Parameter estimation and uncertainty quantification applied to advection-diffusion problems arising in atmospheric source inversion

by

Juan Gabriel García Osorio

B.Sc(Physics), National University of Colombia, 2013

B.Sc(Mathematics), National University of Colombia, 2015

Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in the

Department of Mathematics

Faculty of Science

SIMON FRASER UNIVERSITY

Spring 2018

Approval

Name:	**Juan Gabriel García Osorio**
Degree:	**Master of Science (Mathematics)**
Title:	**Parameter estimation and uncertainty quantification applied to advection-diffusion problems arising in atmospheric source inversion**
Examining Committee:	**Chair:** Razvan Fetecau Associate Professor
	John Stockie Senior Supervisor Professor
	Paul Tupper Committee Professor
	Ben Adcock Internal Examiner Assitant Professor
Date Defended:	**April 10, 2018**

Abstract

In this thesis we present a method to obtain an efficient algorithm to perform parameter estimation with uncertainty quantification of mathematical models that are complex and computationally expensive. We achieve this with a combination of emulation of the mathematical model using Gaussian processes and Bayesian statistics and inversion for the parameter estimation and uncertainty quantification. In particular we apply these ideas to a source inversion problem in atmospheric dispersion.

We explain the theory and ideas behind each relevant part of the process in the emulation and parameter estimation. The concepts and methodology presented in this work are general and can be applied to a wide range of problems where it is necessary to estimate parameters but the underlying mathematical model is expensive, rendering more classical approaches unfeasible.

To validate the concepts used, we perform a parameter estimation study in a model that is relatively cheap to compute and whose parameter values are known in advance. Finally we perform a parameter estimation with uncertainty quantification of a much more expensive atmospheric dispersion model using real data from a lead-zinc smelter in Trail, British Columbia. The parameter estimation includes approximating high-dimensional integrals with Markov chain Monte Carlo methods and solving the source inversion problem in atmospheric dispersion using the Bayesian framework.

Keywords: Bayesian Statistics, Gaussian Processes, Parameter Estimation, Experimental Design, Model Emulation, Bayesian Inversion, Atmospheric Dispersion, Source Inversion.

Dedication

To the memory of my late grandma Josefina.

Acknowledgements

Like everything in life, this work is the contribution of many people. I apologize if in the next paragraphs I omit to mention someone that was part of this journey to complete my masters, but rest assure that it is an honest mistake. First I would like to express my gratitude to the people that are closest to my life: my mom and my best friends that coincidentally are also my siblings and girlfriend. Luisa, Maria Jose, Juan Lukas and Paola you are the reason why I do what I do, you are the people that make success feel meaningful and worth sharing. Thanks for being there for me and give me your unconditional love, you guys make me a happy person.

I also want to thank John, for his unconditional support during this process, but most importantly for believing in me and giving me the opportunity to come here to this amazing country and amazing university to keep growing as a person and to look for a better future for me and my family. A very special gratitude goes to Bamdad for his infinite patience with me, and the many hours and expertise he put into helping me assemble this thesis. Last but not least a very special thank you to all my friends at PIMS and SFU that made my way through grad school a total blast and all the professors from whom I had the fortune to learn, not only academically but also how to be a better person.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With raising concerns over climate change and anthropogenic impact on the environment, such as airborne emissions of pollutants from industrial sites, the importance of atmospheric dispersion models has grown over the years. Unfortunately, atmospheric dispersion models are imprecise [6, 20]. Although the precision of a model can be improved at the expense of adding more parameters that explain the missing physics in the model, even in this case there is a limit to how accurate a model can be due to uncertainties in measurement of the relevant parameters. The reason for this comes from the fact that sources of uncertainty are diverse and different in nature, following [32] we have

- Model uncertainty: the model does not explain all of the physical processes of interest.
- Data uncertainty: experimental measurements or parameter estimates are inaccurate.
- Stochastic uncertainty: physical processes in the atmosphere are inherently unpredictable, e.g. turbulence.

Despite this difficulty in obtaining accurate predictions there is a growing necessity to develop more accurate and computationally efficient atmospheric dispersion models in practice [18]. The need for computational efficiency derives mainly from the fact that atmospheric models are usually expressed as systems of partial differential equations, whose simulation is computationally expensive. This situation may hinder the model's ability to be used in real situations. One of the goals of this work is to show how to approximate computationally demanding models with cheap surrogates.

The mathematical model for atmospheric dispersion that we are interested is the advection-diffusion equation for the concentration C of particulate matter (particulate zinc is our main focus in this work). The behavior of C as a function of space and time is governed by the following partial differential equation

$$\frac{\partial C(\mathbf{x},t)}{\partial t} + \nabla \cdot (\bar{\mathbf{v}} C(\mathbf{x},t) + \mathbf{D}\nabla C(\mathbf{x},t)) = \sum_{j=1}^{N} q_j \delta(\mathbf{x} - \mathbf{x}_j) \tag{1.1}$$

Here $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$, t is the time variable, $\bar{\mathbf{v}}$ is the wind velocity field, $\mathbf{D}$ is the diffusivity tensor and the right hand side of the equation is the emission rate. In the scenario we are interested in modeling, we have N zinc sources at known locations $\{\mathbf{x}_j\}_{j=1}^N$ with strengths $\{q_j\}_{j=1}^N$ respectively. The values of the wind velocity field $\bar{\mathbf{v}}$ and the diffusivity tensor $\mathbf{D}$ depend upon certain number of parameters. These parameters will be explained in detail in Chapter 4. For the moment it is enough to consider these parameters to be represented by a k-dimensional vector denoted by Θ. By replacing the linear operator $\nabla \cdot (\mathbf{v} + \mathbf{D}\nabla)$ on the left hand side by $L(\Theta)$, to make the dependence on Θ explicit, we can write equation (1.1) more compactly as

$$\frac{\partial C(\mathbf{x}, t)}{\partial t} + L(\Theta)C(\mathbf{x}, t) = \sum_{j=1}^{N} q_j \delta(\mathbf{x} - \mathbf{x}_j). \tag{1.2}$$

In practice, the concentration of particulate matter is not measured directly, but what is measured instead is the deposition in some region after some interval of time. Given the concentration C we can calculate the deposition in a region $R \subset \mathbb{R}^2$ after T units of time as

$$\int_R \int_0^T C(x, y, 0, t) v_{set} dt\, dx\, dy,$$

where v_{set} is the vertical settling velocity of zinc particles. In this work we are mainly concerned in estimating the source strengths given a finite set of deposition measurement at different locations, which is known as source inversion. More precisely, in this work, we want to estimate the source strength vector (or just sources for short), with four sources

$$\mathbf{q} := (q_1, q_2, q_3, q_4)^T,$$

given measurements of the deposition. This problem is ill-posed[8] in the sense of Hadamard. Uncertainty in a model's prediction capabilities gets worse if the problem is ill-posed. For example in an ill-posed model, small uncertainties in the experimental data or parameters of the model translate into significant variations in the output.

Despite the ill-posed nature of the source inversion problem, different approaches have been proposed in the literature. Lin and Chang [23] used a statistical approach to estimate air trajectory and the strength of different sources of volatile organic compounds from anthropogenic origin. Stockie and Lushi [25] used a Gaussian plume approach to calculate zinc deposition in a lead-zinc smelter and used least-squares to perform the source inversion. Skiba [39] solved the adjoint equation for the advection-diffusion equation and used least-squares with Tikhonov regularization to invert the sources. What these approaches have in common is that they provide a point estimate for the strength of the sources of interest but no measure of uncertainty for the estimate.

The Bayesian framework has been applied to solving the source inversion problem, to obtain point estimates and the uncertainties associated with them. Sohn et al. [41] developed an algorithm to obtain estimates and uncertainties for the location and strength of pollutant sources in buildings. They combined Bayes' rule with data obtained from a COMIS simulation. Hosseini and Stockie [9] used a Gaussian plume model coupled with experimental data to estimate and quantify the uncertainty of airborne fugitive emissions. Using experimental data from the transport and dispersion experiment "The Muck Urban Setting Test" and Keats *et al.* [15] obtained, using the Bayesian framework, probability distributions for source strengths and locations.

In this work we apply the Bayesian framework to estimate the source strength vector $\mathbf{q}$ given a set of field measurements denoted by $\mathbf{y}$. Unlike the previous work mentioned above, we will also estimate and quantify the uncertainties in the estimates for the set of model parameters Θ. This will be achieved by obtaining a joint probability density for the parameters and sources. More precisely, we use Bayes' rule to obtain the posterior distribution for parameters and sources. Getting ahead of the explanation of Bayes' rule in Chapter 2, we formally write the posterior distribution as

$$\mathbb{P}_{post}(\Theta, \mathbf{q}|\mathbf{y}) = \frac{\mathbb{P}_{like}(\mathbf{y}|\Theta, \mathbf{q})\mathbb{P}_{prior}(\Theta, \mathbf{q})}{Z(\mathbf{y})},$$

where $\mathbb{P}_{like}$ is called the likelihood distribution, $\mathbb{P}_{prior}$ the prior distribution and $Z(\mathbf{y})$ is a normalization constant such that the integral of the posterior over the whole space is one, that is

$$\int \mathbb{P}_{post}(\Theta, \mathbf{q}|\mathbf{y}) d\Theta d\mathbf{q} = 1.$$

Traditionally, the process of tuning the parameters in atmospheric dispersion models has been done empirically using Pasquill stability classes [38, 42]. Instead, we propose an approach to estimate the parameters using the information contained in the posterior probability distribution. The advantage of this method is that we are taking into account the experimental data available and we let the data determine the most likely value, given the observations. Furthermore we use a method to deal with computationally expensive models. To simulate equation (1.1) we use a finite volume code, whose computational overhead is high because of the grid resolution in three dimensions, therefore using it several times to obtain an approximation for the likelihood $\mathbb{P}_{like}(\mathbf{y}|\Theta, \mathbf{q})$ is not feasible. Instead we run the code for a small number of different combinations of parameters and sources and construct an emulator using Gaussian process regression to extrapolate the results of the simulation to other combinations of parameters and sources. We then use a Markov Chain Monte Carlo Method to sample the posterior distribution. We use the samples to obtain point estimates and uncertainties for the parameters and emission rates.

To conclude this chapter, we will explain how this work is organized: in Chapter 2 we explain the theory and computations behind the Bayesian framework, sensitivity analysis and Gaussian process regression for emulation. In Chapter 3 we introduce more theory and combine it with the topics developed in the previous chapter to show how parameter estimation works on a toy problem. Finally in Chapter 4 we apply everything from Chapters 2 and 3 to an industrial case study and obtain estimates for the parameters and the sources through sampling the posterior distribution $\mathbb{P}_{post}(\Theta, \mathbf{q}|\mathbf{y})$.

Chapter 2

Theoretical and Computational Framework

The framework of Bayesian statistics is the foundation of our approach to estimate parameters and solve inverse problems. Unlike frequentist statistics, in the Bayesian approach, randomness is a measure of uncertainty or lack of information, not a matter of frequency. Consider a statement such as: the probability of having life in the universe is 0.01. In the frequentist perspective, this number can be interpreted as: in all multiverses similar to ours, on average, one universe out of a hundred, shelters life. In the Bayesian perspective the number 0.01 is interpreted as a measure of how certain we are about life in our universe given the current state of knowledge about the outer space. Clearly there is a philosophical difference between these two approaches that has a direct impact in how far reaching is each point of view in terms of theoretical foundations and applications [12].

When we mention uncertainty we are talking about every possible source of randomness or lack of information. That is, the use of the word uncertainty in this work is related to either [16]

- Epistemic: a phenomenon might not be random but the complete lack of understanding of it makes us see it as random.
- Aleatory: Uncertainty inherent to the nature of the phenomenon. For example, this is the kind of randomness physicists believe is happening in quantum mechanics.

In real life the uncertainty associated with a measurement or quantity of interest is usually connected with the uncertainty of other variables involved in the problem under study. The Bayesian framework provides a rigorous framework to study these uncertainties, using whatever information is available for the underlying problem. The cornerstone of this framework in the mathematical sense is known as Bayes' formula. Before we present it, let us introduce some important definitions taken from [7].

2.1 Definitions and Notation

Definition 1. *A probability space is a triple $(\Omega, \mathscr{F}, \mathbb{P})$, where Ω is a set called the sample space and $\mathscr{F}$ is a collection of subsets of Ω that satisfies*

1. $\emptyset, \Omega \in \mathscr{F}$.
2. *If* $A \in \mathscr{F}$ *then* $A^c \in \mathscr{F}$.
3. *If* $A_1, A_2, \ldots \in \mathscr{F}$ *then* $\bigcup_{i\in\mathbb{N}} A_i \in \mathscr{F}$.

A collection of sets that satisfies properties 1 to 3 is called a σ-algebra and its elements are called events.
The map $\mathbb{P} : \mathscr{F} \to [0, 1]$ is called a probability measure and satisfies

1. $\mathbb{P}(\Omega) = 1$.
2. *If* $A_1, A_2, \ldots \in \mathscr{F}$ *are pairwise disjoint, then*

$$\mathbb{P}\left(\bigcup_{i\in\mathbb{N}} A_i\right) = \sum_{i\in\mathbb{N}} \mathbb{P}(A_i).$$

Definition 2. *Given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and two events $A, B \in \mathscr{F}$, with $\mathbb{P}(B) \neq 0$, we define the conditional probability of A given B by*

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

With the definitions above, we are now in a position to state Bayes' fomula as

$$\mathbb{P}_{post}(A|B) = \frac{1}{Z}\mathbb{P}_{like}(B|A)\mathbb{P}_{prior}(A). \tag{2.1}$$

The sets A and B are subsets of the sample space Ω and are elements of the associated $\sigma-$algebra $\mathscr{F}$. The notation $\mathbb{P}_{like}(\cdot|\cdot)$ or $\mathbb{P}_{post}(\cdot|\cdot)$, denotes conditional probability. Let us introduce some terminology: the term $\mathbb{P}_{like}(B|A)$ is called the *likelihood* of B given A. The term $\mathbb{P}_{prior}(A)$ is called the *prior* probability for A. The prior probability expresses how much we believe the event A to happen without assuming anything about B. The reciprocal of Z is a *normalization constant* defined as

$$Z = \int_\Omega \mathbb{P}_{like}(B|A) d\mathbb{P}_{prior}. \tag{2.2}$$

The integral is understood in the Lebesgue sense as the integral with respect to the measure $\mathbb{P}_{prior}$ [19]. The term $\mathbb{P}_{post}(A|B)$ is the *posterior* probability of A given B. The posterior contains the information that we gained by comparing our beliefs (encoded in the prior

probability) with experimental data (encoded in the likelihood).

Now we look at the connection between Bayesian statistics and the field of inverse problems. Inverse problems are often concerned with finding the cause of an effect, whereas a forward problem is concerned with finding the effect of a cause. If we have information about the forward problem, then we can use it to obtain information about the inverse problem. Bayes' rule puts in a mathematical language the connection between the inverse and forward problems. If we consider the cause to be the event A and the effect the event B, then the information about the forward problem is represented by $\mathbb{P}_{like}(B|A)$. The information about the inverse problem is encoded in $\mathbb{P}_{post}(A|B)$. That is why in the Bayesian framework, the posterior probability is the *solution to an inverse problem.*

Often, inverse problems are ill-posed, which means that these problems might not satisfy one or more of the following properties [17]:

- Existence: There exists a solution for the problem.
- Uniqueness: The problem has a unique solution.
- Stability: Small changes in inputs result in small changes in outputs.

Any such lack of well-posedness is a serious issue. For example, if the problem under study has at least one solution but is unstable to small perturbations, how can we assess the accuracy of the solution to the problem? Therefore an statistical or non-deterministic approach is called for. As explained before, the Bayesian framework is useful in this context. Bayes' rule connects the inverse problem of finding the cause of an effect through the posterior with the forward problem of finding the effect of a cause through the likelihood in a way that is possible to quantify the uncertainty about the solution of the problem. Let us clarify with an example of how the Bayesian framework can be used to solve inverse problems.

Consider the problem of finding the launch location of a rock that impacts (and cracks) a window. We can start by considering the following events

$$A = \text{Coordinates of the launching location.}$$
$$B = \text{Coordinates of the impact location on the window.}$$

Here we assume we know B, but A is unknown. We can use Bayes' rule to estimate A through the posterior $\mathbb{P}_{post}(A|B)$. In this case we need to find the connection between A and B via the forward problem, that is, given the launch coordinates find the impact location. This connection is encoded in the likelihood $\mathbb{P}_{like}(B|A)$. In addition we also need to set the prior probability for A.

Let us explain how we could estimate the different probabilities mentioned in the previous paragraph. First, to find the likelihood we need to know how the rock's impact position in the window is related to the launch location. We can use the kinematic equations for parabolic trajectories to get [2]

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2, \tag{2.3}$$

where $\mathbf{r}$ and $\mathbf{r}_0$ are the final and initial position of the rock, $\mathbf{v}_0$ is the initial velocity, and $\mathbf{g}$ is a vector that points to the center of the earth and has a magnitude equal to the acceleration of gravity. The scalar t represents time. In a more physical language, to compute the likelihood it is necessary to estimate $\mathbf{r}$ (where the rock hit the window) assuming we know $\mathbf{r}_0$ (where it was thrown), and the initial velocity of the rock $\mathbf{v}_0$. Once all the other variables are identified the value of t can be computed in a straightforward manner.

Equations in physics are just models of reality and as such are just an approximation to it. For example, equation (2.3) does not consider air resistance or the Coriolis force. To take this into account we add an extra layer to the model by adding a random parameter that accounts for the discrepancy of our model with reality. We propose

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2 + \epsilon, \tag{2.4}$$

where ϵ is a *random vector* distributed as *multivariate Gaussian*. Before we define what a multivariate Gaussian distribution it is necessary to define more terminology and mathematical objects that are going to be used throughout the rest of the text.

Definition 3. *Given a set Ω, for any subset $T \subset \Omega$, we define the $\sigma-$algebra generated by T as the smallest $\sigma-$algebra in Ω that contains T.*

Definition 4. *Let O be the set of all open sets in $\mathbb{R}^n$. The $\sigma-$algebra generated by O is called the Borel $\sigma-$algebra and is denoted by $\mathcal{B}^n$. If $n = 1$ we denote $\mathcal{B}^1 := \mathcal{B}$.*

Definition 5. *Given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, a function $X : \Omega \to \mathbb{R}$ is called a random variable if $X^{-1}(C) \in \mathscr{F}$ for all $C \in \mathcal{B}$.*

Definition 6. *An n-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ in $(\Omega, \mathscr{F}, \mathbb{P})$ is a function $\boldsymbol{X} : \Omega \to \mathbb{R}^n$ such that each component is X_i is a random variable. Note that a single random variable can be considered as a one dimensional random vector.*

Definition 7. *Given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and an n-dimensional random vector $\mathbf{X} : \Omega \to \mathbb{R}^n$, the distribution of $\mathbf{X}$ is the probability measure*

$$\mu : \mathcal{B}^n \to [0, 1],$$

where μ is defined by

$$\mu := \mathbb{P} \circ \boldsymbol{X}^{-1}.$$

Definition 8. *Given a random vector $\boldsymbol{X} : \Omega \to \mathbb{R}^n$ with probability distribution μ, we say that $\mathbf{X}$ is absolutely continuous with respect to the Lebesgue measure if there exists a real valued, integrable function ρ such that for all $C \in \mathcal{B}^n$ we have*

$$\mu(C) = \int_C \rho(x)dx.$$

We say that ρ is the density function for $\mathbf{X}$.

