uniform distribution (two between -3 and -1, two between -1 and 1, and two between 1 and 3). Knowing that there were two discontinuities, $\hat{\eta}(\mathbf{x}) = 0.5$ will have two solutions. From Figure 5.25 and 5.26, both PDFs were found to be similar to the Laplace distribution.

Figure 5.27 shows the regions, A, B, C and D. Each region is defined using table 5.1.

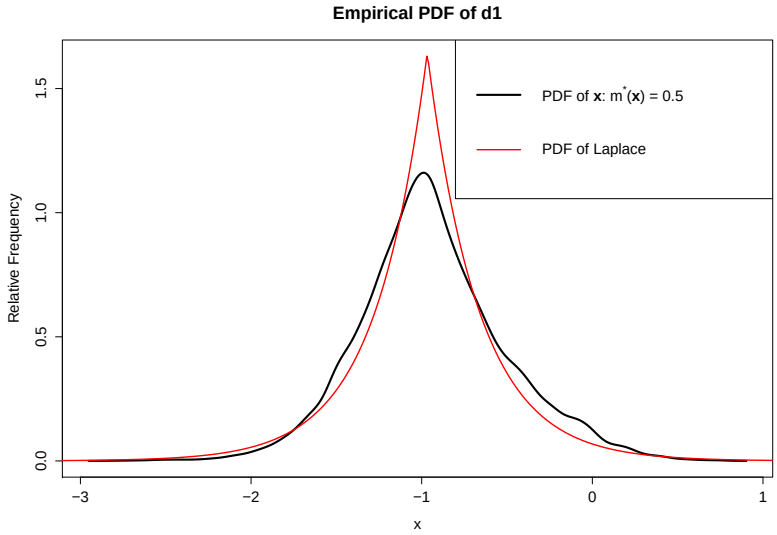

Figure 5.25: PDF of x from $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d_1})$ over a Laplace PDF

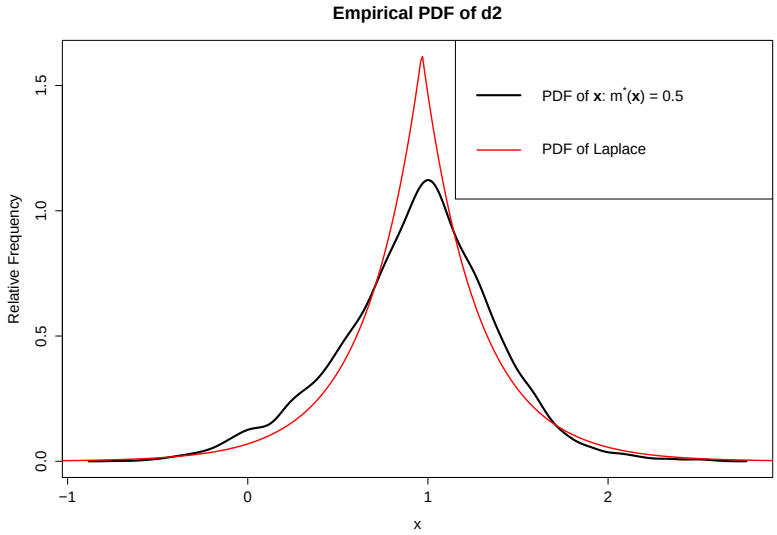

Figure 5.26: PDF of x from $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d_2})$ over a Laplace PDF

	$\hat{\eta}(\mathbf{x}) \geq 0.5$	$\hat{\eta}(\mathbf{x}) < 0.5$
$\eta(\mathbf{x}) = 1$	B	A
$\eta(\mathbf{x}) = 0$	C	D

Table 5.1

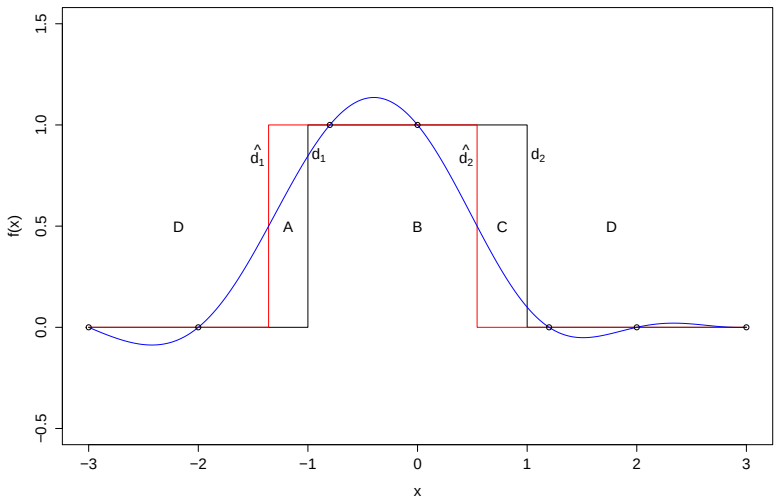

Figure 5.27: Example of Jaccard index regions for one dimension with two discontinuities. note that the red line represents the approximation of where the discontinuous is based on the emulator (the black curved line) $\hat{\eta} = 0.5$

To measure the accuracy of the emulator correctly predicting where $\eta = 1$, the Jaccard similarity index was applied by $\frac{B}{A+B+C}$. It was found that from 10,000 runs, 95% of the time B would be between 68% and 97% correct compared to the true function. When increasing the number of data points to nine (three between -3 and -1, three between -1 and 1, and three between 1 and 3), the region of B showed that the accuracy of the emulator was increased. 95% of the time, B would be between 71% and 96% correct.

When the number of data points is increased, the accuracy as expected would also increase, as Figures 5.30 shows.

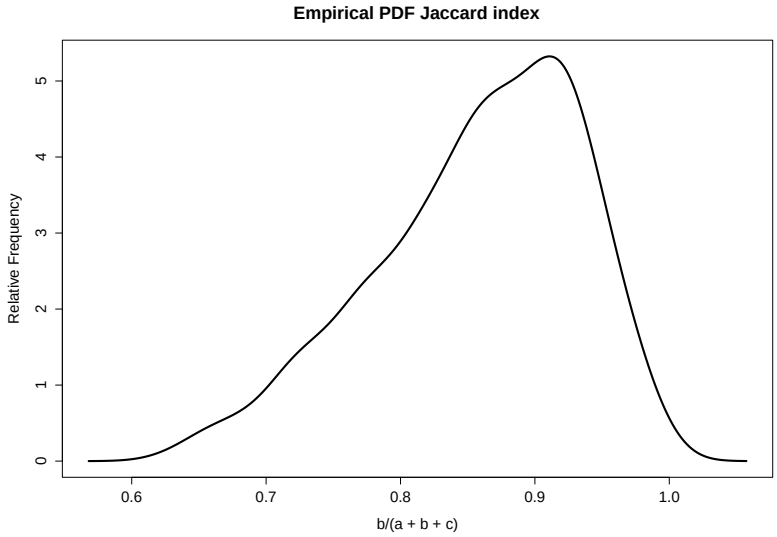

Figure 5.28: PDF of accuracy using Jaccard index

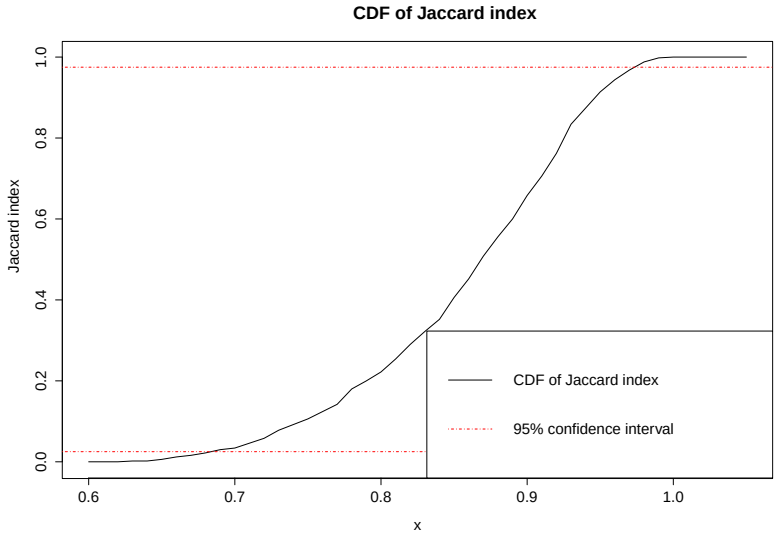

Figure 5.29: CDF of Jaccard index similarity

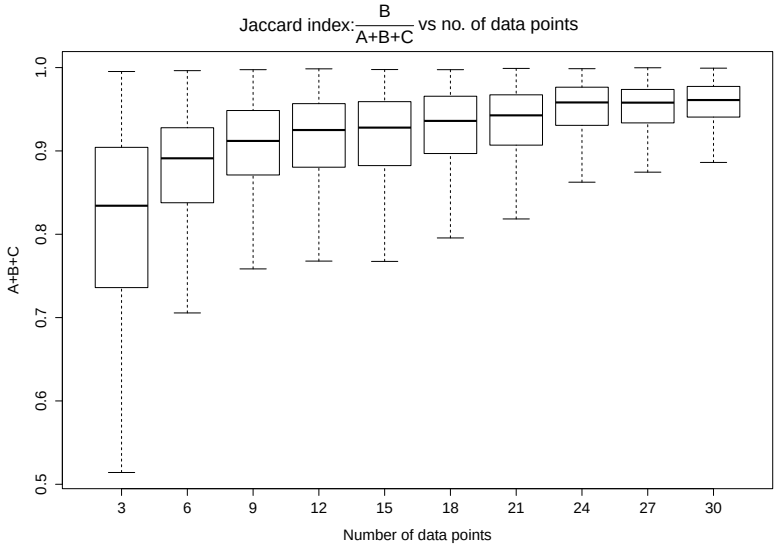

Figure 5.30: Average accuracy vs data points, using Jaccard index

5.3.3 Comparing $\hat{\eta}(-1)$ to $\eta(-1)$ and $\hat{\eta}(1)$ to $\eta(1)$

Similar to section 5.2.3, it is expected that $\hat{\eta}(-1) = \hat{\eta}(1) = 0.5$; however unlike in section 5.2.3, there were cases where the emulator would overshoot (no undershooting was found). Figures 5.31 and 5.32 show the PDF and CDF of this: the PDF shows that there are extreme cases when it would overshoot above 1 (from 10,000 runs there was a case $\hat{\eta}(-1) = 189$). From this research with two discontinuities, it might suggest that this goodness-of-fit technique is impractical.

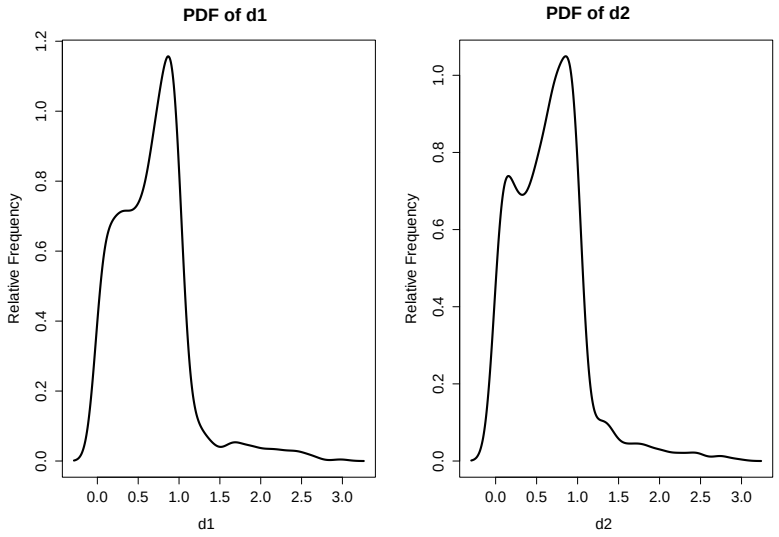

Figure 5.31: PDF of $\hat{\eta}(d_1)$ and $\hat{\eta}(d_2)$ refer to Figure 5.27

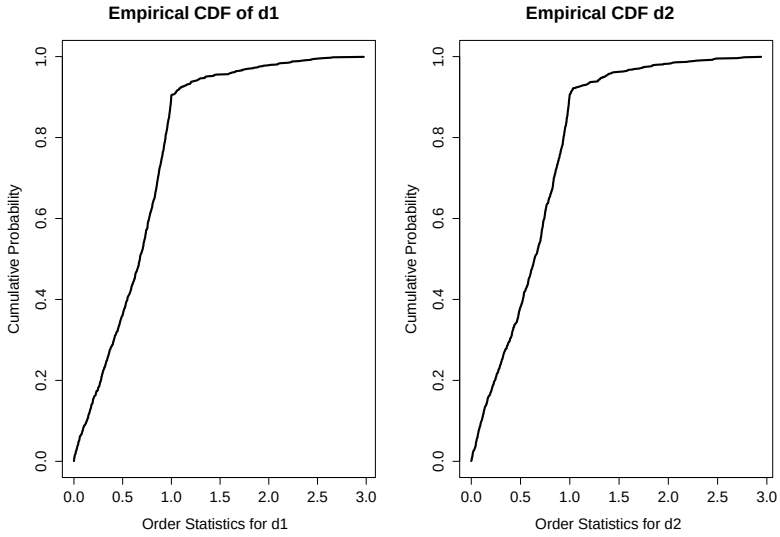

Figure 5.32: CDF of $\hat{\eta}(d_1)$ and $\hat{\eta}(d_2)$

5.3.4 Summary

The results from the mean square error (MSE) were very similar to the one discontinuity Heaviside function in Section 5.2, with the MSE being occasionally high, causing the PDF to be skewed to the right, again indicating the importance of the data point selection process. Because there is more than one discontinuity it was decided that a Jaccard similarity index would be used to measure how accurate the emulator is. A post-process of the emulator was done (Equation 4.4 on page 40) to determine the location of the emulator's estimation of the simulator's region.

Figure 5.27 (see p. 60) is an example illustrating this, separating the emulator results into four different regions (see table 5.1 (see p. 59). Region B is where the emulator correctly estimated the simulator to be at 1 and region D is where the emulator correctly estimated the simulator to be at 0. Regions A and C are seen as regions which the emulator has incorrectly estimated the simulator. The equation $\frac{B}{A+B+C}$ is used to measure this accuracy, where A, B, and C are the area or length of the regions described above. Region D is not included in this formula as (for this example) region D could become very large. Figure 5.30 (see p. 61) is the average of this error with respect to the number of data points. As expected, with more data points it is more likely the data points will be closer together, resulting in a better prediction of where the discontinuity is.

Unlike in Section 5.2, where there was no over/undershooting with one discontinuity, with two discontinuities there was overshooting (which was very extreme in some cases); however, there was no indication of undershooting.[3] 90% of the time $\hat{\eta}$ would be between 0 and 1 (see Figure 5.32, page 62).

[3]Numerically concluded with over 10,000 runs with random data points for every run.

5.4 Two Dimensions

For the next part of the experiments, the emulator is to model a function in two-dimensional space, containing discontinuities at the boundary of a region, using Bayesian emulation. The function is motivated by the physical problem of heavy gas dispersion (HGD) travelling across a terrain.

Let D be the domain or region in two-dimensional space where the gas is present, representing a function:

$$f(x,y) = \begin{cases} 1 & \text{if } (x,y) \in D \\ 0 & \text{if } (x,y) \notin D \end{cases} \tag{5.1}$$

where the discontinuity exists at the boundary of the domain D, and if the function; $f(x,y) = 1$, then the gas is present.

One of the simplest regions that could represent a gas cloud is a circle of radius r. The gas is present if $f(x,y) = 1$, meaning that the location (x,y) is inside the circle domain. For this investigation, the radius $r = 3$.

This can be represented as: $f(x,y) = \begin{cases} 1 & \text{if } \sqrt{x^2+y^2} \leq 3^2 \\ 0 & \text{if } \sqrt{x^2+y^2} > 3^2 \end{cases}$ where the discontinuity exists at the edge of the circle.

Figure 5.33 is a plot of this function.

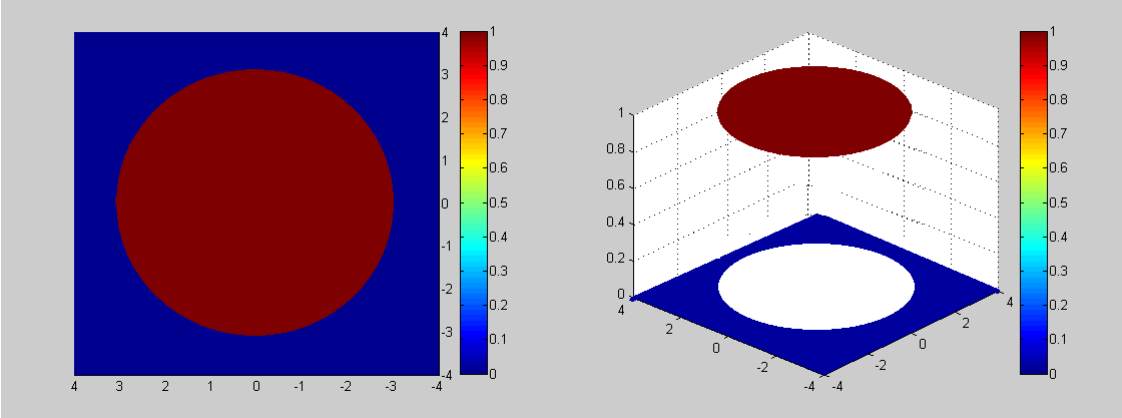

Figure 5.33: Simple example of a 2D region

An empirical PDF and a CDF will be used in the goodness-of-fit technique applied to the data obtained from 1,000 runs[4]. To start with, each run will contain 50 observation points from the true function. The data points are drawn from a uniform distribution within a domain (between -5 and 5 for both planes) so that every run should contain information about a change in the function, meaning that there may exist discontinuities yet the location and the knowledge that there is discontinuity is withheld from the emulator. The goodness-of-fit technique will then

[4]1,000 was chosen, as it is a representation of a large sample size that is a reasonable compromised based on the computer resources available.

be compared to the number of data points from the true function, and it should be expected that, on the average, more information will result in a more accurate emulator.

5.4.1 Mean Square Error (MSE)

For each run, a uniform distribution of 50 data points was taken between -5 and 5 for both planes. For the MSE, Δ[5] was 0.1 due to limited computer power (each run would take approximately 7 seconds). After 1000 runs, a PDF and a CDF of the MSE were generated as shown in Figures 5.34 and 5.35. Similar to the one-dimensional case, there were some runs when the MSE value would be extremely high compared to most of the other runs; this led to the PDF being skewed to the right as Figure 5.34 shows. As the number of data points increases the MSE decreases, as Figures 5.36 and 5.37 show; however, some simulators are costly to run and therefore there is a real-world limitation on the number of runs of the simulator.

