Design and Analysis of Computer Experiments: Assessing and Advancing the State of the Art

by

Hao Chen

M.Sc., Statistics, The University of British Columbia, 2013
B.Sc., Statistics, Renmin University of China, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Statistics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

March 2018

Abstract

Computer experiments have been widely used in practice as important supplements to traditional laboratory-based physical experiments in studying complex processes. However, a computer experiment, which in general is based on a computer model with limited runs, is expensive in terms of its computational time. Sacks et al. (1989) proposed to use Gaussian process (GP) as a statistical surrogate, which has become a standard way to emulate a computer model. In the thesis, we are concerned with design and analysis of computer experiments based on a GP. We argue that comprehensive, evidence-based assessment strategies are needed when comparing different model options and designs.

We first focus on the regression component and the correlation structure of a GP. We use comprehensive assessment strategies to evaluate the effect of the two factors on the prediction accuracy. We also have a limited evaluation on empirical Bayes methods and full Bayes methods, from which we notice Bayes methods with a squared exponential structure do not yield satisfying prediction accuracy in some examples considered. Hence, we propose to use hybrid and full Bayes methods with flexible structures. Through empirical studies, we show the new Bayes methods with flexible structures not only have better prediction accuracy, but also have a better quantification of uncertainty.

In addition, we are interested in assessing the effect of design on prediction accuracy. We consider a number of popular designs in the literature and use several examples to evaluate their performances. It turns out the performance difference between designs is small for most of the examples considered. From the evaluation of designs, we are motivated to use a sequential design strategy. We compare the performances of two exist-

ing sequential search criteria and handle several important issues in using the sequential design to estimate an extreme probability/quantile of a floor supporting system. Several useful recommendations have also been made.

In summary, the thesis concerns both the design and analysis of computer experiments: we not only assess the performance of existing methods, but also propose new methods motivated by the assessment and provide insights into issues faced by practitioners.

Lay Summary

Computer experiments have been widely used in practice as important supplements to traditional physical experiments in studying complex processes. However, a computer experiment is expensive in terms of its computational time. Hence, Gaussian process (GP) was proposed to used as a statistical surrogate. The scope of the thesis is rather broad: we are concerned with design and analysis of computer experiments based on a GP. We use comprehensive assessment strategies to evaluate the effect of several factors on the prediction accuracy of the GP model. In addition, we propose new methods motivated by the assessment. Working with an engineering computer model, we also provide insights into issues faced by practitioners.

Preface

The dissertation is written up under the supervision of Prof. William J. Welch. The research questions and the proposed new methods are discussed with Prof. Welch during our weekly research meetings. I am responsible for all major areas of concept formation, mathematical derivation, simulation study and thesis composition. Prof. Welch spend a lot of time in helping me formulate the problems, find solutions and edit the draft. Prof. Jim Zidek also spent much time in providing helpful suggestions and editing the draft.

Chapter 2 is based on a published paper: Chen, H., Loeppky, J., Sacks, J. and Welch, W. (2016), "Analysis Methods for Computer Experiments: How to Assess and What Counts?", *Statistical Science*, 31, 40-60. I conducted the simulation study and created results for the key tables and figures in the article.

Chapter 3 is based on a published paper: Chen, H., Loeppky, J. and Welch, W., "Flexible Correlation Structure for Accurate Prediction and Uncertainty Quantification in Bayesian Gaussian Process Emulation of a Computer Model", *SIAM/ASA Journal on Uncertainty Quantification*, 5, 598-620. The paper comes as an extension to my MSc thesis. I searched and found a suitable prior distribution for the smoothness parameter in the Bayesian analysis. In addition, I drafted the article.

Chapter 4 is based on a manuscript : Chen, H., Loeppky, J., Sacks, J. and Welch, W., "Evaluation of Designs for Computer Experiments". The manuscript will be submitted for journal publication in future.

Chapter 5 will be converted to a manuscript for journal publication in future: Chen, H., and Welch, W., "Sequential Computer Experimental Design for Estimating an Extreme Probability or Quantile". This manuscript is based on a collaborative project between the Department of Statistics of

UBC and FP Innovations. The computer model of a floor system used in the article was suggested by Conroy Lum from FP Innovations.

Table of Contents

List of Tables

List of Figures

Glossary

CGP Composite Gaussian Process.

CP Corner Peak.

EI Expected Improvement.

FL Full Linear.

GP Gaussian Process.

LHS Latin Hypercube Sampling.

MLE Maximum Likelihood Estimation.

mLHD maximin Latin Hypercube Design.

OA Orthogonal Array.

OALHD Orthogonal Array based Latin Hypercube Design.

rLHD random Latin Hypercube Design.

RMSE Root Mean Squared Error.

SGD Sparse Grid Design.

SL Select Linear.

trLHD transformed Latin Hypercube Design.

Acknowledgements

First, I would like to acknowledge my special debts to my supervisor, Professor William J. Welch, who lead me to the intriguing field of analysis of computer experiments and is always strongly supportive of my research career. It is my great pleasure to work with him. I also want to express my heartfelt gratitude to Professor James Zidek, Professor Alexandre Bouchard-Cote, who are my supervision committee members and provided lots of guidance and help to me. In addition, I owe a debt of gratitude to Professor Jerome Sacks and Professor Jason Loeppky, who are my co-authors for several important journal publications.

I would also express my gratitude to Professors Ying MacNab, Lang Wu, Bruce Dunham, Jiahua Chen, Rollin Brant, Paul Gustafson, Matas Salibin-Barrera and Yew-Wei Lim for their constant mentoring and excellent teaching during my stay at UBC. I am also grateful to Peggy Ng, Elaine Salameh, Ali Hauschildt and Andrea Sollberger for their hard work and kind help. Thanks to everyone for making the department such an amazing place.

I feel extremely grateful to my girlfriend — Miss Yumian Hu, who is also my best friend. I would like to thank her for all of the support as well as the sacrifice she has made for me. I would not be able to accomplish the PhD degree without her company.

Last but not least, I owe my special thanks to my parents for their support and understanding of my PhD study in Canada.

Dedication

To my beloved father and mother: Mr. Qindong Chen
and Mrs. Xiuying Zhao, who are so proud.

Chapter 1

Introduction

In this chapter, we outline the research and briefly review some previous work in the relevant fields. The chapter can be divided into three parts: in the first part, we talk about what are computer experiments and the objectives of analyzing a computer model. In the second part, designs for computer experiments are discussed. The structure of the thesis is outlined in the last part.

1.1 Computer Models

Many complex phenomena are extremely difficult to investigate through controlled physical experiments. Instead, computer experiments become important alternatives to provide insights. Nowadays, computer experiments have been successfully applied in many sciences and engineering fields, for instance climate change, where traditional laboratory-based experiments are impossible to conduct. In general, a computer experiment is a designed set of runs of a computer model, which usually has the following two distinguishing features: (1) it is deterministic, that is, repeating an identical set of inputs does not change the output (2) it is time-consuming, a single run may take several hours or even days to complete.

Consider a typical computer model, which was described by Gramacy and Lee (2008). NASA was developing a new reusable rocket booster called the Langley Glide-Back Booster (LGBB). NASA had built a computer model to model the flight characteristics (lift, drag, pitch, side-force, yaw and roll) of the LGBB as a function of 3 inputs—side slip angle, mach number and angle of attack. For each input configuration triplet, the computer model will yield six response variables as described above. However, even for one

set of inputs, it takes the computer model 5-20 hours on a high-end workstation to solve the sets of Euler equations. A small modification of the input configuration means another 5-20 hours of running time. Hence, it would help if one could approximate the output of the computer model with adequate accuracy and much less computational time.

Without loss of generality, we assume each set of inputs is a $1 \times d$ row vector, where d denotes the input dimension and the corresponding output is a scalar. If there is more than one output, one can treat each independently or reduce more complex output to a few scalar summaries. Now, suppose that we have n input configurations as well as the corresponding n outputs from a computer model. The primary goal in the analysis of computer experiments is to predict the output of untried sets of inputs via a statistical surrogate model.

A popular choice for the surrogate model is the Gaussian process (GP) (Sacks et al., 1989), and GP is the only surrogate model we consider in this thesis. Assuming the correlation parameters of a GP are known, the best linear unbiased predictor (BLUP) can be obtained by minimizing the mean squared error (MSE) of the predictor. However, in practice one has to estimate all of the unknown parameters using the available data (n sets of inputs and n outputs). In addition, since most real computer models are expensive to evaluate, the sample size of the available data, n, is usually not big. Therefore, how to carefully design the computer experiment to provide as much information as possible within a limited sample size is of great importance. We will elaborate on each point throughout the thesis.

1.2 Analysis of Computer Experiments

1.2.1 Gaussian Process

Sacks et al. (1989) treated the output of a deterministic computer model as if it is a realization from the following possibly non-stationary regression model:

$$Y(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\boldsymbol{\beta} + Z(\mathbf{x}). \tag{1.1}$$

Here, $\mathbf{x}$ is a vector of inputs to the computer model, $\mathbf{f} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$ contains k known regression functions, $\boldsymbol{\beta}$ is a $k \times 1$ column vector of unknown regression parameters and $Z(.)$ is a Gaussian process defined on the input space $\mathcal{X}$ with zero mean and unknown variance σ^2. The above formulation defines a random function on the $\mathcal{X}$ domain. The interpretation of (1.1) is straightforward: the mean of the stochastic process depends on $\mathbf{x}$ in a standard regression manner, while the residual function is assumed to follow a stationary GP.

The output of a computer model is deterministic, i.e., no independent random errors exist. Hence least-squares fitting of a response surface might not be appropriate. Treating a computer model as a realization from a stochastic process provides a statistical framework for design and analysis. In addition, it gives a basis for uncertainty quantification.

Let $\mathbf{x}$ and $\mathbf{x}'$ be two sets of inputs. The correlation between $Z(\mathbf{x})$ and $Z(\mathbf{x}')$ is denoted by $R(\mathbf{x}, \mathbf{x}')$. Following common practice, $R(\mathbf{x}, \mathbf{x}')$ is taken to be a product of 1-d correlation functions in the distances $h_j = |x_j - x'_j|$, i.e., $R(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^{d} R_j(h_j)$. In this thesis, we mainly consider four choices of R_j and they will be introduced and discussed in detail in chapter 2. In general, we use $\boldsymbol{\psi}$ to denote the set of unknown correlation parameters. The correlation matrix $\mathbf{R}$ is fully specified by $\boldsymbol{\psi}$. In addition, no matter which specific correlation structure is used, the correlation matrix $\mathbf{R}$ is a positive-definite $n \times n$ symmetric matrix with all its diagonal elements equalling 1.

In terms of the regression component in (1.1), we specify three choices for $\mathbf{f}(\mathbf{x})$ in the thesis and the details will be discussed in chapter 2 as well. In practice, without meaningful prior information, how to choose the correlation structure and the regression component remain an open research question. These are the topics of chapter 2.

1.2.2 Prediction

Running a computer model n times at input vectors $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, ..., $\mathbf{x}^{(n)}$ produces n outputs $\mathbf{y} = \left(y(\mathbf{x}^{(1)}), y(\mathbf{x}^{(2)}), \ldots, y(\mathbf{x}^{(n)})\right)^T$. Given a new input

configuration $\mathbf{x}^*$, we wish to predict $y(\mathbf{x}^*)$ whatever correlation structure and regression term is used. According to Sacks et al. (1989), the predictive distribution of $y(\mathbf{x}^*)$ conditional on $\boldsymbol{\beta}$, σ^2, $\boldsymbol{\psi}$ and $\mathbf{y}$ is Gaussian:

$$N\left(m_{\boldsymbol{\psi}}(\mathbf{x}^*), v_{\boldsymbol{\psi}}(\mathbf{x}^*)\right), \tag{1.2}$$

where

$$m_{\boldsymbol{\psi}}(\mathbf{x}^*) = \mathbf{f}^T(\mathbf{x}^*)\boldsymbol{\beta} + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \tag{1.3}$$

and

$$v_{\boldsymbol{\psi}}(\mathbf{x}^*) = \sigma^2\left(1 - \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)\right). \tag{1.4}$$

Here, $\mathbf{F}$ is the $n \times k$ matrix with row i containing $\mathbf{f}^T(\mathbf{x}^{(i)})$, the $n \times 1$ vector $\mathbf{r}(\mathbf{x}^*)$ is obtained from one of the four correlation structures considered in the thesis and depends on which structure is used, with element i given by $R(\mathbf{x}^*, \mathbf{x}^{(i)})$ for all $i = 1, \ldots, n$. The subscript $\boldsymbol{\psi}$ in the notation for $m_{\boldsymbol{\psi}}(\mathbf{x}^*)$ and $v_{\boldsymbol{\psi}}(\mathbf{x}^*)$ emphasizes that these quantities depend on $\boldsymbol{\psi}$. All parameters are unknown and hence require estimation in practice.

1.2.3 Parameter Estimation

In practice, the parameters $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{\psi}$ have to be estimated, leading to further variability in the predictive distribution. In general, the inference paradigm can be classified as being in one of two categories: the first, the frequentist's viewpoint, treats the unknown parameters as constants, which leads to the use of method of maximum likelihood estimation (MLE). It is a special case of the empirical Bayes method. The second is to take a fully Bayesian perspective by assuming each parameter has a specified prior distribution and use the available data to generate its posterior distribution. We briefly introduce the empirical Bayes approach in the sequel along with its relationship to frequency theory.

In the case of interest to us, empirical Bayes estimates the unknown parameters by maximizing the joint likelihood function. Based on (1.1), the

likelihood function for $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\psi}$ is multivariate normal:

$$L(\mathbf{y}|\boldsymbol{\beta},\sigma^2,\boldsymbol{\psi}) = \frac{1}{(2\pi\sigma^2)^{n/2}\det^{1/2}\mathbf{R}}\exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{F}\boldsymbol{\beta})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{F}\boldsymbol{\beta})\right\}. \tag{1.5}$$

For any fixed $\boldsymbol{\psi}$ and hence $\mathbf{R}$, maximizing the likelihood with respect to $\boldsymbol{\beta}$ and σ^2 gives their MLEs:

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{R}^{-1}\mathbf{y} \tag{1.6}$$

and

$$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}}). \tag{1.7}$$

Note that the MLEs for $\boldsymbol{\beta}$ and σ^2 depend on the parameters $\boldsymbol{\psi}$ through $\mathbf{R}$. Plugging $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ into (1.5) gives the profile likelihood,

$$\frac{1}{(2\pi\hat{\sigma}^2)^{n/2}\det^{1/2}(\mathbf{R})}\exp\left\{-\frac{n}{2}\right\}.$$

It needs to be numerically maximized to yield MLEs for $\boldsymbol{\psi}$.

To obtain predictions, the MLEs of $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{\psi}$ are substituted for the true parameter values in (1.2). Extra uncertainty is thereby introduced by use of $\hat{\boldsymbol{\psi}}, \hat{\sigma}^2, \hat{\boldsymbol{\beta}}$, but only the contribution from $\hat{\boldsymbol{\beta}}$ to that added uncertainty is easily quantified. The estimated predictive mean is

$$\hat{m}_{\boldsymbol{\psi}}(\mathbf{x}^*) = \hat{\boldsymbol{\beta}} + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}}) \tag{1.8}$$

and the estimate of the predictive variance in (1.4) becomes

$$\begin{aligned}\hat{v}_{\boldsymbol{\psi}}(\mathbf{x}^*) = {}& \hat{\sigma}^2[1-\mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)]+ \\ & \hat{\sigma}^2[f(\mathbf{x}^*)-\mathbf{F}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)]^T(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}[f(\mathbf{x}^*)-\mathbf{F}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)].\end{aligned} \tag{1.9}$$

An approach takes a Bayesian perspective, in which each unknown parameter is assumed to have a prior distribution. The method of Markov chain Monte Carlo (MCMC) is then used to draw samples from its posterior

distribution. Then the mean and standard deviation are calculated from the sample as estimates of the true output and its standard error, respectively. The basic idea of the Bayesian paradigm is straightforward. However, different researchers assume different prior distributions. Moreover, either use of a hybrid of the MLE and Bayesian approaches or of a fully Bayesian approach to estimate all of the unknown parameters of a GP is another topic that needs discussion. Details of two popular Bayesian implementations as well as newly proposed Bayesian methods will be discussed in chapter 3.

1.3 Design of Computer Experiments

We next provide a brief overview of design for computer experiments. Design is essentially the way we select the sets of inputs at which the outputs are evaluated by the computer model. The primary goal is to carefully design an experiment such that it can "train" the best statistical surrogate for achieving a certain goal, such as making predictions at untried sets of inputs. In this thesis, design is classified into the following two categories: (1) a fixed design and (2) a sequential design. Given the same sample size, n, a fixed design is one in which all of its sets of inputs are generated simultaneously at the beginning and the design stays fixed throughout the process, i.e, no more information is added. By contrast, a sequential design works with the same sample size budget, but in a more dynamic way: only part of the budget is used at the beginning to form an initial design and new sets of inputs are sequentially added into the initial design, one per iteration until it reaches the size n. If more than one point is added at each iteration, the design is called batch sequential. It is obvious that a sequential design scheme is more flexible than that of a fixed design, but it is usually more complex.

1.3.1 Fixed Design

Unless otherwise stated in the sequel, the term "design" in the rest of the thesis refers to a fixed design. We will use "sequential design" to refer

to a design that takes the sequential approach.

We first talk about the fixed design. As far as we are concerned, an obvious way that computer experiments differ from the traditional physical experiments is that the computer model is deterministic, that is, the same result will always be observed if the same set of inputs is evaluated. Uncertainty arises from the fact that the true relationship between inputs and outputs is unknown and any functional relationship researchers use, such as the GP model, is only an approximation to the true relationship.

Based on the above observation, Santner et al. (2003) proposed two principles in selecting designs for computer experiments as follows:

1. A design should not take more than one observation at any set of inputs.

2. Because we do not know the true relationship between the response and inputs, a design should allow one to fit a variety of models and should provide information about all portions of the experimental region.

The above two principles require that a good design for computer experiments should be spread over the design space. Thus, it is not surprising to see the need for a design to be "space-filling" is a widely accepted principle nowadays when the primary interest is to emulate a computer model.

Let us now elaborate on the "space-filling" design. Without loss of generality, at least for rectangular regions, designs are defined on $[0,1]^d$. A design based on a random Latin hypercube sample (LHS) was proposed by McKay et al. (1979) to beat a "baseline" design which is constructed by completely random sampling from $U(0,1)$ independently in each dimension. The design is also called a Latin hypercube design (LHD). A Latin hypercube is the generalization of a square grid to an arbitrary number of dimensions, whereby each cell is the only one in each axis-aligned hyperplane containing it. The authors showed the variance of the sample mean $\frac{1}{n}\sum_{i=1}^{n} Y^{(i)}$ smaller using a LHD than using a completely random design, if $y(x_1,\ldots,x_d)$ is monotonic in each of its arguments. Stein (1987) further showed that if $n \to \infty$,

LHD has a smaller variance than a completely random design without the monotonicity condition.

During the past few decades, there are many variants of the LHD, such as the maximin LHD and the minimax LHD. The ideas underlying the maximin design and minimax design were first proposed by Johnson et al. (1990) and later those designs were introduced within the class of LHDs. Those designs can be viewed as a class of "space-filling" designs where such a property is achieved by defining and optimizing a distance measurement, although the specific metric may vary from one design to another.

There exists another class of "space-filling" design in which such a property is obtained by using a low-discrepancy sequence (Niederreiter, 1988). Given a set $P = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$, using Niederreiter's notation, the discrepancy is defined as

$$D_n(P) = \sup_{B \in J} \left| \frac{A(B;P)}{n} - \lambda_d(B) \right|,$$

where λ_d is the d-dimensional Lebesgue measure, $A(B;P)$ is the number of points in P that fall into B, and J is a set of d-dimensional regions in $[0,1]^d$. B is a member of J. The discrepancy is low if the proportion of points in P falling into B is close to the measure of B for all B.

An obvious application of a low-discrepancy sequence is numerical integration. The integral of a function f can be approximated by the average of the function evaluations at n points:

$$\underbrace{\int_0^1 \cdots \int_0^1}_{d} f(\mathbf{x}) d\mathbf{x} \approx \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}^{(i)}).$$

If the points are chosen randomly from a known distribution, this is the Monte Carlo method. However, if the points are chosen as elements of a low-discrepancy sequence, this is the quasi-Monte Carlo method. Niederreiter (1988) showed that the quasi-Monte Carlo method has a rate of convergence close to $O(1/n)$, whereas the rate for the Monte Carlo method is $O(1/\sqrt{n})$. Please see Niederreiter (1992) for more details. To approximate the integral

well, the set $P = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ should minimize the size of holes in the space. A Sobol sequence (Sobol, 1967) is another type of "space-filling" design whose "space-filling" property is achieved by a low-discrepancy sequence. More details about it will be discussed in chapter 4.

1.3.2 Review of Two Fixed Designs Utilized in Chapters 2 and 3

We assess the effect of designs in chapter 4, in which a detailed review of some fixed designs is given in section 4.1. However, there are two fixed designs frequently used in chapters 2 and 3. We, therefore, provide a review of them in this section to help readers better follow the materials in the following chapters.

Random LHD

McKay et al. (1979) introduced the random LHD. Without loss of generality, given a fixed sample size n, a random LHD in 2-d over the unit square can be constructed by the following steps:

- Construct $n \times n$ cells over the unit square.
- Construct an $n \times 2$ matrix, Z, with column independent random permutations of the integers, $\{1, 2, \ldots, n\}$.
- Each row of Z gives the row and column indices for a cell on the grid. For the i^{th} $(i = 1, 2, \ldots, n)$ row of Z, take a random uniform draw from the corresponding cell. The resulting design is the random LHD.

It can easily be extended to more than 2-d scenario. More details about the random LHD will be given in section 4.1.

Maximin LHD

An maximin LHD can be viewed as a combination of an LHD and a maximin design. The idea of a maximin design is that in order for the points to be spread out over the space, no two points are too close to each

other. To be more specific, let $D \subset \mathscr{X}$ be an arbitrary n-point design that consists of distinct inputs $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$. One way to measure the distance between any two points $\mathbf{x}^{(i)}, \mathbf{x}^{(j)}$ is given by

$$\rho(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(\sum_{k=1}^{d} \left| \mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)} \right|^p \right)^{1/p}, \tag{1.10}$$

where $p = 1$ and $p = 2$ are rectangular and Euclidean distances, respectively. The Euclidean distance, $p = 2$ is used in this thesis. A maximin design maximizes the minimal distance between any two points in the design space. Given the design D, its minimal distance is defined.

$$\Phi(D) = \min_{\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in D} \rho(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}). \tag{1.11}$$

Intuitively, it makes sure that no two points are too close, and hence the design is spread out. Johnson et al. (1990) first defined the maximin design. A design that maximizes $\Phi(D)$ in (4.2) within the class of LHDs is called maximin LHD, which is abbreviated as *mLHD* throughout the thesis.

1.3.3 Sequential Design

A sequential design is an adaptive design. It is an active learning process that allows the model to learn itself "on the fly" as new information is added sequentially. The core of a sequential design is the selection criterion, which selects the next point that will be added into the training set. Different research objectives usually have different selection criteria. The expected improvement (EI) criterion (**?**) is a popular one and is considered in the thesis. The basic idea is that given an improvement function, it selects the next point that maximizes the expectation of the improvement function. The expectation can be interpreted as an average over possible realizations of the conditional GP of the "improvement" a new point brings after it is added into the training set.

In general, different improvement functions were proposed for achieving different statistical objectives; for instance **?** used an improvement function

for optimization. Ranjan et al. (2008) proposed a different improvement function for contour estimation. Although the improvement functions are different due to different research goals, the underlying idea is similar, as both take the EI as the selection criterion. A comparison of EI and a hypothesis testing based method is conducted in chapter 5.