Definition 9. *Given an n dimensional random vector $\mathbf{X}$ such that for any $C \in \mathcal{B}^n$ we have*

$$\mu(C) = \int_C \frac{1}{2\pi det(\Sigma)^{-\frac{1}{2}}} \exp\left[(\boldsymbol{x} - \boldsymbol{x}^*)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{x}^*)\right] d\boldsymbol{x}, \tag{2.5}$$

then we say that $\boldsymbol{X}$ has a multivariate Gaussian distribution (or just Gaussian distribution) with mean $\boldsymbol{x}^* \in \mathbb{R}^n$ and covariance matrix Σ. The matrix Σ is symmetric and positive definite. We shall write

$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{x}^*, \Sigma). \tag{2.6}$$

In this case the components of $\boldsymbol{X}$ are said to be jointly Gaussian.

We now return to equation (2.4) and assume $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Here I represents the 3×3 identity matrix and $\sigma > 0$ parametrizes one's belief in quantifying the accuracy of equation (2.3). By introducing a random variable into the model we cause all variables involved in equation (2.3) to be random variables; that is, we now look at the associated stochastic equation. With this notation we can recast equation (2.1) as

$$\mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r}, \mathbf{v}_0) = \frac{\mathbb{P}_{like}(\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0)\mathbb{P}_{prior}(\mathbf{r}_0)}{Z}, \tag{2.7}$$

where we assumed independence between $\mathbf{r}_0$ and $\mathbf{v}_0$. Since ϵ is Gaussian we can readily obtain [14]

$$\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0 \sim \mathscr{N}(\mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2, \sigma^2 I).$$

This last equation gives an explicit density for the likelihood.

We now turn our attention to the prior. Suppose that we suspect the rock was thrown from the bedroom of a neighbor. One way to model this suspicion is to assume a prior distribution on $\mathbf{r}_0$ as

$$\mathbf{r}_0 \sim \mathscr{N}(\mathbf{w}, \lambda^2 I),$$

where $\mathbf{w}$ is the coordinate vector of the center of the neighbor's bedroom and λ represents one's belief the launch location is at the point $\mathbf{w}$. We note that this is only one way to model prior knowledge and other forms of the prior are also possible. Finally the normalization

constant can be found as

$$\begin{aligned} Z &= \int_{\mathbb{R}^3} \mathbb{P}_{like}(\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0)\mathbb{P}_{prior}(\mathbf{r}_0)d\mathbf{r}_0 \\ &= \frac{1}{(2\pi)^6(\sigma\lambda)^3}\int_{\mathbb{R}^3} \exp\left[-\frac{1}{2\sigma^2\lambda^2}\left(\|\mathbf{r} - (\mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2)\|^2 + \|\mathbf{r}_0 - \mathbf{w}\|^2\right)\right] d\mathbf{r}_0. \end{aligned}$$

Having the likelihood, prior, and normalization constant allows us to compute the posterior using Bayes' rule. With these probabilities calculated we can obtain several different estimates for the value $\mathbf{r}_0$. Common choices of pointwise estimates include

$$\mathbf{r}_{MAP} = \underset{\mathbf{r}_0}{\operatorname{argmax}}\, \mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r}, \mathbf{v}_0) \qquad \text{(Maximum a posteriori)}, \tag{2.8}$$

$$\mathbf{r}_{CM} = \int_{\mathbb{R}^3} \mathbf{r}_0 \mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r}, \mathbf{v}_0)d\mathbf{r}_0 \qquad \text{(Conditional mean)}, \tag{2.9}$$

$$\mathbf{r}_{ML} = \underset{\mathbf{r}_0}{\operatorname{argmax}}\, \mathbb{P}_{like}(\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0) \qquad \text{(Maximum likelihood)}. \tag{2.10}$$

Each of these estimates has its own strengths and weaknesses. If the posterior is bimodal, then the conditional mean might point at a value with very low probability, whereas the maximum a posteriori estimate might be more reliable. If the posterior has no critical points then the mean might be used as a point estimate. We can also assess how confident we are about the point estimate. For example, if $\mathbf{r}^*$ is our point estimate we can calculate a number $\alpha > 0$ such that

$$\int_{B(\mathbf{r}^*,\alpha)} \mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r}, \mathbf{v}_0)d\mathbf{r}_0 = 0.95, \tag{2.11}$$

where $B(\mathbf{r}^*, \alpha)$ is the ball centered at $\mathbf{r}^*$ with radius α. This value of α can be thought of as the Bayesian version of the frequentist's 95% confidence interval. Another way to estimate uncertainty is by calculating the covariance matrix of $\mathbf{r}$ around $\mathbf{r}_0$ as

$$\int_{\mathbb{R}^3} (\mathbf{r}_0 - \mathbf{r}^*) \otimes (\mathbf{r}_0 - \mathbf{r}^*) d\mathbb{P}_{post}.$$

The diagonal of this matrix contains the variance of each coordinate of $\mathbf{r}^*$.

Note that the posterior is a probability density and does not necessarily have a closed form, which can make it difficult to calculate the uncertainties we mentioned above. Hence we need a way of extracting information from $\mathbb{P}_{post}$. One approach is to generate independent samples from $\mathbb{P}_{post}$ and do a Monte Carlo integration to obtain the different uncertainty estimates. How to sample from a probability density and do a Monte Carlo integration will be explained in Chapter 3. For the moment, we assume it is possible to evaluate any of the point estimates and the uncertainty measures. With this, we can use a point estimate

from equation (2.8) to obtain a method to infer the launch location of the rock and with equation (2.11) we can estimate how confident we are about that estimate.

Practical problems are often substantially more challenging than in the above example. Often times we have to deal with further issues such as

1. Uncertainties in experimental measurements.
2. Lack of sufficient information about the physics of the problem and experimental data.
3. Computational complexity of physical models that are too expensive to evaluate.
4. Parameters that might belong to high dimensional spaces so the associated probability density is hard to sample from.
5. Evaluating any of the possible point estimates for the quantity of interest might be computationally difficult.

In the problem that we outlined in Chapter 1, we have to deal with all of the above mentioned issues. In this chapter we are going to discuss our approach for dealling with issues 3, 4 and 5 above. We omit 1 and 2, since these are intrinsic to the physics of the problem and the methodology used to obtain the experimental data, and so, they are outside of our control.

2.2 Dealing with the Computational Complexity of the Physical Model

Models of physical processes can be represented in different ways. Following O'Hagan [28], we represent the mathematical model of the physical process as a function $M(\cdot)$ so that $y = M(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}$. Mathematical models are approximations to complex physical processes. Often times these mathematical models are expensive to compute. It is of great advantage if the complexity of the model can be reduced. One way to do this is by performing a sensitivity analysis on the parameters the model depends on. Roughly, we choose a combination of different values of the parameters and then we assess the importance of each parameter in the output. This means that we need to run the model $M(\cdot)$ for a large set of different combination of its parameters. Since realistic mathematical models are typically expensive, this implies that the use of classical methods such as correlation ratios, FAST method, Method of Sobol', etc. are not feasible (see [37] for details).

Here the concept of emulator as defined in [28] comes into play. We approximate the function $M(\cdot)$, which is expensive to evaluate, with a function $\widehat{M}(\cdot)$ that is cheap to evaluate. To construct such an approximation, we associate a probability distribution with each possible value of $M(\mathbf{x})$ and for example take $\widehat{M}(\mathbf{x})$ to be the mean of this distribution. We will refer to $\widehat{M}(\cdot)$ as an emulator. Following [28] we expect the emulator to satisfy the conditions in the following definition

Definition 10. *An emulator $\widehat{M}(\cdot)$ of a function $M(\cdot)$, is a map that:*

- *At points $\{\boldsymbol{x}\}_{k=1}^{N}$ where we know the output of the mathematical model (i.e. we know $M(\boldsymbol{x}_k)$ for $k = 1, 2, \ldots, N$) the emulator should satisfy $\widehat{M}(\boldsymbol{x}_k) = M(\boldsymbol{x}_k)$.*
- *For points $\{\boldsymbol{x}_k^*\}_{k=1}^{T}$ where we don't know the output $M(\boldsymbol{x}_k^*)$, the emulator should return an estimate $\widehat{M}(\boldsymbol{x}_k^*)$ based on the distribution for $M(\boldsymbol{x}_k^*)$. That estimate should reflect the uncertainty associated with the interpolation/extrapolation done at that point.*

From now on in this work we will to refer to the mathematical model or the computationally expensive function to calculate as $M(\cdot)$, and the emulator that approximates this function by $\widehat{M}(\cdot)$.

A popular method to construct an emulator with the desired extrapolation/interpolation properties is known as a Gaussian process regression.

2.2.1 Gaussian Processes

The conditions on the emulator $\widehat{M}(\cdot)$ imply that we need specify a probability distribution for each point $\mathbf{x}$ in the domain of the model $M(\cdot)$. This means that we need to work with a set of random variables with high cardinality. When dealing with several random variables there is one probability density that is computationally tractable and easy to work with: the multivariate Gaussian distribution (see Definition 9). The computational tractability of integrals involving the multivariate Gaussian distribution can be used as a justification to define a Gaussian process.

Definition 11. *A Gaussian process (GP) is a collection of random variables $\{g(\boldsymbol{x})\}_{\boldsymbol{x}\in A}$ for some set A, possibly uncountable, such that any finite subset $\{g(\boldsymbol{x}_k)\}_{k=1}^{N} \subset \{g(x)\}_{\boldsymbol{x}\in A}$ for $\{\boldsymbol{x}_k\}_{k=1}^{N} \subset A$ is jointly Gaussian [33].*

A GP is specified by a mean function and a covariance operator or covariance kernel. Following Rasmussen [33] we define

$$m(\mathbf{x}) := \mathbb{E}(g(\mathbf{x})), \qquad \text{(Mean)}$$
$$k(\mathbf{x}, \mathbf{x}') := \mathbb{E}((g(\mathbf{x}) - m(\mathbf{x}))(g(\mathbf{x}') - m(\mathbf{x}'))) \qquad \text{(Kernel)}.$$

If $\{g(\mathbf{x})\}_{\mathbf{x}\in A}$ is a GP with mean $m(\cdot)$ and covariance $k(\cdot, \cdot)$ we will write

$$g(\mathbf{x}) \sim \mathbf{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

To understand why the notion of a GP is useful for us, recall that our goal is to create an emulator $\widehat{M}(\cdot)$ that approximates a function $M(\cdot)$. For a fixed $\mathbf{x} \in A$, a realization of the random variable $g(\mathbf{x})$ represents a possible value of $M(\mathbf{x})$. The mean function at that point $\mathbf{x}$, that is, $m(\mathbf{x})$ represents the best prediction of the true value of $M(\mathbf{x})$, so we may

set $\widehat{M}(\mathbf{x}) = m(\mathbf{x})$. Later we will show that one way to measure the uncertainty associated with that prediction is given by the quantity $k(\mathbf{x}, \mathbf{x})$.

We will use GPs to fit functions in high dimensional Euclidean spaces, so that we may think of the index set A of Definition 11 as a subset of $\mathbb{R}^n$ for some $n \geq 1$.

The reason why Gaussian processes are useful in practice is that they are completely characterized by the mean $m(\cdot)$ and choosing a covariance kernel $k(\cdot, \cdot)$[21]. For example a common covariance or kernel is the squared exponential (SE) function given by

$$k(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{2}\|\mathbf{x}-\mathbf{x}'\|_2^2}. \tag{2.12}$$

We choose to use the name "squared exponential" instead of Gaussian to avoid confusion with the probability distribution. This covariance function tells us that if $\mathbf{x}$ and $\mathbf{x}'$ are close in the Euclidean metric then they are highly correlated, whereas far away points have a correlation that decays exponentially. How to choose the covariance function depends on the kind of regularity we want for the realizations of the GP. We will discuss this topic in more detail later in this chapter. For reference purposes, we list some of the most common kernels used in practice [33], setting $r = \|x - x'\|_2$:

- Squared-Exponential: $k(r; \theta) = \exp\left[-\frac{1}{2}(\frac{r}{\theta})^2\right]$
- Exponential: $k(r; \theta) = \exp\left[-\frac{r}{\theta}\right]$
- Matérn $\frac{3}{2}$: $k(r; \theta) = (1 + \frac{\sqrt{3}r}{\theta})exp\left[-\frac{\sqrt{3}r}{\theta}\right]$.
- Matérn $\frac{5}{2}$: $k(r; \theta) = (1 + \frac{\sqrt{5}r}{\theta} + \frac{5}{3}(\frac{r}{\theta})^2)\exp\left[-\frac{\sqrt{5}r}{\theta}\right]$.
- Power-Exponential: $k(r; \theta, p) = \exp\left[-(\frac{r}{\theta})^p\right]$.

Gaussian Processes as Distributions Over Function Spaces

Alternatively GPs can be viewed as measures on function spaces, and so we now discuss them in this context following the approach of [21]. Relevant function spaces (e.g. L^p spaces, Sobolev spaces, etc.) are normed vector spaces with a topology inherited from the metric induced by the norm. So the function spaces of interest here are topological vector spaces (TVS).

Let $\mathscr{T}$ be a TVS and let $\mathscr{T}^*$ be its topological dual. We will denote the action of an element $h \in \mathscr{T}^*$ over an element $z \in \mathscr{T}$ by $\langle h, z\rangle$. Moreover we define a random variable taking values in $\mathscr{T}$ as a map

$$X : (\Omega, \mathscr{F}, P) \longrightarrow \mathscr{T},$$

that is measurable with respect to the σ-algebra generated by the open sets of $\mathscr{T}$. This σ-algebra is known as the Borel σ-algebra for $\mathscr{T}$. The triple $(\Omega, \mathscr{F}, P)$ is a probability space as in Definition 1. We use the shorthand notation $X \in \mathscr{T}$ whenever the random variable X takes values in $\mathscr{T}$. For example if $\mathscr{T} = L^2(\mathbb{R})$, then $X \in L^2(\mathbb{R})$ means that X is a measurable map from the probability space $(\Omega, \mathscr{F}, P)$ into $L^2(\mathbb{R})$.

We say that a random variable $X \in \mathscr{T}$ is called Gaussian if $\langle h, X\rangle$ is a Gaussian random variable on the real line for all $h \in \mathscr{T}^*$. An element $a \in \mathscr{T}$ is the expectation of $X \in \mathscr{T}$ if

$$\mathbb{E}(f, X) = \langle f, x\rangle, \qquad \text{for all } f \in \mathscr{T}^*.$$

Also a linear and positive definite operator $K : \mathscr{T}^* \longrightarrow \mathscr{T}$ is called the covariance operator (the covariance matrix in the finite dimensional case) if

$$\operatorname{cov}(\langle f_1, X\rangle, \langle f_2, X\rangle) = \langle f_1, K f_2\rangle,$$

for all $f_1, f_2 \in \mathscr{T}^*$. Then we say that X is distributed as $\mathcal{N}(a, K)$. It is worth mentioning that given a covariance operator L and an element $b \in \mathscr{T}$ the distribution $\mathcal{N}(b, L)$ does not always exist[22]. But if it does exist, the Gaussian measure $\mathcal{N}(a, K)$ is completely identified by a and K.

As an example consider $\mathscr{T} = \mathbb{C}(T)$, the set of continuous real valued functions on T, where T is compact subset of $\mathbb{R}^n$. This is the space of real valued continuous functions on T, which is a Banach space with the norm [4]

$$\|h\| = \max_{x \in T} |h(x)|.$$

The dual space of $\mathscr{T}$ is given by $\mathscr{T}^* = \mathbb{M}(T)$ which is the set of signed measures defined on the Borel σ-algebra of T. In this case the duality pairing is given by

$$\langle \mu, g\rangle = \int_T g d\mu.$$

A GP, $\{g(t)\}_{t \in T}$ (see Definition 11) with mean function $m(t)$ and covariance kernel $k(t, t')$, can be thought of as a Gaussian measure $\mathcal{N}(m, K)$ where [21]

$$\begin{aligned} \mathbb{E}(g) &= m \in \mathbb{C}(T), \\ (K\nu)(t) &= \int_T k(t, t') d\nu(t'), \qquad \text{for } \nu \in \mathbb{M}(T). \end{aligned}$$

The above example illustrates the connection between GPs and distributions over function spaces, and more precisely how is connected to Gaussian measures on function spaces. Next we will explain how to use GPs in practice.

Assume we have data $\{(\mathbf{x}_i, y_i)\}_{i=1}^m \subset \mathbb{R}^n \times \mathbb{R}$ from an expensive function $M(\cdot)$, where $M(\mathbf{x}_i) = y_i$. The set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ is called the *training set*. For simplicity we do not presume any trend in the *training outputs* $\{y_i\}_{i=1}^m$. Given the training set we would like to infer possible values of $M(\cdot)$ on another set of points $\{\mathbf{x}_j^*\}_{j=1}^k$. This set of points is known as the *test set*. For this purpose we construct an emulator $\widehat{M}(\cdot)$ (see introduction to section 2.1) by considering the GP denoted by $\{f(\mathbf{x})\}_{\mathbf{x}\in \mathrm{dom}(M)}$ where $\mathrm{dom}(M)$ is the domain of $M(\cdot)$. By Definition 11, the random vectors

$$\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) & \dots & f(\mathbf{x}_m) \end{bmatrix}^T,$$
$$\mathbf{f}^* = \begin{bmatrix} f(\mathbf{x}_1^*) & \dots & f(\mathbf{x}_l^*) \end{bmatrix}^T,$$

are jointly Gaussian with

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix}\right), \tag{2.13}$$

where the zero mean reflects the assumption of no trend in the training output $\{y_i\}_{i=1}^m$. The element of the submatrices in the covariance matrix are given by

$$(K(X,X))_{ij} = \mathrm{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)),$$
$$(K(X,X^*))_{ij} = \mathrm{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j^*)),$$

and so on. By the requirements of Defintion 10, the realization of the random vector $\mathbf{f}$ is known and is equal to $[y_1, \dots, y_m]^T$. Given this vector, we want to infer the vector $\mathbf{f}_*$. This can be achieved by obtaining the distribution of $\mathbf{f}_*|\mathbf{f}$. By well known properties of the multivariate Gaussian distribution we obtain [22]

$$\mathbf{f}^*|\mathbf{f} \sim \mathcal{N}(\langle \mathbf{f} \rangle, \Sigma), \tag{2.14}$$

where

$$\langle \mathbf{f} \rangle = K(X^*,X)K(X,X)^{-1}\mathbf{f},$$
$$\Sigma = K(X^*,X^*) - K(X^*,X)K(X,X)^{-1}K(X,X^*).$$

Note that if in the above equations we only consider one test point $\mathbf{x}^*$, and we take the limit as $\mathbf{x}^*$ approaches the training input $\mathbf{x}_j$, the matrix $K(\mathbf{x}^*, X)$ reduces to a vector and converges to $K(\mathbf{x}_j, X)$. In this case, it is not hard to see that the mean would be given by

$$K(\mathbf{x}_j,X)K(X,X)^{-1}\mathbf{f} = y_j, \tag{2.15}$$

and the covariance matrix would reduce to an scalar that tends to zero as $\mathbf{x}^* \to \mathbf{x}$. The interpretation is that at a training input $\mathbf{x}_j$ the prediction is exactly equal to the corresponding training output y_j. For a point $\mathbf{x}^*$ that is not part of the training set, with 95% confidence we have

$$M(\mathbf{x}^*) \in [\langle f \rangle(\mathbf{x}^*) - 2\sigma, \langle f \rangle(\mathbf{x}^*) + 2\sigma], \tag{2.16}$$

where

$$\begin{aligned} \langle f \rangle(\mathbf{x}^*) &= K(\mathbf{x}^*, X)K(X, X)^{-1}\mathbf{f} && \text{(Mean at point } \mathbf{x}^*) \\ \sigma^2 &= K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, X)K(X, X)^{-1}K(X, \mathbf{x}^*) && \text{(Variance at point } \mathbf{x}^*). \end{aligned}$$

Equations (2.15) and (2.16) show that if we define

$$\widehat{M}(\mathbf{x}^*) := \langle f \rangle(\mathbf{x}^*), \tag{2.17}$$

then $\widehat{M}(\cdot)$ satisfies the conditions for an emulator laid out in Definition 10.