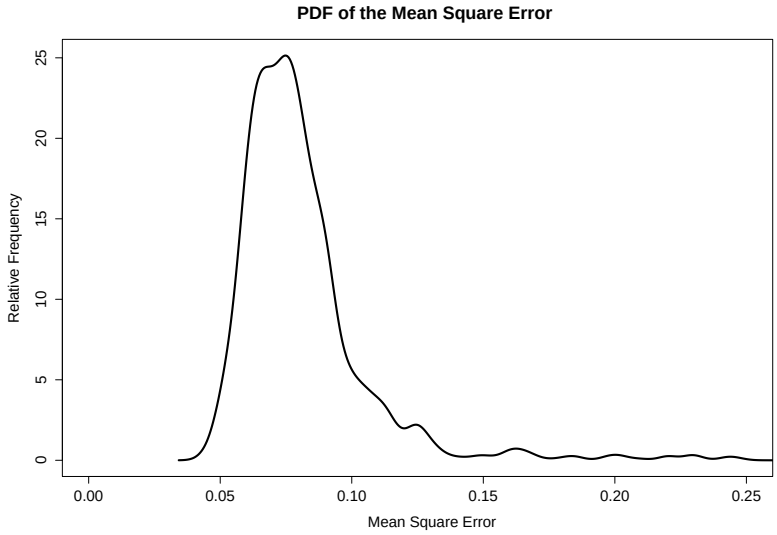

Figure 5.34: PDF of the MSE. Note the skewness is still present

[5] Δ is the nomically step interval change

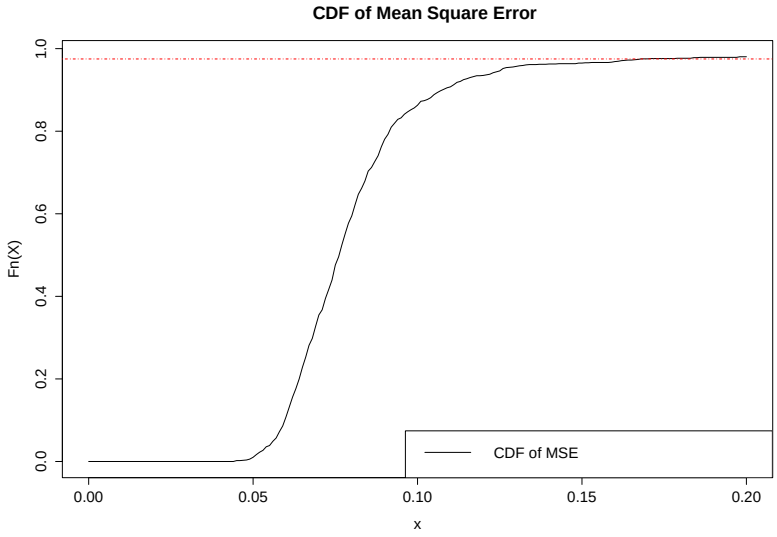

Figure 5.35: CDF of the MSE.

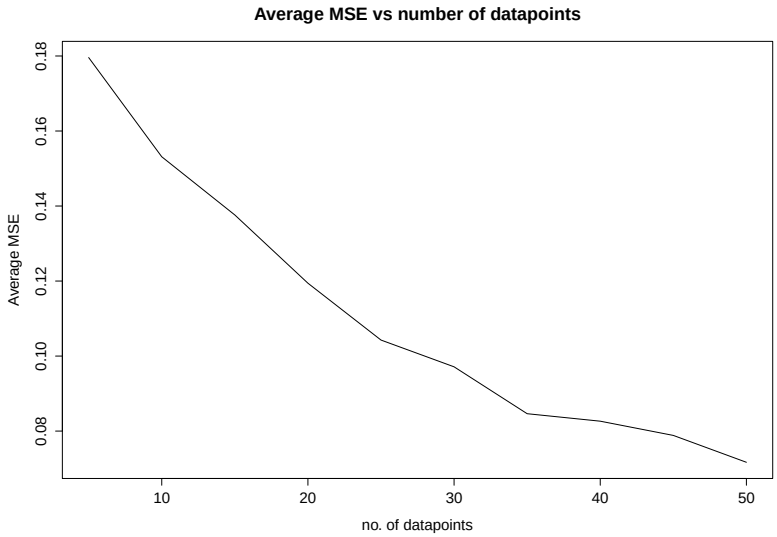

Figure 5.36: Average of MSE vs number of data points

5.4.2 Examining the Discontinuous Region using a Jaccard Approach

For the two-dimensional case the discontinuity is at the boundary of the region of the function (see Figure 5.33), resulting in infinitely many points at which a discontinuity exists. Therefore rather than finding the points when $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d})$, a post-process of the emulator was done (Equation 4.4, page 40).

Figure 5.38 shows an example of one run of the emulator. After the post-process both regions from the emulator and the simulator were compared (one from the emulator and the other from the simulator), Figure 5.39 represents the boundary from the simulator; if the emulator has a high accuracy then the area of the regions A and C should be small, resulting in $\frac{B}{A+B+C}$ being close to 1. After 2000 numerical runs with 50 data points, it was found that on average the overlapping region B was

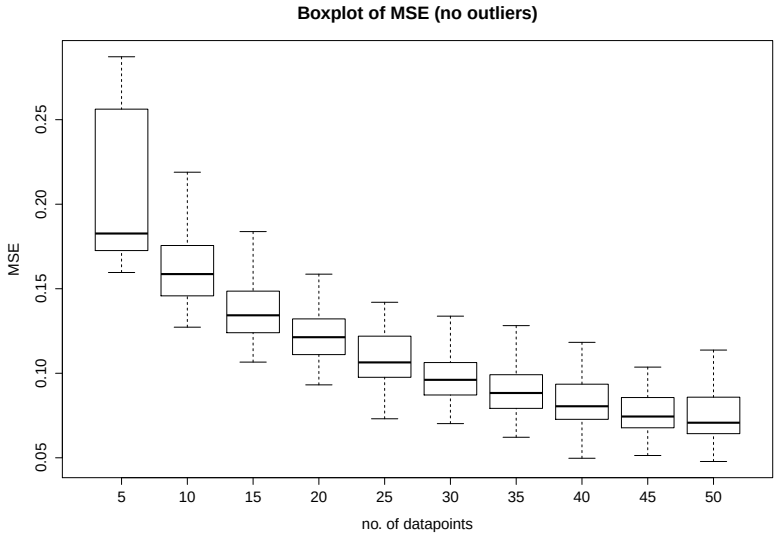

Figure 5.37: Boxplot of MSE vs number of data points

68%. Figure 5.40 shows the PDF of $\frac{B}{A+B+C}$ and Figure 5.41 is the CDF. From 2000 runs, 95% of the time the emulator accuracy (measured by $\frac{B}{A+B+C}$) was between 47% and 82.5%.

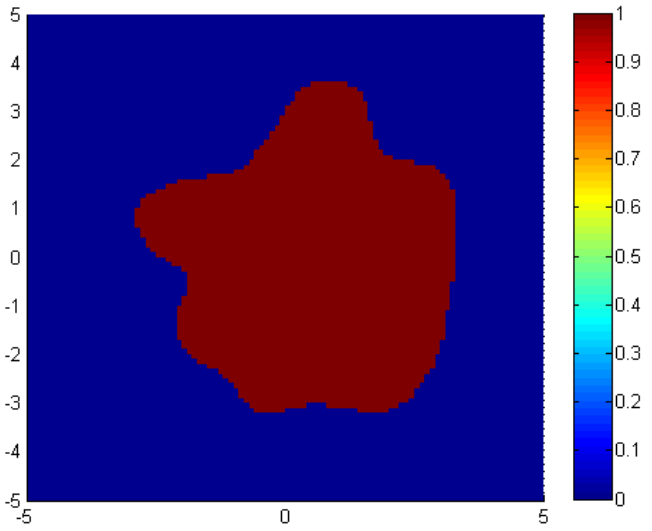

Figure 5.38: Example of the emulator approximating the boundaries in two dimensions

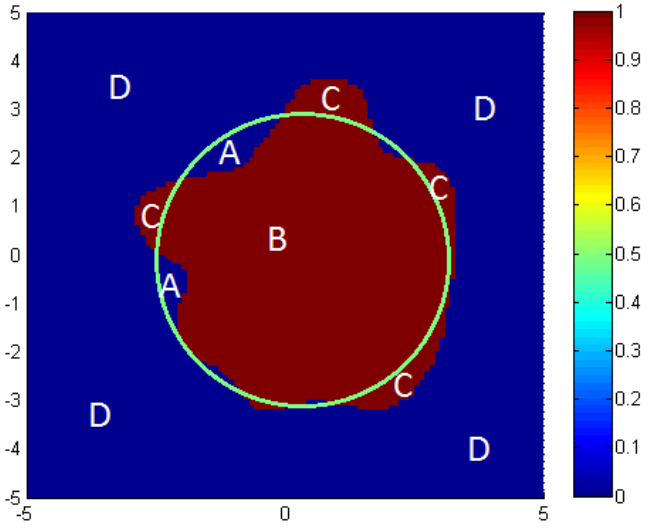

Figure 5.39: Separated in regions for Jaccard index, regions B and D are correct and regions A and C are incorrect, (see Section 4.2.3, page 40)

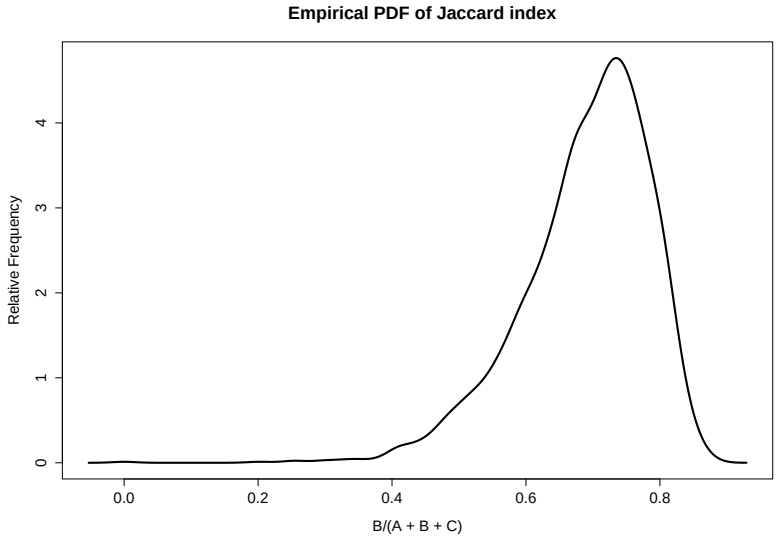

Figure 5.40: PDF of accuracy in two dimensions, using Jaccard index

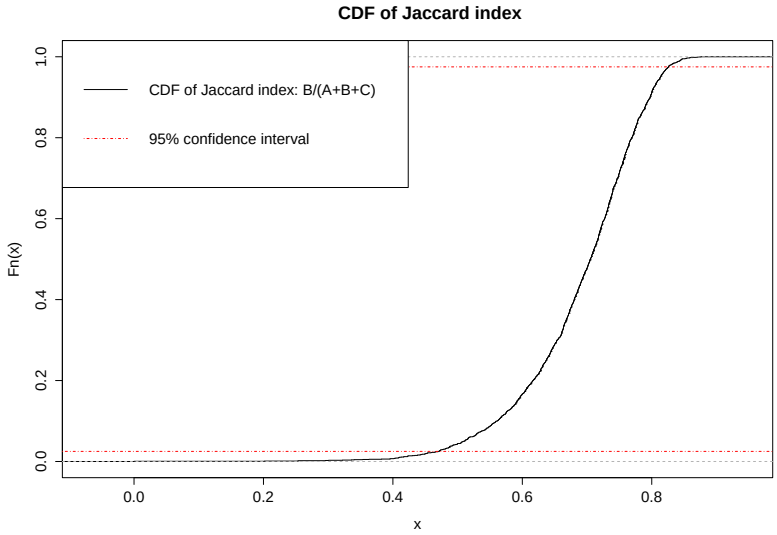

Figure 5.41: CDF of accuracy in two dimensions, using Jaccard index

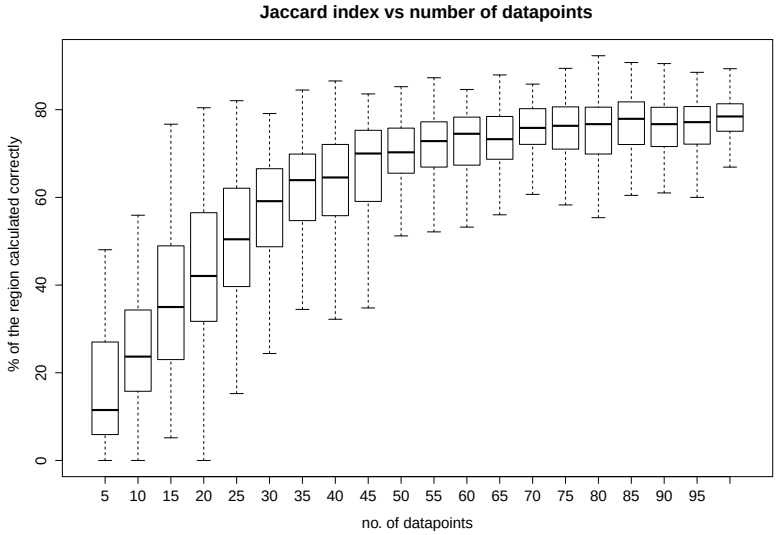

Figure 5.42: Boxplot of accuracy vs number of data points, using Jaccard index

5.4.3 Summary

The function used in section 5.4, equation 5.1, was motivated by the physical problem of heavy gas dispersion (HGD) travelling across a terrain, where 1 represents the presence of the gas and 0 is the absence of the gas (see Figure 5.33).

With small number of data points, there were some cases where the results from the simulator were all 0s or 1s, providing no information that this function was changing, and therefore leading to the emulator only outputting that number. In HGD terms, if the emulator outputs all 1s, it means that the gas is everywhere. This again highlights the importance of data selection, since having too large a gap in the data points leads to higher uncertainty.

The MSE was, in most cases, not too extreme, but in a two-dimensional plane it was shown that unevenly spread data points will result in a lower accuracy of the emulator. Figures 5.36 and 5.37 (see p. 66) show how the MSE changes as the number of data points is increased; as expected from the previous findings, the more data points there are, the lower the MSE is.

When applying the Jaccard index technique, more data points result in a more accurate emulator, and the emulator is better able to estimate the boundary of the circle, where the function would change from 0 to 1. This result is to be expected, as the more data points there are, the closer they are together, resulting in a better estimate of where the boundary is. Figure 5.42 (see p. 69) shows how the accuracy increases the more data points there are.[6] The level of accuracy does begin to level out after a certain number of runs, indicating there would be a "sweet spot" beyond which having more data points will not increase the accuracy of the emulator significantly.

[6]It should be noted that due to the limitations to computation power, the grid size of the emulator for each run is 0.1.

5.5 Two Dimensions + Time

Expanding this investigation, let time and the rate of the gas being released be introduced in the function.

To Start, before the tank leaks, it is assume the tank contained a certain amount of gas T_0 [7]. $S(t)$ is the amount of gas on the surface outside of the tank, before the gas is released it is assume there is none present.

It will be assumed that the gas is released continuously (not instantaneous) and due to pressure the rate of the gas being released is proportional to the remaining present in the tank.

This can be derived as $\frac{dT}{dt} \propto -V$

Expanding this investigation, let time and the rate of the gas being released be introduced in the function. To start, it is assumed that a tank with a limited volume $V_T(t)$ of gas has a leak and the gas is continuously released (not instantaneous).

where:

- $V_T(t)$ is the amount of gas inside of the tank at time t
- $V_S(t)$ is the amount of gas on the surface outside of the tank.
- $\frac{dV_T}{dt}$ is the rate the tank is emptied from the leak.
- The gas release rate is dependent on the pressure of the tank/pipe and how big the hole(s) are where the leakage is coming from.

Note that $V_S(0) = 0$ and $V_T(t) + V_S(t) = V_0$ so $V_S(t) = V_0 - V_T(t)$ and $\frac{dV_S}{dt} = -\frac{dV_T}{dt}$

To start with, let $\frac{dV_T}{dt} = -kV$ where k is some constant

$V_T(t) = V_0 e^{-kt}$

$\therefore V_S(t) = V_0 - \left(V_0 e^{-kt}\right) = V_0\left(1 - e^{-kt}\right)$

$\boxed{V_S(t) = V_0\left(1 - e^{-kt}\right)}$

Note that $k = \frac{-\ln\left(1-\frac{V_S(t)}{V_0}\right)}{t}$

To start with, if the region of the gas is assumed to be a circular shape, and the gas height h is constant, the area of the gas can be calculated as $A = \pi r^2$ and the volume of the gas as $V_S = \pi r^2 h$.

Then, $V_0\left(1 - e^{-kt}\right) = \pi r^2 h$, and rearranging this gives $r(t) = \sqrt{\frac{V_0\left(1-e^{-kt}\right)}{\pi h}}$.

The function to check if the gas is present or not along the region is:

$$f(x,y,t) = \begin{cases} 1 & \text{if } \sqrt{x^2+y^2} \leq r(t) \\ 0 & \text{if } \sqrt{x^2+y^2} > r(t) \end{cases}$$

$$f(x,y,t) = \begin{cases} 1 & \text{if } x^2+y^2 \leq \frac{V_0\left(1-e^{-kt}\right)}{\pi h} \\ 0 & \text{if } x^2+y^2 > \frac{V_0\left(1-e^{-kt}\right)}{\pi h} \end{cases} \tag{5.2}$$

[7] this should be seen as mass rather than volume

where if $f(x, y, t) = 1$ then the gas is present, and if $f(x, y, t) = 0$ then the gas is not present.

With a one-dimensional region, let $y = 0$.

The function now becomes:

$$f(x, 0, t) = \begin{cases} 1 & \sqrt{x^2} \leq \sqrt{\frac{V_0\left(1-e^{-kt}\right)}{\pi h}} \\ 0 & \sqrt{x^2} > \sqrt{\frac{V_0\left(1-e^{-kt}\right)}{\pi h}} \end{cases} \tag{5.3}$$

where if $f(x, y, t) = 1$ then the gas is present, and if $f(x, y, t) = 0$ then the gas is not present.

Example:

let $V_0 = 100m^3$ and in 5 minutes the tank has released $75m^3$

$k = 0.1386294$

From this simulator the graph below (Figure 5.43) is produced, showing the change of the gas's radius over time.