1.4 Brief Overview of the Thesis

The thesis is manuscript-based in the sense that each chapter is either based on a published journal paper or will be converted into a journal paper for future publication. However, the thesis also has strong connections between chapters. The structure of the thesis is as follows.

In chapter 2, we are concerned with some fundamental issues of using a GP as a statistical proxy for complex computer models. First of all, without any prior scientific information, how to choose the regression component and how to model the correlation structure of the GP? We use comprehensive, evidence-based assessment strategies to compare such modelling options. Applying the strategies to several computer models shows that a regression model more complex than a constant mean has little impact on prediction accuracy. The choice of correlation function has modest effect, but there is little to separate two common choices, the power exponential and the Matérn structure. We also have a limited comparison of Bayesian and empirical Bayes methods.

From the evaluation of Bayesian and empirical Bayes methods in chapter 2, we notice that empirical Bayes methods with a squared exponential structure do not yield satisfactory prediction accuracy in some examples considered. Hence, we propose in chapter 3 to use Bayesian methods with the power exponential and Matérn structures, which are much more flexible than the squared exponential structure. Through extensive empirical studies, we show the new methods with flexible structures not only have better prediction accuracy, but also have a better quantification of uncertainty.

In chapter 4, we are interested in assessing the effect of different designs on the prediction accuracy with a GP model. We consider popular

fixed designs in the literature with desirable mathematical properties and use several examples to evaluate their performances. From an intensive simulation study, we find that the performance difference in terms of prediction accuracy of different designs is small. There are other factors that have much bigger impacts on the prediction accuracy: the sample size and the transformation of output, Y.

In chapter 5, with a real computer model in wood engineering, we compare the performances of two existing sequential methods in estimating the failure probability of the engineering system and its corresponding quantile. We also discuss several practical issues that faced by practitioners, closing the gap between the theory of sequential design and its application in practice. Based on our study, we make some useful suggestions.

Finally, in chapter 6, we make some comments and summarize the future work motivated from the dissertation.

From this brief introduction, it should be clear that the thesis covers a very broad area of computer experiments: we not only review and assess the methods that exist for design and analysis of computer experiments, but also extend the existing methods to overcome the drawbacks found from the assessment. In summary, the thesis is concerned with fundamental issues in design and analysis of computer experiments.

Chapter 2

Evaluation of the Analysis Methods for Computer Experiments

Statistical methods based on a regression model plus a zero-mean GP have been widely used for analyzing a deterministic computer model. Our main thrusts in this chapter are the practical questions faced by the GP model users: What regression function and correlation function should be used? Does it matter? Besides the main points, there exist other variations of using a GP, for instance whether an empirical Bayes or a fully Bayesian method should be used to estimate the unknown GP parameters? We also partially address those variations in the chapter.

We will call a GP model with specified regression and correlation function a Gaussian stochastic process (GaSP) model. For example, GaSP(Const, PowExp) will denote a constant mean regression term and the power exponential correlation function. Following common practice, the correlation function is taken to be a product of 1-d correlation functions in the distances $h_j = |x_j - x'_j|$, i.e., $R(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^{d} R_j(h_j)$. The four 1-d correlation functions and three regression models under consideration are as follows.

- **Squared exponential (abbreviated SqExp)**:

$$R_j(h_j) = \exp\left(-\theta_j h_j^2\right) \tag{2.1}$$

  where $\theta_j > 0$. The sensitivity parameter (θ_j) controls how fast the correlation decays when the distance between $\mathbf{x}$ and $\mathbf{x}'$ increases. The

process is infinitely differentiable with respect to each x_j, i.e., it is extremely smooth, because of the fixed exponent of 2 in the distance metric. SqExp has been used in many applications; see Gramacy and Lee (2008), Kennedy and O'Hagan (2001), for instance.

- **Power exponential (abbreviated PowExp)**:

$$R_j(h_j) = \exp\left(-\theta_j h_j^{\alpha_j}\right) \tag{2.2}$$

where $\theta_j > 0$ and $\alpha_j \in [1, 2]$. The introduced parameters, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_d)$, govern smoothness, and hence PowExp is a more flexible family than SqExp.

- **Matérn**:

$$R_j(h_j) = \frac{1}{\Gamma(\nu_j)2^{\nu_j - 1}} \left(\tilde{\theta}_j h_j\right)^{\nu_j} K_{\nu_j}\left(\tilde{\theta}_j h_j\right), \tag{2.3}$$

where Γ is the Gamma function, K_{ν_j} is the modified Bessel function of order ν_j and $\tilde{\theta}_j > 0$. Here $\tilde{\theta}_j$ is a sensitivity parameter, and ν_j controls smoothness. To be consistent with the parameterization of the sensitivity parameters of SqExp and PowExp, we redefine $\tilde{\theta}_j$ as $\theta_j = 2\sqrt{\nu_j}/\tilde{\theta}_j$. For simplicity we consider special cases defined by ν_j. If $\nu_j = \delta_j + \frac{1}{2}$ with integer $\delta_j \geq 0$, there are δ_j derivatives with respect to x_j. The four special cases considered are

$$R_j(h_j) = \begin{cases} \exp(-\theta_j|h_j|) & (\delta_j = 0 \text{ derivatives}) \\ \exp(-\theta_j|h_j|)\,(\theta_j|h_j| + 1) & (\delta_j = 1 \text{ derivative}) \\ \exp(-\theta_j|h_j|)\left(\frac{1}{3}(\theta_j|h_j|)^2 + \theta_j|h_j| + 1\right) & (\delta_j = 2 \text{ derivatives}) \\ \exp(-\theta_j|h_j|^2) & (\delta_j \to \infty \text{ derivatives}). \end{cases}$$

The last case gives SqExp. The Matérn class was recommended by (Stein (1999) , section 2.7) for its control via $\nu_j > 0$ of the differentiability of the correlation function with respect to x_j, and hence that of the prediction function. Also note that different coordinates have different ν_j to be estimated in practice for the Matérn class just like

the PowExp structure.

- **Matérn-2**: Some authors (e.g. Picheny et al. (2013)) fix ν_j in the Matérn correlation function to give some differentiability. The Matérn-2 subfamily sets $\nu_j = 2 + \frac{1}{2}$ for all j, giving 2 derivatives. This is a special case of the Matérn correlation function.

In general, $\boldsymbol{\psi}$ will denote the set of unknown correlation parameters:

$$\boldsymbol{\psi} = \begin{cases} (\theta_1, \ldots, \theta_d) & \text{(SqExp)} \\ (\theta_1, \ldots, \theta_d, \alpha_1, \ldots, \alpha_d) & \text{(PowExp)} \\ (\theta_1, \ldots, \theta_d, \delta_1, \ldots, \delta_d) & \text{(Matérn)} \\ (\theta_1, \ldots, \theta_d) & \text{(Matérn-2).} \end{cases} \tag{2.4}$$

In terms of the regression component in (1.1), we explore three main choices:

- Constant (abbreviated Const): β_0, i.e., $k = 1$, only the intercept.
- Full linear (FL): $\beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d$, i.e., a full linear model in all input variables with $k = d + 1$.
- Select linear (SL): linear in x_j like FL but only includes selected terms.

The proposed algorithm for SL is as follows. For a given correlation family construct a default predictor of outputs for untried inputs with the Const approach. Decompose the predictive function (Schonlau and Welch, 2005) and identify all main effects that contribute more than $100/d$ percent to the total variation. These become the selected coordinates. Typically, large main effects have clear linear components. If a large effect lacks a linear component, little is lost by including a linear term. Inclusion of possible nonlinear trends can be pursued at considerable computational cost; we do not routinely do so, but in section 2.2 we do include a regression model with nonlinear terms in x_j.

Assessing statistical strategies for the analysis of a computer experiment often mimics what is done for physical experiments: a method is proposed, applied in examples and compared to other methods. If it is possible, formal

mathematical comparisons are made, otherwise, one needs to use empirical criteria to assess performance. An initial empirical study for a physical experiment has to rely on the specific data of that experiment and, while different analysis methods may be applied, all are bound by the single data set. There are limited opportunities to vary sample size or design. However, computer experiments provide rich opportunities to investigate the relative merits of an analysis method. One can construct a whole spectrum of "replicate" experiments for a single computer model, avoiding the dangerous "anecdotal" reports.

The danger of being misled by anecdotes can be seen in the following example. The borehole function (Morris et al., 1993) is frequently used as a testbed to assess different models. A 27-run orthogonal array (OA) in the 8 input factors was proposed as a design, following Joseph et al. (2008). The 27 runs were analyzed via GaSP with a specific $\mathbf{R}$ (the SqExp) and with two choices for the regression function: Const versus SL (x_1 is selected). The details are not important for now. A set of $10,000$ test points selected at random in the 8-dimensional input space was then predicted. The resulting values of the root mean squared error (RMSE) measure defined in (2.5) of section 2.1 were 0.141 and 0.080 for the Const and SL regression models, respectively.

While the SL approach reduced the RMSE by about 43% relative to a model with the Const component, does this example provide powerful evidence for using regression terms in the GaSP model? Not quite. We replicated the experiment with the same choices of regression terms and correlation function and the same test-data, but the training data came from a theoretically equivalent 27-run OA design (There are many equivalent OAs, e.g., by permuting the labels between columns of a fixed OA.) The RMSE values for the second analysis were 0.073 and 0.465 for the Const and SL models respectively. The two analyses lead to very different conclusions.

We study the impact on prediction accuracy of the particular model specifications commonly used. The primary goals are two-fold: First, we propose a more evidence-based approach to distinguish what may be important from the unimportant and what may need further exploration. Second,

our application of this approach to various examples leads to some specific recommendations.

The rest of the chapter is organized as follows. In sections 2.1 and 2.2, we use various computer models to assess the effect of the two aforementioned factors on prediction accuracy, along with some choices of sample size and designs. In section 2.3, some other modelling strategies are considered and compared. We make some comments in section 2.4. Finally, some conclusions and recommendations will be made in section 2.5.

2.1 Fast Computer Models

We slightly modify the commonly-used root mean square error (RMSE) to normalized root mean squared error (RMSE) of prediction over the holdout (test) set ($e_{\text{rmse,ho}}$) and normalized maximum absolute error ($e_{\text{max,ho}}$). These are given below:

$$e_{\text{rmse,ho}} = \frac{\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}(\mathbf{x}_{\text{ho}}^{(i)}) - y(\mathbf{x}_{\text{ho}}^{(i)})\right)^2}}{\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\bar{y} - y(\mathbf{x}_{\text{ho}}^{(i)})\right)^2}}, \quad e_{\text{max,ho}} = \frac{\max|\hat{y}(\mathbf{x}_{\text{ho}}^{(i)}) - y(\mathbf{x}_{\text{ho}}^{(i)})|}{\max|\bar{y} - y(\mathbf{x}_{\text{ho}}^{(i)})|}, \tag{2.5}$$

where m is the number of runs in the holdout set, $\mathbf{x}_{\text{ho}}^{(i)}$ is point i in the holdout set, $\hat{y}(\mathbf{x}_{\text{ho}}^{(i)})$ is the predicted value, and $\bar{y}$ is the trivial predictor: sample mean of y in the training set. The normalization in the denominator puts RMSE roughly on $[0, 1]$ whatever the scale of y, with 1 indicating no better performance than $\bar{y}$. Similarly, worst-case performance can be defined as the normalized maximum absolute error.

What are tolerable levels of error? Clearly, these are application-specific so that tighter thresholds would be demanded, say, for optimization than for sensitivity analysis. For general purposes we take the rule of thumb that $e_{\text{rmse,ho}} < 0.10$ is useful. For normalized maximum error it is plausible that the threshold could be much larger, say 0.25 or 0.30. These speculations are consequences of the experiences we document later, and are surely not the

last word. The value of having thresholds is to provide benchmarks that enable assessing when differences among different methods or strategies are practically insignificant versus statistically significant.

For fast codes under our control, large holdout sets can be obtained. Hence, in this section performance is measured through the use of a holdout (test) set of $10,000$ points, selected as a random LHD on the input space. We consider three fast computer models in this section.

2.1.1 Borehole Model

The first model we look at is the borehole model (Morris et al., 1993). It is an 8-dimensional computer model that has served as a test-bed in many contexts (e.g., Chen (2013)). The detailed description is put in the appendix A.1. Three different designs are available for the experiment: a 27-run, 3-level OA; a 27-run maximin LHD (mLHD); and a 40-run mLHD. As permuting the columns of an OA or an mLHD design does not change the internal structure of that design, we generate 25 equivalent designs by permuting the columns of the available designs to avoid "anecdotal" comparisons. In addition, GP parameters are estimated by the method of MLE.

There are 12 possible modelling combinations from the four correlation structures and three regression models outlined previously. The SL choice for $\boldsymbol{\beta}$ here is always the term x_1. Its main effect accounts for approximately 80% of the variation in predictions over the 8-dimensional input domain, and all analyses with a constant regression function choose x_1 and no other terms across all designs and all repeat experiments. The experimental results are shown in Figure 2.1 for the normalized RMSE and in Figure 2.2 for the normalized maximum absolute error.

The top row of Figure 2.1 shows results with the 27-run OA design. For a given modelling strategy, 25 random permutations of the columns of the 27-run OA lead to 25 repeat experiments and hence a reference set of 25 values of $e_{\text{rmse, ho}}$ shown as a boxplot. The results are striking. Relative to a constant regression model, the FL regression model has apparently larger $e_{\text{rmse, ho}}$ values for all correlation functions. The SL regression also performs

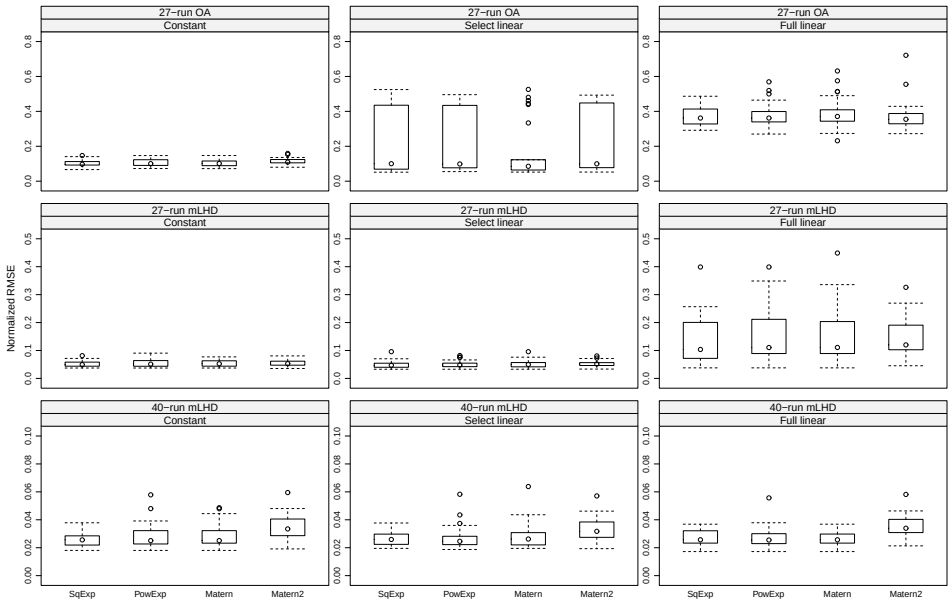

Figure 2.1: Borehole function: Normalized RMSE of prediction, $e_{\text{rmse,ho}}$, for GaSP with all combinations of three regression models and four correlation functions. There are three base designs: a 27-run OA (top row); a 27-run mLHD (middle row); and a 40-run mLHD (bottom row). For each base design, 25 random permutations of its columns give the 25 values of $e_{\text{rmse, ho}}$ in a boxplot.

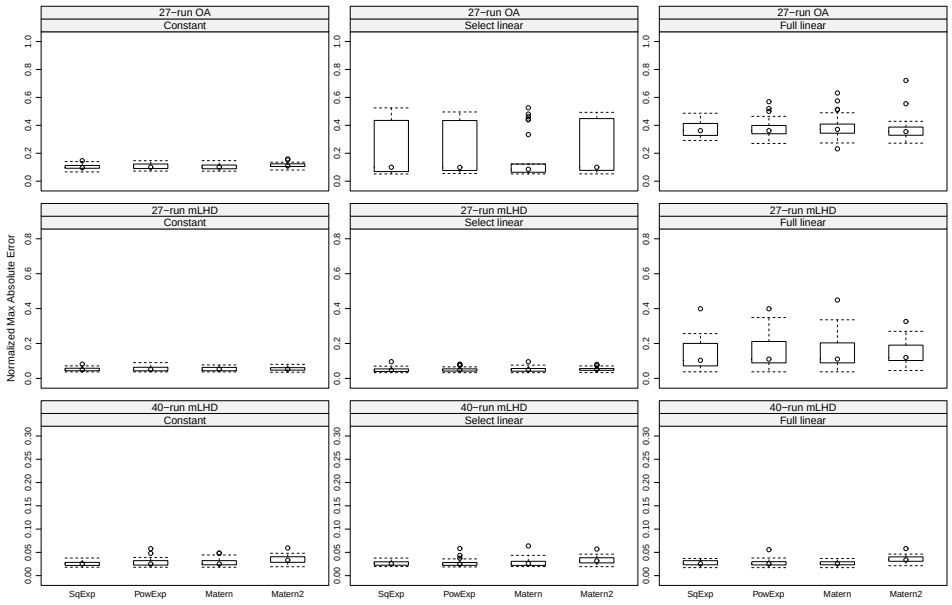

Figure 2.2: Borehole function: Normalized maximum absolute error of prediction, $e_{\max,\mathrm{ho}}$, for GaSP with all combinations of three regression models and four correlation functions. There are three base designs: a 27-run OA (top row); a 27-run mLHD (middle row); and a 40-run mLHD (bottom row). For each base design, 25 random permutations of its columns give the 25 values of $e_{\max,\ \mathrm{ho}}$ in a boxplot.

very poorly sometimes, but not always. In addition, the top row of Figure 2.1 also shows that the choice of correlation function is far less important than that of regression model.

The results for the 27-run mLHD in the middle row of Figure 2.1 show that design can have a large effect on accuracy: every analysis model performs better for the 27-run mLHD than for the 27-run OA. Note the vertical scale is different for each row of the figure. The SL regression now performs about the same as the constant regression. There is no substantial difference in accuracy between the correlation functions. Indeed, the impact on accuracy of using the space-filling mLHD design instead of an OA is much more important than differences due to choice of the correlation function.

Increasing the number of runs to a 40-run mLHD (bottom row of Figure 2.1) makes a further substantial improvement in prediction accuracy. All 12 modelling strategies give $e_{\text{rmse, ho}}$ values of about $0.02 - 0.06$ over the 25 repeat experiments. Although there is little systematic difference among strategies, the variation over equivalent designs is still striking in a relative sense. The figure for normalized maximum absolute error (Figure 2.2), which measures the worst case scenario tells the same story as Figure 2.1, and all results for normalized maximum absolute error will be reported in the appendix A.2 to save space from now on.

Stein (1988) showed that as long as the covariance function used to obtain the predictor is compatible with the actual covariance function, the obtained predictor will have the same asymptotic efficiency as the optimal predictor based on the actual covariance function. Stein (1988) defined that two covariance functions are compatible if the probability measures of two GP with equal mean functions and covariance functions are mutually absolutely continuous. However, the "true" covariance function is unknown in practice. In addition, the affordable experimental budget is usually not large for expensive computer models, i.e., n can not go to infinity. Hence, it is valuable to assess the performance of different correlation structures empirically when sample size is limited.

2.1.2 Gprotein Model

The second application, the G-protein model used in Loeppky et al. (2009) and described in the appendix, consists of a system of ODEs with 4-dimensional input.

Figure 2.3 shows $e_{\text{rmse,ho}}$ for the three regression models (here SL selects x_2, x_3 as inputs with large effects) and four correlation functions. The hold-out set comprises $10,000$ points generated from a random LHD. The results in the top row are for a 40-run mLHD. With d = 4, all 24 possible permutations of a single base design lead to 24 data sets and hence 24 $e_{\text{rmse,ho}}$ values. The boxplots in the top row have similar distributions across the 12 modelling strategies. As these empirical distributions have most $e_{\text{rmse,ho}}$ values above 0.1, we try increasing the sample size with an 80-run mLHD, which leads to a substantial decline in the normalized RMSE with all modelling methods giving $e_{\text{rmse,ho}}$ values of about 0.06 or less.

Thus, for the G-protein application, none of the three choices for $\boldsymbol{\beta}$ or the four choices for $\mathbf{R}$ matters. Again, the variation among equivalent designs is alarmingly large, dwarfing any impact of modelling strategy. The normalized maximum absolute error results are reported in Figure A.1 in the appendix.

2.1.3 PTW Model

Results for a third fast-to-run code, PTW (Preston et al., 2003), are reported in Figure 2.4 for the normalized RMSE. The model has 11 inputs. We took an mLHD with $n = 110$ as the base design. Prior information from engineers suggested incorporating linear x_1 and x_2 terms; SL also included x_3. No essential differences among $\boldsymbol{\beta}$ or $\mathbf{R}$ emerged, but again there is a wide variation over equivalent designs. The results of normalized maximum absolute error are in the appendix A.2.

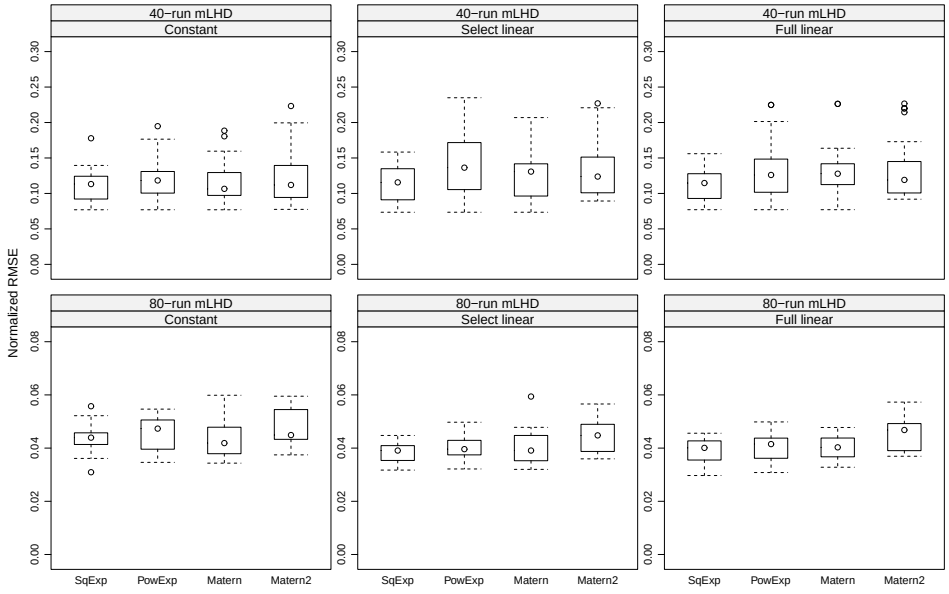

Figure 2.3: G-protein function: Normalized RMSE of prediction, $e_{\text{rmse,ho}}$, for GaSP with all combinations of three regression models and four correlation functions. There are three base designs: a 40-run mLHD (top row); and a 80-run mLHD (bottom row). For each base design, 24 random permutations of its columns give the 24 values of $e_{\text{rmse, ho}}$ in a boxplot.