In Figure 2.1 is shown an example that summarizes the discussion above. We consider the problem of emulating the model $M(x) = \cos(2\pi x)$ having five training points. The 95% confidence region shows that in the training input $\mathbf{x}_j$ the variance is zero and $\widehat{M}(\mathbf{x}_j) = y_j$, as predicted by equation (2.15).

The covariance kernel is the quantity that defines the mean and covariance for the Gaussian distribution obtained when we look at finitely many random variables in a Gaussian Process. Therefore choosing it is a crucial step in the fitting process. We next discuss the properties of kernels and how to choose them depending on the data and the smoothness properties we are looking for in the emulation process.

Covariance Kernels

The covariance kernel cannot be any arbitrary function $k(\mathbf{x}, \mathbf{x}^*)$. To see why, consider the matrix in equation (2.13) given by

$$C := \begin{bmatrix} K(X, X) & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix}.$$

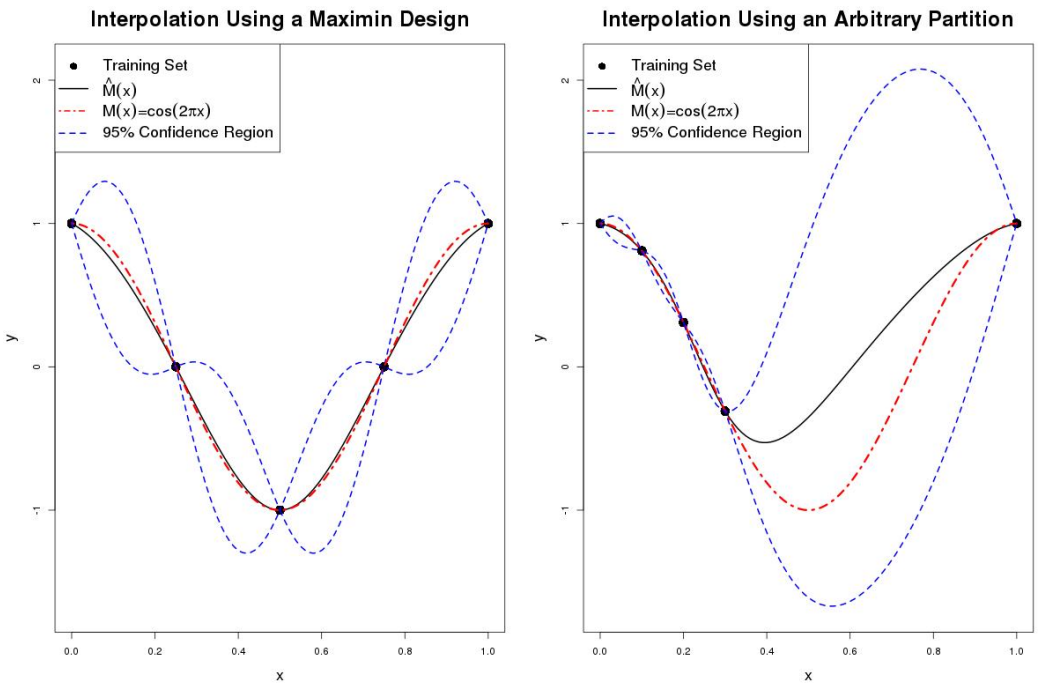

Figure 2.1: Comparison between the approximation quality of the emulator $\widehat{M}$ (solid line) for for the model $M(x) := \cos(2\pi x)$ (dashed-dotted line) in the interval $[0, 1]$ for two different partitions. On the left, the emulation is performed on the partition $\{0, 0.25, 0.5, 0.75, 1\}$. On the right on the partition $\{0, 0.1, 0.2, 0.3, 1\}$. The dashed line represents the 95% confidence region. The black points are the training set.

This is the covariance matrix of a multivariate Gaussian distribution and is obtained by evaluating the covariance kernel at different points. The matrix C must be symmetric and positive definite for any set of training and test inputs. This implies that the covariance kernel has to be symmetric, or in other words, for all $\mathbf{x}$ and $\mathbf{x}'$ in the domain of $k(\cdot, \cdot)$ we must have

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x}).$$

We also need that for any set of inputs $\{\mathbf{x}_i\}_{i=1}^n$ the Gram matrix defined by $K_{jk} := k(\mathbf{x}_j, \mathbf{x}_k)$, must be positive definite. If k is just a function of $\mathbf{x} - \mathbf{x}'$, which is common for many kernels of practical interest, then $k(\cdot, \cdot)$ is said to be *stationary*.

To understand the role of the covariance kernel in the continuity and differentiability of the mean function, let us define some concepts first.

Definition 12. *Let $\boldsymbol{y}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ be a sequence of points in $\mathbb{R}^n$, such that*

$$\|\boldsymbol{x}_n - \boldsymbol{y}\|_2 \to 0 \qquad \text{as } n \to \infty.$$

Then the collection of real valued random variables $\{f(\boldsymbol{x})\}$ *defined in a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *are said to be continuous at* y *in the mean-squared sense if [33]*

$$\mathbb{E}(|f(\boldsymbol{x}_n) - f(\boldsymbol{y})|^2) \to 0 \qquad as\ n \to \infty,$$

where

$$\mathbb{E}(f(x)) := \int_\Omega f(x) d\mathbb{P}.$$

We also have a definition for differentiability.

Definition 13. *The mean square derivative of the collection* $\{f(\boldsymbol{x})\}$ *in the* i*-th direction at a point* $\boldsymbol{y}$ *is*

$$\frac{\partial f(\boldsymbol{y})}{\partial x_i} = \lim_{h \to 0} \mathbb{E}\left(\left| \frac{f(\boldsymbol{y} + h\boldsymbol{e}_i) - f(\boldsymbol{y})}{h} \right|^2 \right),$$

whenever the limit exists. Here $\boldsymbol{e}_i$ *is the* i*-th canonical vector of the standard basis in* $\mathbb{R}^n$*. The mean square* n*-th derivative is given by*

$$\frac{\partial^n f(\boldsymbol{y})}{\partial x_i^n} = \lim_{h \to 0} \mathbb{E}\left(\left| \frac{\frac{\partial^{n-1} f(\boldsymbol{y} + h\boldsymbol{e}_i)}{\partial x_i^{n-1}} - \frac{\partial^{n-1} f(\boldsymbol{y})}{\partial x_i^{n-1}}}{h} \right|^2 \right),$$

whenever the limit exists.

In the definitions above if we take $\{f(\mathbf{x})\}$ to be a Gaussian process with stationary covariance kernel k then, it can be shown that the process is continuous in the mean at a point $\mathbf{y}$ if and only if k is continuous at $(\mathbf{y}, \mathbf{y})$. Also the kernel function for the n-th derivative is given by [1]

$$\frac{\partial^{2n} k(\mathbf{x}, \mathbf{x}')}{\partial^2 x_1 \dots \partial^2 x'_m}.$$

Therefore the continuity and differentiability properties of the mean function in a Gaussian process depends exclusively in the continuity and differentiability properties of the covariance kernel.

Another important aspect of covariance kernels is that they are defined in terms of parameters. The way we choose the values of these parameters in practice is based on the data we are analyzing. To see how this works, let us return to the problem of of approximating $M(\cdot)$ by $\widehat{M}(\cdot)$ using Gaussian processes. Let $k(x, x'; \theta)$ be the covariance kernel for the GP that depends on the parameter θ, where θ could be a scalar, vector, etc. In this case to predict the output $\mathbf{y}^* = \{M(\mathbf{x}_1^*), \dots M(\mathbf{x}_m^*)\}$ given the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, we can try different approaches. One of the most common is maximum likelihood optimization (MLE),

where we pick a parameter $\hat{\theta}$ such that

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \mathbb{P}(\mathbf{y}^* | \{(\mathbf{x}_i, y_i)\}_{i=1}^m, \theta).$$

By Definition 11 we know that the conditional probability for $\mathbf{y}^*$ has to be distributed as a multivariate Gaussian distribution. More precisely

$$p(\mathbf{y}^* | \{(\mathbf{x}_i, y_i)\}_{i=1}^m, \theta) = \frac{1}{(2\pi)^{\frac{m}{2}} \det(K_{\mathbf{y}^*}(\theta))^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{y}^{*T} K_{\mathbf{y}^*}(\theta)^{-1} \mathbf{y}^*)\right], \tag{2.18}$$

where $K_{\mathbf{y}^*}(\theta)$ is the matrix $K(X, X)$ in equation (2.13). To find the value of $\hat{\theta}$ we have to maximize (2.18) with respect to θ. This goal is unchanged if we take the logarithm of both sides and minimize the following function instead[1]

$$L(\theta) = -\log(p(\mathbf{y}^* | \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m, \theta)) = \frac{1}{2}\mathbf{y}^{*T} K_{\mathbf{y}^*}(\theta)^{-1} \mathbf{y}^* + \frac{1}{2}\log |K_{\mathbf{y}^*}(\theta)|. \tag{2.19}$$

A minimizer of $L(\theta)$ gives a possible value for $\hat{\theta}$ that explains the best the data $\mathbf{y}^*$ given the training set $\{\mathbf{x}_i, y_i\}_{i=1}^m$. Another common way to tune the parameters, is using K-fold cross validation, but will not use this approach here (the interested reader is referred to [27] for details).

So far we have not discussed how to choose the training inputs $\{\mathbf{x}_i\}_{i=1}^m$. Clearly this choice has a profound impact on the accuracy of the emulator. To see this, let us assume that the function $M(\cdot)$ is supported in $[0, 1]$ and we have computational resources to calculate the output of only five training points. If we pick the points $\{0, 0.1, 0.2, 0.3, 1\}$ the interpolation error of the emulator $\widehat{M}(\cdot)$ for points between 0.3 and 1, will be large, compared to the error associated with the partition $\{0, 0.25, 0.5, 0.75, 1\}$ as shown in Figure 2.1.

Ideally we would like to pick as many training points as possible to improve the fit, but picking too many points to create the training set can result in a very high computational cost. On the other hand, if we pick just few points to create the training set, then it is possible to end up with unreliable predictions. Thus we need a systematic way to choose the number and distribution of the training points. One strategy is to fill as much of the space as possible given a fixed number (possibly small) of training points. This can be accomplished through space-filling designs which we discuss next.

[1]The reason for taking the logarithm is because most software packages for optimization search for the minimum, not the maximum.

2.2.2 Design of Experiments

We assume there is a fixed computational budget. In this case, we need to decide how to choose the training inputs $\{\mathbf{x}_j\}_{j=1}^m$ to obtain reliable predictions of the emulator for points different than the training points. As shown in Figure 2.1, the quality of the emulation depends heavily on the distribution of the training inputs. Intuitively we want to spread the training inputs as much as possible in the parameter space while covering as much space as possible. Distributions of points that achieve this are called *space filling designs*.

Given an set $T \subset \mathbb{R}^n$, there are several ways to create space filling designs. In this work we focus on maximin designs [13]. We note that there are other ways to obtain space filling designs and we refer to the reader to [30]. Consider a metric space (T, d) (e.g. $T \subset \mathbb{R}^n$, compact and d the Euclidean distance) and a subset S of T, with finite (fixed) cardinality, say $|S| = n$. A maximin distance design S^o is a collection of points of T such that

$$\max_{(S\subset T,\ |S|=n)} \min_{(s,s'\in S, s\neq s')} d(s,s') = \min_{s,s'\in S^o, s\neq s'} d(s,s') = d^o.$$

That is, we are looking for a set S^o of cardinality n that maximizes the minimum distance among its elements. As an example consider $T = [0,1]^3$, the unit cube in $\mathbb{R}^3$ and $n = 8$. In this case the design that maximizes the minimum distance among its elements is given by choosing the 8 vertices of the cube or as shown in Figure 2.1 (right), if $T = [0,1]$ and $n = 5$, the maximin design is given by a uniform partition of the set T.

The problem of finding the optimal maximin design is difficult to solve in general. In practice we use computational tools to find a design that is close to optimal. Different algorithms can be used for the optimization of the design, such as genetic algorithms, simulated annealing, particle swarm, etc. A survey on the subject can be found in [43]. In Chapter 4 we will see how the particle swarm algorithm can be used to create a maximin design for a five dimensional parameter space.

To conclude this section, we note that there is a conection between maximin designs and Gaussian processes. Consider a GP $\{f(x)\}_{x\in T}$, fix $S = \{s_1, \ldots, s_n\} \subset T$, and consider the random vector

$$\mathbf{f} = [f(s_1), \ldots, f(s_n)],$$

where $\mathbf{f}$ is assumed to be jointly Gaussian. Let K_s be the correlation matrix for the probability distribution of $\mathbf{f}$. Then it can be shown that the minimax design minimizes the quantity

$$D(S) = -\det(K_s),$$

where the matrix K_s is the same as the covariance matrix in equation (2.13). A survey of the theory behind maximin distance designs can be found in [13].

2.2.3 Sensitivity Analysis

Having a space filling design for the training input permits us to create an emulator $\widehat{M}(\cdot)$ that closely approximates $M(\cdot)$ over its whole domain. By "closely" we mean within a tolerable uncertainty in the output of the emulator for all points in the domain (see Figure 2.1). If we have a reliable fitting, then we can confidently assess what parameters in the model $M(\cdot)$ are relevant and which ones are not. This ultimately allows to approximate the model with a simpler one. For example, if our model is given by

$$M(x_1, x_2, x_3) = x_1 + x_2 + 10^{-8}x_3, \qquad (x_1, x_2, x_3) \in T = [0, 1]^3,$$

then clearly the variable x_3 is not as relevant as x_1 or x_2. We need to formalize in what sense x_3 is irrelevant. One way to achieve this is by doing a sensitivity analysis. In summary, the goal of a sensitivity analysis is to assess how the output of a function $M(\cdot)$ depends on variations of its arguments. There is a great number of methods to perform a sensitivity analysis, such as adjoint methods, local methods, and variance based methods, to name a few. The primary difference between each of these methods is how they measure the importance of each variable. For example, in local methods the sensitivity at a point in a given direction is the slope of the function, whereas in variance based methods what matters is the magnitude of the area under the curve when fixing all parameters but one. For a survey of techniques in sensitivity analysis, the reader is referred to [37].

In this work we use variance-based Monte Carlo methods (VBMCM) as described in [40]. The idea of VBMCMs is to use the variance produced by the inputs of a function as an indicator of their importance. More precisely we will use the method of Sobol', which we outline next

The functions of interest in this work have compact support. This implies that without loss of generality we may assume that the domain of these functions is the n-dimensional unit cube Ω^n. Let us consider a generic square integrable function

$$\varphi : \Omega^n \to \mathbb{R},$$

and start by decomposing φ as

$$\varphi(x_1, \ldots, x_n) = \varphi_0 + \sum_{k=1}^{n} \varphi_k(x_k) + \sum_{1 \le k < l \le n} \varphi_{kl}(x_k, x_l) + \ldots + \varphi_{1,2,\ldots,n}(x_1, \ldots, x_n).$$

This decomposition is not unique, but it can be shown that if each term $\varphi_{i_1,\ldots,i_j}$ in the expansion satisfies

$$\int_{[0,1]} \varphi_{i_1,\ldots,i_j} dx_{i_k} = 0 \qquad \text{if } i_k \in \{i_1, \ldots, i_j\}, \tag{2.20}$$

then the decomposition is unique and all terms in the expansion are orthogonal in $L^2(\Omega^k)$. To demonstrate the orthogonality property, we may consider the functions $g = \varphi_{i_1,\dots,i_j}$ and $h = \varphi_{\ell_1,\dots,\ell_k}$ with arbitrary indices $(i_1,\dots,i_j) \neq (\ell_i,\dots,\ell_k)$. Without loss of generality we may assume $i_1 \neq \ell_1$. In this case we have

$$\langle g, h\rangle = \int_{[0,1]} \dots \int_{[0,1]} \underbrace{\left(\int_{[0,1]} \varphi_{i_1,\dots,i_j} dx_{i_1}\right)}_{=0 \text{ by } (2.20)} \left(\int_{[0,1]} \varphi_{l_1,\dots,l_k} dx_{l_1}\right) dx_{\sim i_1,l_1} = 0,$$

where we used Fubini's theorem to split the integrals [19]. The symbols to the right of $\sim$ represent the variables omitted in the integration. Another consequence of (2.20) is

$$\int_{\Omega^n} \varphi dx = \varphi_0.$$

This allows us to find the other functions in the decomposition recursively, given φ_0. For example, for $i \in \{1,\dots,n\}$ we have

$$\varphi_i(x_i) = -\varphi_0 + \int_{[0,1]^{n-1}} \varphi(x) dx_{\sim i}.$$

Having $\varphi_i(x_i)$ we can then proceed to find $\varphi_{ij}(x_i, x_j)$ using

$$\varphi_{ij}(x_i, x_j) = -\varphi_0 - \varphi_i(x_i) - \varphi_j(x_j) + \int_{\Omega^{n-2}} \varphi(x) dx_{\sim ij}.$$

By knowing all of the functions in the decomposition of φ we are able to assess how each variable affects the output of φ in the following way. The total variance D of φ is defined as

$$D = \int_{\Omega^n} \varphi^2(x) dx - \varphi_0^2,$$

and similarly we can compute the partial variances as

$$D_{i_1,\dots,i_s} = \int_{[0,1]^{n-1}} \varphi^2_{i_1,\dots,i_s} dx_{i_1} \dots dx_{i_s}.$$

With these variances we define the s-th order Sobol' index

$$S_{i_1,\dots,i_s} = \frac{D_{i_1,\dots,i_s}}{D},$$

which is a measure of the contribution of the variables $x_{i_1},\dots,x_{i_s}$ to the total variance D. If we want to know the separate effect of each variable $x_1,\dots,x_n$ in the total variance D, we look at the first order Sobol' indices $S_1,\dots,S_n$ given by

$$S_i = \frac{D_i}{D}, \qquad \text{for } i = 1,\dots,n.$$

Finally if we want to assess the full effect of a variable on the total variance D, we calculate a quantity known as the total effect index. For example if we want to calculate the total effect index for the variable x_i we would do so by calculating

$$S_i + S_{i1} + S_{i2} + \ldots + S_{i12} + S_{i13} + \ldots + S_{12\ldots,i,\ldots,n}.$$

Note that to calculate each Sobol' index, it is necessary to perform high dimensional integrals. Therefore integration using quadratures is not feasible. It is necessary to resort to other numerical integration techniques. A common tool to perform high dimensional integrals is known as Monte Carlo integration. We will not go into details of Monte Carlo integration in this chapter, but rather postpone them for Chapter 3. What is important at this time is that to apply Monte Carlo integration, it is necessary to evaluate the integrand a large number of times at different points in its domain. If the integrand is the expensive model $M(\cdot)$, then the computational cost of estimating the Sobol' indices is prohibitive. If instead we calculate the Sobol' indices of the emulator $\widehat{M}(\cdot)$, we can use them as an approximation for the Sobol' indices of $M(\cdot)$. In this way we can estimate what arguments of the model are relevant and what arguments are not. This will allow us to reduce the complexity of the model.