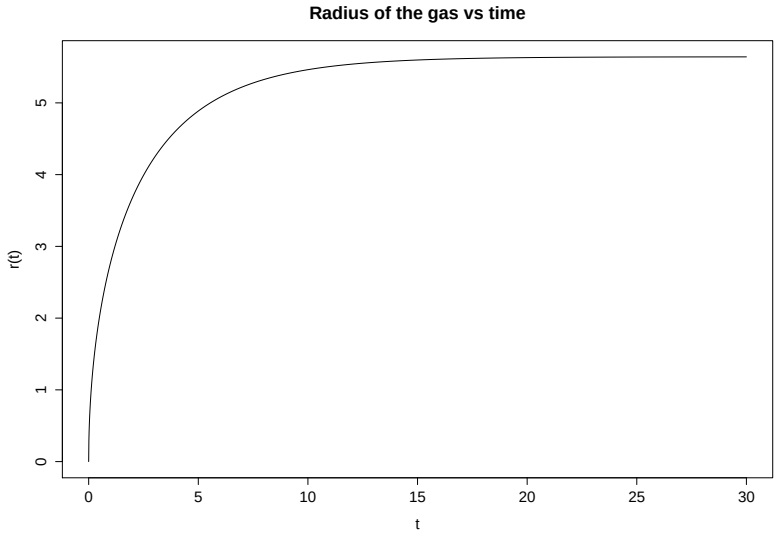

Figure 5.43: Simple example of a HGD model, how the radius would increase over time.

Figure 5.44 is what is expected from the simulator, showing that as time increases, that radius (in the figure the value x) also increases. However because the amount of gas is limited, it is assumed that the radius would also be limited.[8] A short demonstration was done of the emulator, before progressing to measuring the accuracy: 300 data points were generated within the region of -10 to 10 for both directions and time from 0 to 10. It was found that, as time increases, the radius of the gas cloud also increases; however, it does not stop increasing and at $t = 10$ the gas has gone beyond 20 meters where the simulator shows a maximum of 5.6. This

[8]Factors like wind and gravity are not taken into account. This model is designed to have a small amount of added complexity to the previous function in Section 5.4

is because there is no information (i.e. data points) from the simulator given to the emulator beyond this time.

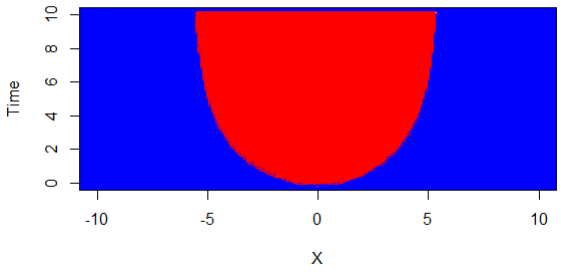

Figure 5.44: Example of the simple HGD model, how the radius increases over time; red: gas is present & blue: gas is not present.

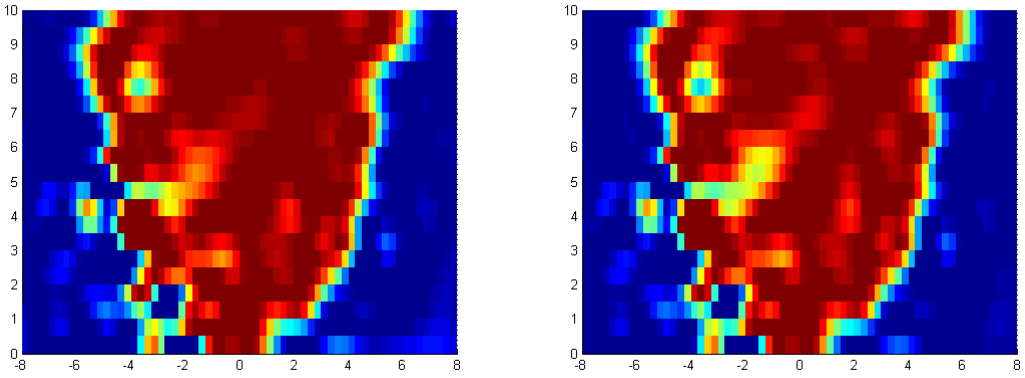

Figure 5.45: Example output of the emulator, modelling simple HGD model, Left: with prior, Right: prior as 0 (300 data points)

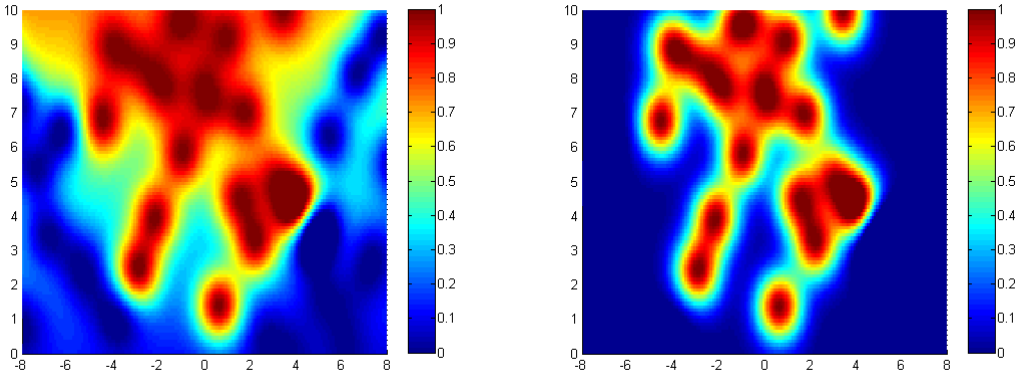

Figure 5.46: Example of the output from the emulator, modelling simple HGD model, Left: with prior, Right: prior as 0 (50 data points)

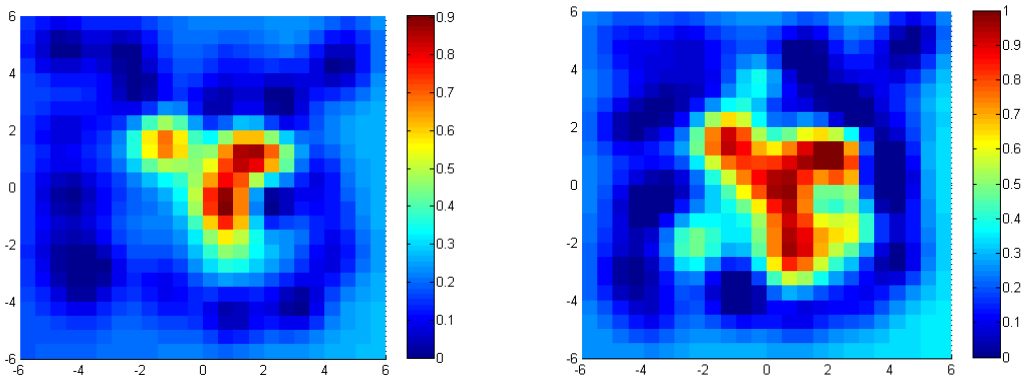

Figure 5.47: $t = 0$ (left) $t = 1$ (right)

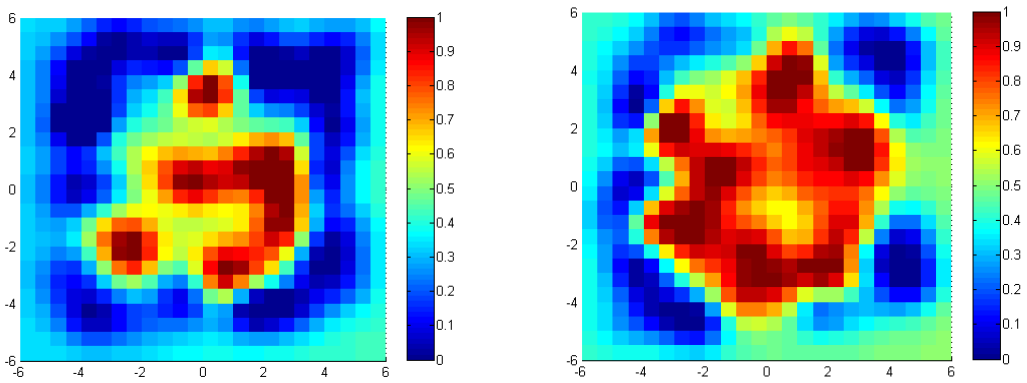

Figure 5.48: $t = 2$ (left) $t = 3$ (right)

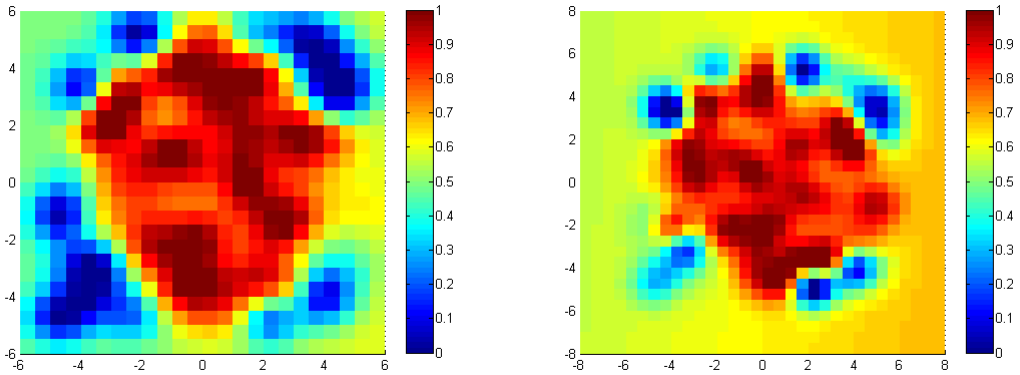

Figure 5.49: $t = 4$ (left) $t = 5$ (right)

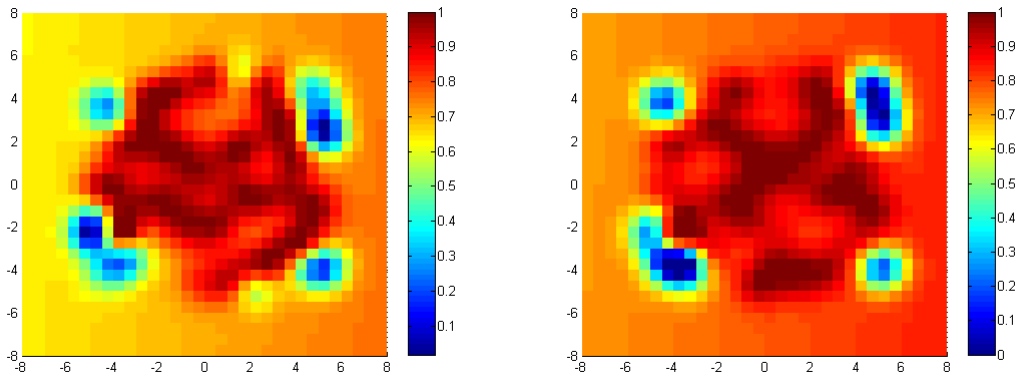

Figure 5.50: $t = 7$ (left) $t = 8$ (right)

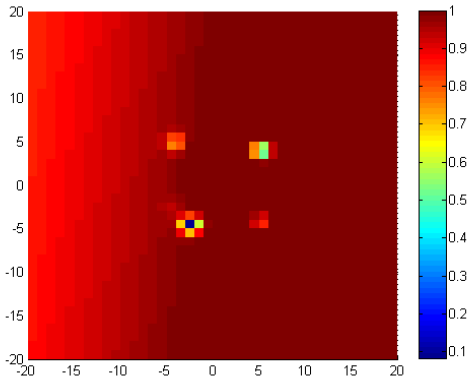

Figure 5.51: $t = 10$

5.5.1 Mean Square Error (MSE)

For each run, a uniform distribution of 50 data points was taken between -5 and 5 for both planes and the time element between 0 and 20. For the MSE, Δ was 0.5 due to limited computer power. After 1000 runs, a PDF and a CDF of the MSE were generated as shown in Figures 5.52 and 5.53. The MSE does not have as many extreme MSE values, which could be due to the sample size. When changing the number of data points, as expected the MSE decreases as the number of points is increased, as Figure 5.54 shows.

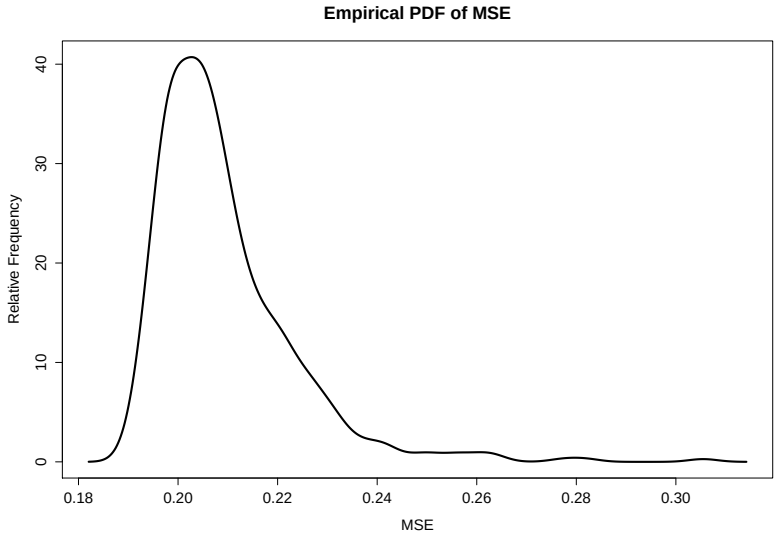

Figure 5.52: PDF of the MSE. Note the skewness is still present.

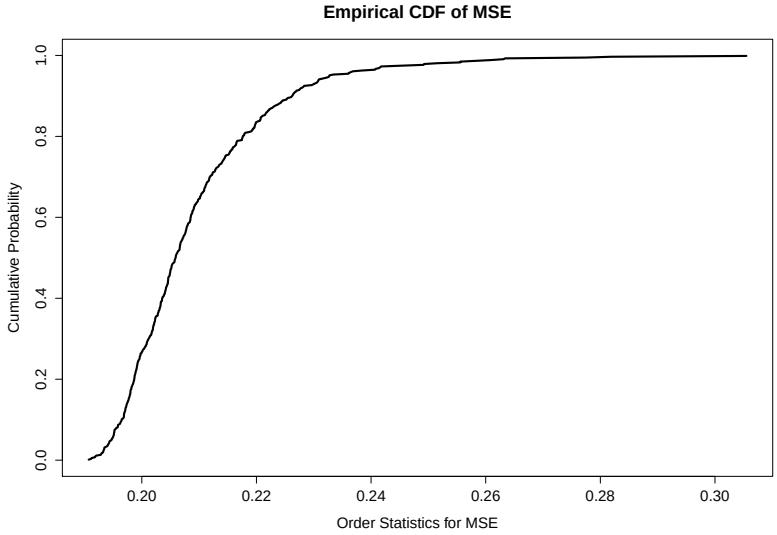

Figure 5.53: CDF of the MSE.

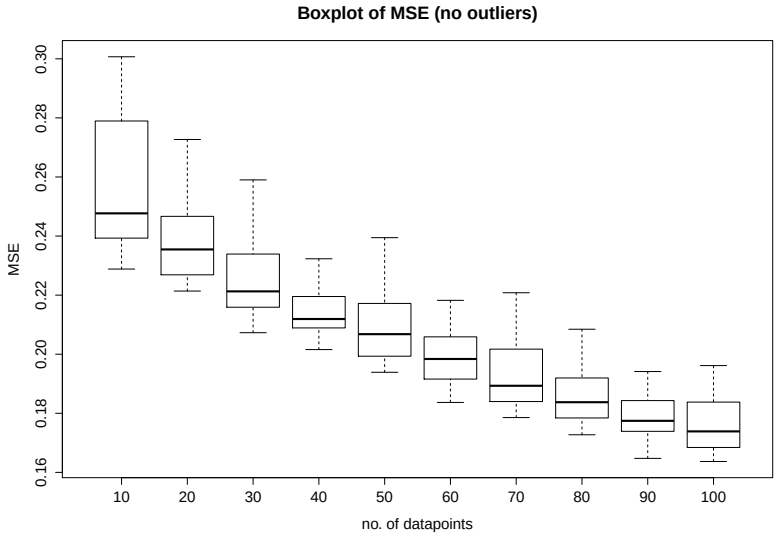

Figure 5.54: Boxplot of MSE vs number of data points

5.5.2 Examining the Discontinuous Region using a Jaccard index Approach

Similar to the case in Section 5.4, a post-process of the emulator was done (Equation 5.4).

$$\begin{array}{ll} 1 & \text{if } \hat{\eta}\left(\mathbf{x}\right) \geq 0.5 \\ 0 & \text{if } \hat{\eta}\left(\mathbf{x}\right) < 0.5 \end{array} \tag{5.4}$$

The emulator is then separated into the same four regions as Figure 5.39 shows, and the equation below is used.

$$\frac{B}{A+B+C}$$

Figures 5.55 and 5.56 are the results from 1000 runs with 50 data points for each run and time between 0 and 10. As a result it was found that on average the overlapping region B was 62%. From 1000 runs, 95% of the time the emulator accuracy (measured by $\frac{B}{A+B+C}$) was between 47% and 82.5%. From Figure 5.57 it is seen that the more data points there are, the more accurate the emulator is.