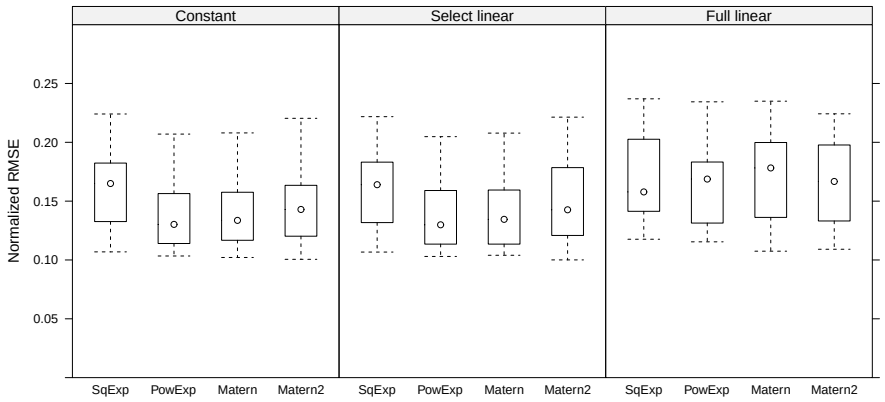

Figure 2.4: PTW function: Normalized RMSE of prediction, $e_{\text{rmse,ho}}$, for GaSP with all combinations of three regression models and four correlation functions. There is one base design: a 110 mLHD. 25 random permutations of its columns give the 25 values of $e_{\text{rmse, ho}}$ in a boxplot.

2.2 Slow Computer Models

For complex costly-to-run models, generating substantial holdout data or output from multiple designs is infeasible. Forced to depend on what data are at hand leads us to rely on cross-validation methods for generating multiple designs and holdout sets. Our approach in this section is to delete a subset from the full data set, use the remaining data to produce a predictor, and calculate the normalized RMSE and normalized maximum absolute error from predicting the output in the deleted (holdout) subset. Repeating this for a number (25 is what we use) of subsets gives some measure of variability and accuracy. In effect, we create 25 designs and corresponding holdout sets from a single data set and compare consequences arising from different choices for predictors. The details described in the applications below differ somewhat depending on the particular application. We report three slow computer models in this section.

2.2.1 Nilson-Kuusk Model

An ecological code modelling reflectance for a plant canopy developed by Nilson and Kuusk (1989) was used by Bastos and O'Hagan (2009) to illustrate diagnostics for GaSP models. With 5-dimensional input, two computer experiments were performed: the first using a 150-run random LHD and the second with an independently chosen LHD of 100 points.

We carry out three studies based on the same data. The first treats the 100-point LHD as the experiment and the 150-point set as a holdout sample. The second study reverses the roles of the two LHDs. A third study, extending one done by Bastos and O'Hagan (2009), takes the 150-run LHD, augments it with a random sample of 50 points from the 100-point LHD, takes the resulting 200-point subset as the experimental design for training the statistical model, and uses the remaining 50 points from the 100-run LHD to form the holdout set in the calculation of $e_{\text{rmse,ho}}$. By repeating the sampling of the 50 points 25 times we get 25 replicate experiments each with the same base 150 runs but differing with respect to the additional 50 training points and the holdout set.

Plotting y against x_5 shows an obvious non linear trend. Therefore, in addition to the linear regression choices we have studied so far, we also incorporate a regression model identified by Bastos and O'Hagan (2009): an intercept, linear terms in the inputs $x_1, \ldots, x_4$, and a quartic polynomial in x_5. We label this model "Quartic". All analyses are carried out with the output y on a log scale. The rationale for a log transformation is based on standard diagnostics for GaSP models (Jones et al., 1998). The details are omitted here. Results on $e_{\text{rmse, ho}}$ are reported in Tables 2.1, 2.2 and Figure 2.5. Results on $e_{\text{max, ho}}$ are reported in Tables A.3, A.4 and Figure A.3 in the appendix.

Table 2.1 summarizes the results of the study with the 100-point LHD as training data and the 150-point set as a holdout sample. It shows the choice for $\boldsymbol{\beta}$ is immaterial: the constant mean is as good as any. For the correlation function, SqExp is inferior to the other choices, there is some evidence that Matérn is preferred to Matérn-2, and there is little difference

	$e_{\text{rmse, ho}}$			
Regression	Correlation function			
Model	SqExp	PowExp	Matern2	Matern
Constant	0.116	0.099	0.106	0.102
Select linear	0.115	0.099	0.106	0.105
Full linear	0.110	0.099	0.104	0.104
Quartic	0.118	0.103	0.107	0.106

Table 2.1: Nilson-Kuusk model: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$, for four regression models and four correlation functions.The experimental data are from a 100-run LHD, and the hold-out set is from a 150-run LHD.

	$e_{\text{rmse, ho}}$			
Regression	Correlation function			
Model	SqExp	PowExp	Matern2	Matern
Constant	0.111	0.080	0.087	0.078
Select linear	0.113	0.080	0.091	0.079
Full linear	0.109	0.079	0.090	0.076
Quartic	0.104	0.079	0.088	0.078

Table 2.2: Nilson-Kuusk model: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$, for four regression models and four correlation functions. The experimental data are from a 150-run LHD, and the hold-out set is from a 100-run LHD.

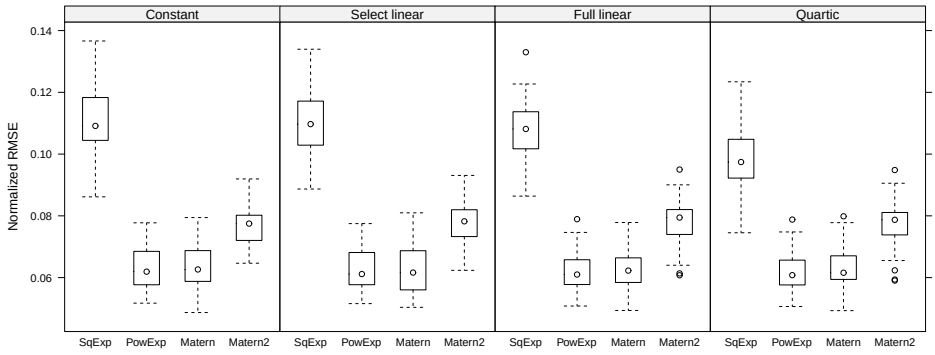

Figure 2.5: Nilson-Kuusk model: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$, for four regression models and four correlation functions. Twenty-five designs are created from a 150-run LHD base plus 50 random points from a 100-run LHD. The remaining 50 points in the 100-run LHD form the holdout set for each repeat.

between PowExp and Matérn, the best performers. Similar results pertain when the 150-run LHD is used for training and the 100-run set for testing in Table 2.2. The results for the normalized maximum absolute errors convey a similar message.

The boxplots in Figures 2.5 for the third study are even more striking in exhibiting the inferiority of $\mathbf{R} =$ SqExp and the lack of advantages for any of the non-constant regression functions. The large variability in performance among designs and holdout sets is similar to that seen for the fast code replicate experiments of section 2.1. The perturbations of the experiment, from random sampling here, appear to provide a useful reference set for studying the behaviour of model choices.

The large differences in prediction accuracy among the correlation functions, not seen in section 2.1, deserves some attention. An overly smooth correlation function — the SqExp — does not perform as well as the Matérn and PowExp functions here. The latter two have the flexibility to allow needed rougher realizations. With the 150-run design and the constant regression model, for instance, the maximum of the log likelihood increases

by about 50 when the power exponential is used instead of the squared exponential, with four of the α_j in (2.2) taking values less than 2.

2.2.2 Volcano Model

A computer model studied by Bayarri et al. (2009) models the process of pyroclastic flow (a fast-moving current of hot gas and rock) from a volcanic eruption. The inputs varied are: initial volume, x_1, and direction, x_2, of the eruption. The output, y, is the maximum (over time) height of the flow at a location. The unit of measurement of y is meter. A 32-run data set provided by Elaine Spiller (different from that reported by Bayarri et al. (2009) but a similar application) is available. Plotting the data shows the output has a strong trend in x_1, and putting a linear term in the GaSP surrogate, as modelled by Bayarri et al. (2009), is natural. But is it necessary?

The nature of the data suggests a transformation of y could be useful. The one used by Bayarri et al. (2009) is $\log(y + 1)$,i.e., 1 meter is added to avoid zero before a log transformation is taken. Diagnostic plots (Jones et al., 1998) from using $\boldsymbol{\beta}$ = Const and $\mathbf{R}$ = SqExp shows that the log transform is reasonable, but a square-root transformation is better still. We report analyses for both transformations.

The regression functions considered are constant, select linear ($\beta_0 + \beta_1 x_1$), full linear, and quadratic ($\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2$), because the trend in x_1 appears stronger than linear when looking at main effects from the surrogate obtained using $\sqrt{y}$ and GaSP(Const, PowExp).

Analogous to the approach in the Nilson-Kuusk example, repeat experiments are generated by random sampling of 25 runs from the 32 available to comprise the design for model fitting. The remaining 7 runs form the holdout set. This is repeated 25 times, giving 25 $e_{\text{ermse, ho}}$ values in the boxplots of Figure 2.6. The $e_{\text{max, ho}}$ values are reported in Figure A.4 in the appendix A.2. The conclusions are much like those in the Nilson-Kuusk example: there is no need to go beyond $\boldsymbol{\beta}$ = Const, and PowExp is preferred to SqExp. The failure of SqExp in the two "slow" examples considered thus far is surprising in light of commonly held beliefs that the SqExp structure

is adequate.

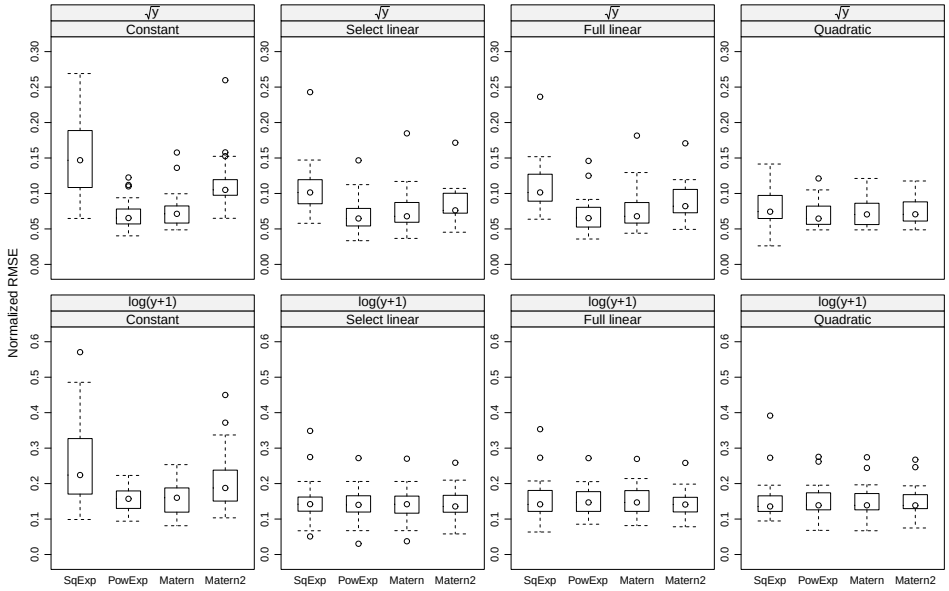

Figure 2.6: Volcano model: Normalized holdout RMSE, $e_{\text{rmse, ho}}$, for four regression models and four correlation functions. Two transformations are considered: $\log(y)$ and $\sqrt{y}$.

2.2.3 Sea Ice Model

The arctic sea-ice model studied in Chapman et al. (1994) and in Loeppky et al. (2009) has 13 inputs, 4 outputs, and 157 available runs. The previous studies found modest prediction accuracy of GaSP(Const, PowExp) surrogates for two of the outputs (ice mass and ice area) and poor accuracy for the other two (ice velocity and ice range). The question arises whether use of linear regression terms can increase accuracy to acceptable levels. We randomly select 130 points to form a training set and the remaining 27 points is the hold-out set. The whole process is repeated 25 times, which leads to the normalized RMSE results in Figure 2.7 and the normalized maximum absolute error in Figure A.5. The answer to that question is no: there is no

help from $\boldsymbol{\beta} =$ SL or FL, nor from changing $\mathbf{R}$. Indeed, FL makes accuracy much worse sometimes.

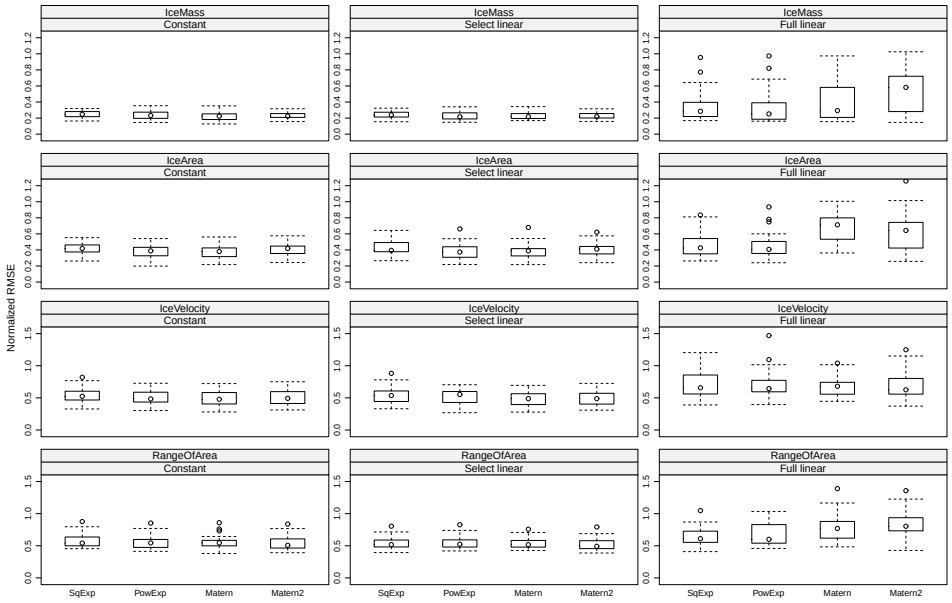

Figure 2.7: Seaice model: Normalized holdout RMSE, $e_{\text{rmse, ho}}$, for three regression models and four correlation functions. The outputs are: IceMass, IceArea, IceVelocity and RangeOfArea, which are modelled independently.

2.3 Other Modelling Strategies

It is clear that we have not studied all possible paths to GaSP modelling that one might take in a computer experiment. In this section we address three others in the same fashion described above.

2.3.1 Full Bayes

A number of full Bayesian approaches have been employed in the literature. They go beyond the statistical formulation using a GP as a prior on the class of functions and assign prior distributions to all parameters,

particularly those of the correlation function. For illustration, we consider a Bayesian implementation by Kennedy (2004), labelled as SqExp-Full (K) method. The method is based on the SqExp correlation structure and the details of the priors it assumes can be found in section 3.1 of chapter 3.

For the borehole application, 25 repeat experiments are constructed for three designs, as in section 2.1. The boxplots of $e_{\text{rmse,ho}}$ in Figure 2.8 compare SqExp-Full (K) with the SqExp and PowExp structures in section 2.1 based on MLE of all parameters. (The method CGP and its boxplot will be discussed in section 2.3.2 later.) Unless stated otherwise, the Const is used as the regression term. SqExp-Full (K) is less accurate than either GaSP(Const, SqExp) or GaSP(Const, PowerExp). The boxplots of $e_{\text{max, ho}}$ are similar to Figure 2.8 and are reported in the appendix (Figure A.6).

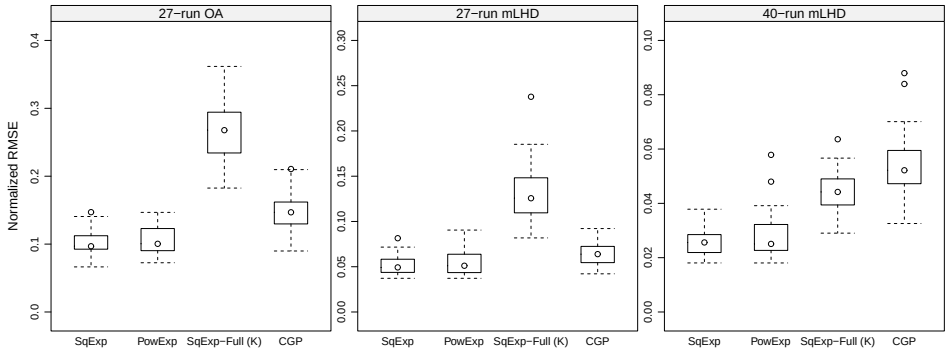

Figure 2.8: Borehole function: Normalized holdout RMSE of prediction, $e_{\text{rmse, ho}}$, for GaSP(Const, SqExp), GaSP(Const, PowerExp), SqExp-Full (K), and CGP. There are three base designs: a 27-run OA (left), a 27-run mLHD (middle); and a 40-run mLHD (right). For each base design, 25 random permutations of its columns give the 25 values of $e_{\text{rmse, ho}}$ in a boxplot.

Figure 2.9 similarly depicts results for the Gprotein model. The boxplots of $e_{\text{max, ho}}$ are similar to Figure 2.9 and are reported in the appendix (Figure A.7). With the 40-run mLHD, the fully Bayesian and empirical Bayes methods all perform about the same, giving only fair prediction accu-

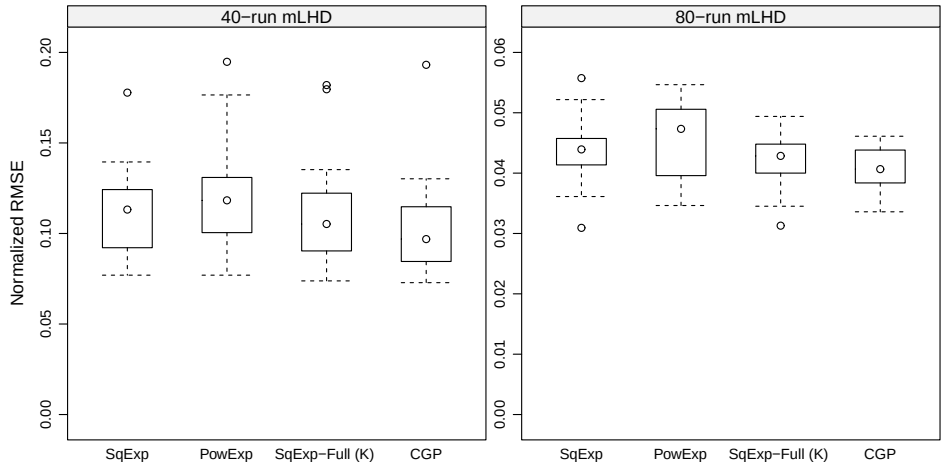

Figure 2.9: G-protein: Normalized holdout RMSE of prediction, $e_{\text{rmse, ho}}$, for GaSP(Const, SqExp), GaSP(Const, PowerExp), SqExp-Full (K), and CGP. There are two base designs: a 40-run mLHD (left); and an 80-run mLHD (right). For each base design, all 24 permutations of its columns give the 24 values of $e_{\text{rmse, ho}}$ in a boxplot.

racy. Increasing n to 80 improves accuracy considerably for all methods (the scales of the two plots are very different), far outweighing any systematic differences between their accuracies.

As we have observed so far, SqExp-Full (K) performs as well as the GaSP methods for Gprotein, not so well for Borehole with $n = 27$ but adequately for $n = 40$. Turning to the slow codes in section 2.2 a different message emerges. Figure 2.10 for the Nilson-Kuusk model is based on 25 repeat designs constructed as for Figure 2.5 with a base design of 150 runs plus 50 randomly chosen from 100. The distributions of $e_{\text{rmse,ho}}$ for SqExp-Full (K) and GaSP(Const, SqExp) are similar, with GaSP(Const, PowExp) showing a clear advantage. Moreover, few of the Bayes $e_{\text{rmse,ho}}$ values meet the 0.10 threshold, while all the GaSP(Const, PowExp) $e_{\text{rmse,ho}}$ values do. The boxplots of $e_{\text{max, ho}}$ are reported in the appendix (Figure A.8) and are similar to Figure 2.10. SqExp-Full (K)) uses the SqExp correlation function,

which performed relatively poorly in section 2.2. The disadvantage carries over to the Bayesian method here.

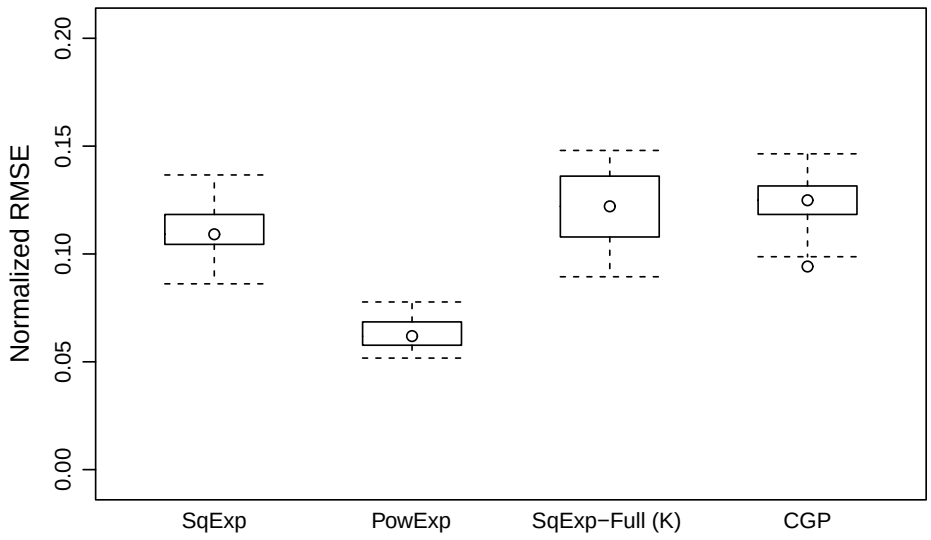

Figure 2.10: Nilson-Kuusk model: Normalized holdout RMSE of prediction, $e_{\text{rmse, ho}}$, for GaSP(Const, SqExp), GaSP(Const, PowerExp), SqExp-Full (K), and CGP.

The results in Figure 2.11 for the volcano model are for the 25 repeat experiments described in section 2.2. Here again GaSP(Const, PowerExp) dominates Bayes and for the same reasons as for the Nilson-Kuusk model. For the $\sqrt{y}$ transformation, all but a few GaSP(Const, PowerExp) $e_{\text{rmse,ho}}$ values meet the 0.10 threshold, in contrast to Bayes where all but a few do not. (Note also that error according to the $e_{\text{rmse,ho}}$ criterion is much smaller on the $\sqrt{y}$ scale, supporting its choice.) The results of $e_{\text{max,ho}}$ are reported in the appendix (Figure A.9).

These results are striking and suggest that Bayes methods relying on **R** = SqExp need extension to have the needed roughness determined by the data. The hybrid Bayes-MLE approach employed by Bayarri et al. (2009)

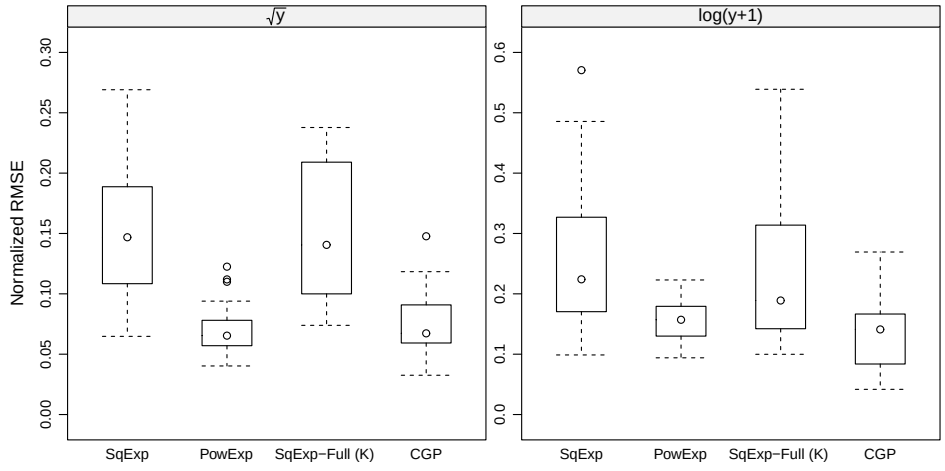

Figure 2.11: Volcano model: Normalized holdout RMSE of prediction, $e_{\text{rmse, ho}}$, for GaSP(Const, SqExp), GaSP(Const, PowerExp), SqExp-Full (K), and CGP.

estimates the correlation parameters in PowExp by MLE, fixes them, and takes objective priors for $\boldsymbol{\beta}$ and σ^2. The mean of the predictive distribution for a holdout output value gives the same prediction as GaSP(Const, PowExp). Whether this is a good remedy requires further exploration. We will discuss it in the next chapter.