Chapter 3

Toy Problem: How Theory Works in Practice

In the previous chapter we reviewed some of the theoretical and computational tools needed to solve a Bayesian inverse problem. In this chapter we will to present a toy problem to illustrate how the theory can be applied in practice. We begin by considering the forward problem, given by the following partial differential equation (PDE)

$$\begin{cases} \Delta u = e^{-b\|\mathbf{x}\|_2}, & \text{for } x \in \Omega = [0,1] \times [0,1] \subset \mathbb{R}^2, \\ u = 0, & \text{for } x \in \partial\Omega, \end{cases} \tag{3.1}$$

where b is some real positive parameter. For us, the function u represents the mathematical approximation of a quantity $\tilde{u}$ that has a physical realization. For example we may think of $\tilde{u}$ as the actual difference in electric potential in Ω relative to a reference point and u as the mathematical approximation to it. Since mathematical models of the physical world are a simplification of reality, it is convenient to make a clear distinction between physics ($\tilde{u}$) and mathematics (u).

In Section 2.1, we explained how to build an emulator $\hat{M}(\cdot)$ that approximates the output y of a computationally expensive function $M(\cdot)$ at a point in its domain. In this chapter, the function $M(\cdot)$ takes as input a point $(\mathbf{x}, b) \in \Omega \times (0, \infty)$. The output is the value of the solution u at that point, that is $u(\mathbf{x}; b) = M(\mathbf{x}, b)$. Now we proceed to explain the associated inverse problem and how we will to construct $\hat{M}(\cdot)$.

Assume that we have ten experimental measurements of $\tilde{u}$ at the points $P := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{10}\} \subset \Omega$. That is, we know the vector of measurements $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$. We want to estimate the value of b that explains the experimental data $\mathbf{y}$ the best. This is our inverse problem. A simple approach to estimate b would be to solve equation (3.1) for a large number of values b in the interval $(0, L]$ where L is chosen in a manner that there exists a $b^* \in (0, L]$ such that the vector $(u(\mathbf{x}_1; b^*), \ldots, u(\mathbf{x}_{10}; b^*))$ has 'small' discrepancy with the experimental data $\mathbf{y}$. This approach is not feasible if solving the forward model is computationally expensive.

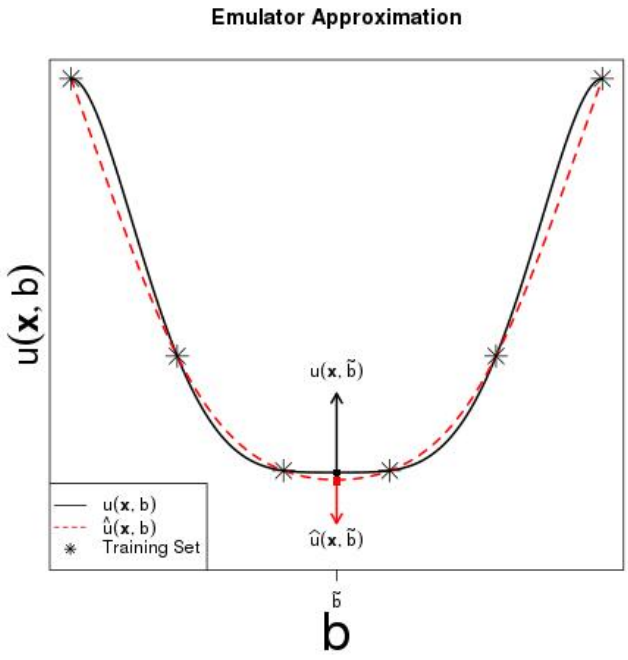

Figure 3.1: Approximation of a model $u(\mathbf{x};\cdot)$ by the mean of a Gaussian process trained with six different outputs from the model. The mean of the Gaussian process at a point $\tilde{b}$ is taken as the value $\widehat{u}(\mathbf{x};\tilde{b})$ of the emulator.

Let us assume that solving equation (3.1) is computationally expensive and repeating the calculation for a big range of different values of b is not feasible. One way to get around that is by constructing an emulator $\widehat{u}(\cdot)$ that approximates $u(\cdot)$ and is cheap to compute. The way we will construct $\widehat{u}(\cdot)$ is as follows: for a fixed $\mathbf{x} \in \mathbb{R}^2$ we solve equation (3.1) for n different values of b. We pick the value of n in a way that the computational cost of computing (3.1) n times, does not exceed our computational and time budget. Then use the data $\{b_j, u(\mathbf{x}, b_j)\}_{j=1}^{n}$ as a training set to create a Gaussian process, as explained in Section 2.1.1. Finally for any value $\tilde{b}$ we use the mean of the Gaussian process at that point as $\widehat{u}(\mathbf{x}, \tilde{b})$. An sketch from the result for approximating an arbitrary model $u(\mathbf{x};\cdot)$ with an emulator $\widehat{u}(\mathbf{x};\cdot)$ is shown in Figure 3.1.

For clarity in the exposition, the table below summarizes the notation we will use throughout the rest of the chapter.

Symbol	Meaning
$\tilde{u}(\mathbf{x}; b)$	Value of the physical variable at the point $\mathbf{x}$ with parameter b.
$u(\mathbf{x}; b)$	Numerical solution of equation (3.1) at $\mathbf{x}$ with parameter b.
$\hat{u}(\mathbf{x}; b)$	Value of the interpolation of the emulator $\hat{M}(\cdot)$ at the point $\mathbf{x}$ with parameter b.
$P := \{\mathbf{x}_1, \ldots, \mathbf{x}_{10}\}$	Points where the experimental measurements were taken .
$\mathbf{y} := (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$	Values of the experimental measurements for the variable $\tilde{u}$.

Table 3.1: Summary of symbols used in Chapter 3.

Let us return to our original goal: to estimate the value of b that explains the experimental data $\mathbf{y}$ as best as possible. To create the experimental data $\mathbf{y}$ we assume that the true value of b is 0.925. Then, for this value of b, we solve equation (3.1) using a finite difference five point stencil approximation for the Laplacian. Next we pick ten points at random in Ω and save the value of the numerical solution u at those locations (see Figure 3.2). Finally we add noise from a normal distribution with mean zero and standard deviation 0.01 to each of the ten values. The resulting numbers are what we use as the experimental data $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$. Note that the noise added to the data obtained from the numerical solution of equation (3.1) plays the role of possible errors in the experimental measurements plus inaccuracies of the model to describe the true behavior in the physical variable $\tilde{u}$.

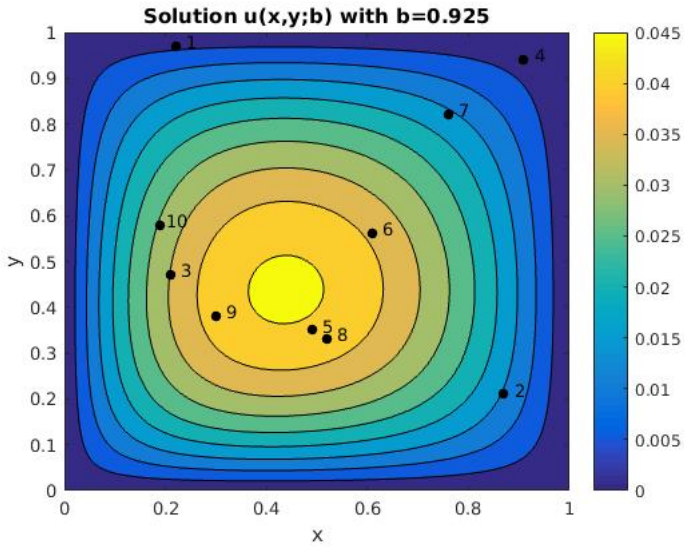

Figure 3.2: Numerical solution of the system (3.1) using a five point stencil finite difference approximation for the Laplacian. The mesh size used in x and y was 0.01 and $b = 0.925$. The black dots in the plot represent the points used to generate the experimental data $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$.

3.1 Computing the Posterior

With the experimental data $\mathbf{y}$ created, we now proceed to obtain a point estimate value of b that produced that data. To that end we first compute the posterior distribution. To calculate the posterior we use Bayes' rule to get

$$\mathbb{P}_{post}(b|\mathbf{y}) = \frac{\mathbb{P}_{like}(\mathbf{y}|b)\mathbb{P}_{prior}(b)}{Z(\mathbf{y})}. \tag{3.2}$$

Note that finding the posterior enables us to obtain any of point estimate from equation (2.8) and the uncertainty associated with that estimate. To compute $\mathbb{P}_{post}(b|\mathbf{y})$ we need to choose a prior distribution and the likelihood for b.

3.1.1 Choosing the Prior

For the sake of the example assume that the parameter b cannot be greater than 2. In this case one way to choose a prior distribution for b that does not assume any other knowledge than $b \in (0, 2]$, is the *uniform distribution*. In this case we have

$$\mathbb{P}_{prior}(b) = \frac{1}{2}\mathbf{1}_{(0,2]}(b), \qquad \text{for all } b \in \mathbb{R}, \tag{3.3}$$

where $\mathbf{1}_{(0,2]}$ is the indicator function of the set $(0, 2]$. The indicator function for a Borel measurable set C is defined as

$$\mathbf{1}_C(y) = \begin{cases} 1 & \text{if } y \in C \\ 0 & \text{if } y \in C^c. \end{cases}$$

3.1.2 Finding the Likelihood

To calculate the likelihood, first we need to know how the set of possible measurements $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$ is related to b when b is allowed to vary. Since we don't know the experimental values of the physical variable $\tilde{u}$ for different values of b, it is necessary to approximate the relation between $\tilde{u}$ and b by the relation between u and b. To obtain such a relation we need to solve equation (3.1). By solving this equation explicitly we can find a functional relation between u and b for each one of the ten locations depicted in Figure 3.2. It is possible to solve analytically equation (3.1), however the relation between u and b is given by an infinite series. Indeed equation (3.1) is Poisson's equation with homogeneous boundary conditions. This equation can be solved using an eigenfunction expansion [24]. The eigenfunctions of the Laplacian in the unit square are given by

$$\phi_{mn} = \sin(n\pi x)\sin(m\pi y), \qquad \text{for } m, n \in \mathbb{N},$$

with eigenvalues

$$\lambda_{mn} = (n\pi)^2 + (m\pi)^2.$$

The eigenfunction expansion for u in equation (3.1) is

$$u = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} a_{mn}\phi_{mn}$$

where

$$a_{mn}\lambda_{nm} = -\frac{\int_\Omega e^{-b\|x\|^2}\phi_{mn}d\mathbf{x}}{\int_\Omega \phi_{mn}^2 d\mathbf{x}} = -\frac{\langle e^{-b\|x\|^2}, \phi_{mn}\rangle}{\|\phi_{mn}\|^2_{L^2(\Omega)}}. \tag{3.4}$$

The symbol $\langle \cdot, \cdot \rangle$ represents the standard inner product in $L^2(\Omega)$.

Having a functional relation given by an infinite series is often not very useful. For example in equation (3.4) the integral in the numerator does not have a closed form. Hence we need a different approach to gain insight into the relation between $\mathbf{y}$ and b. The approach we will use is the same as the one that allowed us to obtain Figure 3.1. First we solve equation (3.1) for n different values of b. For the sake of the example assume $n = 10$. Then for each $\mathbf{x}$ in $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_{10}\}$ we use the set $\{b_j, u(\mathbf{x}_k; b_j)\}_{j=1}^{10}$ to train a Gaussian process for each $k = 1, 2, \ldots, 10$. Finally for any $\tilde{b} \in (0, 2]$ we use the mean of the Gaussian process at that point as the value $\widehat{u}(\mathbf{x}_k; \tilde{b})$. By proceeding in this manner we obtain a cheap method to approximate the behavior of $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$ when we allow b to vary.

The next step is to choose the values of b for which the PDE (3.1) is solved in a way that the uncertainty associated with the emulator is as small as possible. We shall denote the points we choose as $\{b_1, \ldots, b_{10}\}$. To choose the points we use a maximin design as explained in Section 2.2.2. In this case it is straightforward to check that a maximin design is the set of equidistant points

$$\{b_1 = 0.2, b_2 = 0.4, \ldots, b_{10} = 2\}.$$

By solving equation (3.1) for these values of b and for each $\mathbf{x}$ in P, we know the values in the set $\{u(b_j, \mathbf{x}_k)\}_{j,k=1}^{10}$. We use this set to train ten Gaussian Processes. With these processes we define the functions

$$G_k : (0, 2] \to \mathbb{R} \qquad \text{for } k = 1, 2, \ldots 10,$$

such that for each k and b, the value of the mean of the k-th GP is given by $G_k(b)$. That is, $G_k(\cdot)$ is the emulator for $u(\mathbf{x}_k, \cdot)$. More precisely

$$G_k(b) = \hat{u}(\mathbf{x}_k; b).$$

The functions $G_k(\cdot)$ are cheap to evaluate and are a good approximation of $\tilde{u}(\mathbf{x}_k, \cdot)$. Now it is possible to approximate the value of b that explains $\mathbf{y} = (\tilde{u}(\mathbf{x}_1, b), \ldots, \tilde{u}(\mathbf{x}_{10}, b))$ by trying a large number of different values of b and then compare with the experimental data, to see what choice of b gives the smallest discrepancy. To this end, we calculate the values of $G_k(\cdot)$, for $k = 1, \ldots, 10$ in the set

$$\{0.01, 0.02, \ldots, 1.99, 2\}.$$

In Figure 3.3 are plotted the emulator at these points, the true value of b, the experimental measurement $\tilde{u}(\mathbf{x}_k; b)$ and the training data $\{u(\mathbf{x}_k; b_j)\}_{j=1}^{10}$ for each of the ten sites.

We are now ready to make the mathematical connection between $\tilde{u}, u$ and $\hat{u}$. Recall that u is the mathematical approximation of the physical variable $\tilde{u}$ and $\hat{u}$ is an emulator for u. Hence if $\hat{u}$ approximates u well, we would expect that $\hat{u}$ approximates $\tilde{u}$. For any point $\mathbf{x}_k \in P$ we do not know exactly how $\hat{u}(\mathbf{x}_k, \cdot) = G_k(\cdot)$ differs from $\tilde{u}(\mathbf{x}_k, \cdot)$. If we define $y_k(b) := \tilde{u}(\mathbf{x}_k, b)$, then a possible relation that connects these quantities is given by the following Gaussian additive model [14]

$$y_k(b) = G_k(b) + \epsilon_k, \qquad \text{with } \epsilon_k \sim \mathcal{N}(0, \lambda^2), \tag{3.5}$$

where λ is a positive number that models how much we believe the emulator prediction differs from $\tilde{u}$. We chose the value $\lambda = 5.4 \times 10^{-3}$ to get a signal to noise ratio of 1:10. By

defining the vector $\mathbf{G}(b) = (\hat{u}(\mathbf{x}_1; b), \ldots, \hat{u}(\mathbf{x}_{10}; b))$ and the definition of $\mathbf{y}$ (see table 3.1), equation (3.5) can we written more compactly as

$$\mathbf{y} = \mathbf{G}(b) + \epsilon, \qquad \text{with } \epsilon \sim \mathcal{N}(0, \lambda^2 I_{10\times 10}). \tag{3.6}$$

Since the random vector ϵ has a Gaussian distribution, we can use equation (3.6) to conclude

$$\mathbf{y}|b \sim \mathcal{N}(\mathbf{G}(b), \lambda^2 I_{10\times 10}),$$

that is

$$\mathbb{P}_{like}(\mathbf{y}|b) \propto e^{-\frac{1}{2\lambda^2}\|\mathbf{G}(b)-\mathbf{y}\|_2^2}, \tag{3.7}$$

where the proportionality constant normalizes the distribution on the right hand side to one.

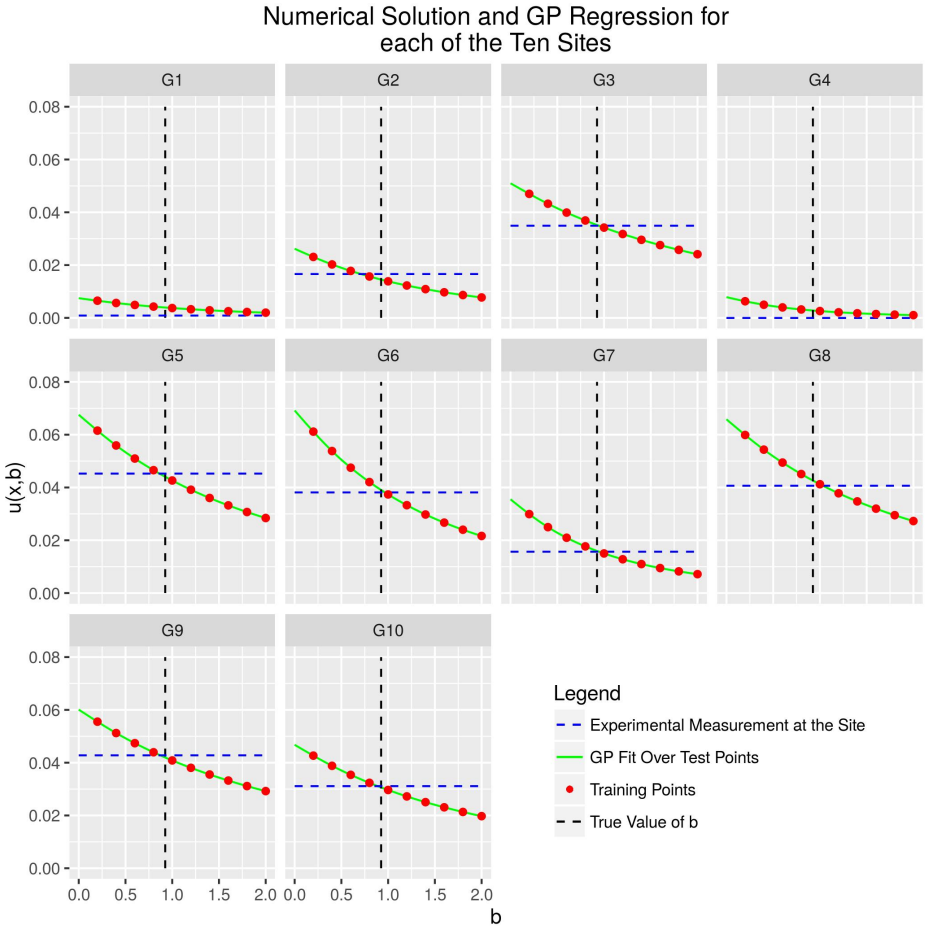

Figure 3.3: Training points, GP regression, true value of b and experimental measures for each one of the ten sites labeled from 1 to 10 in Figure 3.2

Now that we have explicit expressions for the prior and likelihood distributions, we can compute the posterior probability for b. Since the denominator in Bayes' rule (3.2) is independent of b, we can use equations (3.3) and (3.7) to write

$$\mathbb{P}_{post}(b|\mathbf{y}) \propto \mathbb{P}_{like}(\mathbf{y}|b)\mathbb{P}_{prior}(b) \propto \mathbf{1}_{(0,2]}(b)e^{-\frac{1}{2\lambda^2}\|\mathbf{G}(b)-\mathbf{y}\|_2^2}. \tag{3.8}$$

An interpretation of this result is that before taking experimental measurements we only knew that $b \in (0, 2]$. After weighting this prior belief with the data $\mathbf{y}$, our current state of knowledge about the parameter b is encoded in the posterior distribution. Figure 3.4 shows this updated distribution.