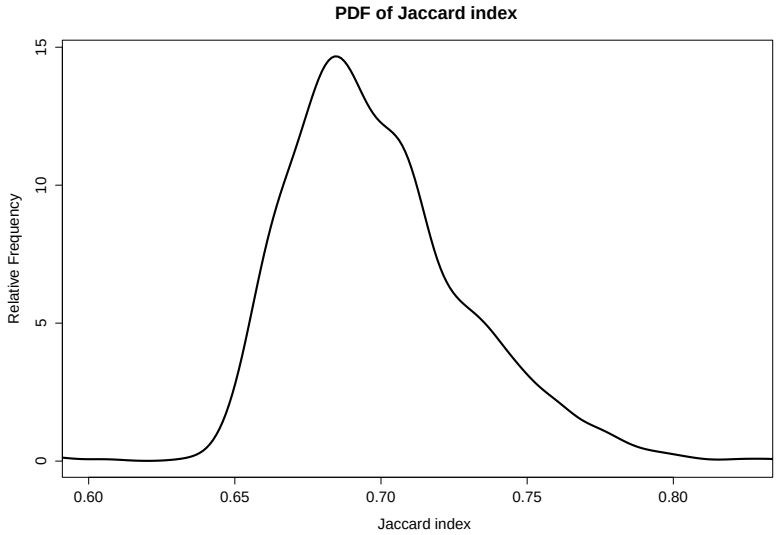

Figure 5.55: PDF of accuracy using Jaccard index

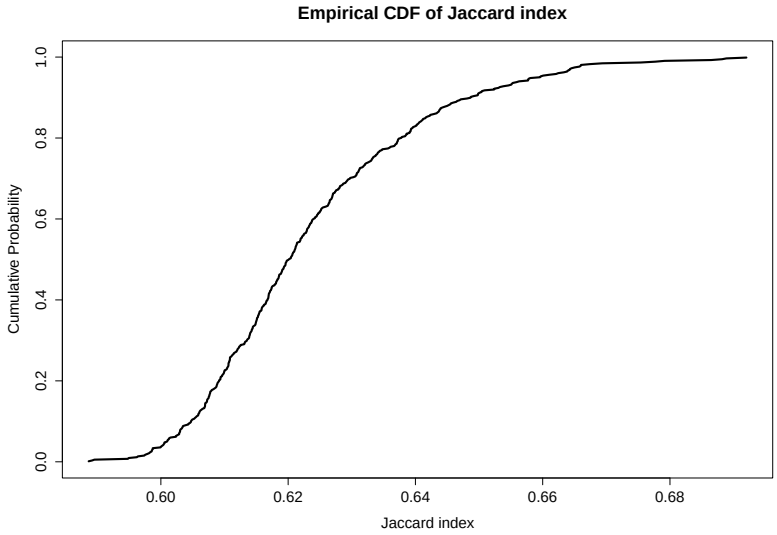

Figure 5.56: CDF of accuracy using Jaccard index

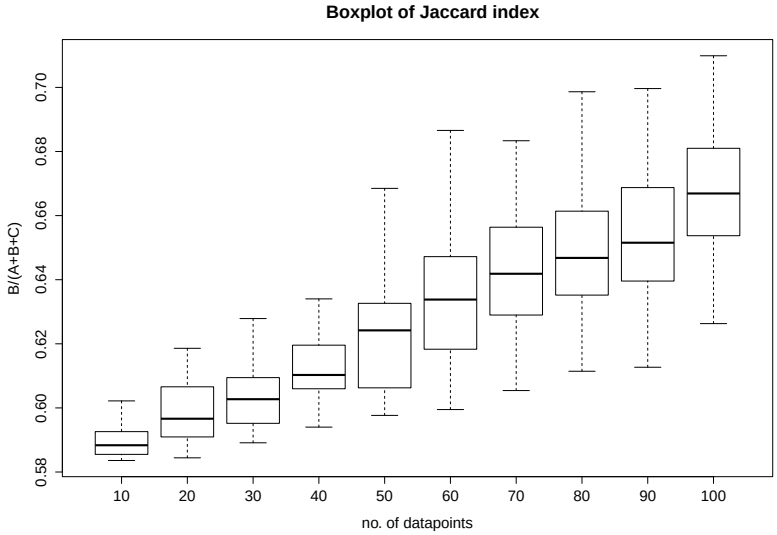

Figure 5.57: Boxplot of accuracy vs number of data points, using Jaccard index

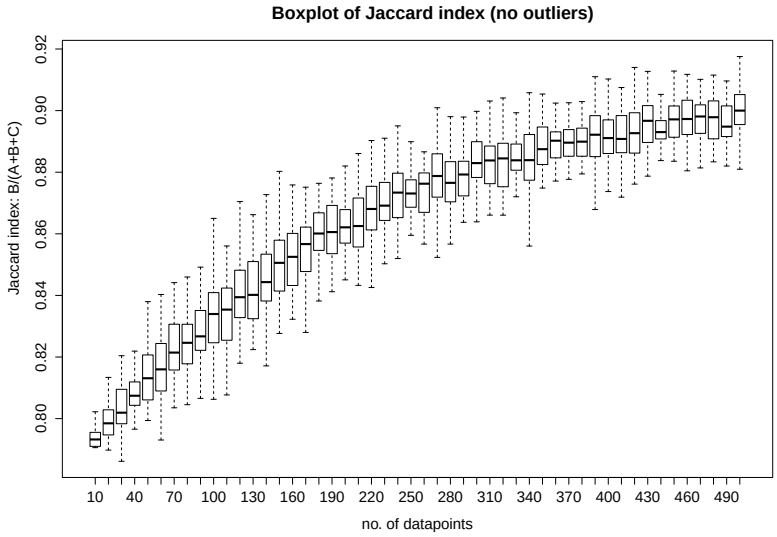

Figure 5.58: Boxplot of accuracy vs number of data points, using Jaccard index

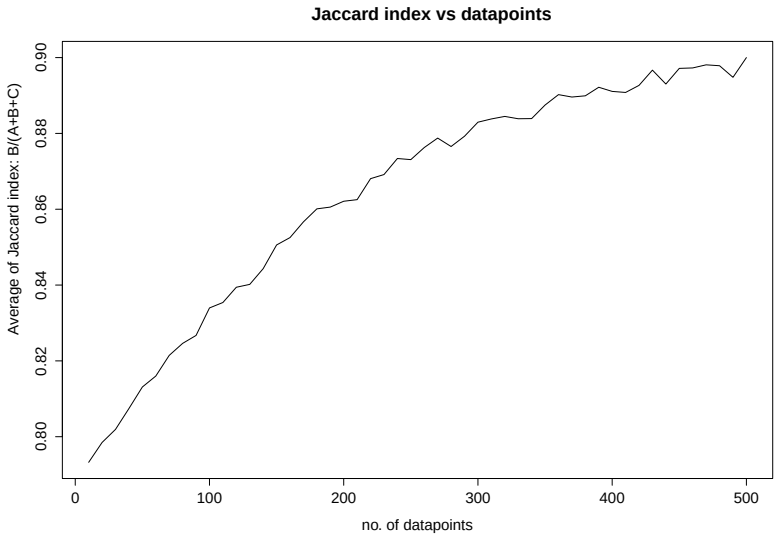

Figure 5.59: Average accuracy vs number of data points, using Jaccard index

5.5.3 Summary

In this section, it was demonstrated how the emulator is able to model a more detailed function that could represent a simple HGD simulator. The emulator was used to model the function with the boundary of the gas increasing over time. Figures 5.47 to 5.50 (see p. 74 to 75) show an example of how the emulator was able to model this; however, with Figure 5.51 (see p. 75) as an example, if the emulator were to attempt to model the simulator outside of the data points (time being greater than 10) then the accuracy of the emulator would decrease significantly.

The MSE, on average, was higher than in Section 5.4, however, this is due to the added complexity of the model and the added variable of time. Figure 5.54 (see p. 77) shows again that MSE decreases as the number of data points are increased.

When applying the Jaccard index technique, more data points result in a more accurate emulator. However, there is a limit when more data points would not significantly increase the accuracy of the emulator as Figure 5.59 (see p. 79) shows, but this increase is not as great as it was in Section 5.4 (see Figure 5.42 on page 69) and more data points are required before the emulator's accuracy levels out (over 300 data points).

Chapter 6

Discussion and Conclusion

6.1 Discussion and Analysis

6.1.1 Introduction

The objective of this research was to enhance the understanding of Bayesian emulation in modelling simulators containing one or more discontinuities (at unknown positions) that represent physical problems. This was done by using one of the simplest functions that contains a discontinuity (the Heaviside function) and then progressing the investigation to slightly more complex functions.

6.1.2 Research Contribution

As expected from Oakley (1999), Bayesian emulation is best suited for continuous smooth functions; therefore, it has come as no surprise that the emulator would have some difficulties in modelling discontinuous functions. When the positions of the discontinuities are unknown, using Bayesian emulation to model discontinuous functions may not be the most practical approach due to the fact that the emulator is treating the discontinuous function as continuous (see Figure 6.1 as an example). A search of the literature indicates there has been little research until now on how Bayesian emulation is able to model discontinuous functions with unknown discontinuity positions. Because of the lack of past research, most of the findings in this thesis were obtained through experiments. Oakley (1999) demonstrated the use of the emulator on smooth and continuous simulators, and therefore most researchers would make the assumption that the simulator was "a smooth function of the inputs without discontinuities" (Chang et al., 2015, p. 17). However Caiado and Goldstein (2015), admitting it would be an expensive operation, attempted to find the discontinuity directly by running the simulator multiple times and then running the emulator for each side of the discontinuity (i.e. two separate emulators). However, from the current research of this thesis, the Bayesian emulator was able to model the discontinuous functions to some degree using goodness-of-fit techniques to measure

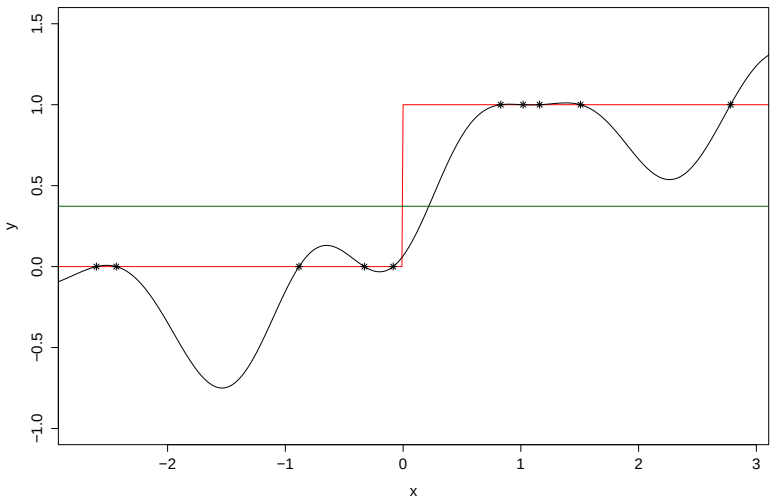

Figure 6.1: Simple example of the emulator modelling the Heaviside function

the accuracy of the emulator, and from this, some research techniques were obtained to model the location of the discontinuities.

During the research, the question arose as to whether it is possible to give information to the emulator that the function contains a discontinuity using the prior/regression function. In Section 3.5 it was shown how a discontinuous function was used as a prior and how it is impractical, as assumptions had to be made about the location of the discontinuities. If the location of the discontinuity is incorrect and there is data given that conflicts with the prior, the emulator will be very inaccurate, as shown for the example in Figure 3.9. If the discontinuity's location is known with complete certainty, then the emulator can still be used to model the simulator. However, as seen in Section 3.5, it may be best to use only one emulator rather than splitting the simulator and using multiple emulators just to avoid the discontinuities (like the approach used in Caiado and Goldstein (2015)). This is because by splitting the simulator, information would be lost between the emulations.

During the one-dimensional experiments with one discontinuity, it was found that there were no over/undershoots between the two data points before and after the discontinuity. Initially it was expected that the emulator would behave similarly to Figure 6.2. However, when experimenting with two or more discontinuities, there was some overshooting.

One concern about this research is how much the correlation function is influencing the results compared to the emulator. It is possible that a different correlation function would change the results (for better or worse). The correlation function used in this research represented part of a Gaussian function (the exponential part), and it was taken directly from Oakley's thesis (Oakley, 1999). Other research papers have used similar exponential functions (Caiado & Goldstein, 2015; Chen et al., 2015; Conti et al., 2009; Hankin, 2012; M. C. Kennedy & O'Hagan, 2001; Montagna

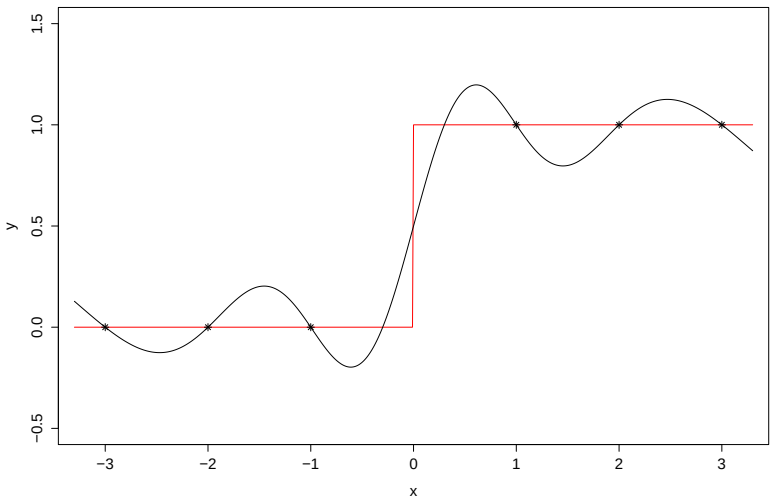

Figure 6.2: An initial expectation of the emulator, containing overshooting and undershooting between the two closest data points either side discontinuity (this is an intentionally incorrect representation of the emulator)

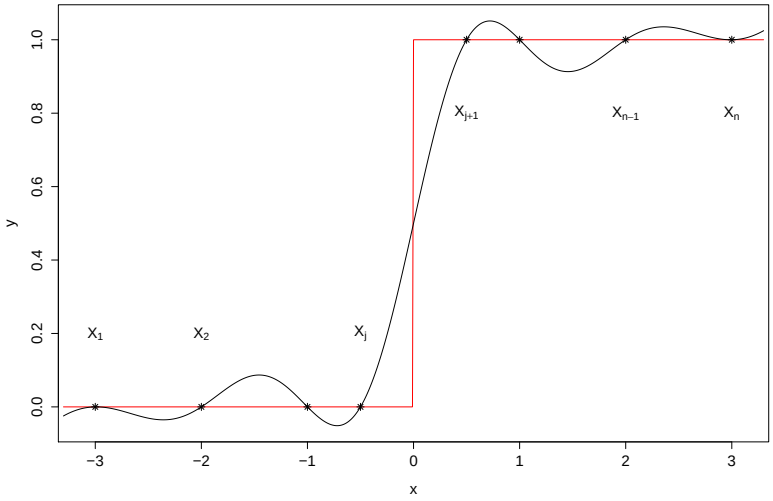

Figure 6.3: Example of the emulator, containing no overshooting and undershooting between the two closest data points either side discontinuity

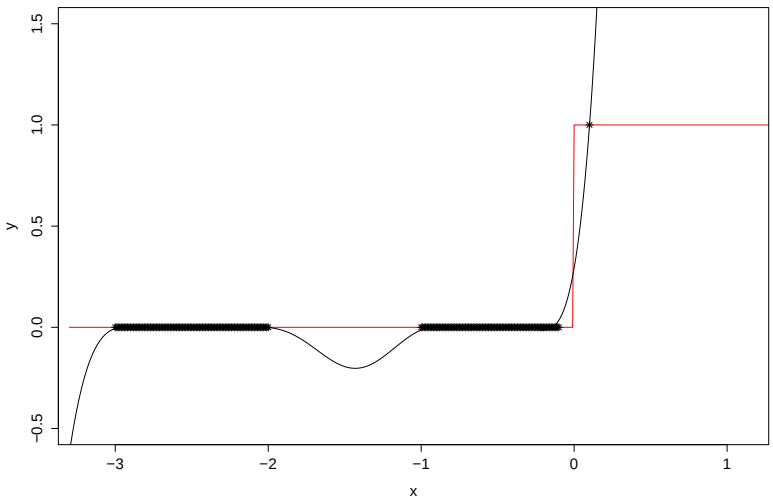

Figure 6.4: Even in the extreme case with many data points, the emulator has no over/undershooting between the two data points on either side of the discontinuity.

& Tokdar, 2016; Oakley, 2002; Zhang et al., 2015).

6.2 Conclusions

Goodness of Fit

Several techniques were used to measure how well the emulator would perform with step discontinuity functions. These goodness-of-fit techniques were applied for each of the following cases:

- A one-dimensional function with only one discontinuity (Heaviside function)
- A one-dimensional function with two discontinuities
- A two-dimensional function
- A two-dimensional function with time (the two-dimensional shape would change over time)

The goodness-of-fit techniques were:

Mean Square Error (MSE)

The **Mean Square Error** is measured only across the sampled region as outside the data points region, the emulator becomes inaccurate resulting in a higher MSE the further away $\hat{\eta}(\mathbf{x})$ is from the data points.

In all the experiments it was found that the PDF of the MSE would occasionally be extremely high, which was dependent on the simulator runs (see Figure 5.3, page

45). This large error of the emulator would increase if there were a large distance between two data points (Figure 5.7 on page 47 is one example when the MSE would be high). This is because the further away the position from which the emulator is approximating the data points, the more uncertain the emulator is, leading to the emulator to lose its accuracy. To avoid having a large error, it was found that the data points should be as evenly spaced as possible and runs of the emulator outside the data points range should be limited. Throughout all experiments, the MSE would decrease the more data points from the simulator were obtained; however, there is a limit to the number of runs of the simulator in a real-world example due to the cost.

Solving $\hat{\eta}(\mathbf{x}) = \eta(\mathbf{d})$ for x and Jaccard index

The function with one discontinuity, $\mathbf{x}$ was then compared to the true location of the discontinuity ($d = 0$); however, with multiple discontinuities, the emulator was post-processed to be:

$$\begin{array}{ll} 1 & \text{if } \hat{\eta}(\mathbf{x}) \geq 0.5 \\ 0 & \text{if } \hat{\eta}(\mathbf{x}) < 0.5 \end{array} \tag{6.1}$$

Then a Jaccard approach was also used, by looking at how much the emulator's prediction of the region correctly matched the simulator (see Figure 5.39 on page 68 for an example).

To measure the accuracy of the emulator, the equation $\frac{B}{A+B+C}$ was used. Region D is not used in the equation, as by increasing the sample space the accuracy of the emulator can be easily influenced.

In the one-dimensional case it was found that the distribution of the discontinuity's location was similar to a Laplace distribution: $f(x|\mu, b) = \frac{1}{2b} e^{\left(-\frac{|x-\mu|}{b}\right)}$. Given a limited amount of data points, a leave-one-out bootstrap method could be used to approximate the parameters of the Laplace distribution, b and μ, and then a confidence interval of the location of the discontinuity may be calculated. When increasing the complexity of the function to have multiple discontinuities, a Jaccard similarity index was applied. Figures 5.30, 5.42 and 5.58 (see pages 61, 69 and 79) show a summary of the results, showing an increase of accuracy when more data points are given to the emulator. The more complex the simulator is, the more data points are needed to achieve a high accuracy.

Comparing $\hat{\eta}(0)$ to $\eta(0)$

In the first two experiments the location of the discontinuity is at 0, therefore it is expected that $\hat{\eta}(0)$ will be 0.5. For the first function of the experiments with only one discontinuity (the Heaviside Function), it was found that the emulator did not over or undershoot between the two data points (from the Heaviside function / "simulator")

of either side of the discontinuity. This result has yet to be explained. After many runs from the emulator, there was no example of over/undershooting between these two data points. This was not the case with two or more discontinuities as there were examples when it would over/undershoot However, typically $\hat{\eta}(\mathbf{x})$ near the discontinuities was between 0 and 1.

Investigating the steepness of $\hat{\eta}(0)$

This goodness-of-fit technique was applied for the Heaviside function. It was expected that $\frac{d\eta(0)}{d\mathbf{x}}$ should be extremely high at the discontinuity. However, this was only the case if the distance of two between the data points either side of the discontinuity was extremely small. This goodness-of-fit technique did not provide new information about the performance of the emulator and was not applied for the more complex functions.