2.3.2 Non-stationarity

The use of stationary GPs as priors in the face of "non-stationary appearing" functions has attracted a measure of concern despite the fact that L_2-differentiable functions can be approximated using PowExp with enough data. Of course there never are enough data. A relevant question is whether other priors, even stationary ones different from those in sections 2.1 and 2.2, are better reflective of conditions and lead to more accurate predictors.

Ba and Joseph (2012) advanced a "composite" GP (CGP) approach. These authors used two GPs, both with SqExp correlation. The first has

correlation parameters θ_j in (2.1) constrained to be small for gradually varying longer-range trend, while the second has larger values of θ_j for shorter-range behaviour. The second, local GP also has a variance that depends on $\mathbf{x}$, primarily as a way to cope with apparent non-stationary behaviour. Does this composite approach offer an effective improvement to the simpler choices of sections 2.1 and 2.2?

We can apply CGP via its R library to the examples studied in sections 2.1 and 2.2, much as was just done for SqExp-Full (K). The comparisons in Figure 2.8 for the borehole function show that GaSP and CGP have similar accuracy for the two 27-run designs. GaSP has smaller error than CGP for the 40-run mLHD, though both methods achieve acceptable accuracy. The results in Figure 2.9 for G-protein show little practical difference between any of the methods, including CGP. For these two fast-code examples, there is negligible difference between CGP and the GaSP methods. For the models of section 2.2, however, conclusions are somewhat different. GaSP(Const, PowerExp) is clearly much more accurate than CGP for the Nilson-Kuusk model (Figure 2.10) and roughly equivalent for the volcano model (Figure 2.11). All in all, we conclude that no evidence for the effectiveness of a composite GaSP approach. These findings are in accordance with the earlier study by West et al. (1995).

2.3.3 Adding a Nugget Term

A nugget augments the GaSP model in (1.2.1) with an uncorrelated ϵ term, usually assumed to have a normal distribution with mean zero and constant variance σ_ϵ^2, independent of the correlated process $Z(\mathbf{x})$. This changes the computation of $\mathbf{R}$ and $\mathbf{r}(\mathbf{x})$ in the conditional prediction (1.3), which no longer interpolates the training data. A nugget term has been widely used for statistical modelling of deterministic computer codes without random error. The reasons offered are that numerical stability is improved, so overcoming computational obstacles, and a nugget can produce better predictive performance or better confidence or credibility intervals. The evidence — in the literature and presented here — suggests, however, that

for deterministic functions the potential advantages of a nugget term are modest. More systematic methods are available to deal with numerical instability if it arises (Ranjan et al., 2011), adding a nugget does not convert a poor predictor into an acceptable one, and other factors may be more important for good statistical properties of intervals (section 2.4). On the other hand, we also do not find that adding a nugget (and estimating it along with the other parameters) is harmful, though it may produce smoothers rather than interpolators. We now elaborate on these points.

First of all, a small nugget is often included to improve the numerical properties of the correlation structure, $\mathbf{R}$. However, for the space-filling designs used in sections 2.1 and 2.2, Ranjan et al. (2011) showed that ill-conditioning in a no-nugget GaSP will only occur for low-dimensional $\mathbf{x}$, high correlation, and large n. These conditions are not commonly met in initial designs for applications. For instance, none of the computations in this chapter failed due to ill-conditioning, and those computations involved many repetitions of experiments for the various functions and GaSP models. In addition, when a design is not space-filling, matrix ill-conditioning may indeed occur. For instance, if the sequential design proposed by Bingham et al. (2014) to approximate a contour or to optimize a function is used, the newly added point may get close to the existing points, $\mathbf{x}$, causing numerical problem. If ill-conditioning does occur, however, the mathematical solution by Ranjan et al. (2011) is an alternative to adding a nugget.

Secondly, a nugget term is also sometimes suggested to improve predictive performance. Andrianakis and Challenor (2012) showed mathematically, however, that with a nugget the RMSE of prediction can be as large as that of a least squares fit of just the regression component in (1.2.1). Our empirical findings, choosing the size of σ_ϵ^2 via its MLE, are similarly unsupportive of a nugget. We repeat the calculations leading to Figure 2.1 for the borehole function, but fitting a nugget term in all models, shows essentially no difference. The results are reported in Figure 2.12 for normalized RMSE. The results for maximum absolute error are reported in Figure A.10. The MLE of σ_ϵ^2 is either zero or very small relative to the variance of the correlated process. These findings are consistent with those of Ranjan et al.

(2011), who found for the borehole function and other applications that constraining the model fit to have at least a modest value of σ_ϵ^2 deteriorated predictive performance.

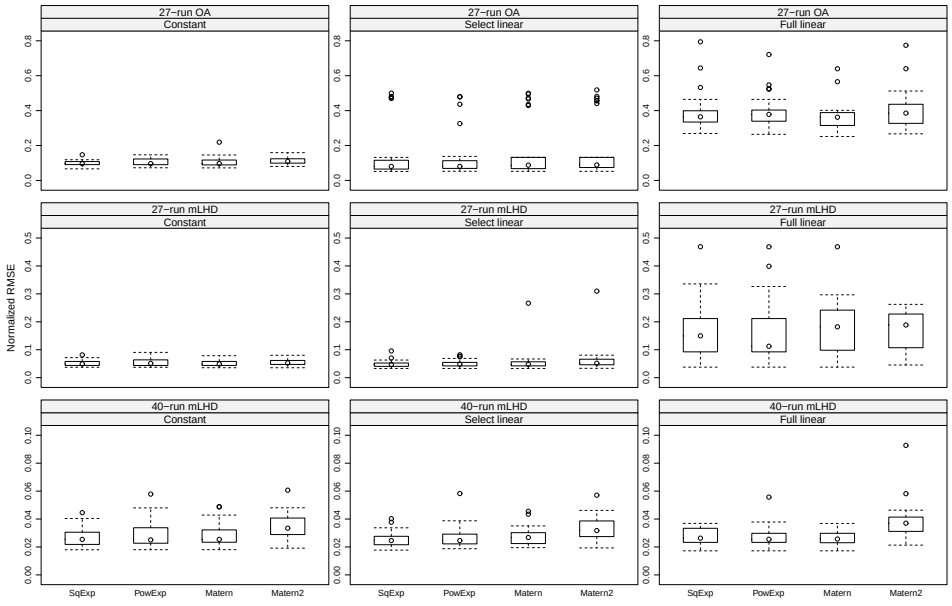

Figure 2.12: Borehole model: Normalized RMSE of prediction, $e_{\text{rmse,ho}}$, for GaSP adding a nugget estimated by its MLEs, and with all combinations of three regression models and four correlation functions. There are three base designs: a 27-run OA (top row); a 27-run mLHD (middle row); and a 40-run mLHD (bottom row). For each base design, 25 random permutations of its columns give the 25 values of $e_{\text{rmse, ho}}$ in a boxplot.

Another example we consider is the Friedman function

$$y(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5, \tag{2.6}$$

with $n = 25$ runs, was used by Gramacy and Lee (2012) to show potential advantages of including a nugget term. However, their context —performance criteria, analysis method and design — differs in all respects from ours. Our results in the top row of Figure 2.13 show that the GaSP(Const, SqExp) and GaSP(Const, PowExp) models with $n = 25$ have highly variable accuracy,

with $e_{\text{rmse, ho}}$ values no better and often much worse than 20%. The results for maximum absolute errors are reported in Figure A.11 and are consistent to Figure 2.13. The effect of the nugget is inconsequential. Increasing the sample size to $n = 50$ makes a dramatic improvement in prediction accuracy, but the effect of a nugget remains negligible.

2.4 Comments

2.4.1 Uncertainty of Prediction

As noted in sections 2.1 and 2.2, our attention is directed at prediction accuracy, the most compelling characteristic in practical settings. For example, where the objective is calibration and validation, the details of uncertainty, as distinct from accuracy, in the emulator of the computer model are absorbed (and usually swamped) by model uncertainties and measurement errors (Bayarri et al., 2007). But for specific predictions it is clearly important to have valid uncertainty statements. Currently, a full assessment of the validity of emulator uncertainty quantification is unavailable. But, it has long been recognized that the standard error of prediction can be optimistic when MLEs of the parameters in the correlation functions of sections 2.1 and 2.2 are "plugged-in" because the uncertainty in the parameter values is not taken into account and that uncertainty is non-trivial (Abt, 1999).

Bayes credible intervals with full Bayesian methods carry explicit and valid uncertainty statements; hybrid methods using priors on some of the correlation parameters (as distinct from MLEs) may also have reliable credible intervals. But for properties such as actual coverage probability (ACP), the proportion of points in a test set with true response values covered by intervals of nominal (say) 95% confidence or credibility, the behaviour is far from clear. Chen (2013) compared several Bayes methods with respect to coverage. The results showed variability with respect to equivalent designs like that found above for accuracy, a troubling characteristic pointing to considerable uncertainty about the uncertainty.

In Figure 2.14 we see some of the issues. It gives ACP results for the

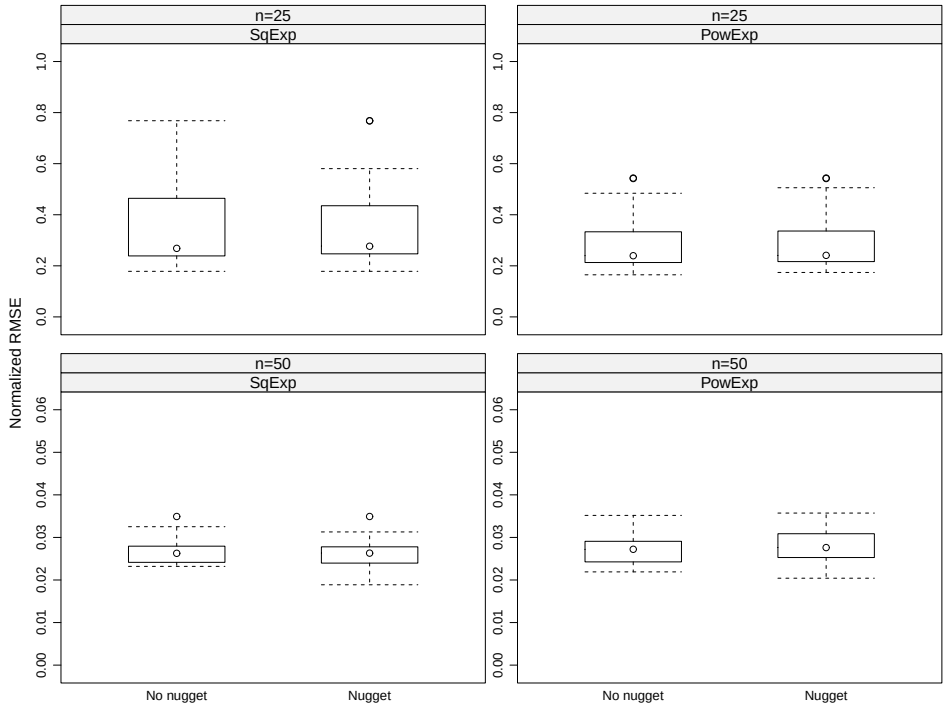

Figure 2.13: Friedman function: Normalized RMSE of prediction, $e_{\text{rmse,ho}}$, for GaSP(Const, SqExp) and GaSP(Const, PowExp) models with no nugget term versus the same models with a nugget. There are two base designs: a 25-run mLHD (top row); and a 50-run mLHD (bottom row). For each base design, 25 random permutations between columns give the 25 values of $e_{\text{rmse, ho}}$ in a boxplot.

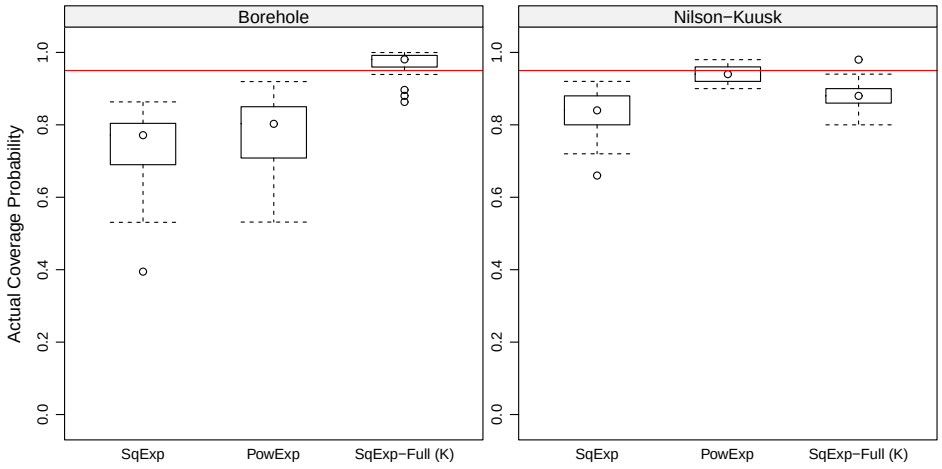

Figure 2.14: Borehole and NilsonKuusk functions: ACP of nominal 95% confidence or credibility intervals for GaSP(Const, SqExp), GaSP(Const, PowExp), and SqExp-Full (K). For the borehole function, 25 random permutations between columns of a 40-run mLHD give the 25 values of ACP in a boxplot. For the NilsonKuusk function, 25 designs are created from a 150-run LHD base plus 50 random points from a 100-run LHD. The remaining 50 points in the 100-run LHD form the holdout set for each repeat.

borehole and NilsonKuusk models. The left-hand plot for borehole displays the anticipated under-coverage using plug-in estimates for the correlation parameters. (Confidence intervals here use $n1$ rather than n in the estimate of σ in the standard error and t_{n-1} instead of the standard normal.) PowExp is slightly better than SqExp, and SqExp-Full (K) has ACP values close to the nominal 95%. Surprisingly, the plot for the NilsonKuusk model on the right of Figure 2.14 paints a different picture. Plug-in with SqExp and SqExp-Full (K) both show under-coverage, while plug-in PowExp has near ideal properties here. We speculate that the use of the SqExp correlation function by SqExp-Full (K) is again suboptimal for the NilsonKuusk application, just as it was for prediction accuracy.

We also compare models with and without a nugget in terms of coverage properties for the Friedman function in (2.6). The results in Figure 2.15

show that the problem of substantial undercoverage seen in many of the replicate experiments is not solved by inclusion of a nugget term. A modest improvement in the distribution of ACP values is seen, particularly for $n = 50$.

2.4.2 Designs

The variability in performance over equivalent designs is a striking phenomenon in the analyses in previous sections and raises questions about how to cope with what seems to be unavoidable bad luck. Are there sequential strategies that can reduce the variability? Are there advantageous design types, more robust to arbitrary symmetries. For example, does it matter whether a random LHD, mLHD, or an orthogonal LHD is used? The latter question will be addressed in detail in chapter 4. That design has a strong role is both unsurprising and surprising. It is not surprising that care must be taken in planning an experiment; it is surprising and perplexing that equivalent designs can lead to such large differences in performance that are not mediated by good analytic procedures.

2.4.3 Larger Sample Sizes

As noted in sections 2.1 and 2.2, our attention is on experiments where n is small or modest at most. With advances in computing power it becomes more feasible to mount experiments with larger values of n while, at the same time, more complex codes become feasible but only with limited n. Our focus continues to be on the latter and the utility of GaSP models in that context.

As n gets larger, Figure 2.1 illustrates that the differences in accuracy among choices of $\mathbf{R}$ and $\boldsymbol{\beta}$ begin to vanish. Indeed, it is not even clear that using GaSP models for large n is useful; standard function fitting methods such as splines may well be competitive and easier to compute. In addition, when n is large nonstationary behaviour can become apparent and encourages variations in the GaSP methodology such as decomposing the input space (as in Gramacy and Lee (2008)) or by using a complex $\boldsymbol{\beta}$ together

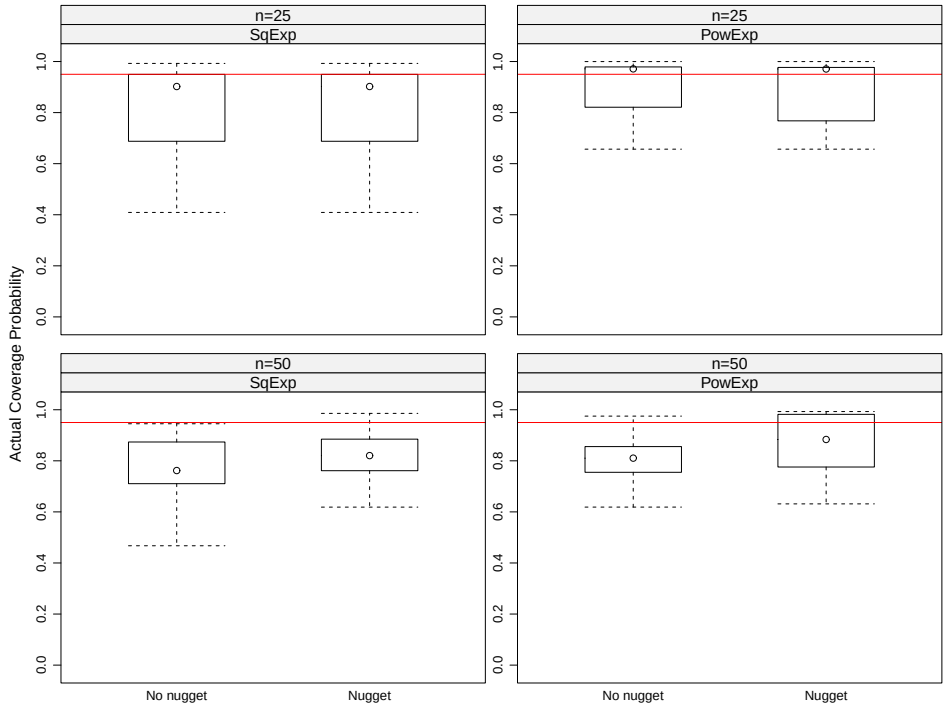

Figure 2.15: Friedman function: ACP of nominal 95% confidence intervals, for GaSP(Const, SqExp) and GaSP(Const, PowExp) models with no nugget term versus the same models with a nugget. There are two base designs: a 25-run mLHD (top row); and a 50-run mLHD (bottom row). For each base design, 25 random permutations between columns give the 25 values of ACP in a boxplot.

with a computationally more tractable $\mathbf{R}$ (as in Kaufman et al. (2011)). Comparison of alternatives when n is large is yet to be considered.

2.5 Conclusions and Recommendations

We have stressed the importance of going beyond "anecdotes" in making claims for proposed methods. While this point is neither novel nor startling, it is one that is commonly ignored, often because the process of studying consequences under multiple conditions is more laborious. The borehole example (Figure 2.1), for instance, employs 75 experiments arising from 25 repeats of each of 3 base experiments.

The studies in the previous sections lead to the following conclusions:

- There is no evidence that GaSP(Const, PowExp) is ever dominated by use of regression terms, or other choices of $\mathbf{R}$. Moreover, we have found that the inclusion of regression terms makes the likelihood surface multi-modal, necessitating an increase in computational effort for empirical Bayes or full Bayesian methods. This appears to be due to confounding between regression terms and the GP paths. In addition, GaSP(Const, PowExp) is the method that we recommend when faced with a particular model and a set of runs and the primary goal is to make predictions for untried sets of inputs.

- Choosing $\mathbf{R}$ = SqExp, though common, can be unwise. The Matérn function optimized over a few levels of smoothness is a reasonable alternative to PowExp.

- Design matters but cannot be controlled completely. Variability of performance from equivalent designs can be uncomfortably large.

There is not enough evidence to settle the following questions:

- Are full Bayes methods ever more accurate than GaSP(Const, PowExp)? Full Bayes methods relying on $\mathbf{R}$ = SqExp were seen to be sometimes inferior, and extensions to accommodate less smooth $\mathbf{R}$

such as PowExp are needed. This will be addressed in the next chapter.

- Are composite GaSP methods ever better than GaSP(Const, PowExp) in practical settings where the output exhibits nonstationary behaviour? This is still an open question.

Chapter 3

Flexible Correlation Structures in Bayesian Gaussian Process

This chapter is a follow-up of chapter 2, where we observed the prediction accuracy of a SqExp structure based GP model can be worse than that of a GP model with a more flexible structure, such as the PowExp and the Matérn in section 2.2. In practice, the GP parameters, $\boldsymbol{\beta}$, σ^2, and the correlation parameters, are unknown, and hence need to be estimated using the available data. Well-known estimation methods can be broadly classified into two categories based on the underlying paradigm. The first is the empirical Bayes method, where MLEs are used; see, for example, the algorithm of Welch et al. (1992). The second category is Bayesian posterior inference, such as that proposed by Higdon et al. (2008) and the treed GP method used by Gramacy and Lee (2008). The Bayesian methods, in principle, take account of parameter-estimation uncertainty and produce better coverage probabilities for credible intervals. However, in chapter 2, we noted that this is not always true when a GP with a squared-exponential correlation function is used. In this chapter, we propose new Bayesian methods with more flexible, power-exponential or Matérn, correlation structures.

The focus here is quantifying the uncertainty from the statistical emulator of the computer model, including the contribution from parameter estimation. There are other sources of uncertainty, such as the systematic discrepancy between the computer model and physical measurements of the system (Kennedy and O'Hagan, 2001), not explicitly considered here. The pro-

posed methods have relevance, however, because modelling the computer-model data is an important component of analyses addressing other objectives, such as optimization and contour estimation. In addition, in this chapter, we consider the Const regression component only. Therefore, $\boldsymbol{\beta}$ reduces to a scalar μ.

The rest of the chapter is organized as follows. We first review two full Bayes methods in section 3.1. An example motivates new Bayesian methods is analyzed in section 3.2. In section 3.3, we propose Bayesian methods with power-exponential or Matérn correlation structure. They are evaluated in section 3.4 through application examples and a simulation study. Some concluding remarks are made in section 3.5.

3.1 Review of Two Full Bayes Methods

Bayes with a squared exponential correlation (version K)

The first fully Bayes method is by Kennedy (2004), and hence we call it SqExp-Full (K). It has the following independent prior distributions on the GP parameters:

- An improper uniform distribution (normal with infinite variance) on μ;
- A Jeffreys prior on σ^2; i.e., $\pi(\sigma^2) \propto 1/\sigma^2$; and
- An exponential distribution with rate 0.1 on each sensitivity parameter θ_j.