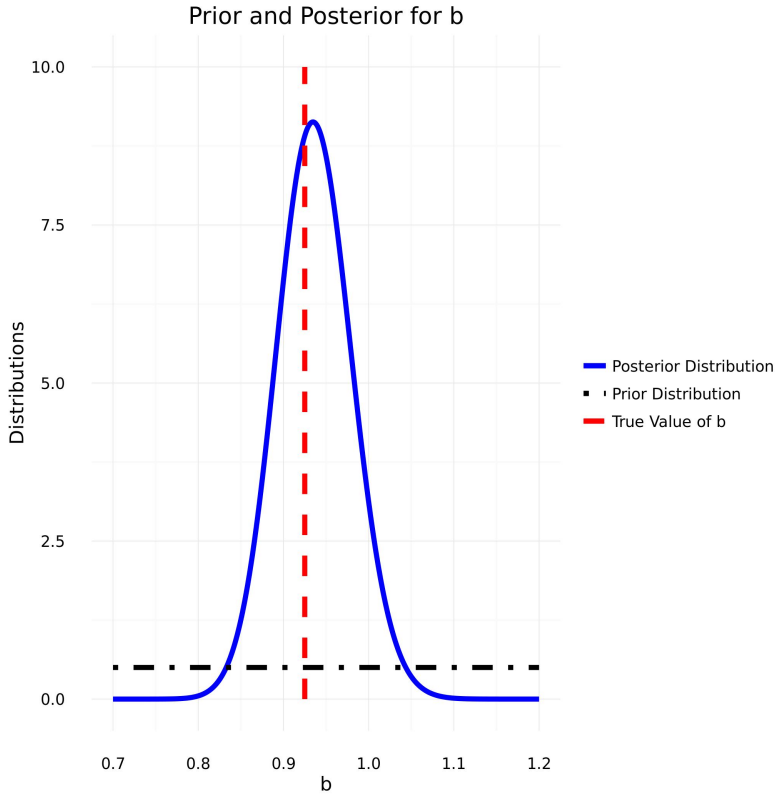

Figure 3.4: Plots of the prior distribution, posterior distribution and true value of the parameter b.

It is not always possible to visualize a probability density so it is necessary to sample from it in order to obtain statistics about the parameters of interest. A family of methods for this purpose is known as Markov Chain Monte Carlo (MCMC). In this work we focus on a particular algorithm known as Metropolis-Hastings (MH). We now proceed to explain how MH works in practice using the posterior for b in equation (3.8) as an example.

Consider the posterior density $\mathbb{P}_{post}(b|\mathbf{y})$. The idea is to construct a Markov chain that

wanders around the support of the posterior in a way that the chain spends more time in regions with high probability. One way to achive that is as follows: if we are at a point q_1 and we want to move to a point q_2 we will accept that move with probability one if $\mathbb{P}_{post}(q_1|\mathbf{y}) \leq \mathbb{P}_{post}(q_2|\mathbf{y})$ and with probability $\frac{\mathbb{P}_{post}(q_2|\mathbf{y})}{\mathbb{P}_{post}(q_1|\mathbf{y})}$ otherwise. We choose in what direction to move, randomly, using some probability distribution that is easy to sample from. For simplicity, in this and the next Chapter we choose the uniform distribution to decide in what direction to move. The pseudocode for the MH algorithm as described above is given below [14]

Algorithm 1 Metropolis-Hastings Algorithm

pick a point q_1 in the support of the distribution
for j=2:N **do**
    Draw $u \sim U([0, \alpha])$
    $q_j \leftarrow q_{j-1} + u$
    $\beta \leftarrow \min(1, \frac{\mathbb{P}_{post}(q_j|D)}{\mathbb{P}_{post}(q_{j-1}|\mathbf{y})})$
    Draw $w \sim U([0, 1])$
    if $w < \beta$ **then**
        $q_{j-1} = q_j$ (Accept the move)
    else
        $q_{j-1} = q_{j-1}$ (Reject the move)
    end if
end for

The rule of thumb for choosing the parameter α in the scheme above is that the proportion of times we accept a move is about 0.25 [34]. It can be shown that the sequence $q_1, q_2, \ldots, q_N$ is a realization of a Markov chain that in the limit as $N \to \infty$ is distributed according to the distribution $\mathbb{P}_{post}(b|\mathbf{y})$. This convergence result works under mild conditions over the distribution that is being sampled. For more details about the theory behind MCMC methods we refer the reader to [34]. Since we do not have the computational power to let $N \to \infty$ we let the chain run for a large number of steps until it converges. Then, we throw away the *burn-in* portion of the chain and compute statistics using the remaining samples. The burn-in portion of the chain consists of samples obtained before the chain is close to converging. A common choice is to discard the first $\frac{N}{2}$ samples.

Using Algorithm 1, we sample from the posterior distribution $\mathbb{P}_{post}(b|\mathbf{y})$ and take the values $\alpha = 0.23$ and $N = 10000$. The burn-in period is set to be the first 5000 samples. A histogram of the last 5000 is shown in Figure 3.5.

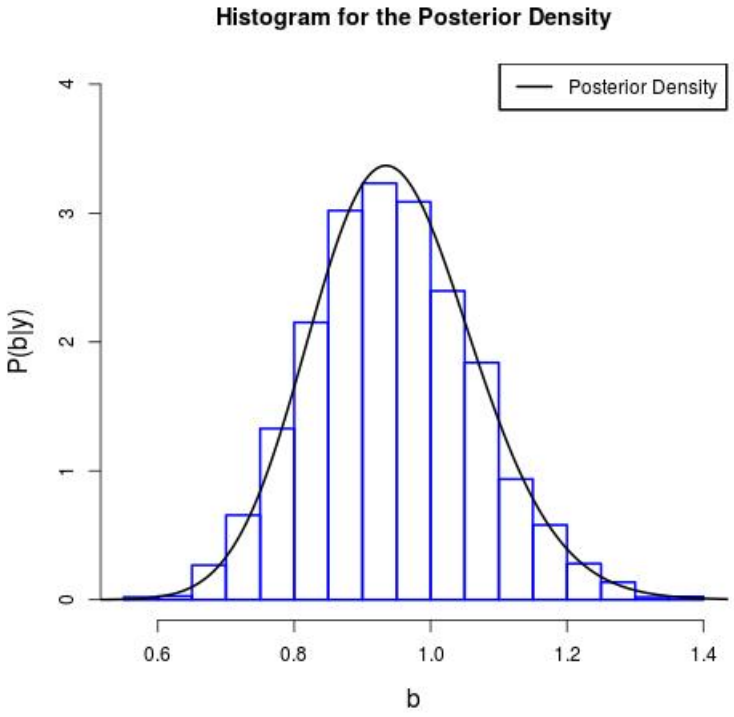

Figure 3.5: Histogram obtained for the posterior distribution (3.2) from 5000 samples from MH algorithm with step size $\alpha = 0.23$. The solid line is the graph for the posterior $\mathbb{P}_{post}(b|\mathbf{y})$.

With the samples obtained we readily obtain useful statistics for b. For example, we can estimate the conditional mean using [34]

$$b_{cm} = \int_{(0,2]} b\mathbb{P}_{post}(b|\mathbf{y})db \approx \frac{1}{5000}\sum_{j=1}^{5000} b_j = 0.9247042, \tag{3.9}$$

where the summands b_j are the samples obtained after the burn-in period of 5000 samples. We can also estimate the variance of the samples as

$$\int_{(0,2]} (b - b_{cm})^2\mathbb{P}_{post}(b|\mathbf{y})db \approx \frac{1}{5000}\sum_{j=1}^{5000} (b_j - b_{cm})^2 = 0.01427.$$

With these values we can compute a 95% confidence interval for b. In this case the interval is given by

$$[0.9247042 - 2\sqrt{0.01427}, 0.92470422 + 2\sqrt{0.01427}] = [0.68579, 1.163618].$$

Let us make a short digression about the idea behind Monte Carlo integration. Consider the generic problem of evaluating the n-dimensional integral

$$\int_{\mathbb{R}^n} h(x)\rho(x)dx, \tag{3.10}$$

where ρ is the Lebesgue density of some probability measure $\mathbb{P}$. This means that calculating (3.10) is equivalent to calculating the expected value of h, that is

$$\mathbb{E}[h] = \int_{\mathbb{R}^n} h(x)\rho(x)dx.$$

If $X_1, \ldots, X_n$ are independent random variables with density ρ, then by the strong law of large numbers, the sequence of random variables

$$h_n = \frac{1}{n}\sum_{k=1}^{n} h(X_k),$$

converges to $\mathbb{E}[h]$[7]. Furthermore if $\mathbb{E}[h^2] < \infty$ we can assess the speed of convergence and the quality of the approximation h_n for $\mathbb{E}[h]$. By the central limit theorem the sequence of random variables h_n

$$\frac{h_n - \mathbb{E}[h]}{\sqrt{\sigma_n}} \to \mathcal{N}(0,1),$$

where

$$\sigma_n = \frac{1}{n}\sum_{k=1}^{n}(h(X_k) - h_n)^2.$$

This means that the uncertainty in the approximation h_n for $\mathbb{E}[h]$ goes to 0 as $\mathcal{O}(\frac{1}{\sqrt{n}})$. Note that the convergence rate is independent of the dimension of the problem. That is the reason why Monte Carlo integration is used in high dimensional problems, where quadrature methods are prohibitively expensive to implement. In Chapter 4 we apply this method to calculate integrals of real valued functions supported in a seven dimensional space.

The estimate for b in equation (3.9) depends on the choice of the prior. At this point it is unclear how choosing a different prior would give a different estimate for b. To close this chapter we discuss the role that the prior has in inference in the Bayesian Framework.

3.2 Importance of the Prior

Once again consider problem of estimating the value of the parameter b, whose real value is, as before, 0.925. This time we assume the parameter b can be any real number (not just $0 < b \leq 2$ as before) and the prior distribution for b to be

$$b \sim \mathcal{N}(b^*, \sigma_b^2),$$

where b^* and σ_b are parameters to be set later. With this new prior the formula for the posterior is

$$\mathbb{P}_{post}(b|\mathbf{y}) \propto \underbrace{\exp\left(-\frac{\|\mathbf{y}-\mathbf{G}(b)\|_2^2}{2\sigma^2}\right)}_{\text{Likelihood}} \underbrace{\exp\left(-\frac{(b-b^*)^2}{2\sigma_b^2}\right)}_{\text{Prior}}.$$

To illustrate the role that the prior has in the inference of the value of the parameter given the data $\mathbf{y}$, suppose that

$$b \sim \mathcal{N}(4, 2.5).$$

This prior assumes that, with 95% confidence, the value of b is in the interval $[1.8, 8.2]$. Clearly, there is a mismatch between the true value of b and the range of values that the prior distribution assigns high probability. Let us evaluate how the posterior distribution for b evolves as we consider more and more experimental data from the measurements of $\tilde{u}$. Figure 3.6 shows how the posterior evolves when we calculate the likelihood with more and more data. The first frame shows the result when only the measurement $\tilde{u}(\mathbf{x}_1; b)$ is taken into account, the second frame when the measurements $\tilde{u}(\mathbf{x}_1; b), \tilde{u}(\mathbf{x}_2; b)$ are taken into account. In each new frame we proceed by adding one more measurement to calculate the likelihood.

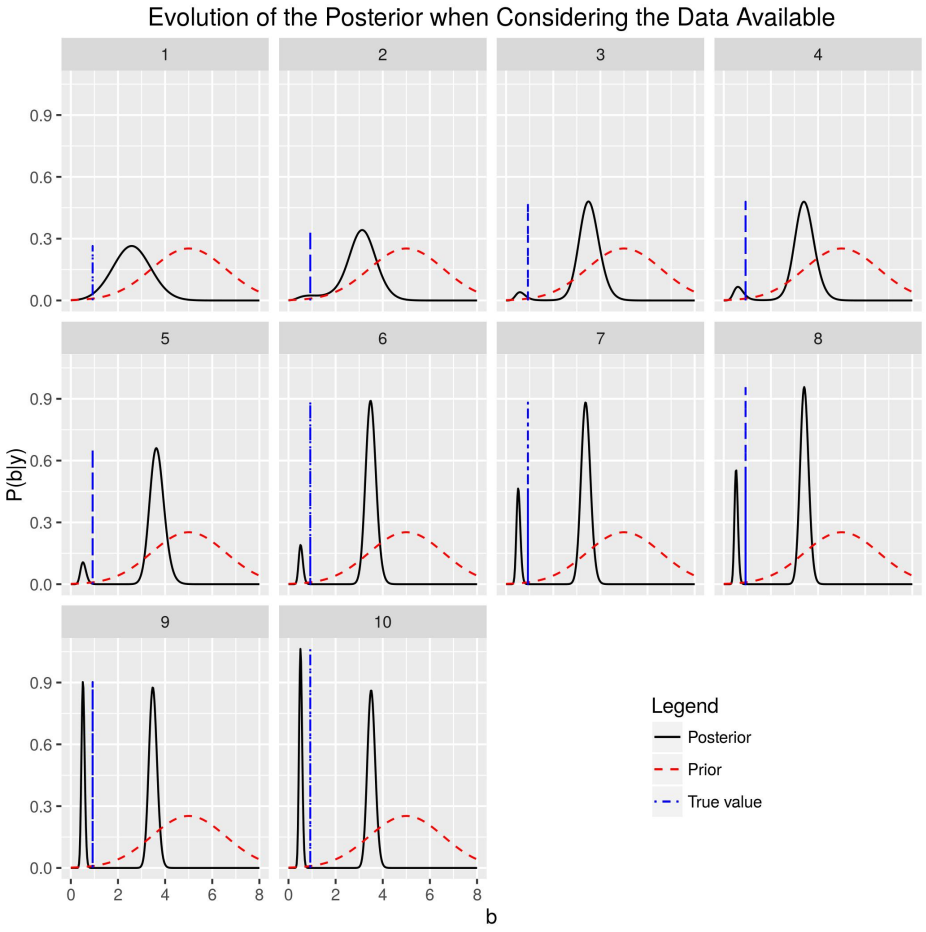

Figure 3.6: Evolution of the posterior distribution when more experimental data is taken into account

The sequence of plots in Figure 3.6 shows that the experimental data creates a new mode in the posterior distribution that is close to the true value of b. In the end of the sequence where we consider all 10 experimental measurements, the mode that is close to the true value of b is bigger than the mode originated by the prior at the point $b = 4$. The explanation for this behavior is that the prior has a high value near $b = 4$, but it is close to zero for values around $b = 0.925$. Then, when the experimental data is used, the likelihood distribution has a higher value for points close to $b = 0.925$ than points close to $b = 4$. The more data are used, the higher the value of the likelihood around $b = 0.925$ and the closer to zero away from it. However since the prior distribution gives negligible probability to values close to the true value of b, when all data are used the product $\mathbb{P}_{prior}(\mathbf{y}|b)\mathbb{P}(b)$ will be non-negligible only in regions close to $b = 4$ or $b = 0.925$.

The above example is a warning example. If we know how to choose the prior distribution in a way that is meaningful to the problem, reliable inference can be done even with a small

amount of data. On the contrary if the prior distribution is not realistic, inference may not be reliable even with a large amount of data.

Chapter 4

Industrial Case Study

In this chapter we study the dispersion of zinc from a lead-zinc smelter in Trail, British Columbia, Canada, operated by Teck Resources Ltd. We consider four zinc sources. Our ultimate goal is to estimate the contribution of each source to the total amount of zinc that is being released by the smelter. We have access to measurements of zinc depositions (in terms of total mass of zinc that reaches ground-level over a monthly period within a dust-fall jar) at nine different locations as well as wind field velocity data at a location near the smelter. We also know prior engineering estimates of the zinc emission provided by the company. An aerial photograph of the region of interest with the location of the sources and the measurement devices is shown in Figure 4.1. The sources are represented by the letters Q_1 to Q_4. The deposition measurement sites are marked as R_1 to R_9.

In order to estimate the emission rates, we shall use a similar approach to that of Section 3.1. In Chapter 3 we assumed we had access to a mathematical model that approximates the physics of processes of interest. Here, it is necessary to develop such a model for the pollutant dispersion in the atmosphere from scratch.

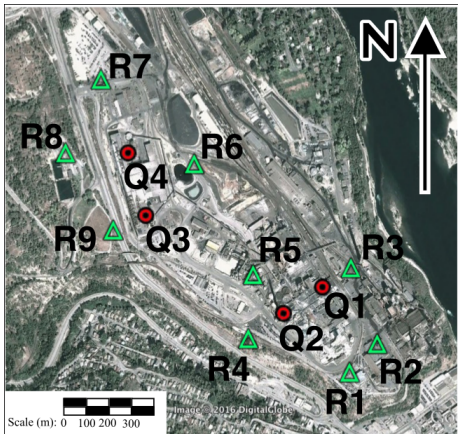

Figure 4.1: Aerial photograph taken from [10] of the lead-zinc smelter in Trail, British Columbia, Canada. The points Q_1 to Q_4 represent the sources of zinc. The green triangles R_1 to R_9 represent the location of the measurement devices.

4.1 A Mathematical Model for Pollutant Dispersion

Our starting point is the conservation of mass for particulate zinc in the atmosphere. Consider a closed region $\Lambda \subset \mathbb{R}^3$ with a mass m of zinc within it. Assume that in the interior of Λ there is a source of zinc and that zinc is flowing throughout the boundary of Λ (denoted by $\partial\Lambda$) due to an advection field (see Figure 4.2). It can be shown that net mass per unit time of zinc that is flowing through the boundary is given by [38]

$$\int_{\partial\Lambda} c(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)\cdot\hat{\mathbf{n}}dA,$$

where $\mathbf{v}(\mathbf{x},t)$ represents the wind velocity at a point $\mathbf{x}$ at time t and $c(\mathbf{x},t)$ in units of mass per units of volume is the concentration of zinc at a point $\mathbf{x}$ at time t. The vector $\mathbf{n}$ is the unit normal vector to $\partial\Lambda$ and dA is an area element. On the other hand the rate of change of total mass m at a time t inside Λ can be written as

$$\frac{dm(t)}{dt} = \frac{d}{dt}\int_{\Lambda} c(\mathbf{x},t)dV,$$

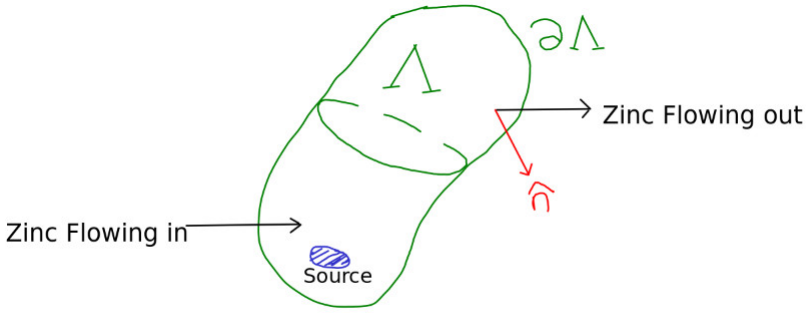

Figure 4.2: Schematic representation of the region Λ in space containing a source of zinc.

where dV is an element of volume in Λ. Finally, the amount of zinc that comes from the source at a time t is given by

$$\int_{\Lambda} s(\mathbf{x},t)dV,$$

where $s(\mathbf{x},t)$ is the source density with units of mass per unit volume per unit time. Conservation of mass states that the total mass inside Λ should be conserved. Therefore for all times t the rate of change of the mass inside m should equal all sources of its variation. Thus we can write

$$\frac{d}{dt}\int_{\Lambda} c(\mathbf{x},t)dV = -\oint_{\partial\Lambda} c(\mathbf{x},t)\mathbf{v}\cdot\mathbf{n}dA + \int_{\Lambda} s(\mathbf{x},t)dV.$$

Since we picked the orientation of $\partial\Lambda$ with the normal pointing outwards, it is necessary to put a minus sign in front of the surface integral for consistency. Assuming that the concentration and the velocity field are continuous functions of time and space, a straightforward application of the divergence theorem and the Leibniz rule for integrals gives

$$\int_{\Lambda} \left(\frac{\partial c(\mathbf{x},t)}{\partial t} + \nabla \cdot (c\mathbf{v}) - s(\mathbf{x},t) \right) dV = 0.$$