6.3 Further Research

From this thesis, the following topics could be addressed as future research. There is a wide range of fields and applications which this thesis has not covered. The purpose of this thesis was to be a foundation of how emulators model discontinuities and how the accuracy of the emulator can be measured.

Correlation Function

For this thesis, $c(\mathbf{x}, \mathbf{x}') = e^{-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')}$ was used for the correlation function, as it is the most common correlation function used in a wide range of research (for example, Caiado & Goldstein, 2015; Chen et al., 2015; Conti et al., 2009; Hankin, 2012; M. C. Kennedy & O'Hagan, 2001; Montagna & Tokdar, 2016; Oakley, 2002; Zhang et al., 2015). However, there are examples where this correlation function performed poorly, Figure 3.4 is an example but may also be corrected with the prior/regression function. There may exist a correlation function more suitable and accurate for discontinuous functions, such as a Laplace-type equation. However these more "better" correlation functions that has yet to be discovered and/or tested.

A full study on the optimisation of B or other smoothness parameters in the correlation functions has yet to be conducted with discontinuous functions (some preliminary work on optimising B with the popular correlation function is presented in the appendix of this thesis, in Appendix B).

Regression Function

It was found that a regression function for the prior; $H = [1]$ to suitable when modelling step-discontinuous functions using Bayesian emulation, using a discontinuous prior may only be practical if the position of the discontinuities are known beforehand. It could be suggested to fix the prior to a constant (e.g. 0 or 0.5), however this will require altering the BACCO package (more specifically the interpolant function) for this to work.

More Complex Simulators

One of the motivations of this research was the flow of heavy gas dispersion (HGD) travelling across a terrain. This is a real-world application and can be seen as an example that contains a discontinuous region at the boundary of the gas. Inside the region there is a high concentration on the gas (which could be seen as toxic) and just outside of the region there is no gas present. During the experiments with Bayesian emulation, the complexity of the step discontinuous function was increased from a Heaviside function to a simple model that could be represented as an HGD

model. This demonstrated how the emulator was able to model one of the simplest discontinuous functions and move up to a more complex function.

Another field where this research can be expanded is the use of more complex simulators. A suggested simulator could be TWODEE (Hankin & Britter, 1999), which is also used to model the flow of heavy gases across a terrain.

Other Types of Discontinuous Functions

This thesis was mainly focused on step (or jump) discontinuous functions, such as the Heaviside step function. This could be advanced to a function such as equation 6.2, Figure 6.5. These functions may be piecewise, and could contain multiple discontinuities (the number of discontinuities might also be unknown). Other non-smooth functions have not been tested in this thesis. Functions with complex numbers are possible with the emulator (Hankin, 2014); however, this also has not yet been tested with discontinuous functions.

$$\eta(x) = \begin{matrix} \sin(e^{-x}) & \text{if } x < 1 \\ x^2 & \text{if } x \geq 1 \end{matrix} \tag{6.2}$$

Discontinuous Functions with known Discontinuity position

It has been shown in Section 3.5, (page 21 of Chapter 3) that the emulator is able to model more complex discontinuous functions more accurately when the discontinuity position is known, this would normally require altering the prior. However, further investigation were outside the scope of this study.

If the location of the discontinuity is known then it maybe be possible to use separate regions from the data points space, or the use of a nugget.

Comparing The Accuracy of The Emulator to RDD

Regression Discontinuity Design (RDD) is one technique that is currently being used to model discontinuous functions in some fields, for example economics (Imbens & Lemieux, 2008), and pharmacology (Moscoe et al., 2015). Regression Discontinuity Design can also be used when the location of the discontinuity is unknown (Porter & Yu, 2015). RDD could be compared to the Bayesian emulator using the same goodness-of-fit techniques (or other appropriate ones).

Goodness-of-fit

This study includes some examples of goodness-of-fit measures that can be used to measure how accuracy the emulator was compared to the true function, theses included a very simple version Jaccard which is normally used in ecology and biology to measure population differences. Some of these goodness-of-fit techniques are

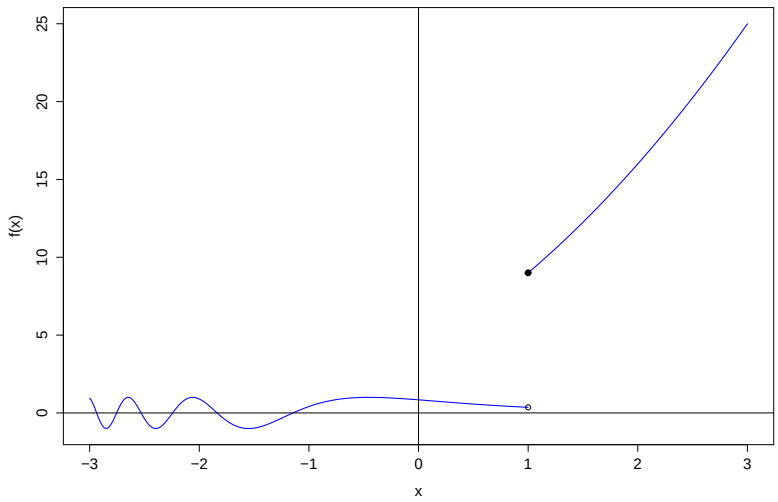

Figure 6.5: Example of a more complex piecewise function, as a suggest for further research

designed from key properties of a step discontinuous functions, however these techniques are not perfect as they alone may not answer what makes a "good" emulator when fitting a discontinuous functions.

6.3.1 Conclusion

Bayesian emulation can be seen as a powerful statical tool to represent simulators. Before this study very little research was done in the implications of using Bayesian emulation for discontinuous functions and it was found that, with the positions being unknown the Bayesian emulator will be less accurate; however, this accuracy may be increased by experimenting with the prior/regression and correlation function and measuring using some goodness-of-fit techniques.

Appendix A

Emulator Properties and Proofs

All of these properties of the emulation are important features of finding a discontinuity or calculating how accurate the emulator is when modelling a function with a discontinuity. Some of the following properties have no analytical solution; in this case, an attempt to find a solution analytically has been done for demonstration and further research may be required. For the research in this thesis the properties that were not solvable analytically will be solved numerically.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} : \mathbf{x}_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \in \mathbb{R}^k$ be a suitable amount of fixed unique input parameters, known as the data input and $\mathbf{y} = [\eta(\mathbf{x}_1), \eta(\mathbf{x}_2), \ldots, \eta(\mathbf{x}_n)]^T$ be a set of results of the simulator.

Let $\mathbf{x}$ be a free variable from the domain of $\mathbf{y}$

$$\mathbb{E}(\eta(\cdot)|\mathbf{y}) = m^*(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \mathbf{t}(\mathbf{x})^T \Sigma^{-1} \left(\mathbf{y} - H\hat{\beta}\right) \text{ Eq: 2.30 (Oakley, 1999, p. 14)}$$

where $\mathbf{t}(\mathbf{x}) = [\mathrm{c}(\mathbf{x}, \mathbf{x}_1), \mathrm{c}(\mathbf{x}, \mathbf{x}_2), \ldots, \mathrm{c}(\mathbf{x}, \mathbf{x}_n)]^T$

$$\Sigma = \begin{bmatrix} \mathrm{Var}(\eta(\mathbf{x}_1)) & \mathrm{Cov}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_2)) & \cdots & \mathrm{Cov}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_n)) \\ \mathrm{Cov}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_2)) & \mathrm{Var}(\eta(\mathbf{x}_2)) & & \mathrm{Cov}(\eta(\mathbf{x}_2), \eta(\mathbf{x}_n)) \\ \vdots & & \ddots & \vdots \\ \mathrm{Cov}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_n)) & \cdots & & \mathrm{Var}(\eta(\mathbf{x}_n)) \end{bmatrix}$$

$$m^*(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \mathbf{t}(\mathbf{x})^T \Sigma^{-1} \left(\mathbf{y} - H\hat{\beta}\right)$$

or

$$m^*(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \sum_{i=1}^{n} r_i(x) \left(y_i - \mathbf{h}(x_i)^T \hat{\beta}\right) \text{ Eq: 2.40 (Oakley, 1999, p. 21)}$$

where $r_i(x)$ is the i-th element of $\mathbf{t}(\mathbf{x})^T \Sigma^{-1}$

Property 1 Given $\mathrm{c}(\mathbf{x}, \mathbf{x}') = e^{-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')}$ and $m^*(\mathbf{x}) = \gamma$, where γ is known. As the number of data points increase (n); it becomes very difficult to find a non-trivial solution of $\mathbf{x}$ analytically.

*Proof of **Property 1**.*

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2\} : \mathbf{x}_i = [x] \in \mathbb{R}$ and $\mathbf{y} = [\eta(\mathbf{x}_1), \eta(\mathbf{x}_2)]^T$

let $\mathrm{c}(\mathbf{x}, \mathbf{x}_i) = e^{-(\mathbf{x}-\mathbf{x}_i)^T B(\mathbf{x}-\mathbf{x}_i)}$ $B = 1$

$m^*(\mathbf{x}) = \gamma$

$m^*(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \mathbf{t}(\mathbf{x})^T \Sigma^{-1} \left(\mathbf{y} - H\hat{\beta}\right)$

m^* is then expanded to get the following:

$$m^*(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \begin{bmatrix} e^{-(\mathbf{x}-\mathbf{x}_1)^T B(\mathbf{x}-\mathbf{x}_1)} \\ e^{-(\mathbf{x}-\mathbf{x}_2)^T B(\mathbf{x}-\mathbf{x}_2)} \end{bmatrix}^T \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} y_1 - \mathbf{h}(\mathbf{x}_1)\hat{\beta} \\ y_2 - \mathbf{h}(\mathbf{x}_2)\hat{\beta} \end{bmatrix} \right)$$

Σ is then inverted, and B is replaced, and m^* is then simplified:

$$m^*(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \frac{1}{\Sigma_{11}\Sigma_{22} - \Sigma_{12}\Sigma_{12}} \begin{bmatrix} e^{-(\mathbf{x}-\mathbf{x}_1)^2} \\ e^{-(\mathbf{x}-\mathbf{x}_2)^2} \end{bmatrix}^T \begin{bmatrix} \Sigma_{22} & \Sigma_{21} \\ \Sigma_{12} & \Sigma_{11} \end{bmatrix} \left(\begin{bmatrix} y_1 - \mathbf{h}(\mathbf{x}_1)\hat{\beta} \\ y_2 - \mathbf{h}(\mathbf{x}_2)\hat{\beta} \end{bmatrix} \right)$$

The matrix is then expanded:

$$m^*(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \hat{\beta} + \frac{1}{\Sigma_{11}\Sigma_{22} - \Sigma_{12}\Sigma_{12}} \left\{ \begin{array}{l} \left(y_1 - \mathbf{h}(\mathbf{x}_1)\hat{\beta}\right)\left(\Sigma_{22}e^{-(\mathbf{x}-\mathbf{x}_1)^2} + \Sigma_{12}e^{-(\mathbf{x}-\mathbf{x}_2)^2}\right) + \\ + \left(y_2 - \mathbf{h}(\mathbf{x}_2)\hat{\beta}\right)\left(\Sigma_{21}e^{-(\mathbf{x}-\mathbf{x}_1)^2} + \Sigma_{11}e^{-(\mathbf{x}-\mathbf{x}_2)^2}\right) \end{array} \right\}$$

$m^*(\mathbf{x}) = \gamma$

$$\mathbf{h}(\mathbf{x})^T \hat{\beta} + \frac{1}{\Sigma_{11}\Sigma_{22} - \Sigma_{12}\Sigma_{12}} \left\{ \begin{array}{l} \left(y_1 - \mathbf{h}(\mathbf{x}_1)\hat{\beta}\right)\left(\Sigma_{22}e^{-(\mathbf{x}-\mathbf{x}_1)^2} + \Sigma_{12}e^{-(\mathbf{x}-\mathbf{x}_2)^2}\right) + \\ + \left(y_2 - \mathbf{h}(\mathbf{x}_2)\hat{\beta}\right)\left(\Sigma_{21}e^{-(\mathbf{x}-\mathbf{x}_1)^2} + \Sigma_{11}e^{-(\mathbf{x}-\mathbf{x}_2)^2}\right) \end{array} \right\} = \gamma$$

the above equation will now be reduced to be as simple as possible with constants

$$\mathbf{h}(\mathbf{x})^T \hat{\beta} + k_1\left(\Sigma_{22}e^{-(\mathbf{x}-\mathbf{x}_1)^2} + \Sigma_{12}e^{-(\mathbf{x}-\mathbf{x}_2)^2}\right) + k_2\left(\Sigma_{21}e^{-(\mathbf{x}-\mathbf{x}_1)^2} + \Sigma_{11}e^{-(\mathbf{x}-\mathbf{x}_2)^2}\right) = \gamma$$

$$\mathbf{h}(\mathbf{x})^T \hat{\beta} + (k_2\Sigma_{11} + k_1\Sigma_{22})\left(e^{-(\mathbf{x}-\mathbf{x}_1)^2}\right) + (k_1\Sigma_{12} + k_2\Sigma_{21})\left(e^{-(\mathbf{x}-\mathbf{x}_1)^2}\right) = \gamma$$

$$\mathbf{h}(\mathbf{x})^T \hat{\beta} + k_1^* e^{-(\mathbf{x}-\mathbf{x}_1)^2} + k_2^* e^{-(\mathbf{x}-\mathbf{x}_2)^2} = \gamma$$

$\mathbf{h}(\mathbf{x})^T$ is commonly seen as$[\mathbf{x} + 1]$

$k_1^* e^{-(\mathbf{x}-\mathbf{x}_1)^2} + k_2^* e^{-(\mathbf{x}-\mathbf{x}_2)^2} = a\mathbf{x} + b$

let $\mathbf{x} = \mathbf{x} - \mathbf{x}_1$

$e^{-\mathbf{x}^2} + \varphi e^{-(\mathbf{x}-\alpha)^2} = a\mathbf{x} + b$

$$e^{-\mathbf{x}^2} + \varphi e^{-(\mathbf{x}-\alpha)^2} = a\mathbf{x} + b \tag{A.1}$$

This becomes the simplest form for $n = 2$,[1] and because of the exponentials, it would be difficult to inverse the above equation analytically[2] to solve for $\mathbf{x}$. Because of this, finding the analytically solution of $m^*(\mathbf{x})$ is beyond the scope of this thesis.

□

[1] $\sum_{i=1}^{n} k_i^* e^{-(\mathbf{x}-\mathbf{x}_i)^2} = \gamma - \mathbf{h}(\mathbf{x})^T \hat{\beta}$ would be used for n many data points

[2] An attempt to solve this using Wolfram Alpha and Matlab were unsuccessful

Property 2 Given $\mathrm{c}\left(\mathbf{x}, \mathbf{x}'\right) = e^{-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')}$

when $n > 3$ the equation: $\frac{d}{d\mathbf{x}} m^*\left(\mathbf{x}\right) = \mathbf{0}$ has no analytical solution to find any non-trivial for $\mathbf{x}$.

*Proof of **Property 2**.* $\frac{d}{d\mathbf{x}} m^*\left(\mathbf{x}\right) = \mathbf{0}$

$$\frac{d}{d\mathbf{x}} m^*\left(\mathbf{x}\right) = \frac{d}{d\mathbf{x}}\left(\mathbf{h}\left(\mathbf{x}\right)^T \hat{\beta}\right) + \frac{d}{d\mathbf{x}}\mathbf{t}\left(\mathbf{x}\right)^T \Sigma^{-1}\left(\mathbf{y} - H\hat{\beta}\right) = 0$$

$$\frac{d}{d\mathbf{x}}\mathbf{t}\left(\mathbf{x}\right)^T = \left[\frac{d}{d\mathbf{x}}\mathrm{c}\left(\mathbf{x}, \mathbf{x}_1\right), \frac{d}{d\mathbf{x}}\mathrm{c}\left(\mathbf{x}, \mathbf{x}_2\right), \ldots \frac{d}{d\mathbf{x}}, \mathrm{c}\left(\mathbf{x}, \mathbf{x}_n\right)\right]$$

$$\text{let } \Sigma^{-1} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \cdots & \Gamma_{1n} \\ \Gamma_{12} & \Sigma_{22}^{-1} & & \Gamma_{2n} \\ \vdots & & \ddots & \vdots \\ \Gamma_{1n} & \cdots & & \Gamma_{nn} \end{bmatrix}$$

$$\frac{d}{d\mathbf{x}}\mathbf{t}\left(\mathbf{x}\right)^T = -\left[\begin{bmatrix} \left(\sum_{i=1}^{k} B_{i1}(x_1 - x_{11}) + \sum_{i=1}^{k} B_{1i}(x_1 - x_{11})\right) e^{-(\mathbf{x}-\mathbf{x}_1)^T B(\mathbf{x}-\mathbf{x}_1)} \\ \left(\sum_{i=1}^{k} B_{i2}(x_2 - x_{12}) + \sum_{i=1}^{k} B_{2i}(x_2 - x_{12})\right) e^{-(\mathbf{x}-\mathbf{x}_1)^T B(\mathbf{x}-\mathbf{x}_1)} \\ \vdots \\ \left(\sum_{i=1}^{k} B_{ik}(x_k - x_{1k}) + \sum_{i=1}^{k} B_{ki}(x_k - x_{1k})\right) e^{-(\mathbf{x}-\mathbf{x}_1)^T B(\mathbf{x}-\mathbf{x}_1)} \end{bmatrix}, \begin{bmatrix} \left(\sum_{i=1}^{k} B_{i1}(x_1 - x_{21}) + \sum_{i=1}^{k} B_{1i}(x_1 - x_{21})\right) e^{-(\mathbf{x}-\mathbf{x}_2)^T B(\mathbf{x}-\mathbf{x}_2)} \\ \left(\sum_{i=1}^{k} B_{i2}(x_2 - x_{22}) + \sum_{i=1}^{k} B_{2i}(x_2 - x_{22})\right) e^{-(\mathbf{x}-\mathbf{x}_2)^T B(\mathbf{x}-\mathbf{x}_1)} \\ \vdots \\ \left(\sum_{i=1}^{k} B_{ik}(x_k - x_{2k}) + \sum_{i=1}^{k} B_{ki}(x_k - x_{2k})\right) e^{-(\mathbf{x}-\mathbf{x}_2)^T B(\mathbf{x}-\mathbf{x}_2)} \end{bmatrix}, \ldots, \begin{bmatrix} \left(\sum_{i=1}^{k} B_{i1}(x_1 - x_{\mathbf{n}1}) + \sum_{i=1}^{k} B_{1i}(x_1 - x_{\mathbf{n}1})\right) e^{-(\mathbf{x}-\mathbf{x}_n)^T B(\mathbf{x}-\mathbf{x}_n)} \\ \left(\sum_{i=1}^{k} B_{i2}(x_2 - x_{n2}) + \sum_{i=1}^{k} B_{2i}(x_2 - x_{\mathbf{n}2})\right) e^{-(\mathbf{x}-\mathbf{x}_n)^T B(\mathbf{x}-\mathbf{x}_n)} \\ \vdots \\ \left(\sum_{i=1}^{k} B_{ik}(x_k - x_{\mathbf{n}k}) + \sum_{i=1}^{k} B_{ki}(x_k - x_{\mathbf{n}k})\right) e^{-(\mathbf{x}-\mathbf{x}_n)^T B(\mathbf{x}-\mathbf{x}_n)} \end{bmatrix}\right]$$