By integrating out μ and σ^2, Handcock and Stein (1993) gave the marginal posterior distribution of the SqExp correlation parameters $\boldsymbol{\psi}$ as

$$\begin{aligned} \pi(\boldsymbol{\psi}|\mathbf{y}) &\propto \int\int \pi(\mu)\pi(\sigma^2)\pi(\boldsymbol{\psi})L(\mathbf{y}|\mu,\sigma^2,\boldsymbol{\psi})\,d\mu\,d\sigma^2 \\ &\propto \frac{\prod_{i=1}^{d}\pi(\theta_j)}{\left(\hat{\sigma}^2_{\boldsymbol{\psi}}\right)^{\frac{n-1}{2}}\det^{1/2}(\mathbf{R})\det^{1/2}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})}, \end{aligned} \tag{3.1}$$

where $\pi(\theta_j)$ is the prior distribution of θ_j, $\hat{\mu}$ is as in (1.6) and $\hat{\sigma}^2_{\boldsymbol{\psi}}$ is given by

$$\hat{\sigma}^2_{\boldsymbol{\psi}} = \frac{1}{n-1}(\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}). \tag{3.2}$$

Except for a change in degrees of freedom, $\hat{\sigma}^2_{\boldsymbol{\psi}}$ in (3.2) is the same as $\hat{\sigma}^2$ in (1.7). The Metropolis-Hastings algorithm is then applied to sample from the posterior distribution of θ_j. The predictive distribution conditional on $\boldsymbol{\psi}$ is a non-central t distribution with $n-1$ degrees of freedom (Kennedy, 2004). That is,

$$y(\mathbf{x}^*|\boldsymbol{\psi}, \mathbf{y}) \sim t(\hat{m}_{\boldsymbol{\psi}}(\mathbf{x}^*), \hat{v}_{\psi}(\mathbf{x}^*)), \tag{3.3}$$

where $\hat{m}_{\boldsymbol{\psi}}(\mathbf{x}^*)$ and $\hat{v}_{\boldsymbol{\psi}}(\mathbf{x}^*)$ were given in (1.8) and (1.9).

Bayes with a squared-exponential correlation (version H)

A second full Bayes implementation based on the SqExp correlation function was described by Higdon et al. (2008), and we call it SqExp-Full (H). The independent prior distributions on the GP parameters are now:

- An improper uniform distribution (normal with infinite variance) on μ;
- An inverse-gamma distribution $\text{IG}(\phi_1, \phi_2)$ on σ^2 with shape parameter $\phi_1 = 1$ and scale $\phi_2 = 0.0001$; and
- A $\text{beta}(1, 0.1)$ distribution on $\rho_i = \exp(-\theta_j/4)$.

A beta prior on ρ_j is equivalent to a log-beta distribution on θ_j. The parameters μ and σ^2 can be marginalized out of the posterior (Higdon et al., 2008), and a Markov chain Monte Carlo (MCMC) algorithm is applied to sample the ρ_j. The predictive distribution at an untried input vector is the same as (3.3) but with $n - 1 + 2\phi_1$ as the degrees of freedom in the denominator of $\hat{\sigma}^2_{\boldsymbol{\psi}}$ in (3.2) and in the non-central t distribution in (3.3).

Chen (2013) showed that the different priors on σ^2 in SqExp-Full (K) and SqExp-Full (H) do not substantially affect predictive properties, but the priors for θ_j are an important distinction. Besides these two fully Bayes

methods, there exist several other fully Bayes implementations, for example, the treed GP (TGP) method (Gramacy and Lee, 2008). The aim of the chapter, however, is not to compare existing estimation methods; a comprehensive comparison of Bayesian methods can be found in Chen (2013). Rather, the purpose is to introduce Bayesian implementations with flexible correlation structure.

3.2 A Motivating Example: Nilson-Kuusk Model

Abt (1999) noted that the extra uncertainty from the use of empirical Bayes plug-in MLEs of the correlation parameters can be non-trivial. In principle, a Bayesian method should quantify parameter-estimation uncertainty and thus produce better coverage probabilities of credible intervals. We show in this section, however, that empirical Bayes versus fully Bayes may be less important for uncertainty quantification than the correlation-function family.

We consider an ecological computer code that models reflectance for a plant canopy. It was developed by Nilson and Kuusk (Nilson and Kuusk, 1989) and analyzed by Bastos and O'Hagan (2009) to illustrate diagnostics for GP emulators. Two computer experiments are available for this code with $d = 5$ inputs: the first is a 150-point LHD and the second is an independent 100-point random LHD. We randomly sample 100 points from the 150-point set as the training set to train a GP model. The remaining 50 points are added to the 100-point set to form a 150-point holdout set. Twenty-five different random samples are used so that each method considered below will produce 25 repeated experiments and 25 sets of results. This approach is similar to that of chapter 2, in which we noted that the variation in results from one design to another can be considerable. Hence, throughout the chapter we report results from multiple designs, either by random sampling or by permuting the columns of a base design.

As recommended in chapter 2, a GP with the Const regression term is chosen as the statistical model to emulate the NK model. We consider the following four methods, which have different combinations of correlation

structures and inference paradigm:

- SqExp-Emp (further abbreviated as Sq-Emp in figures), i.e., SqExp correlation and empirical Bayes;
- SqExp-Full (K) (further abbreviated as Sq-Full (K) in figures), i.e., SqExp correlation and full Bayes with priors from Kennedy (2004), reviewed in section 3.1 ;
- SqExp-Full (H) (further abbreviated as Sq-Full (H) in figures), i.e., SqExp correlation and full Bayes with priors from Higdon et al. (2008), reviewed in section 3.1; and
- PowExp-Emp (further abbreviated as Pow-Emp in figures), i.e., Pow-Exp correlation and empirical Bayes.

To measure the prediction accuracy, we use normalized RMSE and normalized maximum absolute error in (2.5). How well a method quantifies the uncertainty is measured by the actual coverage probability (ACP). We set $\alpha = 0.05$ throughout for nominal (correct) coverage probability of 0.95. A number much less than 0.95 is called under-coverage, which indicates the method does not fully quantify uncertainty.

Results for the Nilson-Kuusk example are presented in Figure 3.1. In the left panel the boxplots of normalized RMSE over the 25 repeat experiments show that the best prediction accuracy among the four competing methods is from PowExp-Emp; it clearly outperforms all three methods with SqExp structure. The normalized maximum error results in the appendix show the same pattern. The ACP results in the right panel of Figure 3.1 show that the three methods with SqExp structure dramatically under-cover. PowExp-Emp has the best performance again.

These accuracy and coverage results are likely due to the effect of the fifth input to the Nilson-Kuusk model. Its estimated main effect (Schonlau and Welch, 2005) from a PowExp-Emp analysis accounts for about 90% of the total variance of the predicted output over the 5-dimensional input space. The median of the 25 MLEs of α_5 in the PowExp correlation function is 1.85, which deviates from the fixed power of 2 in SqExp. All of this

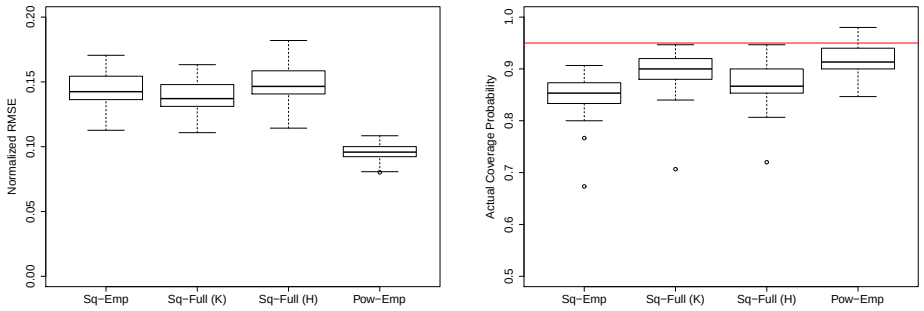

Figure 3.1: Normalized RMSE (left panel) and actual coverage probability (right panel) for the Nilson-Kuusk model from four methods: SqExp-Emp, SqExp-Full (K), SqExp-Full (H) and PowExp-Emp. The boxplots show the results from 25 random training-test data splits. The horizontal line in the right panel is the nominal coverage probability, 0.95.

suggests that full Bayes methods with SqExp structure need to be generalized to incorporate more flexibility to deal with a computer model like Nilson-Kuusk, for better accuracy and better uncertainty quantification.

Unfortunately, PowExp-Emp is not the panacea: it does not always work as well as it does for the Nilson-Kuusk example. In chapter 2, we noted that its ACP is much worse than that of the SqExp-Full (K) method for the borehole model. New methods are needed.

3.3 New Bayesian Methods

In this section we outline new Bayesian methods for the PowExp and Matérn correlation functions. We first describe the priors for the parameters treated in a fully Bayesian way; any further correlation parameters for a particular method are estimated by empirical Bayes. We then describe a general approach to implement all methods by outlining the marginal posterior distribution of all correlation parameters estimated by Bayes' rule and the algorithm for sampling them via MCMC.

3.3.1 Priors for μ, σ^2 and $\boldsymbol{\theta}$

Similar to SqExp-Full (Version H), all the new methods treat μ, σ^2 and the sensitivity parameters $\boldsymbol{\theta}$ in a fully Bayesian way. The independent priors are:

- A uniform distribution, $U(-1000, 1000)$, on μ, which approximates an improper normal distribution;
- An inverse-gamma IG(ϕ_1, ϕ_2) distribution on σ^2, where $\phi_1 \to 0$ and $\phi_2 \to 0$ (equivalent to the Jeffreys prior); and
- An independent beta$(1, 0.5)$ distribution on $\rho_j = \exp(-\theta_j/4)$, for $j = 1, \ldots, d$.

The Jeffreys prior is an improper prior distribution on σ^2. However it does not cause any problems when integrating σ^2 out of the joint posterior distribution of all parameters. See the derivation for the marginal posterior distribution of the correlation parameters in Appendix B.2.

A beta (1,0.5) prior on the ρ_j scale is equivalent to assuming a log-beta distribution for θ_j:

$$\pi(\theta_j) = \frac{1}{8}\exp(-\theta_j/4)\frac{1}{\sqrt{1-\exp(-\theta_j/4)}} \quad (j = 1, \ldots, d).$$

Figure 3.2(a) shows that this log-beta prior places heavy weight on small θ_j values.

The preference for small values of θ_j reflects prior belief that the underlying function is predictable, since a small θ_j value means strong correlation between neighbouring points in the design space.

When implementing the algorithm, we actually work with $\lambda_j = \ln(\rho_j/(1-\rho_j))$. After the logistic transformation, λ_j is unconstrained, facilitating proposals in MCMC. The implied prior density on λ_j is

$$\pi(\lambda_j) = \frac{1}{2}\left(\frac{\exp(-\lambda_j)}{1+\exp(-\lambda_j)}\right)^{-\frac{1}{2}}\left(\frac{\exp(\lambda_j)}{(1+\exp(\lambda_j))^2}\right) \quad (j = 1, \ldots, d). \quad (3.4)$$

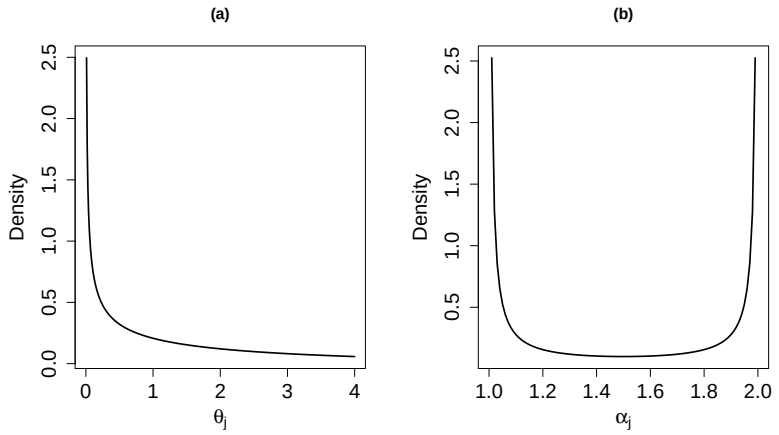

Figure 3.2: Prior densities: (a) prior on θ_j and (b) prior on α_j.

The priors above are closely related to the SqExp-Full (H) method of Higdon et al. (2008). Chen (2013) found through simulation that the log-beta prior on θ_j, which favours small θ_j, is highly preferred. The improper normal prior on μ and inverse-gamma prior on σ^2 are well-known conjugate priors, which makes it easy to integrate them out when deriving the marginal posterior of the θ_j, or equivalently the λ_j.

3.3.2 Power-Exponential-Hybrid

We explore two approaches to deal with the smoothness parameters, $\boldsymbol{\alpha}$, of PowExp. The first takes the empirical Bayes paradigm for these parameters only, by plugging in the MLEs of α_j. As the remaining parameters are treated in a fully Bayesian way (see section 3.3.1), we call this method PowExp-Hybrid (further abbreviated as Pow-Hyb in figures).

Spiller et al. (2014) recently proposed a Bayesian method in which they fix $\alpha_j = 1.9$ and estimate $\log(\theta_j)$ by its posterior mean using a reference prior (Paulo, 2005). This approach differs from the PowExp-Hybrid method we propose here in two aspects: (1) we estimate the smoothness parameters instead of fixing them, and (2) we use a different prior distribution on the sensitivity parameter, θ_j.

3.3.3 Power-Exponential-Full

The second new PowExp implementation is fully Bayesian and hence abbreviated PowExp-Full (further abbreviated as Pow-Full in figures). Besides the priors specified above, it needs priors on the smoothness parameters α_j.

We actually work on the logistic scale γ_j, where

$$\alpha_j = 1 + \frac{1}{1 + \exp(-\gamma_j)}.$$

Like the λ_j for the sensitivity parameters, γ_j is unrestricted, always giving $\alpha_j \in (1, 2)$. In practice, we take an independent uniform prior, $U(-20, 20)$, on γ_j. After a transformation of variables, the implied prior distribution for α_j is

$$\pi(\alpha_j) = \frac{1}{40} \frac{1}{(\alpha_j - 1)(2 - \alpha_j)} \quad (j = 1, \ldots, d), \tag{3.5}$$

as plotted in Figure 3.2(b). It assigns heavy weight to α_j values up to about 1.2 and from about 1.8.

The specification of this prior on α_j (via γ_j) was the result of much trial and error. We tried independent uniform priors on the α_j, which is straightforward, but results were not satisfactory for the borehole example in section 3.4.3. We also considered independent uniform $U(0, \pi)$ priors on γ_j, where $\alpha_j = 1 + \sin(\gamma_j)$. This implied prior for α_j has heavy weight on values close to the boundary 2, the qualitative feature we were seeking, but it did not work well for the Nilson-Kuusk model. Several other priors were considered, but only the prior in (3.5) from a logistic transformation worked for all the examples in section 3.4. It is used for for all results reported for PowExp-Full.

3.3.4 Matérn-Hybrid

A Matérn-Hybrid implementation parallels PowExp-Hybrid: the Matérn smoothness parameters δ_j in (2) are estimated by MLE, with each dimension allowed its own estimate from the four levels of smoothness. It is applied in section 3.4.1 and section 3.4.2 to two computer codes where the correlation

structure is known to be important from the analysis in chapter 2. The method is further abbreviated as Mat-Hyb in figures.

3.3.5 Marginal posterior distribution of the correlation parameters

For any method, denote by $\boldsymbol{\psi}_B$ the correlation parameters treated in a fully Bayesian way and sampled by MCMC. Any remaining correlation parameters are estimated by empirical Bayes. Thus, for the methods to be compared,

$$\boldsymbol{\psi}_B = \begin{cases} \boldsymbol{\lambda} & \text{(SqExp-Full, with all } \alpha_j \text{ fixed at 2)} \\ \boldsymbol{\lambda}, \boldsymbol{\gamma} & \text{(PowExp-Full)} \\ \boldsymbol{\lambda} & \text{(PowExp-Hybrid, with all } \alpha_j \text{ replaced by MLEs)} \\ \boldsymbol{\lambda} & \text{(Matérn-Hybrid, with all } \delta_j \text{ replaced by MLEs).} \end{cases}$$

All methods have the priors for μ, σ^2, and the λ_j prescribed in section 3.3.1. Recall that λ_j and γ_j are logistic transformations of the sensitivity parameter θ_j and the power α_j, respectively.

We want the marginal posterior distribution of $\boldsymbol{\psi}_B$ after integrating out μ and σ^2. With the priors $\pi(\mu) \propto 1$ and $\pi(\sigma^2) = IG(\phi_1, \phi_2)$, as in section 3.3.1, we have

$$\pi(\boldsymbol{\psi}_B|\mathbf{y}) \propto \frac{\pi(\boldsymbol{\psi}_B)}{|\mathbf{R}|^{1/2} \left(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}\right)^{1/2} \left(\phi_2 + \frac{n-1}{2}\hat{\sigma}^2_{\boldsymbol{\psi}_B}\right)^{(\phi_1+\frac{n-1}{2})}}, \tag{3.6}$$

in the appendix where $|\mathbf{R}|$ denotes the determinant of $\mathbf{R}$, and $\hat{\sigma}^2_{\boldsymbol{\psi}_B}$ is as in (1.7) but parameterized in terms of $\boldsymbol{\psi}_B$. A derivation may be found in the in the appendix. The prior $\pi(\boldsymbol{\psi}_B)$ is a product of the independent priors (3.4) for the λ_j and, in the case of PowExp-Full, the independent uniform priors in section 3.3.3 for the γ_j. For PowExp-Hybrid and Matérn-Hybrid, both sides of (3.6) are conditional on the MLEs of the α_j or δ_j, respectively.

The parameters $\boldsymbol{\psi}_B$ are sampled from the marginal posterior distribution by the algorithm in section 3.3.6. According to Santner et al. (2003), the predictive distribution of $y(\mathbf{x}^*|\mathbf{y}, \boldsymbol{\psi}_B)$ is as in (3.3), but again the degrees of

freedom are $2\phi_1 + n - 1$ and the predictive variance changes to

$$\hat{v}_{\boldsymbol{\psi}_B}(\mathbf{x}^*) = \left(\frac{(n-1)\hat{\sigma}^2_{\boldsymbol{\psi}_B} + 2\phi_2}{2\phi_1 + n - 1} \right) \left(1 - \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*) + \frac{\left(1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)\right)^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} \right). \tag{3.7}$$

3.3.6 Metropolis-Hastings algorithm

Taking the PowExp-Full method for illustration, we briefly describe how Metropolis-Hastings (M-H) is used to sample the posterior distribution of $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$.

At iteration i, the algorithm cycles through the input dimensions making a proposal for each parameter in turn. The details are in algorithm 1.

Algorithm 1 Implementation of the M-H Algorithm

At iteration i, to update for dimension j, denote the current values of $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ by $\boldsymbol{\lambda}_j^{[i]} = (\lambda_1^{[i]}, \ldots, \lambda_{j-1}^{[i]}, \lambda_j^{[i-1]}, \lambda_{j+1}^{[i-1]}, \ldots, \lambda_d^{[i-1]})$ and $\boldsymbol{\gamma}_j^{[i]} = (\gamma_1^{[i]}, \ldots, \gamma_{j-1}^{[i]}, \gamma_j^{[i-1]}, \gamma_{j+1}^{[i-1]}, \ldots, \gamma_d^{[i-1]})$.

1. Use the adaptive proposal $Q(\lambda_j^{[i]})$ in (3.8) to generate a candidate λ_j^*.
2. Randomly sample u from $U(0,1)$.
3. Let $\boldsymbol{\lambda}_j^* = (\lambda_1^{[i]}, \ldots, \lambda_{j-1}^{[i]}, \lambda_j^*, \lambda_{j+1}^{[i-1]}, \ldots, \lambda_d^{[i-1]})$. If $\ln(u) < \ln(\pi(\boldsymbol{\lambda}_j^*, \boldsymbol{\gamma}_j^{[i]}|\mathbf{y})) - \ln(\pi(\boldsymbol{\lambda}_j^{[i]}, \boldsymbol{\gamma}_j^{[i]}|\mathbf{y}))$, set $\lambda_j^{[i]} = \lambda_j^*$; otherwise $\lambda_j^{[i]} = \lambda_j^{[i-1]}$. The posterior distribution used here is $\pi(\boldsymbol{\psi}_B|\mathbf{y})$ in (3.6) with $\boldsymbol{\psi}_B = (\boldsymbol{\lambda}, \boldsymbol{\gamma})$. We now have the updated vector $\boldsymbol{\lambda}_{j+1}^{[i]} = (\lambda_1^{[i]}, \ldots, \lambda_{j-1}^{[i]}, \lambda_j^{[i]}, \lambda_{j+1}^{[i-1]}, \ldots \lambda_d^{[i-1]})$.
4. Use the adaptive proposal $Q(\gamma_j^{[i]})$ in (3.8) to generate a candidate γ_j^*.
5. Randomly sample another u from $U(0,1)$.
6. Let $\boldsymbol{\gamma}_j^* = (\gamma_1^{[i]}, \ldots, \gamma_{j-1}^{[i]}, \gamma_j^*, \gamma_{j+1}^{[i-1]}, \ldots, \gamma_d^{[i-1]})$. If $\ln(u) < \ln(\pi(\boldsymbol{\lambda}_{j+1}^{[i]}, \boldsymbol{\gamma}_j^*|\mathbf{y})) - \ln(\pi(\boldsymbol{\lambda}_{j+1}^{[i]}, \boldsymbol{\gamma}_j^{[i]}|\mathbf{y}))$, set $\gamma_j^{[i]} = \gamma_j^*$; otherwise $\gamma_j^{[i]} = \gamma_j^{[i-1]}$. We now have the updated vector $\boldsymbol{\gamma}_{j+1}^{[i]} = (\gamma_1^{[i]}, \ldots, \gamma_{j-1}^{[i]}, \gamma_j^{[i]}, \gamma_{j+1}^{[i-1]}, \ldots \lambda_d^{[i-1]})$.

For an efficient algorithm, the step length for a proposal is important, and we use the adaptive MCMC algorithm (Roberts and Rosenthal, 2009). Let ψ denote the value at iteration i of a parameter under consideration for transition. The candidate value is

$$Q(\psi) = 0.95N(\psi, 2.38^2 s^2) + 0.05N(\psi, 0.1^2), \tag{3.8}$$

where s^2 is the sample variance of the sampled values of ψ up to iteration i, which adapts as the algorithm iterates. Efficiency is achieved by this mixture proposal for all the examples we consider in section 3.4 in the sense that almost all acceptance rates are between 0.15 and 0.50 (Rosenthal, 2014).

More details may be found in the appendix.

For any method, at iteration i after all dimensions have been updated, $\boldsymbol{\psi}_B^{[i]}$ has been sampled from the posterior. The sample leads to a conditional predictive mean, $\hat{m}_{\boldsymbol{\psi}_B^{[i]}}(\mathbf{x})$, and conditional predictive variance, $\hat{v}_{\boldsymbol{\psi}_B^{[i]}}(\mathbf{x})$, computed according to (1.3) and (3.7), respectively. The overall predictor is defined as

$$\hat{m}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} \hat{m}_{\boldsymbol{\psi}_B^{[i]}}(\mathbf{x}),$$

where M is the number of samples. The predictive variance is obtained through the law of total variance. That is,

$$\hat{v}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} \hat{v}_{\boldsymbol{\psi}_B^{[i]}}(\mathbf{x}) + \frac{1}{M-1}\sum_{i=1}^{M} \left(\hat{m}_{\boldsymbol{\psi}_B^{[i]}}(\mathbf{x}) - \hat{m}(\mathbf{x})\right)^2.$$

The M-H algorithm is simpler for SqExp-Full, PowExp-Hybrid, and Matérn-Hybrid, because it needs to iterate over the $\boldsymbol{\lambda}$ sensitivity parameters only. Thus, the other methods take half the computational time per iteration in the M-H algorithm compared with PowExp-Full. (The hybrid methods have to find MLEs of the smoothness parameters, however, before running M-H.)