Since the region Λ was arbitrary, the previous equality holds if and only if [11]

$$\frac{\partial c(\mathbf{x},t)}{\partial t} + \nabla \cdot (c\mathbf{v}) = s(\mathbf{x},t), \text{ almost everywhere in } \mathbb{R}^3 \times \mathbb{R}. \tag{4.1}$$

If we apply this equation to estimate the concentration of zinc using real wind data measured experimentally, we will find that the prediction for the concentration is not completely accurate, even if the model is believed to be an accurate representation of the underlying physics and the model parameters are estimated as accurately as possible. One of the reasons for this is that at small scales, there are random fluctuations in the wind velocity that are very difficult to measure. To model this we follow [38] and write the wind velocity field as

$$\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}', \tag{4.2}$$

where $\bar{\mathbf{v}}$ is the measured wind velocity and $\mathbf{v}'$ is a random variable with zero mean. If we replace $\mathbf{v}$ in equation (4.1) with the expression for the velocity in equation (4.2) we get

$$\frac{\partial c(\mathbf{x},t)}{\partial t} + \nabla \cdot (c(\bar{\mathbf{v}} + \mathbf{v}')) = s(\mathbf{x},t). \tag{4.3}$$

The presence of the random variable $\mathbf{v}'$ transforms the solution c into a random variable as well. In this case we describe c as the contribution of two terms as

$$c(\mathbf{x},t) := \mathbb{E}(c)(\mathbf{x},t) + c(\mathbf{x},t)', \tag{4.4}$$

where c' satisfies $\mathbb{E}(c(\mathbf{x},t)') = 0$. The motivation behind this definition is the following: if we measure the concentration at a point $\mathbf{x}$ and at time t a large number of times under identical initial and boundary conditions, then we expect the measurements to have an underlying average behavior $\mathbb{E}(c)(\mathbf{x},t)$ plus some noise $c(\mathbf{x},t)'$. By substituting equation (4.4) into equation (4.3) we obtain

$$\frac{\partial \mathbb{E}(c)}{\partial t} + \nabla \cdot (\bar{\mathbf{v}}\mathbb{E}(c)) + \nabla \cdot (\mathbb{E}(c'\mathbf{v}')) = s(\mathbf{x},t). \tag{4.5}$$

This equation includes the extra variable $\mathbb{E}(c'\mathbf{v}')$. In this case we have two unknowns and one equation. One way to overcome this issue is to use the so-called mixing-length theory

[38]. We use the constitutive equation

$$\mathbb{E}(c'\mathbf{v}') = \mathbf{D}\nabla(\mathbb{E}(c)). \tag{4.6}$$

The term $\mathbf{D}$ is a rank two tensor called the eddy diffusivity tensor. This tensor is assumed to be symmetric, hence is always diagonalizable. For simplicity we assume that we are working on the principal axes of $\mathbf{D}$ [38], hence

$$\mathbf{D} = \begin{bmatrix} D_{xx} & 0 & 0 \\ 0 & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{bmatrix}.$$

Substituting equation (4.6) in equation (4.5) gives

$$\frac{\partial \mathbb{E}(c)}{\partial t} + \nabla \cdot (\bar{\mathbf{v}}\mathbb{E}(c) + \mathbf{D}\nabla\mathbb{E}(c)) = s(\mathbf{x}, t). \tag{4.7}$$

The variable $\mathbb{E}(c)$ is now a deterministic function of space and time. In order to make the notation simpler, we set $C(\mathbf{x}, t) := \mathbb{E}(c)$. Note that C has the same units as c. We interpret $C(\mathbf{x}, t)$ as the expected concentration that we would measure at a point $\mathbf{x}$ at time t.

We now turn our attention to the source density. We assume point-wise emission sources. Suppose there are n_s sources at the points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_s}$, then we assume that $s(\mathbf{x}, t)$ has the following the form

$$s(\mathbf{x}, t) = \sum_{j=1}^{n_s} q_j(t)\delta(\mathbf{x} - \mathbf{x}_j). \tag{4.8}$$

Here q_j is the rate of emission of the j-th source and has units of mass per unit time. The function $\delta(\cdot)$ is the Dirac delta function. For the case study of this chapter, we have $n_s = 4$, where n_s is the number of sources (see Figure 4.1). Putting together equations (4.7) and (4.8) we finally state our mathematical model for zinc dispersion as

$$\frac{\partial C(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\bar{\mathbf{v}}C(\mathbf{x}, t) + \mathbf{D}\nabla C(\mathbf{x}, t)) = \sum_{j=1}^{4} q_j(t)\delta(\mathbf{x} - \mathbf{x}_j). \tag{4.9}$$

We still need to discuss models for the diffusivity tensor $\mathbf{D}$ and wind velocity distribution $\mathbf{v}$. These quantities are hard to measure so we use empirical models for them. We will discuss in detail the assumptions on the diffusivity tensor and the wind velocity field in the next sections.

4.1.1 Assumptions on the Diffusivity Tensor

Following [26], the vertical diffusion coefficient (in the z direction) D_{zz} is represented by the following expression

$$D_{zz} = \frac{\kappa v_* z}{\phi(z/L)}, \tag{4.10}$$

where κ is the *von Karman* constant whose value in practice is set equal to 0.4. The denominator is defined as the piece-wise continuous function

$$\phi\left(\frac{z}{L}\right) = \begin{cases} 1 + 4.7\frac{z}{L} & \text{for } \frac{z}{L} \geq 0, \\ (1 - 15\frac{z}{L})^{-\frac{1}{2}} & \text{for } \frac{z}{L} < 0, \end{cases}$$

where L is the *Monin-Obukhov length*. The parameter v_* is known as the *friction velocity* and represented by

$$v_*(t) = \frac{\kappa v_r(t)}{\ln(\frac{z_r}{z_0})},$$

where $v_r(t)$ is a reference velocity at a reference height h_r. The variable z_0 is called the *roughness length*, that depends on terrain and surface type.

For the elements D_{xx} and D_{yy}, we assume $D_{xx} = D_{yy}$ and independence of height [26]. A commonly used equation for these parameters is

$$D_{xx} = D_{yy} \approx \frac{v_* z_i^{\frac{3}{4}} (-\kappa L)^{-\frac{1}{3}}}{10}.$$

The variable z_i is known as the *mixing layer height*.

4.1.2 Assumptions on the Wind Velocity Distribution

In practice, the wind velocity measurements are taken using anemometers that measure a two dimensional projection of the velocity vector field in a horizontal plane. Therefore it is necessary to model vertical variations of this vector field. Following [10], we consider a velocity vector field of the form

$$\mathbf{v} = (v_x(z,t), v_y(z,t), v_{set}). \tag{4.11}$$

Observe that we are assuming the wind velocity field is independent of x and y. The reason for this is that the terrain of interest is flat, hence no significant variations of the wind velocity at a given height are expected. In equation (4.11), v_{set} is a constant given by the settling velocity of the zinc particles. By Stokes' law, this velocity for spherical particles is given by

$$v_{set} = \frac{\rho g d^2}{18\mu},$$

where ρ is the particle density, g is the acceleration of gravity, d is the diameter of the particle, and μ is the dynamic viscosity of air. For the x, y components of $\mathbf{v}$ we assume a power law relation of the form

$$\|(v_x(z,t), v_y(z,t))\|_2 = v_r(t) \left(\frac{z}{z_r}\right)^p, \tag{4.12}$$

where $v_r(t)$ is the wind speed at a reference height z_r. The exponent p depends on factors such as the surface roughness and atmospheric stability class. For more details about the power law model for the wind velocity the reader is referred to [38].

Now we have all the information necessary to completely specify the model for the zinc pollutant dispersion. The domain of interest, the boundary and initial conditions in equation (4.9) are given by

$$\Pi := \{(x, y, z) \in \mathbb{R}^3 | z \geq 0\} \qquad \text{(domain of equation (4.9))}.$$

Following [10], the boundary conditions are given by the far-field condition

$$C(\mathbf{x}, t) \to 0 \qquad \text{as } \|x\| \to \infty,$$

and the Robin boundary condition

$$\left(v_{set} C + D_{zz} \frac{\partial C}{\partial z}\right)\Big|_{z=0} = v_{dep} C|_{z=0}\,. \tag{4.13}$$

Observe that the diffusivity in equation (4.10) vanishes at ground level. This is inconsistent with the boundary condition in equation (4.13). Hence, following [10] we assume the diffusivity to be non-zero below a *cutting height* z_{cut}.

The models for the wind velocity and the diffusivity tensor, introduce new parameters whose values need to be set. These parameters are listed below:

- p - the fitting parameter for the wind velocity power law,
- z_0 - roughness length,
- z_i - mixing layer height,
- L - Monin-Obukhov length,
- z_{cut} - cutting height.

In practice we often set a heuristic value for these parameters from a given empirical range [38, 10]. The caveat with this approach is that there is no good reason to choose one value

over a different one or else years of experience from an expert are necessary to make a sound choice. In this work we will use the Bayesian framework in order to decide the values of these parameters and the emission rates using the experimental data available. We will also quantify the uncertainties that are associated with the values of the parameters.

To estimate the parameters p, z_0, z_i, L, z_{cut} and the four sources sources, we use the experimental measurements of the total zinc deposition at the sites $R_1, \ldots, R_9$ (see Figure 4.1) over a one month period.

The mathematical model described so far, concerns the dispersion of particles and not the deposition. Thus is necessary to make the connection between the solution C of equation (4.9) and the deposition of zinc at the ground level. If v_{set} is the settling velocity of zinc particles, then the deposition per unit area at a point $(x, y, 0) \in \Pi$ during the interval of time $(0, T]$ is given by

$$w(x, y, T) = \int_0^T C(x, y, 0, t) v_{set} dt. \tag{4.14}$$

Since the nine measurements $R_1, \ldots, R_9$ were obtained by the placement of identical dust-fall jar collectors with small but non-zero, cross-sectional area, we can readily approximate the total deposition during the interval $(0, T]$ at the i-th site as

$$W(x_i, y_i, T) = \int_{dustfalljar} w(x, y, T) dxdy \approx w(x_i, y_i, T) \Delta A,$$

where ΔA is the cross-sectional area of the dust-fall jar. Furthermore we define

$$M_i = W(x_i, y_i, T) \qquad \text{for } i = 1, \ldots, 9. \tag{4.15}$$

The scalar M_i is a measurement of the zinc deposition at the site R_i. From now on we assume T to be one month.

From equations (4.14) and (4.15), we infer that the mapping from concentration to depositions is linear. Also from equation (4.9) it is straightforward to check that the concentration C is related to the source $s(\mathbf{x}, t)$ by the solution operator of the partial differential equation (4.9), which is also linear. Since composition of linear maps is linear, we conclude that the mapping from the emission rates q_j to the depositions M_j is linear . Therefore we can write

$$M_i = \sum_{j=1}^{4} a_{ij}(p, z_0, z_i, L, z_{cut}) q_j \qquad \text{for } i = 1, \ldots, 9. \tag{4.16}$$

The coefficients a_{ij} capture the dependence of the deposition on the model parameters p, z_0, z_i, L, z_{cut}, which is non-linear in general. Define the vectors

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} M_1 & \ldots & M_9 \end{bmatrix}^T, \\ \mathbf{q} &= \begin{bmatrix} q_1 & q_2 & q_2 & q_4 \end{bmatrix}^T, \end{aligned}$$

and write equation (4.16) more compactly as

$$\mathbf{y} = \mathscr{A}(p, z_0, z_i, L, z_{cut})\mathbf{q}. \tag{4.17}$$

Here $\mathscr{A}$ is a 9×4 matrix whose coefficients are the $a_{ij}(p, z_0, z_i, L, z_{cut})$.

Equation (4.17) models the relationship between deposition values and all other parameters in equation (4.9). However we do not know an expression for the 36 coefficients a_{ij} as functions of $(p, z_0, z_i, L, z_{cut})$. To find such an expression, we need to solve equation (4.9), but the solution does not have a known closed form and a numerical approximation is needed.
If we had unlimited computational budget we could solve equation (4.9) for as many different configurations of the parameters $(p, z_0, z_i, L, z_{cut})$ as we want and get an approximation to the coefficients a_{ij}. Clearly this is not feasible. A more realistic approach is to solve equation (4.9) for a number of different values of $(p, z_0, z_i, L, z_{cut})$ and construct an approximation to a_{ij} using these limited runs. We shall use the use the Gaussian processes approach of Section 2.2.1 to construct such an approximation. Before we construct the Gaussian processes we perform a sensitivity analysis as described in Section 2.2.3 to see if it is possible to reduce the dimensionality of the parameter space in order to reduce the cost of constructing the Gaussian processes.

4.2 Sensitivity Analysis

Our starting point is to define the set of maps $\{\varphi_{x,y}\}_{x,y\in\mathbb{R}}$. The input for each map is a point $(p, z_0, z_i, L, z_{cut})$ in the parameter space and the output is the deposition value at the location (x, y). The domain of each map is the range of allowed values for the set of parameters as shown in Table 4.1, taken from [38].

Parameter (units)	Symbol	Range
Velocity exponent	p	$[0.1, 0.4]$
Roughness length (m)	z_0	$[10^{-3}, 2]$
Height of mixing layer (m)	z_i	$[10^2, 3 \times 10^3]$
Monin-Obukhov length (m)	L	$[-500, -1]$
Cut-off length (m)	z_{cut}	$[1, 5]$

Table 4.1: Parameters of interest to out sensitivity study and their accepted ranges

We perform a sensitivity analysis as explained in Section 2.2.3, on the set of maps $\{\varphi_{x_i,y_i}\}_{i=1}^{9}$, where (x_i, y_i) represent the location R_i in Figure 4.1. To make computations simpler we map bijectively the set of ranges of the parameters into the five-dimensional unit hypercube. Thus, without loss of generality we may assume the deposition maps are of the

form

$$\varphi_{x_i,y_i} : [0,1]^5 \subset \mathbb{R}^5 \to [0,\infty) \qquad \text{for } i = 1, 2, \ldots, 9.$$

Recall from Section 2.2.3 that in order to estimate the Sobol' indices it is necessary to perform numerical integrations over the integrands $\{\varphi_{x_i,y_i}\}_{i=1}^9$. These are five-dimensional integrals, and so usual quadrature methods are not suitable. It is necessary to use Monte Carlo integration. But Monte Carlo integration requires us to evaluate $\{\varphi_{x_i,y_i}\}_{i=1}^9$ at a large number of points, which can also be expensive. To overcome this issue we use Gaussian process emulation as described in Section 2.2.1.

Implementing the routines to emulate and estimate Sobol' indices is a time consuming task. In order to optimize our time budget, we used the R packages DiceKriging and Sensitivity [36, 31] to construct the emulator and to estimate the total effect Sobol' indices for each of the maps $\{\varphi_{x_i,y_i}\}_{i=1}^9$. The DiceKriging package allows us to use five different kernels for the emulation. In order to consider the influence of different kernels on the Sobol' indices, we calculate the indices five times, one for each different kernel and summarize the results in the box plots in Figure 4.3.

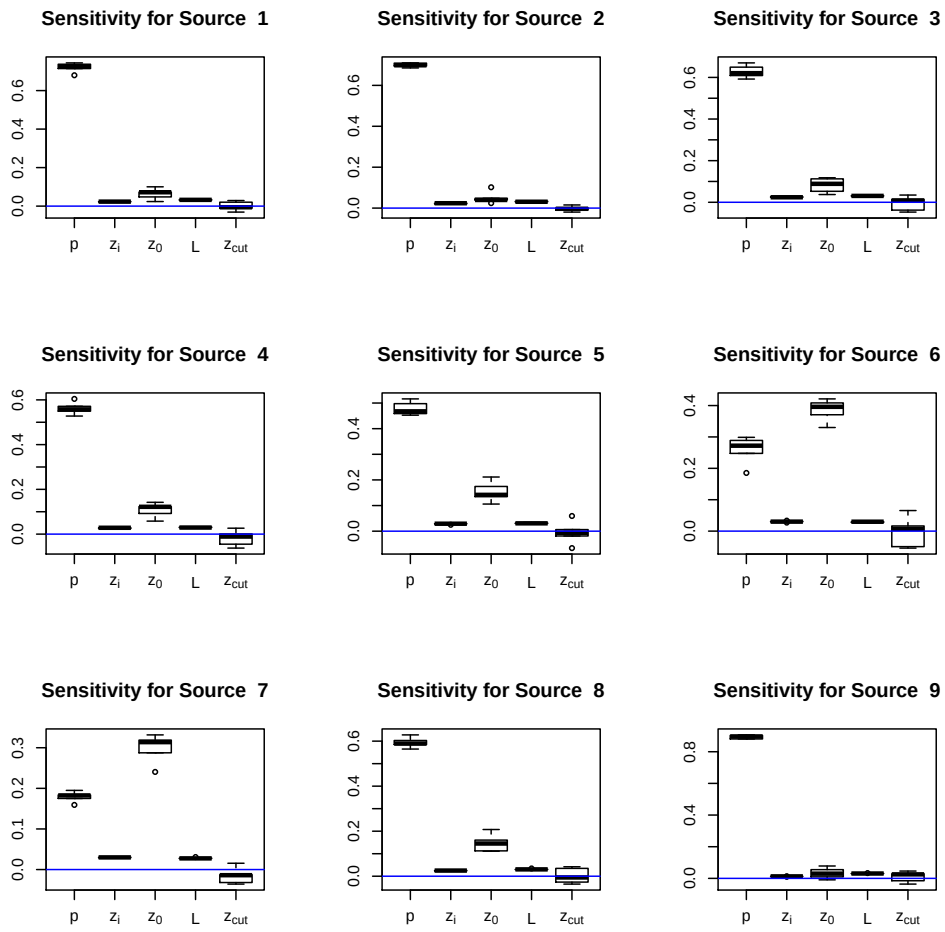

Figure 4.3: Boxplots containing the result for the total effect Sobol' index performed on each of the nine sensors. The dashed line represents a Sobol' index of zero.

This figure shows that two of the five variables, namely z_i and z_{cut}, have very little influence on the variance in the deposition. Therefore we ignore these and reduce the dimensionality of the parameter space from five to three. In this case we can write equation (4.17) as

$$\mathbf{y} = A(p, L, z_0)\mathbf{q}.$$

From now on we use the convention that the parameters z_i and z_{cut} are fixed at the values 100 and 2 respectively (see [10]). We now turn our attention into how to find an approximation for the matrix $A(p, L, z_0)$.

4.3 Building an Emulator for $\mathscr{A}(p, L, z_0)$

To solve equation (4.9) numerically, we use the finite volume solver of [10]. The details are given in [10]. Running the finite volume code on a $30 \times 30 \times 30$ grid takes from 30 minutes up to a few hours, hence the necessity to use Gaussian process regression to create an emulator as shown in Section 2.2.1.

To construct the emulator we need a training set. For this purpose we create a space filling design for the parameter space (p, L, z_0) as explained in Section 2.2.2 using a maximin design. Finding the optimal maximin design is challenging, hence we use a numerical approximation. We use the particle swarm algorithm for the optimization. See [3] for details.

Considering our time and computational budget, we chose to perform the experimental design with $n = 64$ points. The maximin design obtained by using a particle swarm for the optimization is shown in Figure 4.4.