$$\frac{d}{d\mathbf{x}}\mathbf{t}\left(\mathbf{x}\right)^T \Sigma^{-1}\left(\mathbf{y} - H\hat{\beta}\right) =$$

$$= -\begin{bmatrix} \left(\sum_{i=1}^{k} B_{i1}(x_1 - x_{11}) + \sum_{i=1}^{k} B_{1i}(x_1 - x_{11})\right) e^{-(\mathbf{x}-\mathbf{x}_1)^T B(\mathbf{x}-\mathbf{x}_1)} & \ldots & \left(\sum_{i=1}^{k} B_{i1}(x_1 - x_{\mathbf{n}1}) + \sum_{i=1}^{k} B_{1i}(x_1 - x_{\mathbf{b}1})\right) e^{-(\mathbf{x}-\mathbf{x}_n)^T B(\mathbf{x}-\mathbf{x}_n)} \\ \left(\sum_{i=1}^{k} B_{i2}(x_2 - x_{12}) + \sum_{i=1}^{k} B_{2i}(x_2 - x_{12})\right) e^{-(\mathbf{x}-\mathbf{x}_1)^T B(\mathbf{x}-\mathbf{x}_1)} & \ldots & \left(\sum_{i=1}^{k} B_{i2}(x_2 - x_{\mathbf{n}2}) + \sum_{i=1}^{k} B_{2i}(x_2 - x_{\mathbf{n}2})\right) e^{-(\mathbf{x}-\mathbf{x}_n)^T B(\mathbf{x}-\mathbf{x}_n)} \\ \vdots & \ldots & \vdots \\ \left(\sum_{i=1}^{k} B_{ik}(x_k - x_{1k}) + \sum_{i=1}^{k} B_{ki}(x_k - x_{1k})\right) e^{-(\mathbf{x}-\mathbf{x}_1)^T B(\mathbf{x}-\mathbf{x}_1)} & \ldots & \left(\sum_{i=1}^{k} B_{ik}(x_k - x_{\mathbf{n}k}) + \sum_{i=1}^{k} B_{ki}(x_k - x_{\mathbf{n}k})\right) e^{-(\mathbf{x}-\mathbf{x}_n)^T B(\mathbf{x}-\mathbf{x}_n)} \end{bmatrix} \Sigma^{-1} \begin{bmatrix} y_1 - H\hat{\beta} \\ y_2 - H\hat{\beta} \\ \vdots \\ y_n - H\hat{\beta} \end{bmatrix} = 0$$

At this point, if the next step was to expand the matrix, it would become is a similar problem to **Property 1** with exponential functions, and an analytical solution can not be found. □

Appendix B

Investigation into Optimising B

B.1 Introduction

This piece of research was done before the main part of the findings (in Chapter 5). For the correlation function (which includes B): $c(\mathbf{x}, \mathbf{x}') = \exp\left(-(\mathbf{x}-\mathbf{x}')^T B(\mathbf{x}-\mathbf{x}')\right)$ from equation 2.4 comes from Oakley (1999).

For this part of the investigation, the mean square error MSE technique (similar to what was done in section 5.2.1) was used to optimise B in the correlation function of the emulator.

B.2 Findings

One-Dimensional Case

For Figure B.1, with the data points of $[1, 1, 1, 0, 0]$ the MSE between the domain of -3.5 and 3.5 was about 0.098; however, between the domain of -1 and 1 (near the discontinuity) the MSE was 0.31. This will be denoted as $0.098\,(0.31)$.

Next, B was altered from $c(\mathbf{x}, \mathbf{x}')$. Picking any number higher than 1, B is imputed to 10, and as a result the MSE was decreased to $0.081\,(0.23)$ (see Figure B.2). When B was increased over 100 the MSE was decreased more to a minimum of $0.074\,(0.17)$; however, this turns into a linear regression model, questioning whether an MSE is the right approach to comparing models.

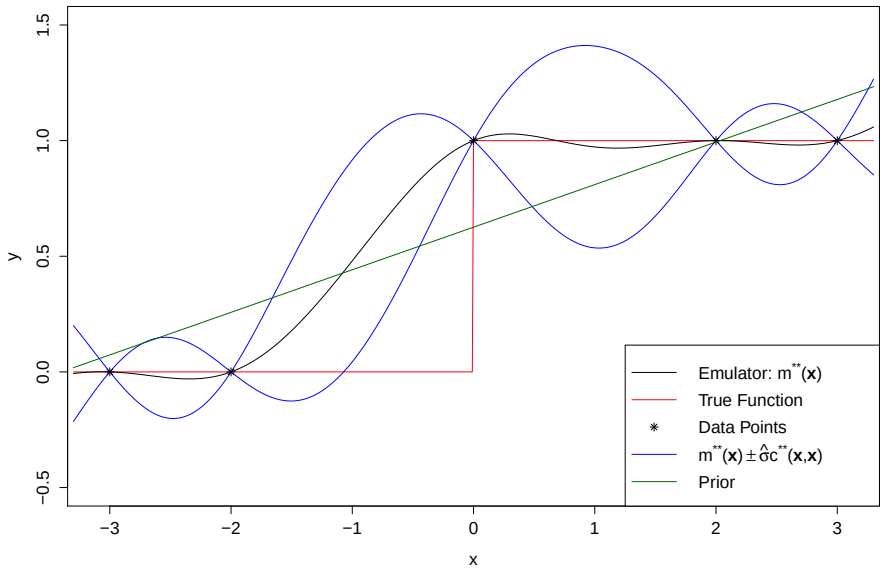

Figure B.1: Bayesian emulator modelling Heaviside function with a linear prior: $H(x) = [1, x]$

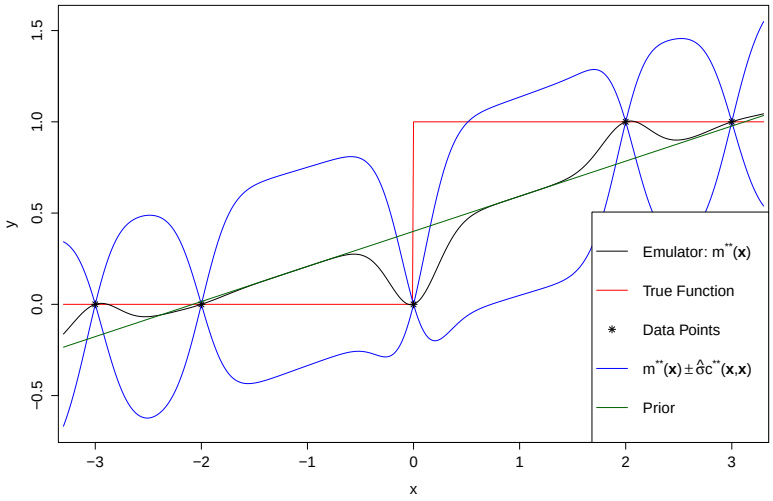

Figure B.2: Bayesian emulation: $H(x) = [1, x]$, $c(\mathbf{x}, \mathbf{x}') = \exp\left(-(\mathbf{x}-\mathbf{x}')^T \times 10 \times (\mathbf{x}-\mathbf{x}')\right)$,MSE:0.081 (0.23)

Two-Dimensional Case

For the two-dimensional investigation, 20 samples were generated as observed data using Latin hypercube sampling. Figure B.3 is one such example of the observed data. The data points are then evaluated from the function (similar to the one-dimensional case) and are stored as observed outputs y to be used for the emulator. Using the observed input and output, the emulator then estimates the function.

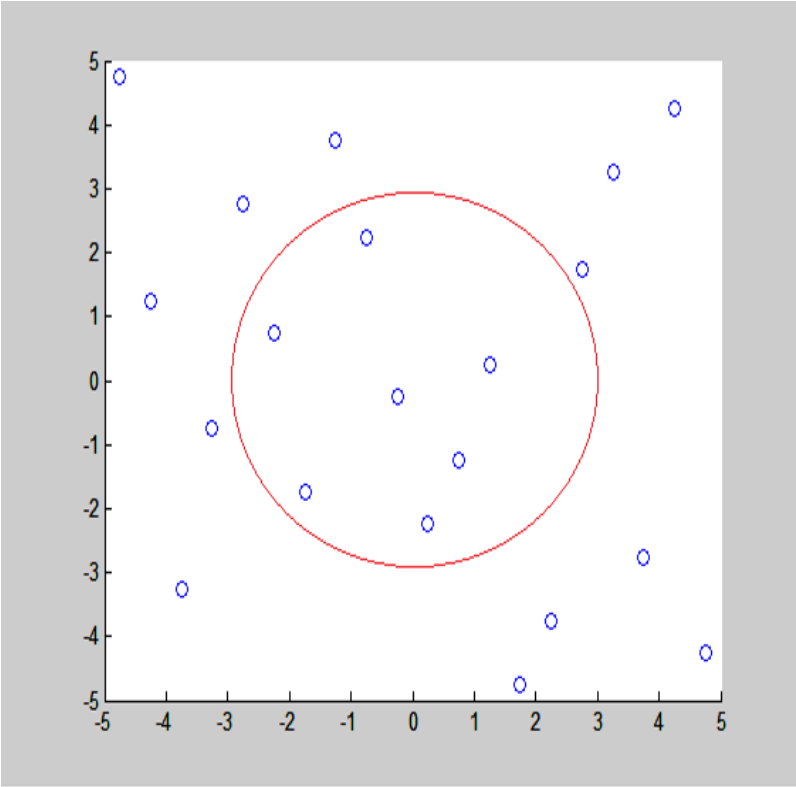

Figure B.3: 20 data points from the simulator, the Red line is boundary of the region

Figure B.4 and Figure B.5 are examples of these outputs with the prior as $H = [1, X]$, and distance function as $c(\mathbf{x}, \mathbf{x}') = \exp\left(-(\mathbf{x}-\mathbf{x}')^T \times B \times (\mathbf{x}-\mathbf{x}')\right)$, where $B = [1, 1]$.

It is observed from Figures B.6, B.7, and B.8 that the smoothness of the function changes as B changes. The leave-one-out bootstrap method could be used later in this investigation to optimise B.

The next part of this investigation is to adjust the prior; by removing the prior (i.e. $H = [0]$) the emulator generates something similar to Figure B.9. Comparing Figure B.9 with the other figures where the prior was $H = [1, X]$, it can be seen that there is an improvement when modelling outside the boundary as the function is pulled towards 0, making it look more stable.

Figure B.10 was calculated with 250 samples, showing, as expected, that the more data points the better the approximation is. It is noted that once there is an approximation of where the boundary is, most of the data points inside and outside the region become redundant. This leads to the following questions: if there are limited runs of the function/simulator available, how can the inputs be chosen effectively, and how can the prior/emulator be changed so there is the understanding that this function contains discontinuities?

To begin to answer these questions, the emulator's output were adjusted as a post-processing feature, so that everything inside the boundary has the same value, such that $\hat{\eta} = \begin{cases} 1 & \text{if } \hat{\eta} \geq \alpha \\ 0 & \text{if } \hat{\eta} < \alpha \end{cases}$. This will provide an approximation of the boundary

of the function in which Figure B.11 demonstrates. of the boundary.

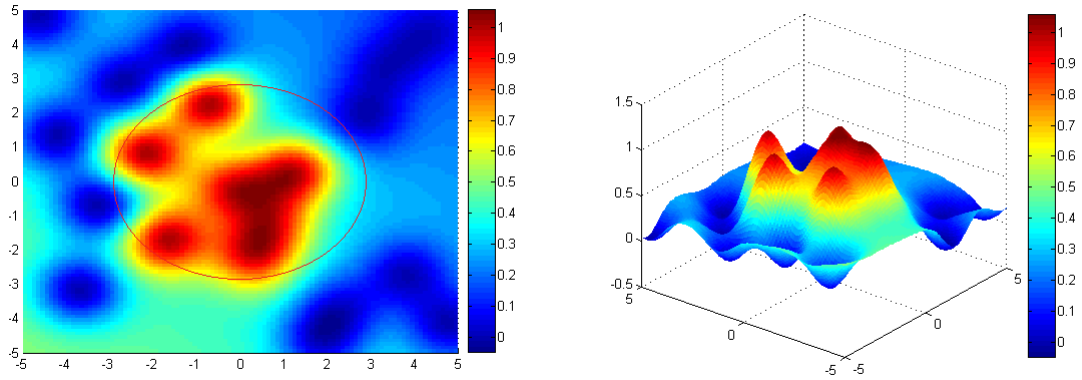

Figure B.4: Output from the emulator with $B = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. 20 data points

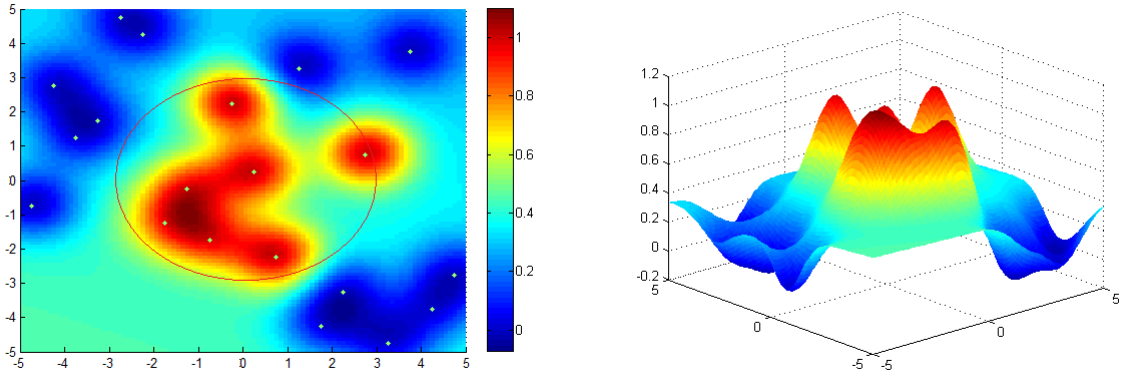

Figure B.5: Output from the emulator with $B = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. 20 data points

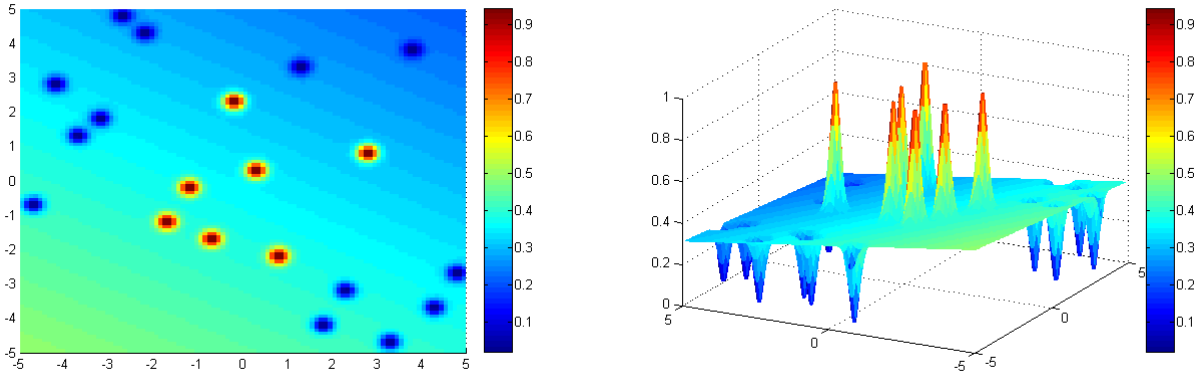

Figure B.6: Output from the emulator with $B = \left[\begin{smallmatrix} 10 & 0 \\ 0 & 10 \end{smallmatrix}\right]$. 20 data points

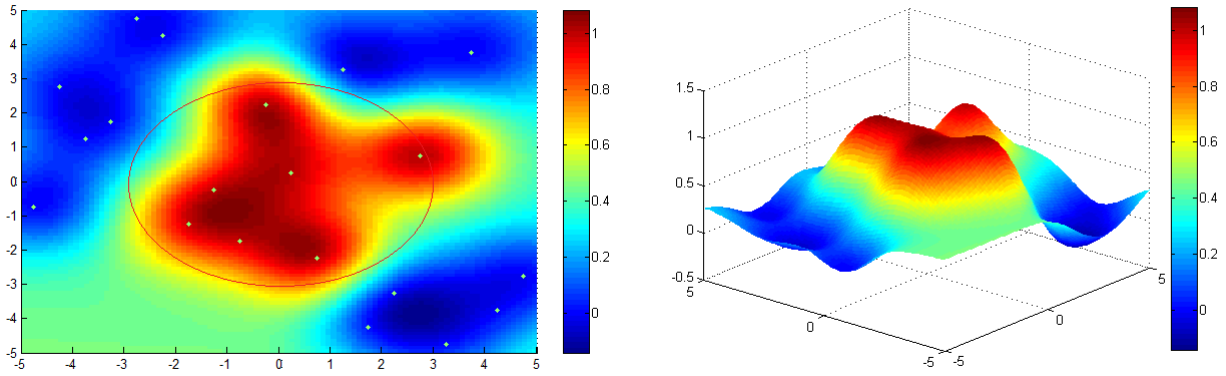

Figure B.7: output from the emulator with $B = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$. 20 data points