3.4 Applications and Simulation Study

We now evaluate the performances of six methods: SqExp-Emp, SqExp-Full, PowExp-Emp, PowExp-Full, PowExp-Hybrid, and Matérn-Hybrid. As in section 3.2, normalized RMSE and normalized maximum absolute error are used to evaluate the prediction accuracy, and the ACP is used to assess the uncertainty quantification. The nominal coverage probability is set to 0.95.

3.4.1 Nilson-Kuusk Model

We revisit the Nilson-Kuusk code with repeat experiments generated as in section 3.2.

The boxplots of normalized RMSE in the left panel of Figure 3.3 show that the PowExp and Matérn correlation functions lead to noticeably better prediction accuracy than SqExp. Empirical Bayes versus full Bayes versus hybrid Bayes makes little difference here. The results for normalized maximum absolute error in the appendix follow a similar pattern.

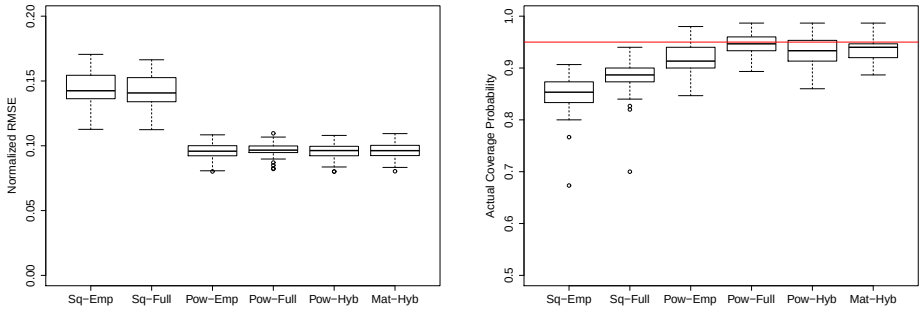

Figure 3.3: Normalized RMSE (left panel) and actual coverage probability (right panel) for the Nilson-Kuusk code from six methods: SqExp-Emp, SqExp-Full, PowExp-Emp, PowExp-Full, PowExp-Hybrid and Matérn-Hybrid. The boxplots show the results from 25 random training-test data splits. The horizontal line in the right panel is the nominal coverage probability, 0.95.

Sample means of normalized RMSE over the repeat experiments are reported in Table 3.1 for the four methods giving clearly superior prediction accuracy here: those using either a PowExp or Matérn correlation function. The small differences in sample means and small standard errors of the mean differences also reported in Table 3.1 indicate that these four methods have similar prediction accuracies on average.

The ACP results in the right panel of Figure 3.3 also demonstrate the advantage of the PowExp and Matérn correlation functions versus SqExp, with the three new Bayesian methods coming closest to the nominal coverage probability of 0.95. Equipping Bayesian methods with PowExp or Matérn structure is advantageous here in terms of uncertainty quantification relative to SqExp, even if the latter has a fully Bayesian treatment. These findings

	PowExp-Emp	PowExp-Full	PowExp-Hybrid	Matérn-Hybrid
PowExp-Emp	0.0955	-0.0010 (0.0005)	-0.0002 (0.0002)	-0.0003 (0.0008)
PowExp-Full	0.0010 (0.0005)	0.0965	0.0008 (0.0005)	0.0007 (0.0007)
PowExp-Hybrid	0.0002 (0.0002)	-0.0008 (0.0005)	0.0957	-0.0001 (0.0008)
Matérn-Hybrid	0.0003 (0.0008)	-0.0007 (0.0007)	0.0001 (0.0008)	0.0958

Table 3.1: Sample means of the 25 normalized RMSEs from repeat experiments with the Nilson-Kuusk model for four methods. The sample means are in the diagonal entries of the table. An off-diagonal entry shows the difference in sample means between the row method and the column method, with the standard error of the mean difference in parentheses.

are confirmed by Table 3.2, which compares the four methods using PowExp or Matérn structure in terms of $|\text{ACP} - 0.95|$, the absolute deviation of ACP from the nominal coverage probability. PowExp-Emp stands out with the

	PowExp-Emp	PowExp-Full	PowExp-Hybrid	Matérn-Hybrid
PowExp-Emp	0.0388	0.0203 (0.0045)	0.0083 (0.0024)	0.0184 (0.0039)
PowExp-Full	-0.0203 (0.0045)	0.0185	-0.0120 (0.0032)	-0.0019 (0.0027)
PowExp-Hybrid	-0.0083 (0.0024)	0.0120 (0.0032)	0.0305	0.0103 (0.0031)
Matern-Hybrid	-0.0184 (0.0039)	0.0019 (0.0027)	-0.0103 (0.0031)	0.0204

Table 3.2: Sample means of $|\text{ACP} - 0.95|$ from 25 repeat experiments with the Nilson-Kuusk model. The sample means of four methods are in the diagonal entries of the table. An off-diagonal entry shows the difference in sample means between the row method and the column method, with the standard error of the mean difference in parentheses.

largest mean absolute deviation from nominal coverage, and the observed mean difference of 0.0203 relative to PowExp-Full is large relative to the standard error of the mean difference. The differences in averages and their standard errors are all fairly small for PowExp-Full, PowExp-Hybrid, and Matérn-Hybrid, indicating that these methods perform about the same, as was seen in Figure 3.3.

To summarize the results for the Nilson-Kuusk code, use of a flexible correlation structure—PowExp or Matérn—leads to non-trivial improvements in prediction accuracy relative to SqExp. Uncertainty quantification is also much more reliable with PowExp or Matérn, with full or hybrid Bayesian implementations adding to the advantage.

3.4.2 Volcano Model

A computer model of pyroclastic flow from a volcano eruption was studied by Bayarri et al. (2009). The two inputs are initial volume, x_1, and direction, x_2, of the eruption, and the output, y, is the maximum height of the flow at a specific location.

The non-negative output suggests transformation may be helpful. Bayarri et al. (2009) transformed y to $\log(y+1)$. In chapter 2, we studied both the $\log(y+1)$ and $\sqrt{y}$ transformations, as we do here. A 32-point data set is available. We sample 25 points out of the 32 points as a training set and leave the remaining 7 points as a hold-out set for testing. Sampling is repeated 100 times.

The normalized RMSE results are shown in Figure 3.4. For both output

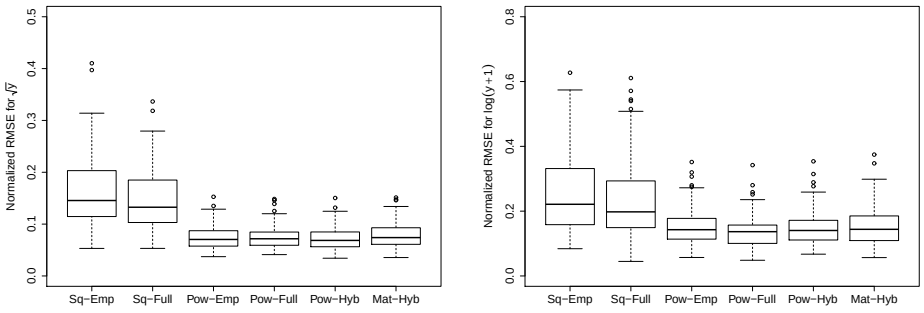

Figure 3.4: Normalized RMSE for the volcano code from six methods: SqExp-Emp, SqExp-Full, PowExp-Emp, PowExp-Full, PowExp-Hybrid and Matérn-Hybrid. Each method has 100 RMSE values from 100 random training-test data splits. Two transformations of y are considered: $\sqrt{y}$ (left panel) and $\log(y+1)$ (right panel).

transformations, methods with a flexible correlation structure—PowExp-Emp, PowExp-Full, PowExp-Hybrid, and Matérn-Hybrid—have higher accuracy than SqExp-Emp and SqExp-Full. Results for normalized maximum absolute error presented in the appendix show a similar pattern.

Table 3.3 reports the ACP of each method, obtained as the number of 95% confidence/credible intervals containing the true output value di-

vided by the total number of predictions, 700. The ACPs of PowExp-Full,

	Average coverage probability	
	$\sqrt{y}$	$\log(y+1)$
SqExp-Emp	0.63	0.71
SqExp-Full	0.78	0.82
PowExp-Emp	0.86	0.85
PowExp-Full	0.92	0.91
PowExp-Hybrid	0.90	0.87
Matérn-Hybrid	0.90	0.88

Table 3.3: Actual coverage probability for the volcano code, computed as the number of 95% confidence/credible intervals containing the true output of each method divided by the total number of predictions, 700. The nominal coverage is 0.95.

PowExp-Hybrid, and Matérn-Hybrid are closer to 0.95, though they slightly under-cover. The SqExp-Emp approach substantially under-covers, with SqExp-Full faring little better. Thus, again the correlation function is more important than the estimation paradigm here, though Bayesian methods, with a full or hybrid implementation, make further improvements.

3.4.3 Borehole Function

We consider the Borehole function again as a test bed. Two base designs, with 80 or 200 points, are explored; both are approximate mLHDs (McKay et al., 1979) with the maximin criterion adapted to have desired space-filling properties in all 2-dimensional projections (Welch et al., 1996). The columns of each base design are permuted at random to generate 25 different but equivalent designs and hence 25 repeat experiments. The hold-out set is 10,000 points drawn from one random LHD.

For $n = 80$ the left panel of Figure 3.5 shows that the prediction accuracies of the six competing methods are similar. The ACP performances of the methods are seen to differ, however, in the right panel of Figure 3.5. SqExp-Emp and PowExp-Emp show under-coverage, while the four Bayesian implementations have near-nominal coverage probabilities, with PowExp-Full and Matérn-Hybrid the best.

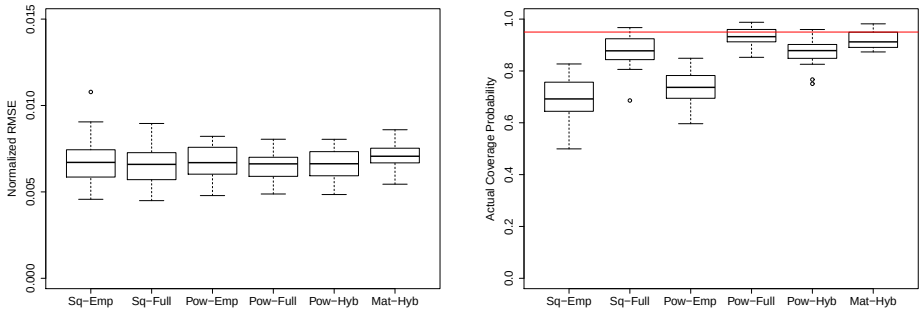

Figure 3.5: Normalized RMSE (left panel) and actual coverage probability (right panel) for the Borehole function and a 80-point mLHD base design. The boxplots show the results from 25 repeat experiments permuting the columns of the base design. The horizontal line in the right panel is the nominal coverage probability, 0.95.

When the sample size is increased to 200, however, the prediction accuracies depicted in the left panel of Figure 3.6 show methods using PowExp correlation structure are noticeably more accurate than those employing SqExp and slightly better than using Matérn. A similar pattern is seen for normalized maximum absolute error in the appendix. The ACP results

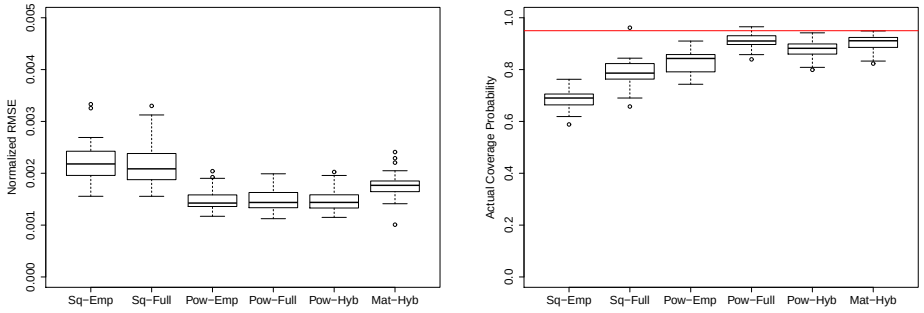

Figure 3.6: Normalized RMSE (left panel) and actual coverage probability (right panel) for the Borehole function and a 200-point mLHD. The boxplots show the results from 25 random training-test data splits. The horizontal line in the right panel is the nominal coverage probability, 0.95.

depicted in the right panel of Figure 3.6 demonstrate that the SqExp correlation structure leads to under-coverage, even with a Bayesian implementation. A possible explanation is that the impact of any inadequacy of the GP statistical model with SqExp in modelling the borehole function increases with sample size. PowExp-Emp also under-covers, though less, while the three Bayesian methods using PowExp or Matérn again have ACP closest to the nominal coverage probability.

3.4.4 PTW Model

Preston et al. (2003) developed the PTW model to describe the plastic deformation of metals. For our purposes the model contains $d = 11$ input parameters. A base design has its columns permuted at random to generate 25 different but equivalent designs. We consider a 110-point mLHD. The hold-out set is $10,000$ points generated from one random LHD.

The results are presented in Figure 3.7. The normalized RMSE values in the left panel show the four methods employing the PowExp or Matérn correlation functions slightly outperforming both methods with SqExp.

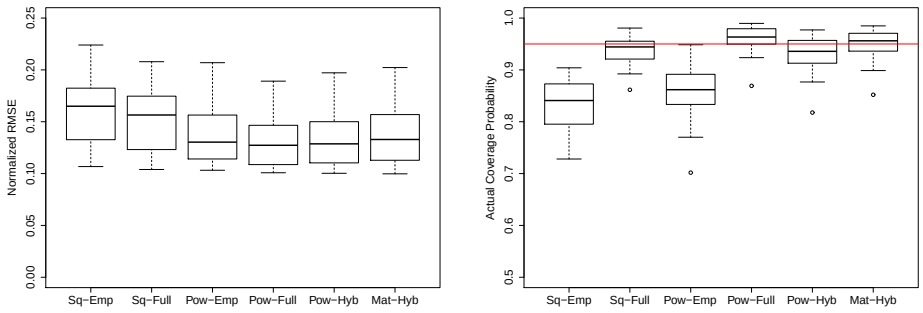

Figure 3.7: Normalized RMSE (left panel) and actual coverage probability (right panel) for the PTW model and a 110-point mLHD. The boxplots show the results from 25 random repeat experiments permuting the columns of the base design. The horizontal line in the right panel is the nominal coverage probability, 0.95.

The normalized maximum absolute error results in the appendix show

a similar pattern. The right panel of Figure 3.7 indicates, however, that the four fully Bayesian or hybrid methods have clearly better uncertainty-quantification properties than the two empirical Bayes methods, which undercover as expected.

3.4.5 Simulations from GPs

The applications so far assess the new Bayesian methods on computer codes, meaning that uncertainty quantification includes model misspecification in representing a code by a statistical GP model. We now consider ideal situations, where functions are realizations of GPs.

The simulation settings are as follows. There are $d = 10$ inputs (results, not shown, for $d = 5$ are similar). The true value of μ is 0 and $\sigma^2 = 1$; there is little loss of generality here as the priors proposed in section 3.3.1 allow much latitude for location and scale. Three true correlation functions are considered: SqExp, PowExp with with all α_j set to 1.8, and Matérn with all $\delta_j = 1$. Thus, the realized functions have various degrees of smoothness. The true values of the θ_j are given in Table 3.4; they are a canonical configuration generated by the method of Loeppky et al. (2009) with $\tau = 1$ and $b = 3$. A training sample size of $n = 100$ gives reasonable prediction accuracy for realizations from all three GP families.

j	1	2	3	4	5	6	7	8	9	10
θ_j	0.271	0.217	0.169	0.127	0.091	0.061	0.037	0.019	0.007	0.001

Table 3.4: Values of the θ_j for simulation.

Given one of the GP settings for generating data, there are 25 repeat experiments from different random LHD training designs. A set of 100 training points is combined with a 2000-point random LHD for the hold-out set (the same hold-out set is used for all 25 repeats), and a GP realization is sampled from the multivariate normal at the $100 + 2000$ points for the training and hold-out output data. All six of the methods SqExp-Emp, SqExp-Full, PowExp-Emp, PowExp-Full, PowExp-Hybrid, and Matérn-Hybrid are fit to the data, with all relevant parameters estimated. For instance, the θ_j and α_j

are estimated for PowExp, by empirical Bayes, full Bayes, or hybrid Bayes.

Figure 3.8 shows the performances of the six methods when data are generated by a GP with SqExp correlation. As SqExp is a special case of PowExp and Matérn, all trained models assume the correct GP family. In the left panel of Figure 3.8 it is seen that distributions of RMSE over the repeat simulations are practically identical, i.e., there is no over-fitting penalty from using PowExp or Matérn with any of the inference methods considered. The ACP results in the right panel of Figure 3.8 indicate that the empirical Bayes methods SqExp-Emp and PowExp-Emp underestimate prediction uncertainty, whereas the full Bayes or hybrid Bayes methods obtain near-nominal ACP.

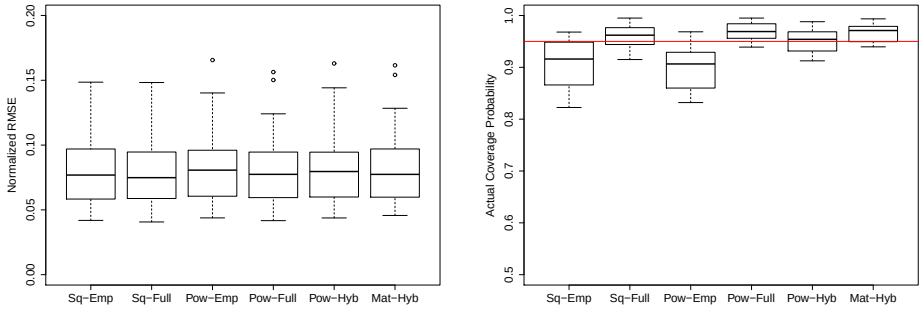

Figure 3.8: Normalized RMSE (left panel) and actual coverage probability (right panel) with $d = 10$ inputs and output simulated from a GP with SqExp correlation. The boxplots show the results from 25 random realizations of the GP. The horizontal line in the right panel is the nominal coverage probability, 0.95.

With output realized from a GP with PowExp correlation and all $\alpha_j = 1.8$, the RMSE results in the left panel of Figure 3.9 show that assuming a PowExp or Matérn correlation function, with any of the inference methods, is now clearly much more accurate for prediction than using the wrong correlation function, SqExp. The ACP results in the right panel of Figure 3.9 are again driven by the inference method: there is substantial under-coverage for the empirical Bayes methods. SqExp-Full and PowExp-

Hybrid slightly under-cover, while PowExp-Full and Matérn-Hybrid have near-nominal ACP.

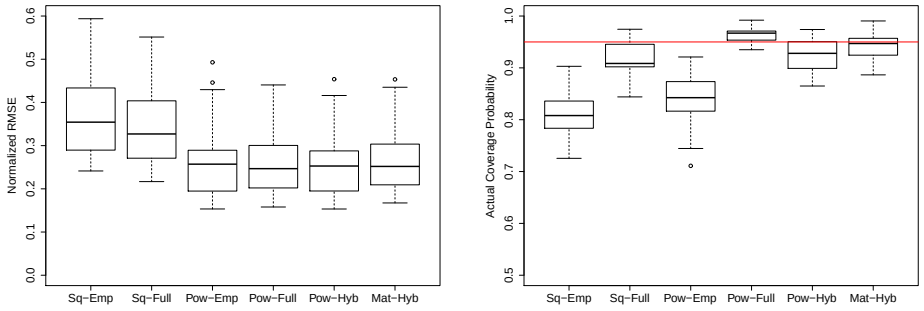

Figure 3.9: Normalized RMSE (left panel) and actual coverage probability (right panel) with $d = 10$ inputs and output simulated from a GP with PowExp correlation and all $\alpha_j = 1.8$. The boxplots show the results from 25 random realizations of the GP. The horizontal line in the right panel is the nominal coverage probability, 0.95.

When the output is realized from a GP with Matérn correlation and all $\nu_j = 1$, the left panel of Figure 3.10 demonstrates that the PowExp and Matérn correlation functions again lead to much smaller RMSEs than does SqExp, regardless of the parameter-estimation method. The right panel of Figure 3.10 shows severe under-coverage for SqExp-Emp, some under-coverage for SqExp-Full, and modest under-coverage for PowExp-Emp. PowExp-Full leads to over-coverage here, while PowExp-Hybrid and Matérn-Hybrid have near-nominal ACP.

Overall, the simulations suggest that the PowExp and Matérn correlation functions are advantageous in terms of prediction accuracy relative to SqExp. When data were generated using SqExp, there was little over-fitting penalty in estimating the extra parameters of the PowExp and Matérn families. On the other hand, the more flexible PowExp and Matérn families sometimes led to much more accurate predictions. The normalized maximum absolute error results reported in the appendix follow similar patterns.

The simulations also point to the empirical-Bayes methods SqExp-Emp

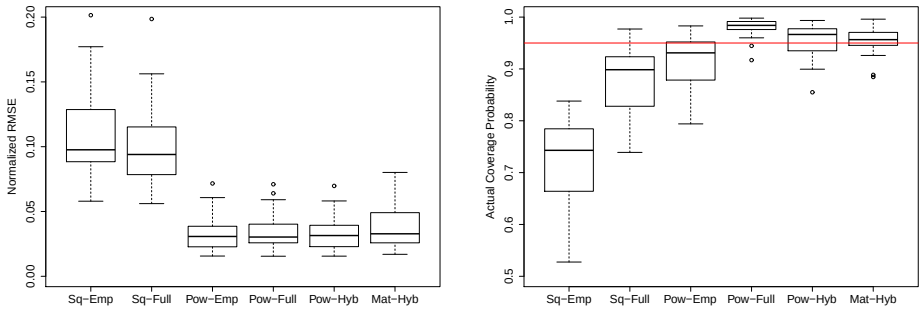

Figure 3.10: Normalized RMSE (left panel) and actual coverage probability (right panel) with $d = 10$ inputs and output simulated from a GP with Matérn correlation and all $\delta_j = 1$. The boxplots show the results from 25 random realizations of the GP. The horizontal line in the right panel is the nominal coverage probability, 0.95.

and PowExp-Emp under-estimating uncertainty, sometimes substantially. SqExp-Full also led to ACP values smaller than nominal, except when data were generated using a SqExp correlation function. A combination of fully Bayesian or hybrid-Bayesian inference with the PowExp or Matérn families, provided more reliable inference, however. Such combinations showed modest over- or under-coverage.

3.5 Conclusions and Discussion

Bayesian implementations with SqExp correlation structure were extended to allow PowExp or Matérn structure. To estimate the additional smoothness parameters α_j in PowExp, hybrid and fully Bayesian approaches were proposed and applied. The hybrid method uses MLEs of the α_j, with all other parameters in a GP model handled via Bayes' rule. The fully Bayesian method employs an uninformative uniform prior for a logistic transform of α_j. Similarly, a hybrid approach for the Matérn correlation function, uses MLEs for the smoothness parameters δ_j.

The work in chapter 2 noted that a fully Bayesian treatment tends to

have better coverage probability than empirical Bayes method for applications such as the borehole function where SqExp is adequate. The previous work also noted that PowExp can give better prediction accuracy for applications such as the Nilson-Kuusk code, where SqExp is less adequate. The applications and simulations considered show that advantages in terms of prediction accuracy and uncertainty quantification tend to go hand in hand when a Bayesian method with flexible correlation function is employed. Moreover, there appears to be little over-fitting penalty from employing PowExp or Matérn when the SqExp special case is adequate.