With the experimental design in hand, we run the finite volume solver for each of the 64 different configurations of parameters. We repeat this step four times. The first time we run the 64 simulations by fixing the first source q_1 to 1 and the other three sources to zero, the second time we fix the second source q_2 to 1 and the other three sources to zero and so on. The reason for this implementation is that we need to connect the output of the finite volume solver, which is the deposition vector $\mathbf{y}$ at the nine sites $R_1, R_2, \ldots, R_9$, with the components of the matrix $\mathscr{A}(p, L, z_0)$. Recall that the relation between these two quantities is given by equation (4.17) which we write explicitly as

$$\mathbf{y} = \mathscr{A}(p, L, z_0) \underbrace{\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}}_{\mathbf{q}}, \tag{4.18}$$

For example if we run the finite volume code with the parameters set at the values p^*, L^*, z_0^* with the i-th source set to one and the other three to zero, then we obtain the output

$$\mathbf{y}^* = \begin{bmatrix} y_1^* \\ \vdots \\ y_9^* \end{bmatrix}.$$

Then from equation (4.18) it is easy to see that the following equality holds

$$\begin{bmatrix} y_1^* \\ \vdots \\ y_9^* \end{bmatrix} = \begin{bmatrix} a_{1i}(p^*, L^*, z_0^*) \\ \vdots \\ a_{9i}(p^*, L^*, z_0^*) \end{bmatrix}$$

where the RHS is the i-th column of $\mathscr{A}$. This result shows why it is necessary to run each of the 64 simulations four times to obtain the training set for the emulator of $\mathscr{A}$.

The emulator for $\mathscr{A}$ will be represented by $\widehat{A}$, similarly its components will be represented by $\hat{a_{ij}}$. The construction of $\widehat{A}$ is obtained by using Gaussian processes as explained in Section 2.2.1, and exemplified in Section 3.1.2. By construction, at the points in the maximin design in Figure 4.4, $\mathscr{A}$ and $\widehat{A}$ coincide. To account for the discrepancies outside this set of points we assume an additive Gaussian noise model between deposition, parameters and sources, that is (cf. Section 3.1.2)

$$\mathbf{y} = \widehat{A}(p, L, z_0)\mathbf{q} + \epsilon, \tag{4.19}$$

with $\epsilon \sim \mathcal{N}(0, \lambda I_{9\times 9})$. The value of the parameter λ will be specified later in this chapter. In equation (4.19) the vector $\mathbf{y}$ could represent either the output of the finite volume code at the dustfall jars positions or the experimental deposition measures. This interpretation of $\mathbf{y}$ implies that the random variable ϵ also accounts for the discrepancy between simulation of equation (4.9), the physics and experimental measurement errors.

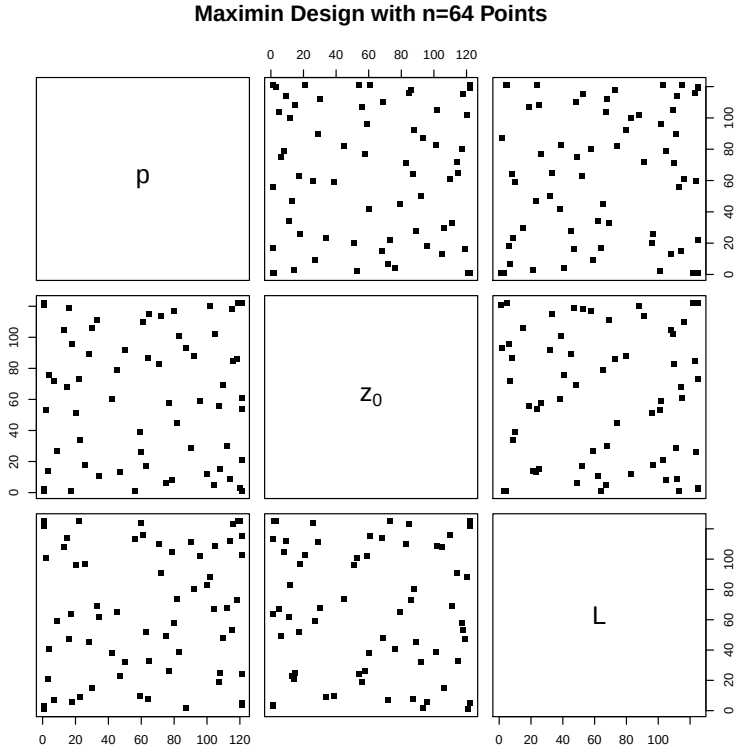

Figure 4.4: Maximin design with 64 points in the parameter space (p, L, z_0).

4.4 Bayesian Framework for the Inverse Problem

With the emulator in hand, we are now in a position to apply the Bayesian framework to estimate the values of the parameters and the source emission rate. Our goal is to characterize the posterior distribution for the parameters and emission rates; more precisely we want to find the posterior of the parameters p, z_0, L and sources $\mathbf{q}$ given the experimental data $\mathbf{y}$. Recall by Bayes' rule in equation (2.1), the posterior distribution is given by

$$\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y}) = \frac{\mathbb{P}_{like}(\mathbf{y}|p, z_0, L, \mathbf{q})\mathbb{P}_{prior}(p, z_0, L, \mathbf{q})}{Z(\mathbf{y})}. \tag{4.20}$$

Given the Gaussian additive noise model in equation (4.19), a straightforward calculation yields the likelihood

$$\mathbb{P}_{like}(\mathbf{y}|p, z_0, L, \mathbf{q}) = \frac{1}{(2\pi\lambda^2)^{\frac{9}{2}}} \exp\left(-\frac{1}{2\lambda^2}\|\widehat{A}q - \mathbf{y}\|^2\right), \tag{4.21}$$

where the constant of proportionality is given by

$$Z(\mathbf{y}) = \int \mathbb{P}_{like}(\mathbf{y}|p, z_0, L, \mathbf{q})\mathbb{P}_{prior}(p, z_0, L, \mathbf{q})dpdz_0dLd\mathbf{q}. \tag{4.22}$$

We now discuss the choice of the prior distribution.

4.4.1 Choosing a Prior

The values of the parameters p, z_0, L depend on the environmental and meteorological conditions of the region, whereas the values of the emission rates $q_1, \ldots, q_4$ do not. Thus, it is reasonable to assume that these variables are independent of each other. This independence can be captured as

$$\mathbb{P}_{prior}(p, z_0, L, \mathbf{q}) = \mathbb{P}_{prior}(p, z_0, L)\mathbb{P}_{prior}(\mathbf{q}).$$

By writing the prior distribution as the product of two distributions, we simplify the problem by working with two different subsets of variables. Let us first introduce the prior for the parameters p, z_0, L. We assume all three parameters are independent of each other, thus

$$\mathbb{P}_{prior}(p, z_0, L) = \mathbb{P}_{prior}(p)\mathbb{P}_{prior}(z_0)\mathbb{P}_{prior}(L).$$

As mentioned in Chapter 1, there is no strong reason to pick a value over the other for these parameters unless sufficient meteorological details are available. To this end, we assume a uniform distribution for each parameter. The allowed range for each parameter is given in

Table 4.1. Explicitly we have the distribution density for each parameter as

$$\begin{aligned} \mathbb{P}_{prior}(p) &= \frac{1}{0.3}\mathbf{1}_{[0.1,0.4]}, \\ \mathbb{P}_{prior}(z_0) &= \frac{1}{2-10^{-3}}\mathbf{1}_{[10^{-3},2]}, \\ \mathbb{P}_{prior}(L) &= \frac{1}{499}\mathbf{1}_{[-500,-1]}. \end{aligned} \tag{4.23}$$

Choosing the prior for the sources $q_1, \ldots, q_4$ requires more analysis since we are not completely ignorant about their possible values. Let us summarize our prior knowledge of these variables.

1. Each source is unrelated to the other three.
2. The q_i are positive and finite.
3. Engineering estimates of emission rates are given in Table 4.2 (these values are taken as a guideline).

Source	Estimated Emission Rate [ton/yr]
q_1	35
q_2	80
q_3	5
q_4	5

Table 4.2: Engineering estimates of the emission rates.

Mathematically, Condition 1 above can be modelled with an independent prior:

$$\mathbb{P}_{prior}(q_1, q_2, q_3, q_4) = \mathbb{P}_{prior}(q_1)\mathbb{P}_{prior}(q_2)\mathbb{P}_{prior}(q_3)\mathbb{P}_{prior}(q_4).$$

The second condition requires that the probability density for each source has to be supported in the set $[0, \infty)$. The third condition can be interpreted as follows: the mode of the prior for each source has to be at the engineering estimates, and 99% of the mass should be contained between 0 and 3 times that value. The reason for this is to give credibility to engineering estimates but not too much credit. We choose to work with the gamma distribution which is a probability distribution that satisfies the above conditions. We denote the probability measure of a gamma random variable by $\mathscr{G}(\alpha, \beta)$. The Lebesgue density of the gamma distribution is given by

$$g(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \mathbf{1}_{[0,\infty)},$$

where α and β are constants. We assume that each source has a gamma distribution

$$q_i \sim \mathscr{G}(\alpha_i, \beta_i) \text{ for } i = 1, 2, 3, 4. \tag{4.24}$$

We choose the values of α_i and β_i in terms of the engineering estimate $q_{eng,i}$ for the i-th source. More precisely we choose the values of α_i and β_i as the solution of the following system of equations

$$\begin{aligned} \frac{\alpha_i - 1}{\beta_i} &= q_{eng,i}, \\ qgamma(0.99, \alpha_i, \beta_i) &= 3q_{eng,i}. \end{aligned}$$

Here *qgamma* is the quantile function for the gamma distribution. The quantile function is the inverse of the cumulative distribution function. By choosing the values of the parameters for the gamma distribution in this manner, we satisfy

$$\max_{q \in [0,\infty)} g(q; \alpha_i, \beta_i) = q_{eng,i} \qquad \text{for } i = 1, 2, 3, 4.$$

and 99% of the mass of the density is concentrated between 0 and 3 times the engineering estimate [7]. Combining the results from equations (4.23) and (4.24) we conclude that the prior distribution for the emission rates has the form

$$\mathbb{P}_{prior}(p, z_0, L, \mathbf{q}) \propto \mathbf{1}_{[0.1,0.4]}(p)\mathbf{1}_{[10^{-3},2]}(z_0)\mathbf{1}_{[-500,-1]}(L)\prod_{i=1}^{4} q_i^{\alpha_i - 1}e^{-\beta_i q_i}\mathbf{1}_{[0,\infty)}(q_i). \tag{4.25}$$

With the prior and the likelihood, we can finally obtain an expression for the posterior by substituting equations (4.21),(4.22) and (4.25) into Bayes' rule (4.20). The posterior $\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y})$ is proportional to the Lebesgue density of

$$\exp\left(-\frac{1}{2\lambda^2}\|\widehat{A}\mathbf{q} - \mathbf{y}\|^2 - \sum_{i=1}^{4}\beta_i q_i\right)\prod_{j=1}^{4} q_i^{\alpha_i - 1}\mathbf{1}_{[0.1,0.4]\times[10^{-3},2]\times[-500,-1]}(p, z_0, L).$$

To make notation simpler we have chosen to represent the vector $[q_1, \ldots, q_4]^T$ as $\mathbf{q}$ or as component-wise in the summation. The indicator function represents the ranges of the allowed values of the parameters in Table 4.1. These values are widely accepted in the literature, however there is not a technically sound reason for why these ranges are acceptable. We will expand the possible values for these parameters in order to test the validity of the values in Table 4.1. The new set of ranges we picked is shown in Table 4.3.

Parameter (units)	Symbol	Range
Velocity exponent	p	$[0, 0.6]$
Roughness length (m)	z_0	$[0, 3]$
Monin-Obukhov length (m)	L	$[-600, 0]$

Table 4.3: Model parameters under study and their new allowed ranges.

Using these new ranges for the parameters the posterior $\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y})$ is now proportional to

$$\exp\left(-\frac{1}{2\lambda^2}\|\widehat{A}\mathbf{q}-\mathbf{y}\|^2-\sum_{i=1}^{4}\beta_i q_i\right)\prod_{j=1}^{4} q_i^{\alpha_i-1}\mathbf{1}_{[0,0.6]\times[0,3]\times[-600,0]}. \tag{4.26}$$

This equation is the solution for the inverse problem. However the formula for the posterior cannot be used directly. It is necessary to extract the information contained in it. In order to extract such information we can obtain point estimates of the parameters and the uncertainty associated with those estimates. Finding these estimates is the topic of the next section.

4.5 Inferring Parameters and the Sources

Recall that there are different types of point estimates, such as maximum a posteriori, or conditional mean (see equation (2.8)). To find any of these estimates and its associated uncertainty, is necessary to perform high dimensional integrals that are not analytically tractable. Thus, we resort to numerical integration techniques. In particular we use the Metropolis-Hastings(MH) algorithm (see Chapter 3, Algorithm 1) to sample the posterior measure and use Monte Carlo integration to approximate the quantities of interest. In order to use MH, the posterior in equation (4.26) has to be fully specified up to a normalizing constant. Up to this point we have not set the value of λ in the likelihood distribution in equation (4.21).

To understand why λ is important, consider the cases when λ is very small or very large. In the former case the exponential term

$$\exp\left(-\frac{1}{2\lambda^2}\|\widehat{A}\mathbf{q}-\mathbf{y}\|^2\right), \tag{4.27}$$

in the likelihood becomes negligible unless $\|\widehat{A}\mathbf{q}-\mathbf{y}\|$ is of the same order as λ. The interpretation is that measurements are so accurate that we are giving all credibility to the model and the data. In this situation the prior has little effect other than imposing certain constraints, and the posterior behaves like the likelihood. When λ is large the exponential in equation (4.27) flattens out, and the likelihood has less influence over the posterior, thus the prior distribution is more credible. In this case the posterior behaves similarly to the

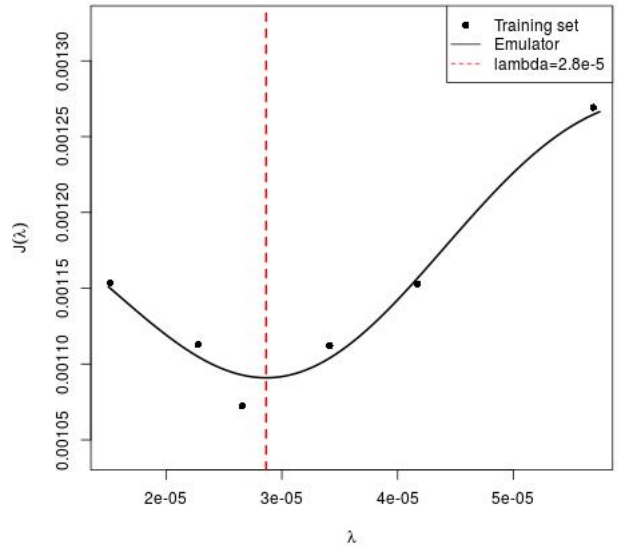

Figure 4.5: Emulator for the $J(\lambda)$ in equation (4.28)

.

prior. Thus, it is necessary to pick a value of λ that weights the importance of the likelihood correctly. That is, we are looking for a value of λ that weights the data, the atmospheric model simulation output and the prior information appropriately according to the quality of the data. To achieve this we choose the value of λ that minimizes the functional

$$J(\lambda) = \frac{1}{2}\int \left(\|\widehat{A}(p, L, z_0)\mathbf{q} - \mathbf{y})\|_2 + \|\mathbf{q} - \mathbf{q}_{est}\|_2 \right) d\mathbb{P}^{\lambda}_{post}, \tag{4.28}$$

where $\mathbf{q}_{est} = [35, 80, 5, 5]^T \frac{Ton}{yr}$. We use the notation $\mathbb{P}^{\lambda}_{post}$ to explicitly show the dependence of the posterior measure $\mathbb{P}_{post}$ with the parameter λ. The motivation to define J as above is the following: consider the expression

$$(1-\delta)\|\widehat{A}(p, L, z_0)\mathbf{q} - \mathbf{y})\|_2 + \delta\|\mathbf{q} - \mathbf{q}_{est}\|_2, \qquad \text{for } \delta \in [0, 1]. \tag{4.29}$$

In this case, depending on the value of δ, we are giving different weights to the atmospheric disperssion model credibility, and the prior information about $\mathbf{q}$. There is no reason to believe the atmospheric model is completely accurate so the engineering emission estimates can be dismissed. Hence, by choosing $\delta = \frac{1}{2}$ we weight both terms equally. Also we want to make the term (4.29) as small as possible, regardless of the values of p, z_0, L and $\mathbf{q}$, and using the information contained in the experimental data $\mathbf{y}$. Taking the expectation of (4.29) accomplishes that.

Estimating $J(\lambda)$ accurately is expensive, since it involves evaluating a seven-dimensional integral. Thus, to approximate a minimizer for λ we will approximate J using a Gaussian processes emulator. To create the training set, we evaluate J for six different values of λ using Markov Chain Monte Carlo integration. The results of the emulation are shown in Figure 4.5. The minimum for $J(\lambda)$ is attained approximately at $\lambda = 2.80008 \times 10^{-5}$, which is the value we pick for λ in the reminder of this chapter. We note that if more accuracy is needed in finding the minimizer it is possible to use Gaussian processes to perform accurate optimization; we refer the reader to [29] for details in this topic.

For the MH implementation we adapt Algorithm 1 in Chapter 3, in a way that the sampling of u (line 3 in the Algorithm) is done adaptively as proposed in [35]. Using the adaptative algorithm we obtained two million samples and as a burn-in period we discarded the first one million samples. Since the acceptance rate was relatively low, around 10%, we thinned the last million samples by picking every hundredth sample. In this way we end up with $10,000$ samples that are weakly correlated in the sense that the autocorrelation function is close to zero.

Using MH to sample from high-dimensional probability distributions is challenging, as is assessing the convergence of the Markov chain to the target distribution. To overcome these difficulties, several heuristics on convergence criteria have been developed, such as graphical methods and non-parametric tests of stationarity. The reader interested in this topic is referred to [34] and the references within. In this work we use the traces of the Markov chain to assess its convergence. A *trace plot* is a graphical display of the motion of the Markov chain in each of the dimensions in the support of the target probability density. Recall that the probability density we are interested in is the posterior in equation (4.26). This posterior is supported in a subset of $\mathbb{R}^7$. To obtain the trace plots, we plot the motion of the Markov chain separately in each of the seven dimensions (see Figure 4.6).

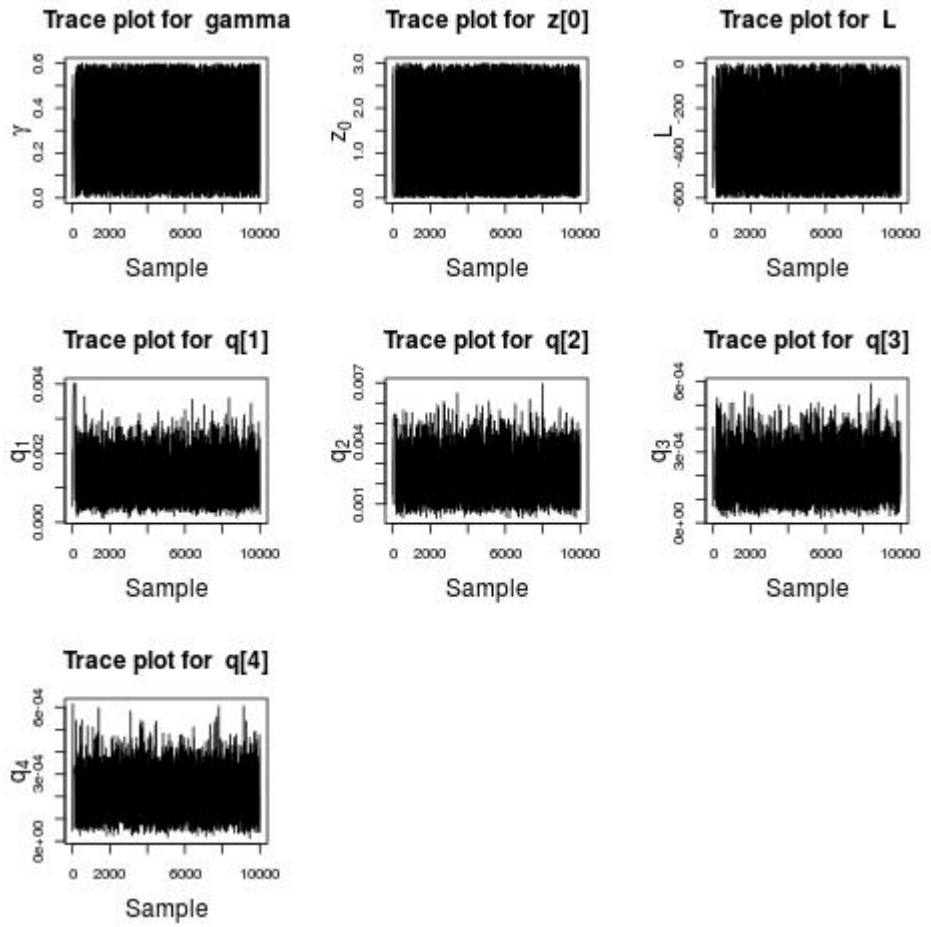

Figure 4.6: Trace plots of the Markov chain for each of the variables p, z_0, L and $\mathbf{q}$. Using the Metropolis-Hastings algorithm of Algorithm 1 in Chapter 3.