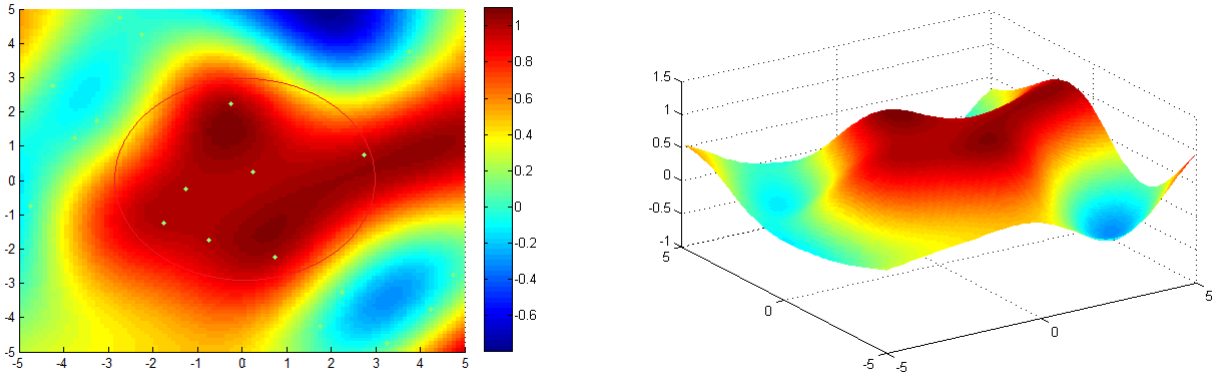

Figure B.8: Output from the emulator with $B = \begin{bmatrix} \frac{1}{10} & 0 \\ 0 & \frac{1}{10} \end{bmatrix}$. 20 data points

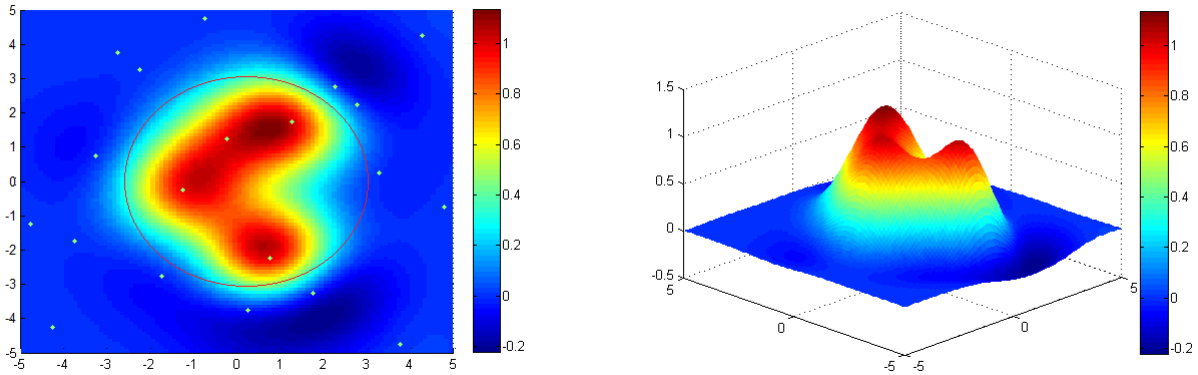

Figure B.9: Output from the emulator with $B = \begin{bmatrix} 0.45 & 0 \\ 0 & 0.45 \end{bmatrix}$, and Prior = 0. 20 data points

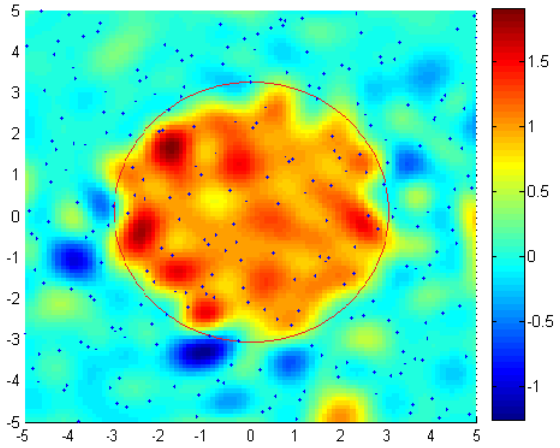

Figure B.10: Output from the emulator with $B = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. 250 data points (large data set case)

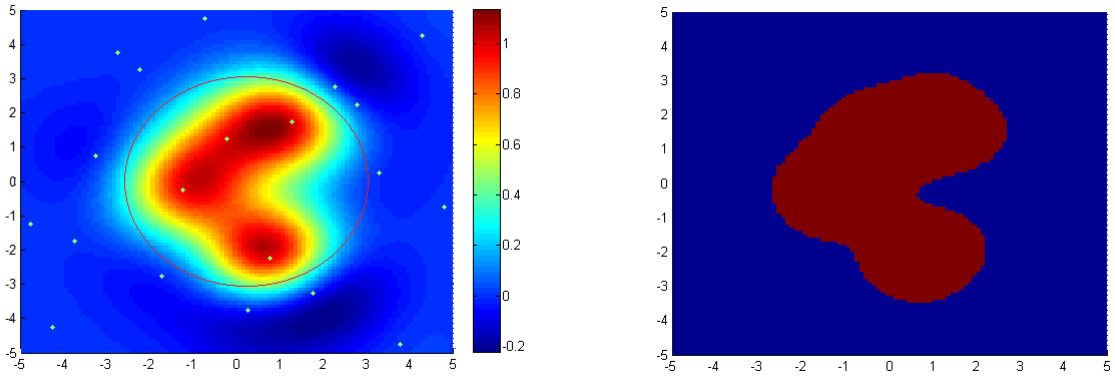

Figure B.11: (Left) emulator before post-processing, (right) after with $\hat{\eta} = \begin{cases} 1 & \text{if } \hat{\eta} \geq 0.2 \\ 0 & \text{if } \hat{\eta} < 0.2 \end{cases}$

Another technique used to optimise B was taking the variance of the emulator into account and calculating the percentage of when the true function is inside the emulator's estimation $\pm$ the variance: $(\hat{\eta} \pm \hat{\sigma^2} c^{**})$. It was found that with the data points of $[1, 1, 1, 0, 0]$, B was optimised to 11.5 with the other parameters fixed. However, changing the observed data to: $(-1.2, -0.7, 0.1, 0.5, 1)$ results in an optimised value for B of 240. This indicates that B is dependent on the observed data, and an optimising technique for B is required for each set of data points.

B.3 Conclusion

Because of the added complexity of B and the need to optimise B for every new run, it was decided to keep B 1 to avoid this problem of having to optimise B, allowing the research to continue focusing on how the emulator models discontinuous functions.

Appendix C

Other Minor Observations

Figure C.1 was obtained by adding more observed data, for example: $(-3, -2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5)$, indicating that more data points may not increase the accuracy of the emulator at the discontinuity.

With the observed data more spread apart from the discontinuity, for example $(-10, -5, -3, 2, 5, 10)$, the error (from both MSE and percentage of the function being between $\hat{\eta} \pm \hat{\sigma^2} c^{**}$) has shown the emulator to be better than before (see Figure C.2). However, this could be because of the selected data points.

With the observed data close to the discontinuity, for example $(-0.2, -0.1, 0.1, 0.2, 0.3)$, being so close together, the $\hat{\sigma^2}$ is large (in the example $\hat{\sigma^2} = 76237.98$)

Figure C.3 is produced by changing the regression function to $H(x) = [1]$ and has not made an improvement to the accuracy. Increasing B pulls the estimation closer to the prior, increases the variance, and is not better at modelling the function at the discontinuity.

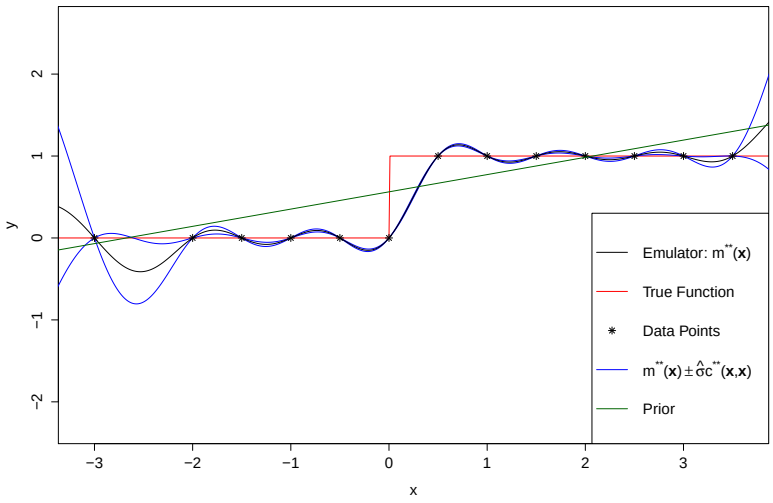

Figure C.1: 13 data points: $H(x) = [1, x]$, $B = 1$

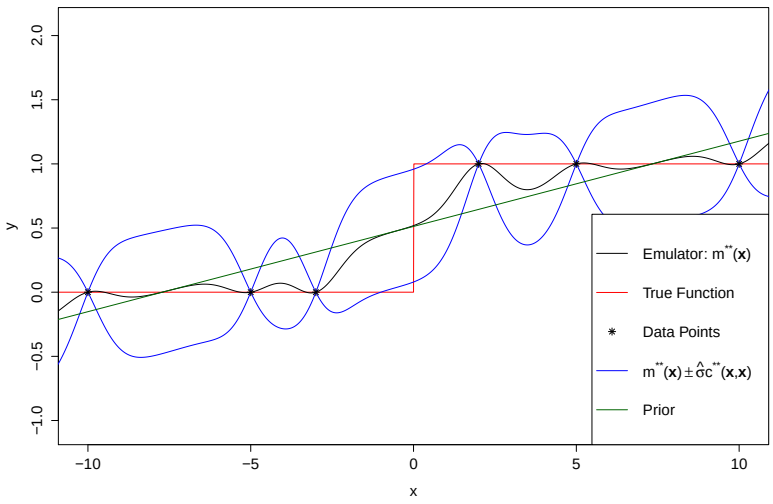

Figure C.2: data points more spread out: $H(x) = [1, x]$, $B = 1$

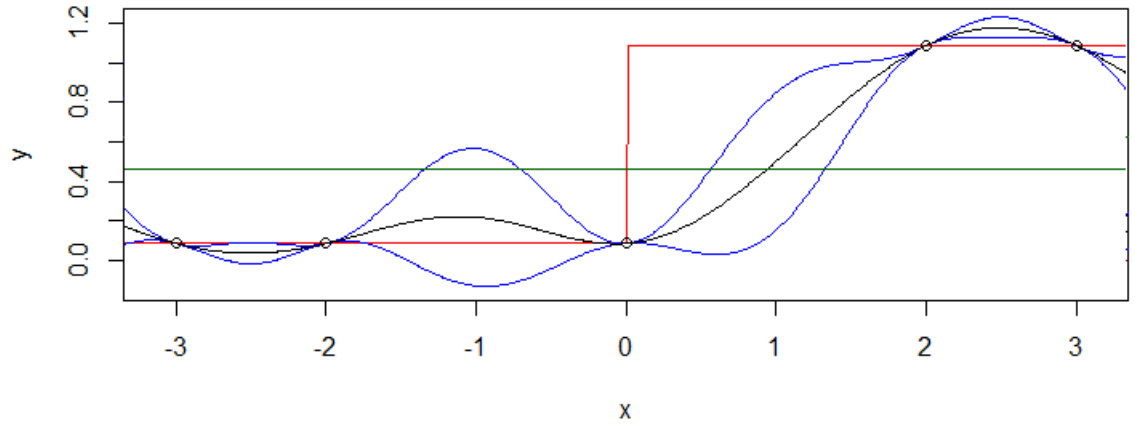

Figure C.3: $H(x) = [1]$, $B = 1$

From Figure C.4 a large change is seen in the emulator's estimation and the variance between the three data points between -2 and 2. Also with data points being symmetric along the x axis; the two graphs are the same if one of them is flipped 180 degrees.

Figure C.5 shows how the emulator behaves when the location of the discontinuity changes. Unless one of the observed data points shows the change, the emulator does not change; however, the residual error does.

Figures C.6 and C.7 are similar to Figures C.4 and C.5 but with the prior $H = [1]$. With the change of prior the variance has increased.

Lastly, Figure C.8 is the emulator with no prior information, i.e. $H = [0]$.

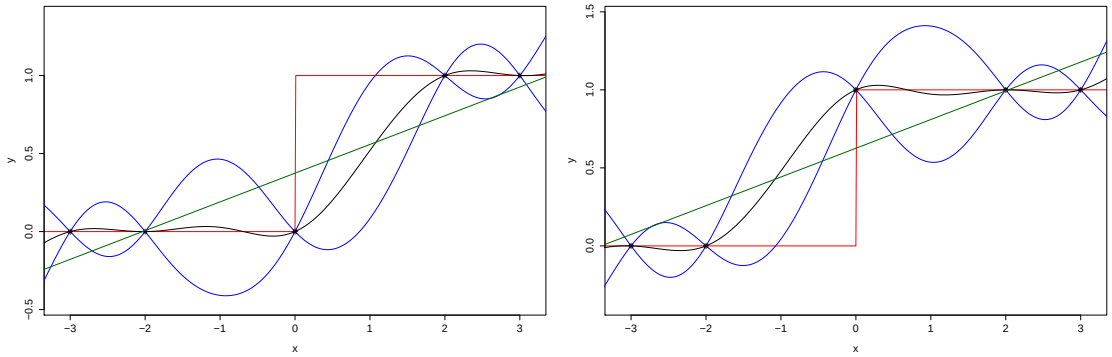

Figure C.4: (Left) $X = [-3, -1, 0, 2, 3]$ (right) $X = [-3, -1, 0.000001, 2, 3]$

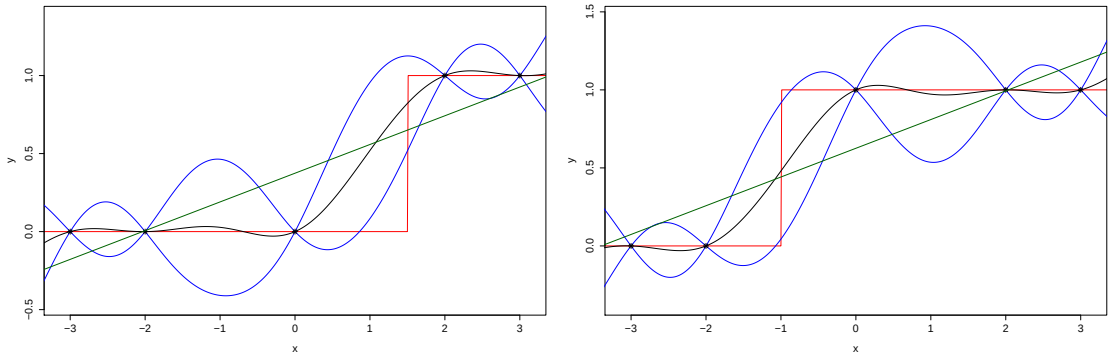

Figure C.5: (Left)$\eta(x) = \begin{cases} 1 & x \leq 1.5 \\ 0 & x > 1.5 \end{cases}$ (right) $\eta(x) = \begin{cases} 1 & x \leq -1 \\ 0 & x > -1 \end{cases}$

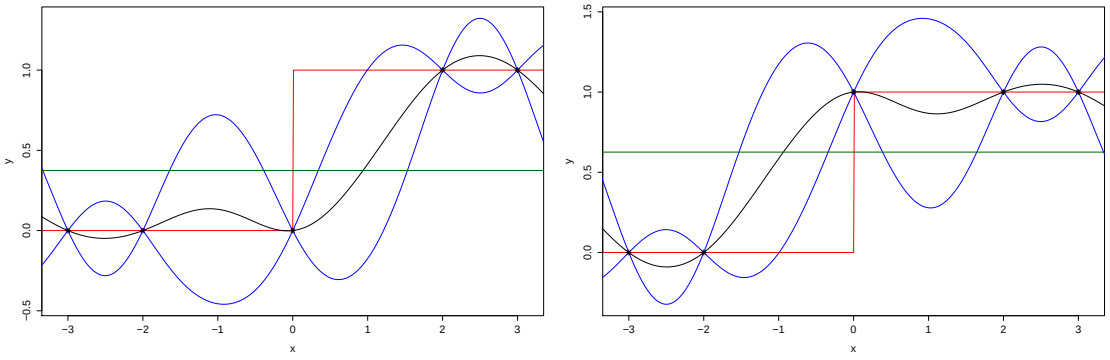

Figure C.6: (left)$X = [-3, -1, 0, 2, 3]$(right)$X = [-3, -1, 0.000001, 2, 3]$, Prior $= [1]$

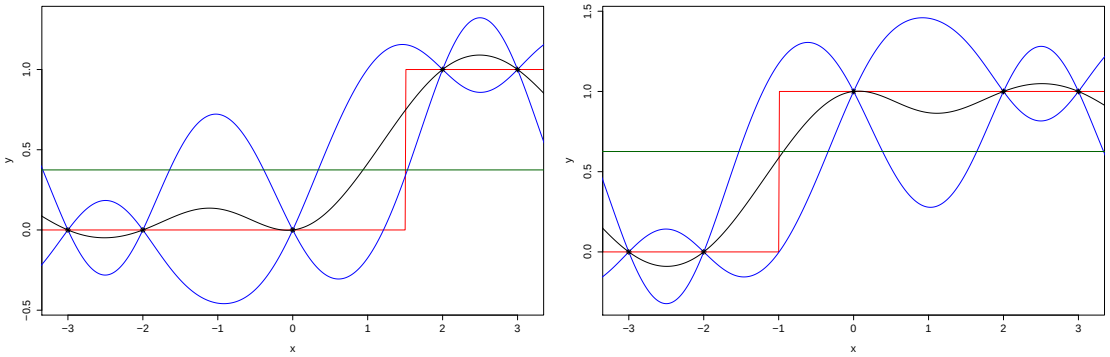

Figure C.7: $(\text{left})\eta(x) = \begin{cases} 1 & x \leq 1.5 \\ 0 & x > 1.5 \end{cases}$ $(\text{right})\eta(x) = \begin{cases} 1 & x \leq -1 \\ 0 & x > -1 \end{cases}$ Prior = [1]

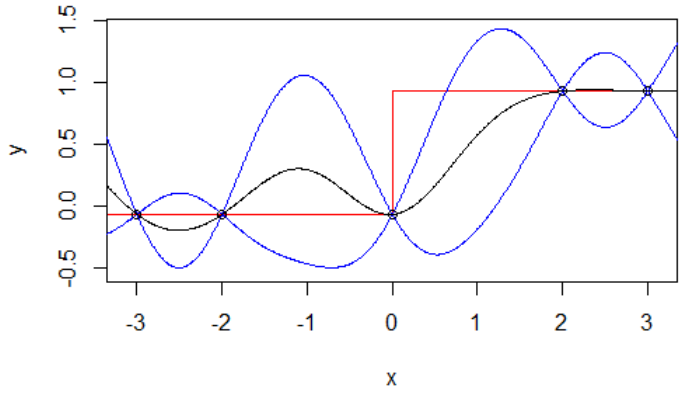

Figure C.8: Without a prior

Appendix D

Software Used

D.1 R

R is a open-source program designed for statistical computing and graphics. The current version used for this thesis is 3.3.1, which was the latest at the time. Additional packages are available through a repository; these packages contain functions.