There are some important details of implementation. First, the priors on the sensitivity parameters θ_j based on previous work (Higdon et al., 2008) assume the inputs x_j are scaled to $[0, 1]$. These priors assign heavier weight to small values of the θ_j, reflecting a prior view that the function can be usefully predicted. Secondly, such priors are particularly appropriate for a constant-mean model. If a regression model with linear trends in the inputs is used, priors giving more weight to moderate values of θ_j (Kennedy, 2004) may be more appropriate. Finally, we found that MCMC sampling of all correlation parameters, for sensitivity and smoothness, was more efficient with parameters transformed to a logistic scale.

Chapter 4

Evaluation of Designs for Computer Experiments

From the analyses in chapters 2 and 3, we observe that design can have a big effect on prediction accuracy. For instance, in chapter 2, with the same sample size for fitting a GP emulating the Borehole function, a 27-run mLHD nearly always yields much smaller RMSE values than a 27-run OA over 25 repeat experiments. The pattern persists when the sample size increases to 40 and 80. This observation shows the importance of design. This chapter concerns the effect of design on prediction accuracy. A simple completely random n by d design can be generated such that every dimension contains n independent realizations from $U(0, 1)$. Such designs are easy to generate, but unless one is lucky, the resulting design might have large holes.

Intuitively, we would like designs to be space-filling when the primary interest is to emulate a computer model using a GP. The reason is that the predictor from a GP (we are assuming the computer model is deterministic and no nugget or a trivially small nugget is used in the GP model) is actually an interpolator. The prediction error can be viewed as a function of the location of the untried point relative to the observed points in the design space. Therefore, the prediction accuracy will deteriorate if the untried point is located in a sub-region that is sparsely observed. We believe this also explains why a space-filling design, in which the design points are evenly spread over the design space, is usually preferred in published analyses. There are several methods available to generate space-filling designs, depending on what one means by "evenly spread". As introduced in section 1.3, chapter 1, there are at least two different ways of achieving the "space-

filling" property, either by defining and optimizing a distance measurement (e.g., mLHD), or by using a low-discrepancy sequence (e.g. Sobol sequence). We will review some of these methods in the next section in this chapter.

Besides what has been covered previously, there are several other types of designs in the literature, such as the maximum utility design Caselton and Zidek (1984), the maximum entropy design (Shewry and Wynn, 1987) and the uniform design (Fang et al., 2000). Pronzato and Müller (2012) provided a comprehensive review. Since there exist many different designs both space-filling (random LHD, e.g.) and less space-filling (completely random design, e.g.) for practitioners to choose from, an interesting question we want to answer is the following. Without prior information, which one can be used as a default design? We are aware that many of the proposed designs have "beautiful" mathematical properties. For instance, Stein (1987) showed that if $n \to \infty$, the variance of the sample mean $\frac{1}{n}\sum_{i=1}^{n} Y^{(i)}$ is smaller using a random LHD than using a completely random design generated from $U(0,1)$ independently in each dimension. That being said, it could be risky to equate attractive theoretical results with better prediction accuracy with a GP. Therefore, the objective of this chapter is practical: first of all, it is fascinating to assess the effect of design on prediction accuracy with a GP in emulating a computer model. Secondly, which design can be used as a default without contextual knowledge. In a nutshell, we are interested in evaluating the prediction performance in practice and a comparison of theoretical properties among different designs is not the focus of the chapter.

The rest of this chapter is organized as follows: we review several popular classes of designs in section 4.1. Empirical studies are conducted in comparing different designs in section 4.2, 4.3 and 4.4 through some real examples. Some comments are made in section 4.5. The conclusions from the research are reported in section 4.6 and are briefly summarized as follows:

- The prediction accuracy differences between different designs are not as big as one might expect. Without prior information, one can use an mLHD as the default design in a computer experiment. MLHDs will be outlined in section 4.1.

- Other factors, such as the sample size and transformation of y if needed, have much larger effects on prediction accuracy than that of designs.

4.1 A Review of Easily Constructed Designs

4.1.1 Random LHD

McKay et al. (1979) introduced the random LHD. Without loss of generality, given a fixed sample size n, a random LHD in 2-d over the unit square can be constructed by the following steps:

- Construct $n \times n$ cells over the unit square.
- Construct an $n \times 2$ matrix, Z, with column independent random permutations of the integers, $\{1, 2, \ldots, n\}$.
- Each row of Z gives the row and column indices for a cell on the grid. For the i^{th} $(i = 1, 2, \ldots, n)$ row of Z, take a random uniform draw from the corresponding cell. The resulting design is the random LHD.

It can easily be extended to more than 2-d scenario. McKay et al. (1979) showed the variance of the sample mean $\frac{1}{n}\sum_{i=1}^{n} Y^{(i)}$ is smaller using a LHD than using a completely random design, if $y(x_1, \ldots, x_d)$ is monotonic in each of its arguments. Stein (1987) further showed that if $n \to \infty$, LHD has a smaller variance than a completely random design without the monotonicity condition. However, it is also well known that a random LHD can still have large holes. An obvious case is if all points lie on the diagonals by chance. We use *rLHD* to denote a random LHD in this chapter.

4.1.2 Maximin LHD

An mLHD can be viewed as a combination of an LHD and a maximin design. The idea of a maximin design is that in order for the points to be spread out over the space, no two points are too close to each other. To be more specific, let $D \subset \mathscr{X}$ be an arbitrary n-point design that consists of

distinct inputs $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$. One way to measure the distance between any two points $\mathbf{x}^{(i)}, \mathbf{x}^{(j)}$ is given by

$$\rho(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(\sum_{k=1}^{d} \left| \mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)} \right|^p \right)^{1/p}, \tag{4.1}$$

where $p = 1$ and $p = 2$ are rectangular and Euclidean distances, respectively. The Euclidean distance, $p = 2$ is used in this thesis. A maximin design maximizes the minimal distance between any two points in the design space. Given the design D, its minimal distance is defined.

$$\Phi(D) = \min_{\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in D} \rho(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}). \tag{4.2}$$

Intuitively, it makes sure that no two points are too close, and hence the design is spread out. Johnson et al. (1990) first defined the maximin design. A design that maximizes $\Phi(D)$ in (4.2) within the class of LHDs is called maximin LHD, which is abbreviated as *mLHD* throughout the thesis. This is an important type of design that will be evaluated in the next section. In this chapter, we use the `maximinSLHD (t=1, ...)` function of the R package `SLHD` (Ba et al., 2015) to generate an mLHD. All function parameters are taken as defaults unless stated otherwise.

In addition, there also exists a variant of mLHD. In (4.1), instead of considering all d dimensions ($d \geq 2$) at once, we consider all projections corresponding to a limited number of input variables. Let $\mathcal{S}$ denote a set of projections, usually all subsets of two variables at a time:

$$\mathcal{S} = \{\{1, 2\}, \{1, 3\}, \ldots, \{d-1, d\}\}.$$

There are $d(d-1)/2$ projections here and we use $|\mathcal{S}|$ to denote the number of projections in $\mathcal{S}$. Therefore, the minimal distance defined in (4.2) changes to:

$$\Phi(D) = \min_{\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in D} \left(\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{1}{n(n-1)/2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\rho_s(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})} \right). \tag{4.3}$$

Joseph et al. (2015) proposed a maximum projection design, which bears the same idea as introduced above. To avoid confusion, we use *mLHD* to denote an mLHD based on (4.2) and use *m2LHD* to denote an mLHD based on (4.3). The m2LHD design is generated by the R function `MaxProLHD()` of the package `MaxPro`.

4.1.3 Transformed LHD

The transformed LHD, which was proposed by Dette and Pepelyshev (2010), is a variant of the LHD. A transformed LHD allocates more experiments near the boundary of the design space. Therefore, it is not as space-filling as other designs. However, since the largest prediction error often occurs at untried points near the boundary, the transformed LHD is expected to have better prediction accuracy than rLHD and mLHD, especially when the prediction accuracy is measured for the worst case. Dette and Pepelyshev (2010) proposed the following steps to generate a transformed LHD:

Step A: Generates an mLHD.

Step B: Define a generalized design supported at the points $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)}$, where $\mathbf{z}^{(i)} = (z^{(i,1)}, \ldots, z^{(i,d)})$, by the transformation of the points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ using the quantile function of the Beta density

$$p_{(1+a)/2}(t) = \frac{1}{B((1+a)/2, (1+a)/2)} t^{(a-1)/2}(1-t)^{(a-1)/2}. \tag{4.4}$$

The transformation from $x^{(i,j)}$ to $z^{(i,j)}$ is effective by defining z as a solution to the equation

$$x^{(i,j)} = \int_0^{z^{(i,j)}} p((1+a)/2)(t)dt, \tag{4.5}$$

where the parameter $a \in [0, 1]$ makes the Beta density u-shaped, $i = 1, \ldots, n, j = 1, \ldots, d$. The parameter a can be considered as a tuning parameter, which specifies the important of the boundary. The smaller a is, the more mass is put at the boundary of the interval $[0, 1]$. The special case of $a = 0$ yields the arc-sine distribution. For that density, the transformation

is explicitly given as follows:

$$z^{(i,j)} = \frac{1 - \cos(\pi x^{(i,j)})}{2}. \tag{4.6}$$

We believe this type of "boundary-focused" design may be useful for some problems where behaviour near a boundary is important. We use *trLHD* to denote a transformed LHD with $a = 0$ based on an mLHD. Note that *trLHD* is based on mLHD only, not on m2LHD from the original paper (Dette and Pepelyshev, 2010).

4.1.4 Orthogonal Array-based LHD

The OA-based LHD was introduced by Tang (1993). The construction of an OA-based LHD depends on the existence of the corresponding OA. We describe orthogonal arrays first. An $n \times d$ matrix, M, is called an OA of strength $r, r \leq d$, level s, size n, index λ, if each $n \times r$ submatrix of M contains $1 \times r$ row vectors of all possible combinations of $1, 2, \ldots, s$ with the same frequency λ. The array is denoted by $OA(n, d, s, r)$. From the definition, it is clear that the sample size $n = \lambda s^r$ and a LHD is a special case of $OA(n, d, n, 1)$.

Consider an $OA(n, d, s, r)$ denoted as M. For each column of M, one replaces the λs^{r-1} locations with entry k by random permutation of $(k - 1)\lambda s^{r-1} + 1, (k-1)\lambda s^{r-1} + 2, \ldots, (k-1)\lambda s^{r-1} + \lambda s^{r-1}$, for all $k = 1, 2, \ldots, s$. It is then mapped to $[0, 1]^d$. The resulting matrix is an OA-based LHD. Tang (1993) showed that an OA-based LHD yields a smaller variance of $\bar{Y}$ than that of a random LHD in an asymptotic sense if $y(x_1, \ldots, x_d)$ is a continuous function in $[0, 1]^d$. However, OA-based LHDs have a strict restriction on sample size $n = \lambda s^r$. For instance, orthogonal arrays ($r \geq 2$) do not exist if n is a prime number. This largely restricts its availability in practice.

4.1.5 Sobol Sequence

A Sobol sequence (Sobol, 1967) is an example of a low-discrepancy sequence (Niederreiter, 1988). Given a set $P = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$, using Niederreiter's notation, the discrepancy is defined as

$$D_n(P) = \sup_{B \in J} \left| \frac{A(B; P)}{n} - \lambda_d(B) \right|,$$

where λ_d is the d-dimensional Lebesgue measure, $A(B; P)$ is the number of points in P that fall into B, and J is the set of d-dimensional regions in $[0, 1]^d$. B is a member of J. The discrepancy is low if the proportion of points in P falling into B is close to the measure of B for all B.

An obvious application of low-discrepancy sequence is numerical integration. The integral of a function f can be approximated by the average of the function evaluations at n points:

$$\underbrace{\int_0^1 \cdots \int_0^1}_{d} f(\mathbf{x}) d\mathbf{x} \approx \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}^{(i)}).$$

If the points are chosen randomly from a known distribution, this is the Monte Carlo method. However, if the points are chosen as elements of a low-discrepancy sequence, this is the quasi-Monte Carlo method. Niederreiter (1988) showed that the quasi-Monte Carlo method has a rate of convergence close to $O(1/n)$, whereas the rate for the Monte Carlo method is $O(1/\sqrt{n})$. Please see Niederreiter (1992) for more details. To approximate the integral well, the set $P = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ should minimize the holes in the space, which results in a "space-filling" design.

In R, the `sobol` function in package `randtoolbox` can be used to generate a Sobol sequence. In the thesis, we use `sobol(scrambling = 3, seed = 4711, ....)` to generate a Sobol sequence.

4.1.6 Uniform Design

Uniform design (Fang et al., 2000) is a kind of "space-filling" design. The basic idea of a uniform design is as follows. Consider d factors over region $[0,1]^d$. We want to find a design, $\mathcal{D}_u$ such that the points are uniformly scattered over the experiment space. This is achieved by minimizing the discrepancy between the distribution function (denoted as $F(\mathbf{x})$) of the uniform distribution on $[0,1]^d$ and the empirical distribution (denoted as $F_n(\mathbf{x})$) of the design points. The L_p discrepancy is usually defined as

$$D_{n,p}(\mathcal{D}_u) = \left[\int_{\mathcal{D}_u} |F(\mathbf{x}) - F_u(\mathbf{x})|^p \, d\mathbf{x}\right]^{1/p}.$$

One popular choice is the L_∞ norm, which yields

$$D_n(\mathcal{D}_u) = \sup_{\mathbf{x}\in\mathcal{D}_u} |F(\mathbf{x}) - F_n(\mathbf{x})|.$$

In practice, a uniform design is usually constructed by number theoretic properties or via algorithms based on a finite candidate set of points. In either case, the construction process is time-consuming and difficult. The difficulty in constructing such a design limits its availability and popularity.

4.1.7 Sparse Grid Design

Before we introduce sparse grid design (SGD), we consider the lattice design. A lattice design is defined as $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2, \times \ldots, \times \mathcal{X}_d$, where each $\mathcal{X}_j$ is a set of one-dimensional points termed a component design. For a set A and B, the Cartesian product, denoted as $A \times B$, is defined as the set of all ordered pairs (a, b), where $a \in A$ and $b \in B$. If the sample size of $\mathcal{X}_j$ is n_j, the sample size of $\mathcal{X}$ is $\prod_{i=1}^{d} n_j$. Given all the component designs, the lattice design is extremely easy to generate. It also has an appealing property that since its covariance matrix takes the form of a Kronecker product of matrices, inverting the correlation matrix R is equivalent to inverting d smaller submatrices, which can dramatically reduce the computational time when the sample size is large.

However, the lattice design has an obvious drawback that if d is large, the sample size will become exponentially large. For instance, if $d = 8$ and each dimension has the same component design $\{0.25, 0.5, 0.75\}$, the total sample size will be $3^8 = 6561$.

Plumlee (2014) proposed the use of sparse grid design as an alternative to a lattice design. To build an SGD, one must first specify a nested sequence of one-dimensional experimental designs for each $j = 1, \ldots, d$ denoted $\mathcal{X}_{j,t}$, where $\mathcal{X}_{j,t} \subseteq \mathcal{X}_{j,t+1}$, $t = 1, 2, \ldots$, $\mathcal{X}_{j,0} = \emptyset$ and $\mathcal{X}_{j,t}$ is a component design. A sparse grid design is, therefore, defined as

$$\mathcal{X}_{SG}(\eta) = \bigcup_{\vec{t} \in \mathcal{G}(\eta)} \mathcal{X}_{1,t_1} \times \mathcal{X}_{2,t_2} \times, \ldots, \times \mathcal{X}_{d,t_d}, \tag{4.7}$$

where $\eta \geq d$ is an integer that represents the level of the construction and $\mathcal{G}(\eta) = \{\vec{t} \in \mathcal{N}^d | \sum_{j=1}^{d} t_j = \eta\}$. Here we use the overhead arrow to distinguish the vector of indices, $\vec{t} = [t_1, \ldots, t_d]$ from a scalar index. It is easy to see that the sample size of a SGD is determined by the following three factors: (1) Dimension, d. (2) Level, η. (3) Component designs. The component designs that we use for this chapter are same as the ones reported in the appendix of Plumlee (2014): $\mathcal{X}_{j,1} = \{0.5\}$, $\mathcal{X}_{j,2} \backslash \mathcal{X}_{j,1} = \{0.125, 0.875\}$, $\mathcal{X}_{j,3} \backslash \mathcal{X}_{j,2} = \{0.25, 0.75\}$, $\mathcal{X}_{j,4} \backslash \mathcal{X}_{j,3} = \{0, 1\}$, $\mathcal{X}_{j,5} \backslash \mathcal{X}_{j,4} = \{0.375, 0.625\}$, $\mathcal{X}_{j,6} \backslash \mathcal{X}_{j,5} = \{0.1875, 0.8125\}$, $\mathcal{X}_{j,7} \backslash \mathcal{X}_{j,6} = \{0.0625, 0.9375\}$. Here $A \backslash B$ means objects that belong to set A and not to B. With the component designs given above, the sample size can be expressed as follows: $n = \sum_{k=0}^{\min(d, \eta-d)} 2^k \binom{d}{k} \binom{\eta-d}{k}$. For other types of component designs and their associated formulas for calculating sample size, please refer to Table 1 of Plumlee (2014).

4.2 Main Comparison

In this section, we consider the following six designs:

- rLHD
- mLHD

- trLHD
- Sobol
- uniform design
- completely random design

The three other designs: OA-based LHD, m2LHD and SGD that are not included in this section. OA-based LHD and m2LHD will be considered together in the next section as they are projection-based. SGD will be discussed in section 4.5. Before any formal comparison is conducted, Figure 4.1 visualizes 8 designs with $n = 27$, $d = 8$. Although OA-based LHD and m2LHD are not discussed in this section, they are included in Figure 4.1 to provide readers with an overall idea of the appearance of these designs. SGD is not include in Figure 4.1 because of its strict restriction on sample size. It shows only the projections onto the first two input variables. Those are the base designs used in the Borehole function, an example we will use as a testbed. The OA-based LHD is index 3, level 3, strength 2. The OA is downloaded from Neil Sloane's online catalog at `http://neilsloane.com/index.html#TABLES`.

We observe from Figure 4.1 that the completely random design has large holes. The random LHD improves but it still does not equally fill the space. As we expect, the trLHD allocates more points to the boundaries and mLHD, uniform design and Sobol are spread out over the region. For the two projection designs, its seems the m2LHD has a better space-filling property than the OA-based LHD.

4.2.1 Borehole Model

We consider the Borehole function as a test-bed to assess the impact of the design classes. There are 25 replicate experiments for each design considered, by random permutation of the columns of the base design visualized in Figure 4.1. A GaSP (Const, PowExp) model is fitted and the hold-out set is $10,000$ points generated by `randomLHS` in R. The assessment criteria are

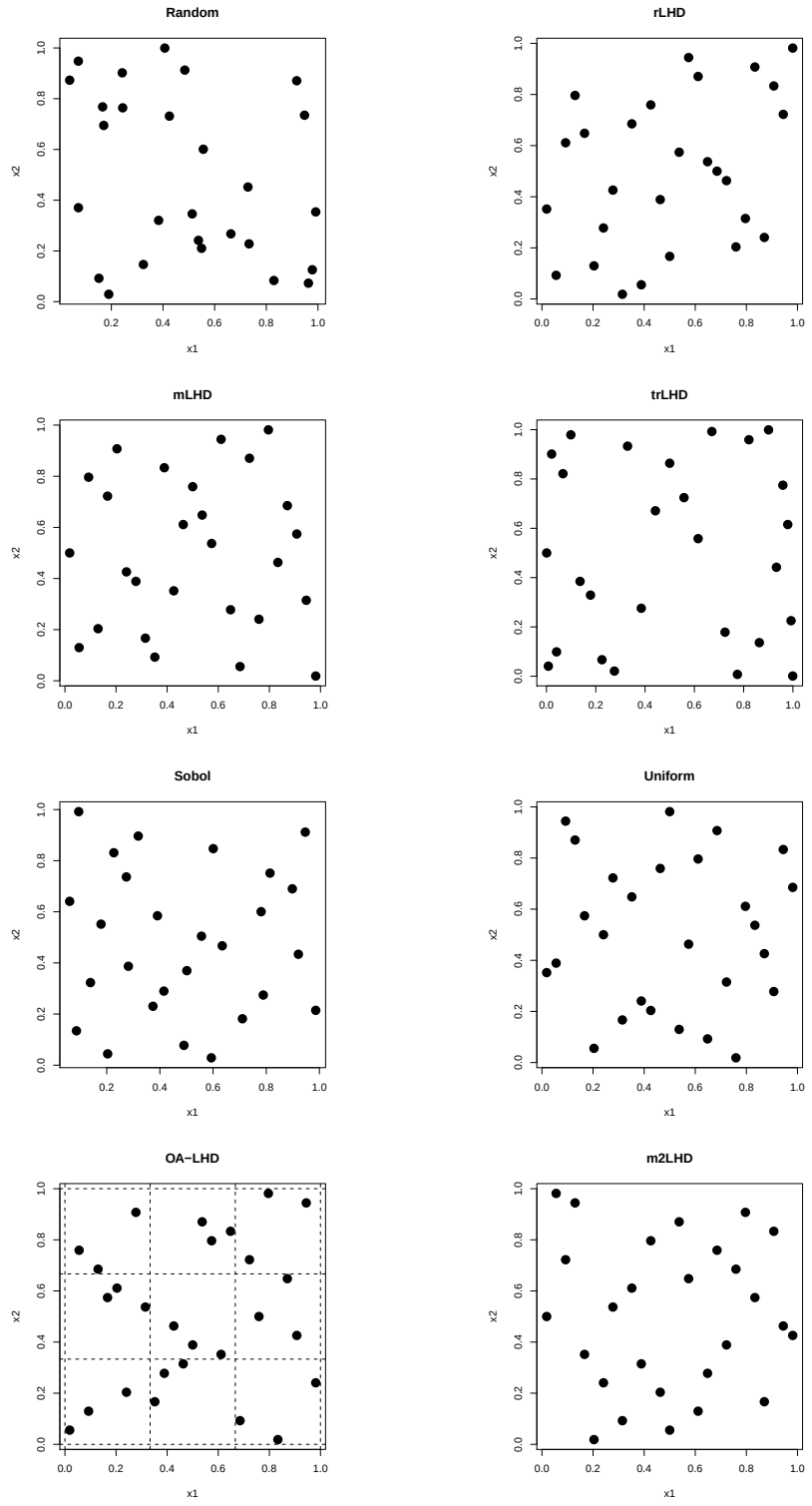

Figure 4.1: Completely random design, rLHD, mLHD, trLHD, Sobol, Uniform, OA-based LHD and m2LHD for $d = 8$ and $n = 27$. OA-based LHD is index 3, level 3, strength 2. The 8-dimensional designs are projected onto their (x_1, x_2) coordinates.

the $e_{\text{rmse, ho}}$ and the $e_{\text{max, ho}}$ in (2.5), while $\bar{y}$ is the mean of the true outputs of the $10,000$ points in the hold-out set to provide the same standardization for all the designs. Figure 4.2 show $e_{\text{rmse, ho}}$ for $n = 27$ for all the 8 designs.