The trace plots in Figure 4.6 show that the Markov chain is moving around the whole support of the posterior density. Another observation is that different regions in the support of $\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y})$ are visited by the chain every so often. This means that the chain is not getting stuck in a local mode of the distribution and is mixing properly. In conclusion the trace plots have the behaviour one would expect of a Markov chain that has converged and whose realizations are taken from the target probability density.

One of the advantages of most sampling algorithm is that the samples obtained in each dimension are distributed as the marginal distribution for that variable. If X and Y are random variables jointly distributed as $\mathbb{P}(X, Y)$, then the *marginal distribution* of the random variable X is given by

$$\mathbb{P}(X) = \int \mathbb{P}(X, Y) dY.$$

For example the marginal distribution for the parameter p is given by

$$\mathbb{P}(p|\mathbf{y}) = \int \mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y}) dz_0 dL d\mathbf{q}.$$

We will use the marginals for each of the variables in the posterior distribution in equation (4.26) to obtain useful statistics about the parameters. The posterior histograms for the marginals of each of the variables of interest are shown in Figure 4.7), where we show the values of the point estimates for each distribution and the fitting curves for the parameters p, z_0, L and the sources q_1 to q_4. We now explain how we chose the point estimates and the associated uncertainties.

The next step for estimating the parameters is to decide which of the point estimates from equation (2.8) we will choose. By looking at Figure 4.7, it is clear that the marginal posterior for the four sources is skewed and with a well-defined mode. Thus, the mode is a reasonable choice for a point estimate. To approximate the value of the mode in the marginal of the sources, we will take advantage of the fact that the histograms for the marginals have a similar shape to the histogram obtained from a gamma distribution, hence we will fit a gamma distribution to the histograms and take the mode of the fitted distribution as our point estimate.

For the parameters p, z_0, L the situation is more subtle. For the parameter p the distribution has no distinctive points. This means that the data has little information regarding the values of p. For z_0 and L there are values that are more distinctive than others, but there is not sharp distinction between them. To pick a point estimate for p, z_0, L we fit a density over the histograms using the R built-in function *density* and then we pick the mode of each histogram to be the chosen point estimate.

Next we consider the uncertainty estimates for our parameters. We consider the 68% Bayesian confidence interval for each of the parameters. Given a point estimate x^* for a random variable X distributed with probability density ρ, a 68% Bayesian confidence interval of that estimate is defined as the ball of radius r centered at x^* such that

$$\int_{B(x^*,r)} \rho(x) dx = 0.68.$$

The reason to choose this uncertainty measure is to replicate the classical "mean plus standard deviation" measurement of confidence when dealing with Gaussian probability distributions [5]. The estimates for the parameters and the uncertainties are shown in Table 4.4.

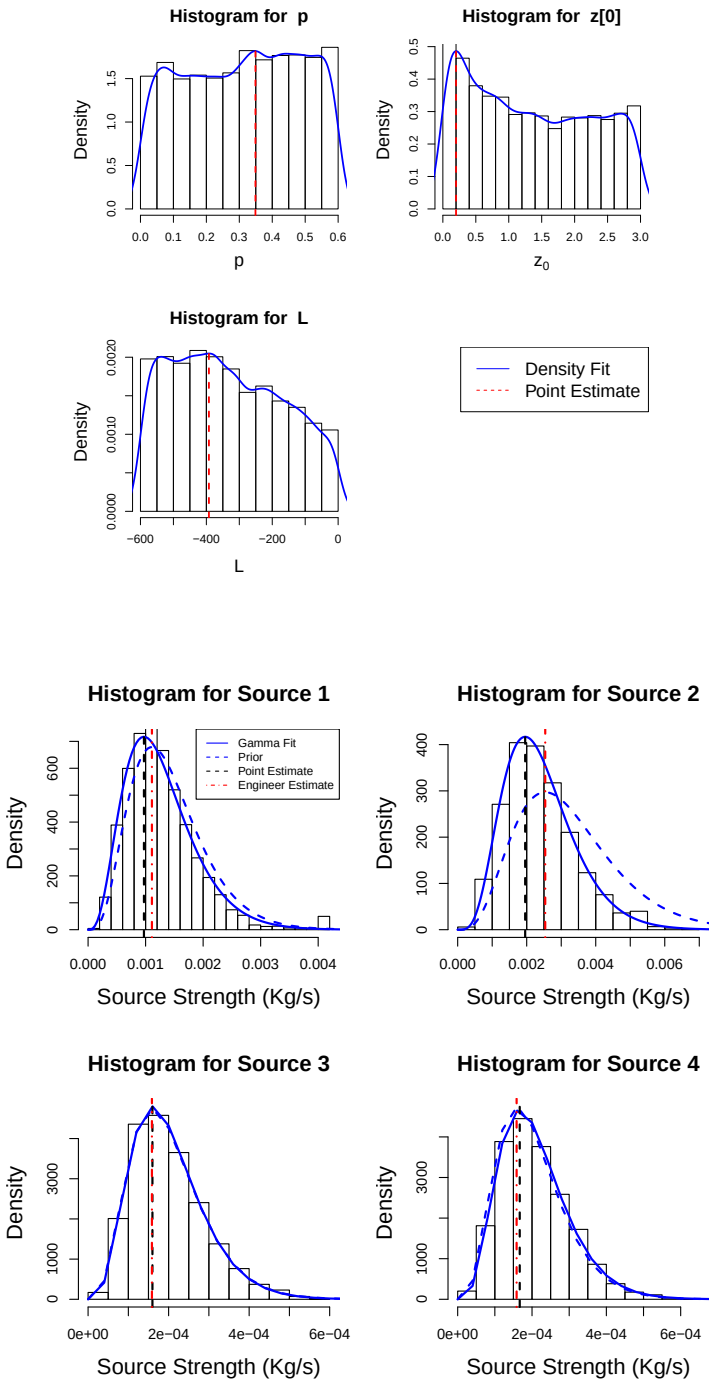

Figure 4.7: Histograms for the marginal posterior distribution for each of the seven unknowns in our model

The spread of the 68% confidence interval for the distribution of the parameters shows that the experimental data is not large enough so the posterior marginal distribution for the parameters is not decisively informative about particular values of the parameters. Despite

this, as we can see in the histograms in Figure 4.7 for L and z_0 that there are region in the support that are more likely than others. Even though we have no prior information about the values of the parameters and only 9 measurements of zinc deposition, our approach is able to distinguish certain regions with high probability. For the value of z_0 the region around the mode agrees with the typical values for this parameter in the literature, but for L the high probability values do not agree with the typical values chosen for this parameter [10, 38].

Parameter	Point Estimate	68% Confidence Interval
p	0.3478	$[0.1498, 0.5458]$
z_0	0.0811	$[0, 1.5781]$
L	-379.45	$[-195,86, -563.04]$

Table 4.4: Parameters along with their point estimates and 68% confidence interval.

For the estimates of the sources the situation is different due to the prior knowledge we have about them. The estimates and uncertainties are shown in Figure 4.8

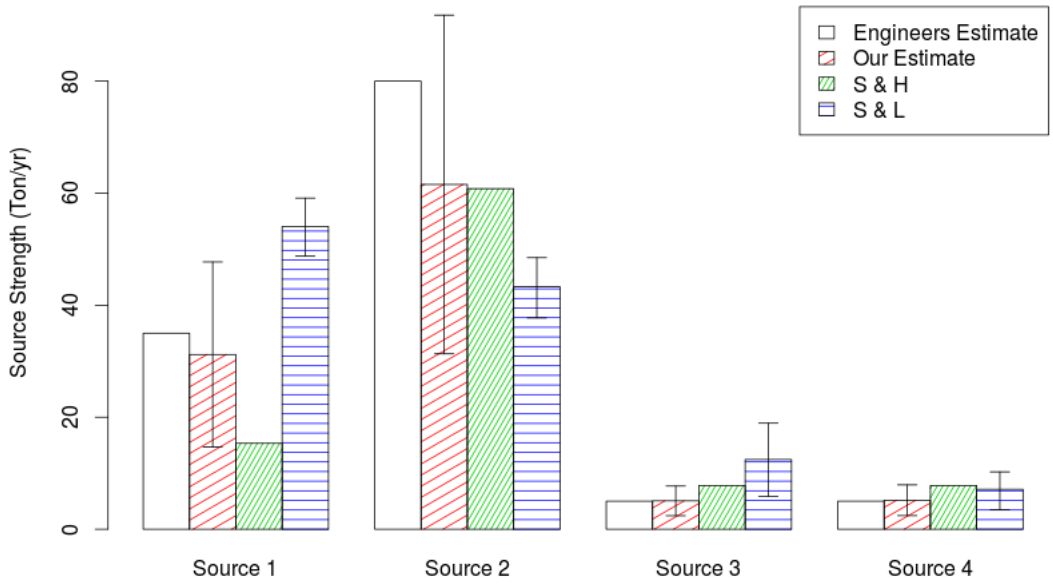

Figure 4.8: Comparison between engineering estimates of the source strength source with our point estimate and the uncertainty associated to it. We also compare with the results obtained in [10](S & H) and [25](S & L).

The results in Figure 4.8, show that sources 1 and 2 are most likely overestimated by the engineers. Whereas sources 3 and 4 are slightly underestimated. If we take into account the error bars for our estimates and the small difference between that estimate and the engineering estimate, we conclude that the experimental data is not necessary to make any strong conclusion about sources 3 and 4. For sources 1, 3 and 4, our results agree with [25] and [10]. However for source 2 the results we obtained agree with the results obtained in [25] for this same source, but in the result obtained in [10], source 2 is severely underestimated.

To conclude this chapter we will compare our results with the results obtained in [25] and [10].

Source	Estimated Emission Rate [ton/yr]
q_1	18
q_2	60
q_3	8
q_4	8

Table 4.5: Emission rates estimated in [25].

The results obtained in Figure 4.8 agree with those obtained in [25] in the sense that Sources 1 and 2 are overestimated by the engineers. Emissions in source 1 are overestimated by 12 and 20 tons per year respectively, whereas for Sources 3 and 4 they are underestimated by 3 tons each (see Table 4.5).

Source	Estimated Emission Rate [ton/yr]
q_1	80
q_2	58
q_3	39
q_4	9

Table 4.6: Emission rates estimated in [10].

When comparing the results in Figure 4.8 with those obtained in [10], listed in Table 4.6, we see that their results for Source 2 partially disagree with the results obtained in this work and in [25]. According to reference [10], Source 1 is severely underestimated by the engineers by 45 tons a year. For the other three sources the results in this work and the results in [10] and [25] agree in the sense that Source 2 is overestimated by the engineers and Sources 3 and 4 are underestimated.

In summary our results agree or partially agree with those obtained in related works. Also it is import to mention that to obtain the results exposed in this chapter, the computational overhead was considerably smaller than simulating equation (4.9) for several different

values of the parameters until getting a reasonable prediction by the model. Plus we have the benefit of assessing the accuracy of the estimates obtained.

Chapter 5

Conclusions

In this thesis we have developed a method to cheaply compute, using Gaussian process, computationally expensive atmospheric dispersion models. Also we show how to combine this with the Bayesian framework to solve the source inversion problem. Furthermore we include in the process, the estimation of the parameters the model depends upon. So far the estimation of parameters in atmospheric dispersion models has been done empirically as a trial and error process by calibrating the parameters involved so to obtain results that agree with the experimental data, one benefit of the Bayesian approach is that there is no trial and error step and the process to find the right parameters of the model is automatized. Also, In this work we use the experimental data available and with the aid of Bayes' rule we obtain point estimates for the parameter and the associated uncertainty.

We stress the conclusion that the methodology explained in this thesis goes beyond atmospheric dispersion models and partial differential equations such as the one used here (see equation (1.2)). As long as there is a differential equation whose parameters we want to estimate but is expensive to calculate, we can apply the methodology described here.

Finally, the results obtained using our proposed approach agree with the results obtained in other papers and books where more classical approaches have been used, showing that the emulation process does not affect the estimations and greatly accelerate calculations. This observation makes the approach used here a good candidate for real time applications. Again, we mention that the possible real time applications are not restricted to atmospheric dispersion models but can also be applied in more general settings.

Bibliography

[1] R. J. Adler. *The Geometry of Random Fields*. Number 62 in Classics in Applied Mathematics. SIAM, Philadelphia, 2010.

[2] Vladimir Igorevich Arnol'd. *Mathematical Methods of Classical Mechanics*, volume 60. Springer Science & Business Media, 2013.

[3] Rajesh K. Arora. *Optimization: Algorithms and Applications*. CRC Press, 2015.

[4] Alberto Bressan. *Lecture Notes on Functional Analysis*. American Mathematical Society, 2013.

[5] Michael G. Bulmer. *Principles of Statistics*. Courier Corporation, 1979.

[6] P.C. Chatwin. The use of statistics in describing and predicting the effects of dispersing gas clouds. *Journal of Hazardous Materials*, 6(1-2):213–230, 1982.

[7] Richard M. Dudley. *Real Analysis and Probability*, volume 74. Cambridge University Press, 2002.

[8] I.G. Enting and G.N. Newsam. Inverse problems in atmospheric constituent studies: II. sources in the free atmosphere. *Inverse Problems*, 6(3):349, 1990.

[9] Bamdad Hosseini and John M. Stockie. Bayesian estimation of airborne fugitive emissions using a Gaussian plume model. *Atmospheric Environment*, 141:122–138, 2016.

[10] Bamdad Hosseini and John M. Stockie. Estimating airborne particulate emissions using a finite-volume forward solver coupled with a Bayesian inversion approach. *Computers and Fluids*, 154:27–43, 2017.

[11] John H. Hubbard and Barbara B. Hubbard. *Vector Calculus, Linear Algebra, and Differential Forms: a Unified Approach*. Matrix Editions, 2015.

[12] Edwin T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[13] Mark E. Johnson, Leslie M. Moore, and Donald Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990.

[14] J. Kaipio and E. Somersalo. *Statistical and computational inverse problems*, volume 160 of *Applied Mathematical Sciences*. Springer Sience & Business Media, New York, 2005.

[15] Andrew Keats, Eugene Yee, and Fue-Sang Lien. Bayesian inference for source determination with applications to a complex urban environment. *Atmospheric Environment*, 41(3):465–479, 2007.

[16] Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B – Statistical Methodology*, 63(3):425–464, 2001.

[17] Leonid P. Lebedev, Iosif I. Vorovich, and Graham Maurice Leslie Gladwell. *Functional Analysis: Applications in Mechanics and Inverse Problems*, volume 41. Springer Science & Business Media, 2012.

[18] Ádám Leelössy, Ferenc Molnár, Ferenc Izsák, Ágnes Havasi, István Lagzi, and Róbert Mészáros. Dispersion modeling of air pollutants in the atmosphere: A review. *Open Geosciences*, 6(3):257–278, 2014.

[19] Nicolas Lerner. *A Course on Integration Theory.* Springer, 2014.

[20] W.S. Lewellen and R.I. Sykes. Meteorological data needs for modeling air quality uncertainties. *Journal of Atmospheric and Oceanic Technology*, 6(5):759–768, 1989.

[21] Mikhail Lifshits. *Lectures on Gaussian Processes.* Springer, 2012.

[22] Mikhail A. Lifshits. *Gaussian Random Functions*, volume 322. Springer Science & Business Media, 2013.

[23] Ching-Ho Lin and Len-Fu W. Chang. relative source contribution analysis using an air trajectory statistical approach. *Journal of Geophysical Research: Atmospheres*, 107(D21):ACH 6–1 – ACH 6–10, 2002.

[24] J. David Logan. *Applied Partial Differential Equations.* Springer, 2014.

[25] E. Lushi and J. M. Stockie. An inverse Gaussian plume approach for estimating atmospheric pollutant emissions from multiple point sources. *Atmospheric Environment*, 44(8):1097–1107, 2010.

[26] A. S. Monin and A. M. Obukhov. Basic laws of turbulent mixing in the surface layer of the atmosphere. *Contributions of the Geophysical Institute of Slovakian Academy of Sciences, USSR*, 151:163–187, 1954.

[27] Kevin P Murphy. *Machine learning: a Probabilistic Perspective.* MIT press, 2012.

[28] Anthony O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, 91(10):1290–1300, 2006.

[29] Michael A. Osborne, Roman Garnett, and Stephen J. Roberts. Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15, 2009.

[30] Luc Pronzato and Werner G. Müller. Design of computer experiments: Space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[31] Gilles Pujol, Bertrand Iooss, Alexandre Janon with contributions from Khalid Boumhaout, Sebastien Da Veiga, Jana Fruth, Laurent Gilquin, Joseph Guillaume, Loic Le Gratiet, Paul Lemaitre, Bernardo Ramos, Taieb Touati, and Frank Weber. *Sensitivity: Global Sensitivity Analysis of Model Outputs*, 2016. R package version 1.12.1.

[32] Shankar Rao. Uncertainty analysis in atmospheric dispersion modeling. *Pure and Applied Geophysics*, 162(10):1893–1917, 2005.

[33] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT press, Cambridge, 2006.

[34] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer texts in Statisitics. Springer Science & Business Media, New York, 2013.

[35] Gareth O. Roberts and Jeffrey S. Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

[36] Olivier Roustant, David Ginsbourger, and Yves Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.

[37] Andrea Saltelli, Karen Chan, and E. Marian Scott. *Sensitivity Analysis*, volume 134 of *Wiley Series in Probability and Statistics*. Wiley, New York, NY, 2000.

[38] John H. Seinfeld and Spyros N. Pandis. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. John Wiley & Sons, New York, 1997.

[39] Yuri N. Skiba. On a method of detecting the industrial plants which violate prescribed emission rates. *Ecological Modelling*, 159(2):125–132, 2003.

[40] Ilya M Sobol. Sensitivity estimates for non-linear mathematical models. *Mathematical Modelling and Computational Experiments*, 1(4):407–414, 1993.

[41] Michael D. Sohn, Pamela Reynolds, Navtej Singh, and Ashok J. Gadgil. Rapidly locating and characterizing pollutant releases in buildings. *Journal of the Air & Waste Management Association*, 52(12):1422–1432, 2002.

[42] D. Bruce Turner. *Workbook of Atmospheric Dispersion Estimates: an Introduction to Dispersion Modeling*. CRC press, 1994.

[43] Felipe A. Viana, Gerhard Venter, and Vladimir Balabanov. An algorithm for fast optimal latin hypercube design of experiments. *International Journal for Numerical Methods in Engineering*, 82(2):135–156, 2010.