The package BACCO was used throughout the experiment. BACCO (Bayesian Analysis of Computer Code Output) was published by (Hankin, 2005), contains the computerised implementations of the ideas of M. Kennedy (2000), M. C. Kennedy and O'Hagan (2001), and Oakley (2002) (see Section 2.5 for equations) some adjustments to the code were made in the interpolation function in the Emulator Package to allow $\mathbf{h}(\mathbf{x}) = 0$:.

D.2 Matlab

Matlab (Version Student R2014a) was used for 3D plotting. All data was exported into a ".txt" file then imported to Matlab.

Appendix E

R code

Below are examples of R code that were used in creating Figures obtained results from the emulator

E.1 Figure B.1

```
library("BACCO")
f = function(x) ifelse(x>0,1,0)



X = as.matrix(c(-3,-2,0,2,3))
Y = f(X);

x = seq(-3.3,3.3,0.01)
y = interpolant.quick(as.matrix(x),Y,X,scales = 1,g=T)


#plotting part
plot(x,f(x),type="l",col=c("red"),xlab = "x",ylab = "y",
ylim = c(min(y$mstar.star-y$Z),max(y$mstar.star+y$Z)),
xlim=c(min(X)-0.1,max(X)+0.1)
);
lines(x,y$mstar.star+y$Z,type="l",col=c("blue"))
lines(x,y$mstar.star-y$Z,type="l",col=c("blue"))
lines(x,y$mstar.star,type='l');
lines(sort(X),f(sort(X)),type="p",pch = 8)
lines(x,y$prior,col=c("darkgreen"))
```

```
legend("bottomright",
       c(expression(paste("Emulator: ",m^'**','(',bold(x),'
          ↪ )')),
       "True Function",
       "Data Points",
       expression(paste(m^'**','(',bold(x),')' %|% hat(
          ↪ sigma),
       c^'**','(',bold(x),',',bold(x),')') ),
       "Prior"),
       lty=c(1,1,NA,1,1),pch=c(NA,NA,8,NA),
       col=c("Black","red","Black","blue","darkgreen"))
```

E.2 Figure 5.1

```
library("BACCO")
f = function(x) ifelse(x>0,1,0)



X = as.matrix(c(-3,-2,0,2,3))
Y = f(X);

x = seq(-3.3,3.3,0.01)
y = interpolant.quick(as.matrix(x),Y,X,scales = 1,g=T,
        func = function(x) 0.5)


#plotting part
plot(x,f(x),type="l",col=c("red"),xlab = "x",ylab = "y",
ylim = c(min(y$mstar.star-y$Z),max(y$mstar.star+y$Z)),
xlim=c(min(X)-0.1,max(X)+0.1)
);
lines(x,y$mstar.star+y$Z,type="l",col=c("blue"))
lines(x,y$mstar.star-y$Z,type="l",col=c("blue"))
lines(x,y$mstar.star,type='l');
lines(sort(X),f(sort(X)),type="p",pch = 8)
lines(x,y$prior,col=c("darkgreen"))

legend("bottomright",
c(expression(paste("Emulator: ",m^'**','(',bold(x),')')),
```

```
"True Function",
"Data Points",
expression(paste(m^'**','(',bold(x),')' %|% hat(sigma),
c^'**','(',bold(x),',',bold(x),')') ),
"Prior"),
lty=c(1,1,NA,1,1),pch=c(NA,NA,8,NA),
col=c("Black","red","Black","blue","darkgreen"))
```

E.3 Figure 5.3, 5.4, 5.5 and 5.6

```
f = function(x) ifelse(x>=0,1,0)
n1=2
n2=2
MSE = numeric(5000)


i = 1
while (i < 5000)
{

X = as.matrix(c(runif(N,-3,3))
Y = f(X);

        if (sum(Y)>0 && sum(Y)<(N))
        {


        x = seq(round(min(X),1)-0.1,round(max(X),1)
          ↪ +0.1,0.01)
        y = interpolant.quick(as.matrix(x),Y,X,scales = 1,g=
          ↪ T,
              func = function(x) 0.5)

        MSE[i] = 1/length(x) * sum((y$mstar.star-f(x))^2)
        }
}

hist(remove_outliers(MSE,boxplot(MSE,outline = F)$stats[5]),
        main = 'Histogram of Mean Square Error (excluding
          ↪ outliers)',
```

```
        xlab = 'MSE',20)

boxplot(MSE,outline = FALSE,horizontal = T); title("Boxplot
    ↪ of MSE (no outliers)")

#################################################################
#################################################################


N = 8
MSE = numeric(5000)

i = 1
while (i < 5000)
{

X = as.matrix(c(runif(N,-3,3))
Y = f(X);

        if (sum(Y)>0 && sum(Y)<(N))
        {
                x = seq(round(min(X),1)-0.1,round(max(X),1)
                    ↪ +0.1,0.01)
                y = interpolant.quick(as.matrix(x),Y,X,
                    ↪ scales = 1,g=T
                ,func = function(x) 0.5)

                MSE[i] = 1/length(x) * sum((y$mstar.star-f(x
                    ↪ ))^2)
                i = i + 1
        }
}

hist(remove_outliers(MSE,boxplot(MSE,outline = F)$stats[5]),
        main = 'Histogram of Mean Square Error (excluding
            ↪ outliers)',
        xlab = 'MSE',20)

boxplot(MSE,outline = FALSE,horizontal = T);
        title("Boxplot of MSE (no outliers)")
```

```
boxplot(MSE, outline = FALSE, horizontal = T); title("Boxplot
    ↪ of MSE (no outliers)")

epdfPlot(remove_outliers(MSE, boxplot(MSE, outline = F)$stats
    ↪ [5]), main = 'PDF of MSE', xlab = 'MSE')


plot(ecdf(remove_outliers(MSE, boxplot(MSE, outline = F)$stats
    ↪ [5])),lwd = 1,main = 'CDF of MSE',xlab="MSE")
abline(h=1-(1-0.95)/2,col=2,lty=4)
abline(h=(1-0.95)/2,col=2,lty=4)

legend("bottomright",
c('CDF of MSE',
"95% confidence interval"),
lty=c(1,4),
col=c("Black","red") ,lwd = c(1,1))
```

E.4 Figure 5.7

```
library("BACCO")
f = function(x) ifelse(x>0,1,0)


X = as.matrix(c(-2.19,-0.02,0.02,0.445))
Y = f(X);

x = seq(-3.3,3.3,0.01)
y = interpolant.quick(as.matrix(x),Y,X,scales = 1,g=T,
func = function(x) 0.5)

#plotting part
# plot(...)
```

same plotting code as Figure 5.1

E.5 Figure 5.9 and 5.10

```
f = function(x) ifelse(x>=0,1,0)
MSE = 0
scales = 1
pos.def.matrix <- diag(scales, nrow = length(scales))
nn = 40
N = 500;
MSE = matrix(rep(0,nn*N),ncol = N)
for (n in 1:nn)
{
        for (i in 1:N)
        {

        X = as.matrix(c(runif(n,-3,0),c(runif(n,0,3))))
        Y = f(X);

        A <- corr.matrix(xold = X, pos.def.matrix = pos.def.
           ↪ matrix,
        distance.function = corr)
        Ainv = ginv(A)


        #x = seq(-3,3,0.01)
        x = seq(min(X)-0.1,max(X)+0.1,0.1)
        y = interpolant.quick(as.matrix(x),Y,X,scales = 1,g=
           ↪ T
        ,func = function(x) 0.5,Ainv=Ainv)


        MSE[n,i] = mean((y$mstar.star-f(x))^2)
        }

}

boxplot(t(MSE[,]),outline = F,names = c(2*1:40),
        xlab='no. of datapoints',ylab="MSE")
plot(2*1:40,boxplot(t(MSE[,]),outline = F,
        names = c(2*1:40))$stats[3,],type='l',
        xlab='no. of datapoints',ylab='Median of MSE')
```

E.6 Figure 5.12 , 5.13, 5.14 and 5.15

```
f = function(x) ifelse(x>=0,1,0)

N = 8
x = numeric(0)
x2 = numeric(0)
d = numeric(0)
i = 1
while (i < 5000)
{
        X = as.matrix(runif(N,-3,3))
        Y = f(X);
        if (sum(Y)>0 && sum(Y)<(N))
        {
                g = function (x) (interpolant(x,Y,X,scales =
                    ↪  1,g=F
                ,func = function(x) 0.5) -0.5)^2
                x[i] = optim(0,g)$par
                x2[i] = (X[n1+1]+X[n1])/2
                d[i] = (X[n1+1]-X[n1])

                i = i + 1
        }
}
hist(x,xlab="x",nclass = 30)
qqnorm(x);abline(mean(x),sd(x))



epdfPlot(x, ylim = c(0,1.4))
mu = mean(x)
b = sqrt(var(x)/2)
dLaplace = function(x,mu,b) (1/(2*b))*exp(-abs(x-mu)/b)
pLaplace = function(x,mu,b) ifelse(x<mu,0.5*exp((x-mu)/b)
   ↪ ,1-0.5*exp(-(x-mu)/b))
lines(xx,dLaplace(xx,mu,b),col = "red", lwd = 2)
```

```
legend("topright",
c(expression(paste('PDF␣of␣',bold(x),':␣',m^'*','(',bold(x),
   ↪ ')␣=␣0.5')),
"PDF␣of␣Laplace"),
lty=c(1,1),
col=c("Black","red") ,lwd = c(3,2))


epdfPlot(x, ylim = c(0,1.4))
mu = mean(x)
b = sqrt(var(x)/2)
dLaplace = function(x,mu,b) (1/(2*b))*exp(-abs(x-mu)/b)
pLaplace = function(x,mu,b) ifelse(x<mu,0.5*exp((x-mu)/b)
   ↪ ,1-0.5*exp(-(x-mu)/b))
lines(xx,dLaplace(xx,mu,b),col = "red", lwd = 2)


legend("topright",
c(expression(paste('PDF␣of␣',bold(x),':␣',m^'*','(',bold(x),
   ↪ ')␣=␣0.5')),
"PDF␣of␣Laplace"),
lty=c(1,1),
col=c("Black","red") ,lwd = c(3,2))
```

E.7 Figure 5.17 and 5.18

```
f = function(x) ifelse(x>=0,1,0)

n1 = 8
n2 = 8
x = numeric(0)
i = 1
while (i < 50000)
{
X = as.matrix(c(runif(n1,-3,0),c(runif(n2,0,3))))
Y = f(X);
x[i] = interpolant(0,Y,X,scales = 1,g=F
,func = function(x) 0.5)

if (x[i]>1)
```

```
{
break;
}
i = i + 1
}

epdfPlot(x)

plot(ecdf(x),lwd = 1)

abline(h=1-(1-0.95)/2,col=2,lty=4)
abline(h=(1-0.95)/2,col=2,lty=4)

legend("bottomright",
c(expression(paste('CDF of ',bold(x),': ',m^'*','(',0,')')),
"95% confidence interval"),
lty=c(1,4),
col=c("Black","red") ,lwd = c(1,1))
```

E.8 Figure 5.20 and 5.21

```
f = function(x) ifelse(x>=0,1,0)

n1=4
n2=4
x = numeric(0)
i = 1
while (i < 30000)
{
X = as.matrix(sort(c(runif(n1,-3,0),c(runif(n2,0,3)))))
Y = f(X);
g = function (x) (interpolant(x,Y,X,scales = 1,g=F
,func = function(x) 0.5))
x[i] = (g(0.01)-g(-0.01))/0.01
if (x[i]< -1.5)
{
break
}
i = i + 1
}
```

```
epdfPlot(remove_outliers(x,15),xlim = c(0,10),main = 'PDF of
    ↪ m(0)\'',xlab='m(0)\'')


plot(ecdf(x),xlim = c(-1,18))
abline(h=1-(1-0.95)/2,col=2,lty=4)
abline(h=(1-0.95)/2,col=2,lty=4)

legend("bottomright",
c('CDF of m(0)\'',
"95% confidence interval"),
lty=c(1,4),
col=c("Black","red") ,lwd = c(1,1))
```

Reference List

Becker, W., Worden, K., & Rowson, J. (2013). Bayesian sensitivity analysis of bifurcating nonlinear models. *Mechanical Systems and Signal Processing*, *34*(1), 57–75.

Blackmore, D., Herman, M., & Woodward, J. (1982). Heavy gas dispersion models. *Journal of Hazardous Materials*, *6*(1-2), 107 - 128. doi: 10.1016/0304-3894(82)80036-8

Caiado, C., & Goldstein, M. (2015). Bayesian uncertainty analysis for complex physical systems modelled by computer simulators with applications to tipping points. *Communications in Nonlinear Science and Numerical Simulation*, *26*(13), 123 - 136. doi: 10.1016/j.cnsns.2015.02.006

Chang, E. T. Y., Strong, M., & Clayton, R. H. (2015). Bayesian sensitivity analysis of a cardiac cell model using a Gaussian process emulator. *PLoS ONE*, *10*, e0130252.

Chen, P., Zabaras, N., & Bilionis, I. (2015). Uncertainty propagation using infinite mixture of Gaussian processes and variational bayesian inference. *Journal of Computational Physics*, *284*, 291 - 333. doi: 10.1016/j.jcp.2014.12.028

Conti, S., Gosling, J., Jeremy E. Oakley, & O'Hagan, A. (2009). Gaussian process emulation of dynamic computer codes. *Biometrika*, *96*(3), 663-676. doi: 10.1093/biomet/asp028

Conti, S., & O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, *140*, 640 - 651. doi: 10.1016/j.jspi.2009.08.006

Currin, C., Mitchell, T., Morris, M., & Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, *86(416)*, 953-963. doi: 10.2307/2290511

Gastwirth, J. L., Gel, Y. R., Hui, W., Lyubchich, V., Miao, W., & Wang, X. (2015). *Test of symmetry.* Retrieved on February 18, 2016, from http://finzi.psych.upenn.edu/library/lawstat/html/symmetry.test.html

Gelman, A. (2009). *Bayesian data analysis* (2nd ed.). Chapman & Hall/CRC.

Gramacy, R. B., & Lee, H. K. H. (2008). Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical*

Association, *103*(483), 1119–1130.

Hankin, R. K. S. (2001). The Euler equations for multiphase compressible flow in conservation form. *Journal of Computational Physics*, *172*(2), 808 - 826. doi: 10.1006/jcph.2001.6859

Hankin, R. K. S. (2005). Introducing BACCO , an r bundle for bayesian analysis of computer code output. *Journal of Statistical Software*, *14*(16). doi: 10.18637/jss.v014.i16

Hankin, R. K. S. (2012). Introducing multivator : A multivariate emulator. *Journal of Statistical Software*, *46*(8). doi: 10.18637/jss.v046.i08

Hankin, R. K. S. (2014). The complex multivariate Gaussian distribution. *r-project*. Retrieved from https://journal.r-project.org/archive/2015-1/hankin.pdf

Hankin, R. K. S., & Britter, R. E. (1999). TWODEE: the health and safety laboratory's shallow layer model for heavy gas dispersion part 1. mathematical basis and physical assumptions. *Journal of Hazardous Materials*, *66*(3), 211 - 226. doi: 10.1016/S0304-3894(98)00269-6

Imbens, G. W., & Lemieux, T. (2008). Regression discontinuity designs: A guide to practice. *Journal of Econometrics*, *142*(2), 615 - 635. (The regression discontinuity design: Theory and applications) doi: 10.1016/j.jeconom.2007.05.001

Kennedy, M. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, *87*(1), 1–13. doi: 10.1093/biomet/87.1.1

Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(3), 425–464. doi: 10.1111/1467-9868.00294

Kim, H.-M., Mallick, B. K., & Holmes, C. (2005). Analyzing nonstationary spatial data using piecewise gaussian processes. *Journal of the American Statistical Association*, *100*(470), 653–668.

Lee, D. S., & Lemieux, T. (2010). Regression discontinuity designs in economics. *Journal of Economic Literature*, *48*(2), 281 - 355. doi: 10.3386/w14723

Montagna, S., & Tokdar, S. T. (2016). Computer emulation with nonstationary gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, *4*(1), 26–47. doi: 10.1137/141001512

Moscoe, E., Bor, J., & Brnighausen, T. (2015). Regression discontinuity designs are underutilized in medicine, epidemiology, and public health: a review of current and best practice. *Journal of Clinical Epidemiology*, *68*(2), 132–143. doi: 10.1016/j.jclinepi.2014.06.021

Oakley, J. E. (1999). *Bayesian uncertainty analysis for complex computer codes* (Ph.D Thesis). University of Sheffield, Sheffield, United Kingdom.

Oakley, J. E. (2002, dec). Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, *89*(4), 769–784. doi: 10.1093/biomet/89.4.769

Oakley, J. E. (2009). Decision-theoretic sensitivity analysis for complex computer models. *Technometrics*, *51*(2), 121–129. doi: 10.1198/tech.2009.0014

O'Hagan, A. (2006). Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering and System Safety*, *91*(1011), 1290 - 1300. doi: http://dx.doi.org/10.1016/j.ress.2005.11.025

Porter, J., & Yu, P. (2015, nov). Regression discontinuity designs with unknown discontinuity points: Testing and estimation. *Journal of Econometrics*, *189*(1), 132–147. doi: 10.1016/j.jeconom.2015.06.002

Rasmussen, C., & Williams, C. (2006). Gaussian processes for machine learning. *the MIT Press*.

Real, R., & Vargas, J. M. (1996). The probabilistic basis of jaccard's index of similarity. *Systematic biology*, *45*(3), 380–385.

Stewart, J. (2010). *Calculus: Concepts and contexts* (4th ed.). South Melbourne Australia: Cengage Learning.

Stover, C., & Weisstein, E. W. (2013). *Discontinuity.* Retrieved on November 1, 2015, from http://mathworld.wolfram.com/Discontinuity.html

Thistlethwaite, D. L., & Campbell, D. T. (1960). Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational Psychology*, *51*(6), 309-317. doi: 10.1037/h0044319

Villarreal, F. (2006). Heaviside generalized functions and shock waves for a burger kind equation. *Integral Transforms and Special Functions*, *17*(2), 213-219. doi: 10.1080/10652460500438128

Villarreal, F. (2012). Generalized Heaviside functions in the Colombeau theory context. *Electronic Journal of Differential Equations*, *2012*(87), 1-20. Retrieved from http://ejde.math.txstate.edu/Volumes/2012/87/villareal.pdf

Weisstein, E. W. (2002). *Heaviside step function.* Retrieved on November 1, 2015, from http://mathworld.wolfram.com/HeavisideStepFunction.html

Zhang, B., Konomi, B. A., Sang, H., Karagiannis, G., & Lin, G. (2015). Full scale multi-output Gaussian process emulator with nonseparable auto-covariance functions. *Journal of Computational Physics*, *300*, 623 - 642. doi: 10.1016/j.jcp.2015.08.006