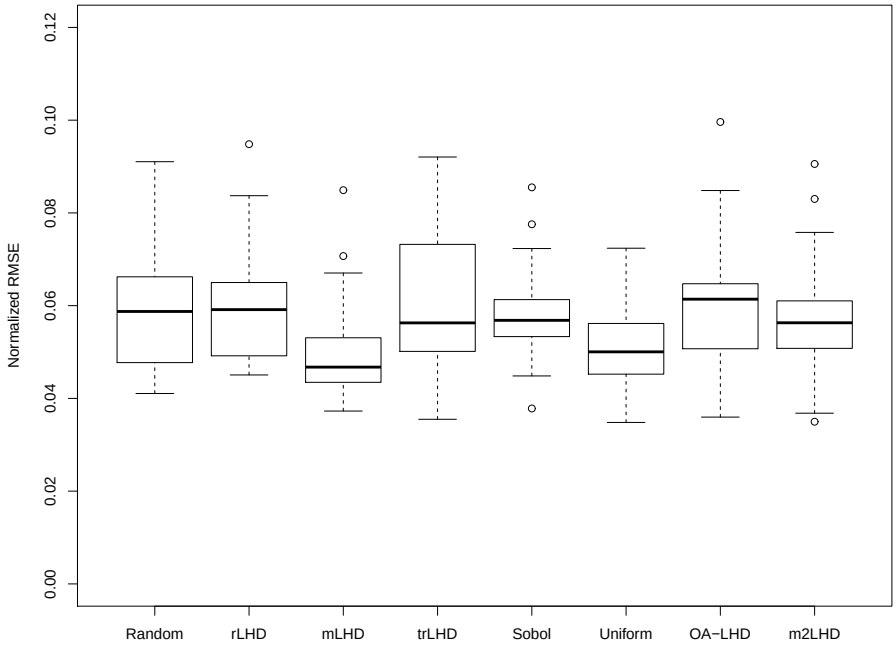

Figure 4.2: Borehole function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$, for 8 base designs, $n = 27$. For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. The OA-based LHD is index 3, level 3, strength 2.

This is a very interesting figure that shows the performance differences between designs are actually small: the uniform design and mLHD have the best overall performance and completely random design, trLHD and rLHD have the largest variations among the designs considered. m2LHD has a sightly better performance than OA-based LHD. However, none of the designs dramatically outperforms others and the overall performance is good as the normalized RMSEs for all designs are below 0.1. The plot for normalized

maximum absolute error shows a similar pattern although the trLHD design and the uniform design have the best overall performances. The results for the maximum error are reported in Figure C.1 in the appendix.

Although the uniform design has a good overall performance with $n = 27$ in the Borehole example, we are not able to find more uniform designs with larger sample sizes. Therefore, from now on, we will focus on the seven other designs only.

In addition, Figure 4.3 shows the $e_{\text{rmse, ho}}$ results for $n = 40, 80$. The results for $e_{\text{max, ho}}$ are reported in Figure 4.4. There is not enough space to report results for the completely random design in the above figures. However, it performs like the other four designs with larger variations though.

There are some similarities in Figures 4.3 and 4.4, that is, given the same sample size, the difference in prediction accuracy of the four designs is marginal, indicating that the effect of sample size overwhelms the impact of design class. Increasing the sample size from $n = 40$ to $n = 80$ reduces the RMSE values from around 0.03 to about 0.01, a factor of about 1/3. Furthermore, we notice that trLHD is worse than mLHD for $e_{\text{rmse, ho}}$, but has a better performance for $e_{\text{max, ho}}$, which measures the worst case. As it assigns more points to the boundaries, it is not surprising to see trLHD has a better performance than the other four design classes in terms of the worst case. Taking all factors into account, mLHD is the recommended design for emulating the Borehole function. Moreover, although we do not report results for the completely random design, its performance is actually not bad, which further reflects sample size is much more important than the choice of design. From now on, we will report the results for the normalized max absolute error in the appendix.

4.2.2 PTW Model

The next model we consider is the PTW model. We consider $n = 128$ and $n = 256$ and with the same settings as in the Borehole example, the results of normalized RMSE and normalized max error are reported in Figure 4.5 and in Figure C.2 in the appendix, respectively.

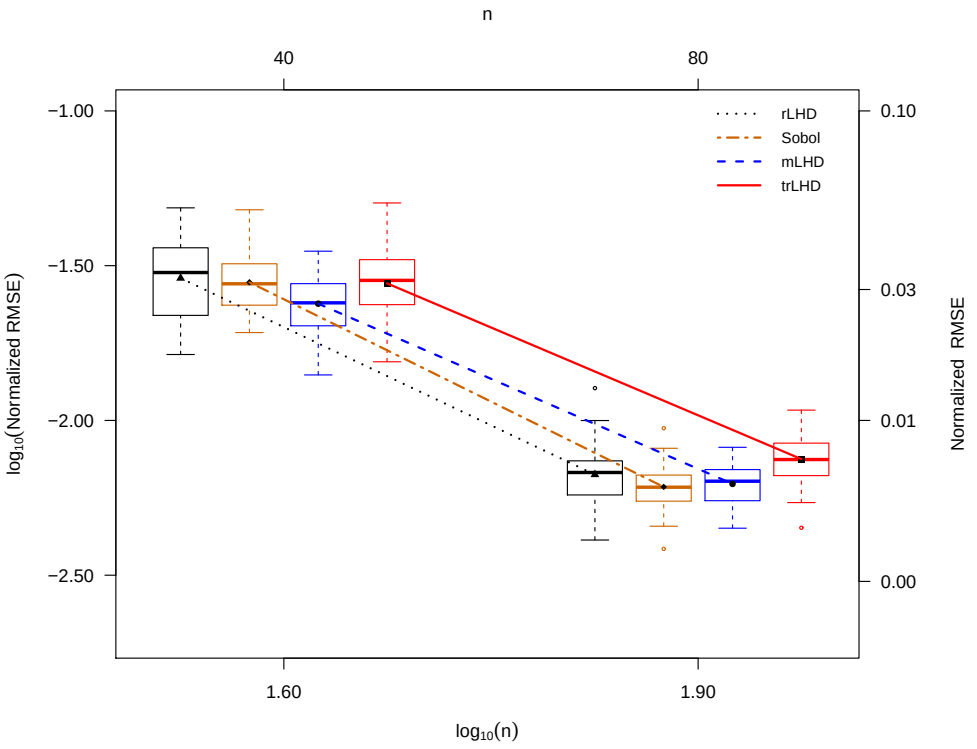

Figure 4.3: Borehole function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 40, 80$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

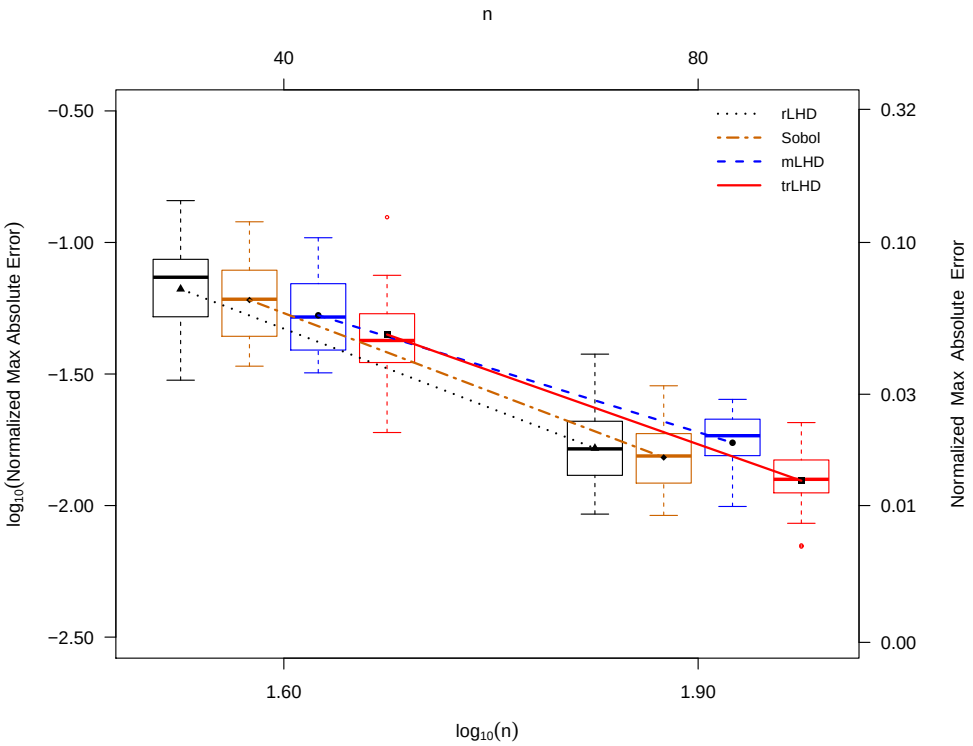

Figure 4.4: Borehole function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 40, 80$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Box plots are joined by their means.

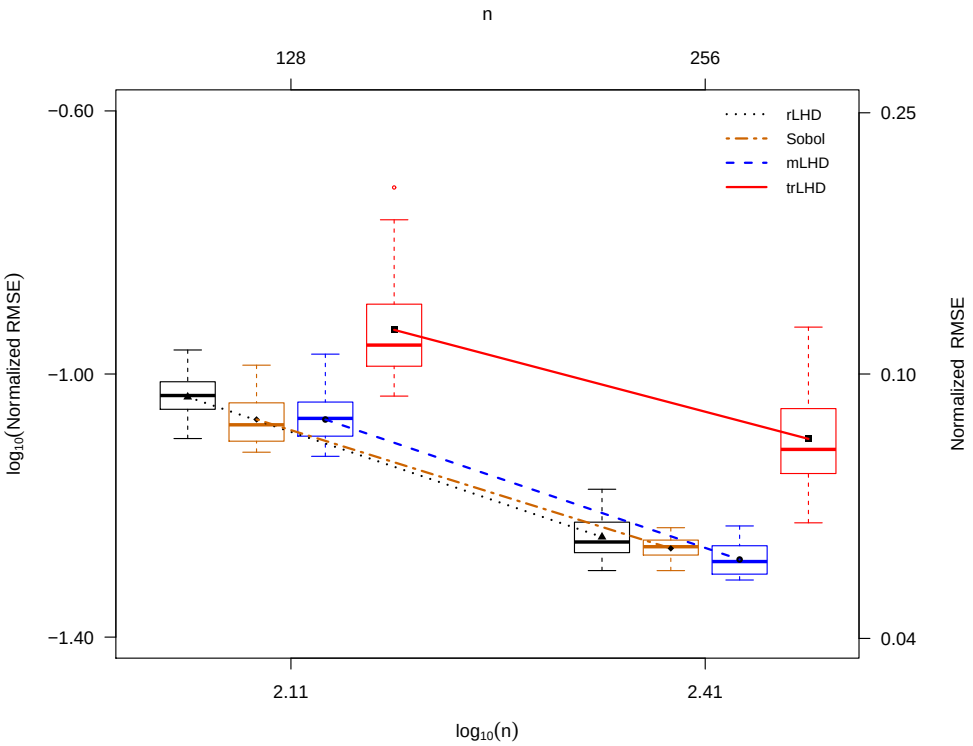

Figure 4.5: PTW function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 128, 256$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

The performances of the designs are similar to those in the Borehole example: trLHD has the worst normalized RMSE among the designs and the mLHD has the best overall performance. By increasing the sample size to 256, the accuracies of all designs have dramatically improved as observed previously. The results of normalized maximum error show a pattern consistent with the Borehole function: trLHD has the best overall performance, which suggests the effect of criterion is also non-trivial. It is, therefore recommended to use several criteria which measure different aspects of accuracy to obtain a more comprehensive conclusion.

4.2.3 Weighted Franke's Function

We consider the Franke's function (Franke, 1979). It was originally a 2-d function used as a test function in interpolation problems. We extend it to a 8 dimension function by adding x_i, x_{i+1}, where $i = 3, 5, 7$. Each pair x_i, x_{i+1} has the same form as x_1, x_2 and we further weight each pair. Our version of the weighted Franke's function is as follows:

$$\begin{aligned} f(\mathbf{x}) = \sum_{i=1,3,5,7} c_i \Bigg(& 0.75 \exp\left(-\frac{(9x_i - 2)^2}{4} - \frac{(9x_{i+1} - 2)^2}{4} \right) \\ & + 0.75 \exp\left(-\frac{(9x_i + 1)^2}{49} - \frac{(9x_{i+1} + 1)}{10} \right) \\ & + 0.5 \exp\left(-\frac{(9x_i - 7)^2}{4} - \frac{(9x_{i+1} - 3)^2}{4} \right) \\ & - 0.2 \exp\left(-(9x_i - 4)^2 - (9x_{i+1} - 7)^2 \right) \Bigg), \end{aligned} \quad (4.8)$$

where $c_1 = 1, c_3 = c_5 = c_7 = 0.1$. By assigning the above weights to the four pairs, the first 2 variables are the most important factors in the function. With sample sizes $n = 80, 200, 500$, we used the same settings as before. The results are presented in Figure 4.6 for normalized RMSE and in Figure C.3 for the normalized max absolute error.

From Figure 4.6, we observe a pattern seen for the Borehole function. That is, given the same sample size, mLHD is recommended, although the

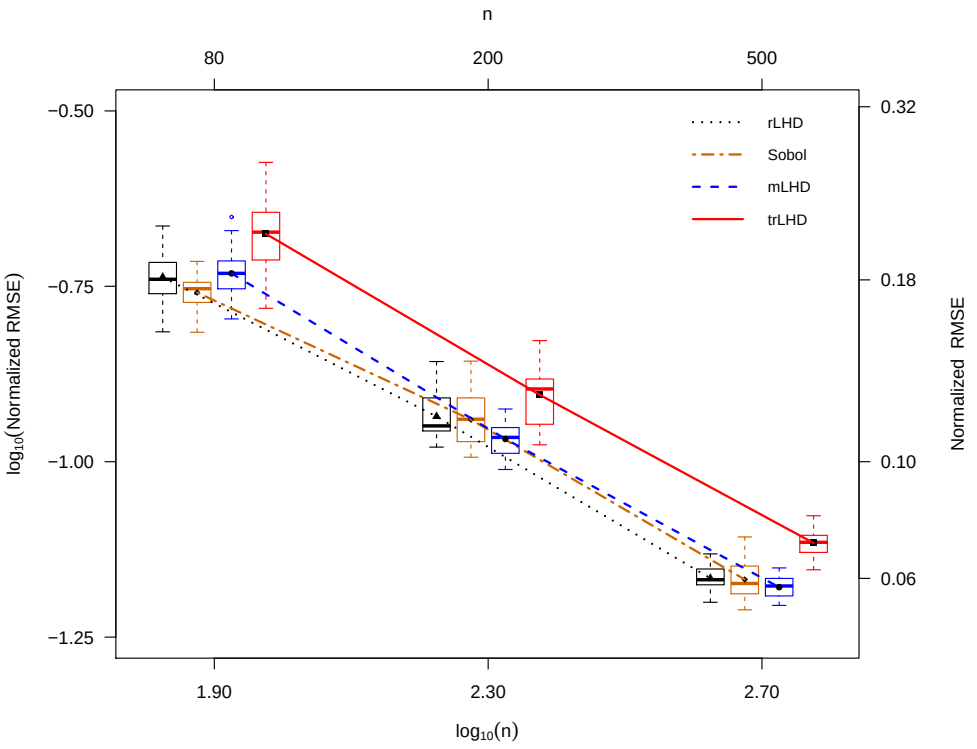

Figure 4.6: Weighted Franke's function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 80, 200, 500$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

performance differences between designs are actually small. trLHD proves to be poorer than the other designs. All the above three examples suggest that sample size has a much bigger effect than the choice of design.

4.2.4 Corner-peak Function: Original Scale

The corner-peak function is

$$y(\mathbf{x}) = \frac{1}{(1 + \sum_{j=1}^{d} c_j x_j)^{d+1}} \quad (0 < x_j < 1, c_j > 0 \text{ for } j = 1, \ldots, d). \tag{4.9}$$

The function is special in a way that it grows rapidly near the origin, if c_j is large. In other words, the difficulty of emulating such a function increases as c_j increases. With $d = 10$, determining the appropriate coefficients c_j requires some effort. After some trial and error, we have chosen to follow Barthelmann et al. (2000) and uniformly sample the c_j from 0 to 1 and rescale them such that $\sum_{j=1}^{10} c_j = 1.85$. By doing so, sparsity among the 10 sampled coefficients will be insured. The sampled coefficients, c_j used are here {0.0014 0.0286 0.0648 0.0714 0.0764 0.0963 0.2553 0.3297 0.4500 0.4761}. Those numbers are kept the same throughout this chapter.

In this section, we work with the corner-peak function on its original scale, i.e., no any transformation is taken on y. The function grows rapidly near the origin and even small designs of $n = 100$ have several orders of magnitude of variation in y. If some care is taken with appropriate statistical modelling, such as the use of the logarithm transformation, we expect the prediction accuracy will greatly improve with the same sample size, n.

We first report results with $n = 50, 100, 250$. As usual, the prediction accuracy is measured by the normalized RMSE ($e_{\text{rmse,ho}}$) and normalized maximum absolute error ($e_{\text{max,ho}}$) on a hold-out set consisting of $10,000$ points generated by a random LHD. The results are reported in Figure 4.7 for the normalized RMSE and in Figure C.4 in the appendix for the normalized maximum absolute error.

Figure 4.7 demonstrates that accuracy is poor with $n = 250$ or less for all

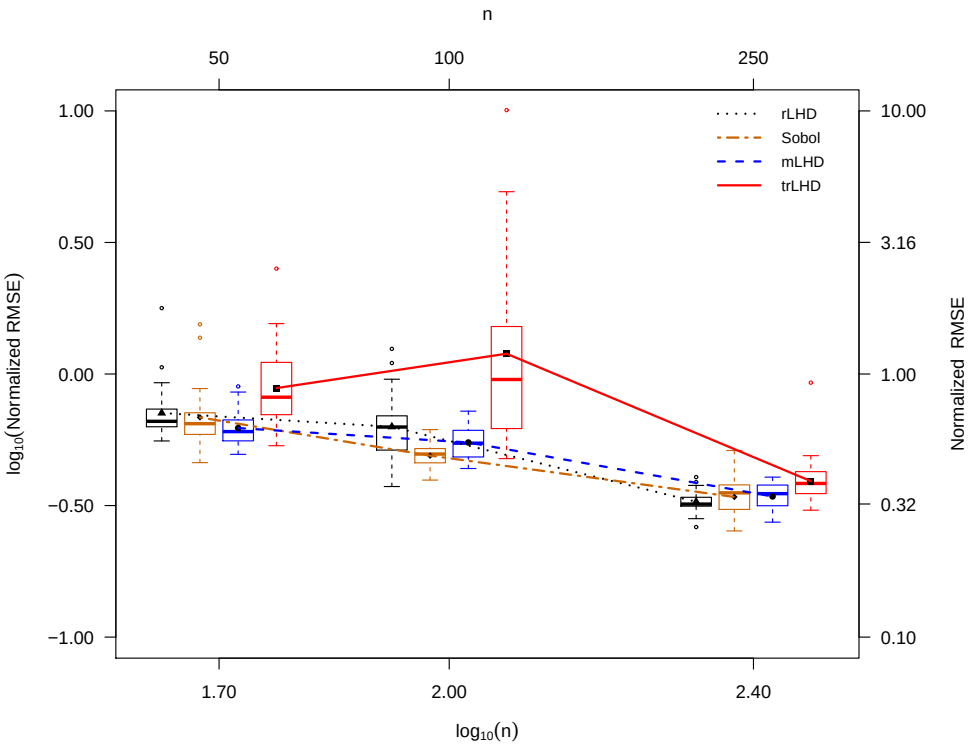

Figure 4.7: Corner-peak function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 50, 100, 250$ for for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

design classes. With $n = 250$, the normalized RMSEs are above 0.25, which indicates the function is difficult to predict on its original scale. However, by increasing the sample size from 50 to 250, the normalized RMSE decreases from about 0.7 to around 0.3 for most of the designs, suggesting the sample size has a non-trivial effect on the prediction accuracy even if the function is relatively hard to emulate. The differences between mLHD and rLHD are small and the trLHD performs very poor when sample size is $n = 50, 100$. However, as we observed before, the trLHD has the best performance in terms of the max error when $n = 250$ in Figure C.4. It is also worth noting that the variation among equivalent designs within the class of trLHD is huge.

4.2.5 Corner-peak Function: Logarithmic Scale

The corner-peak function in (4.9) is difficult to model because it grows rapidly near the origin. We report results on the corner-peak function after a logarithm transformation in section 4.2.5. In practice, it is often impossible to clearly identify the bottleneck. But one can easily generate an $n = 100$ mLHD design and compute the ratio of the maximum of y to the minimum of y. Permuting the columns of the mLHD 25 times will yield 25 of the ratios and the average of the 25 ratios we observed is 390.2. Such a huge number also suggests a logarithm transformation is needed for decent prediction accuracy.

Therefore, instead of working on the original y scale, we consider working with $\log(y(\mathbf{x}))$. When computing RMSE, however we convert back to the original scale. Using the same model settings, the results are reported in Figure 4.8 for the normalized RMSE and in Figure C.5 in the appendix for the normalized maximum absolute error. Sample sizes considered here are $n = 50, 100$ only, as $n = 100$ already yields very good accuracies for all the designs.

From Figure 4.8, it is clear that a huge improvement has been attained from working on the logarithmic scale: Even with $n = 50$, the normalized RMSE is already smaller than 0.1, which was impossible to obtain even with

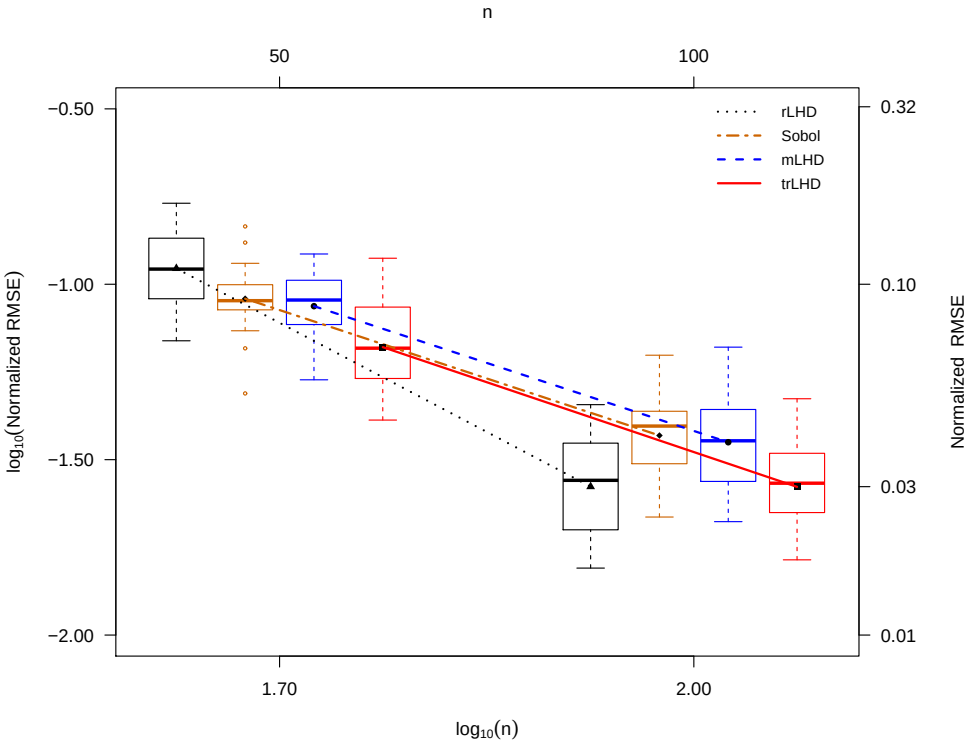

Figure 4.8: Corner-peak function analyzed at the logarithmic scale: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 50, 100$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

$n = 250$ at the original scale. By increasing the sample size to $n = 100$, all designs considered yield normalized RMSE that are around 0.03. Although trLHD performs relatively poorly on the original scale, it outperforms other designs on the logarithmic scale. Relative to the other designs, Sobol does not perform as well as it does on the original scale. The performance of the rLHD is actually very good when $n = 100$ in Figure 4.8. In terms of the maximum absolute error in Figure C.5, the trLHD still has the best overall performance.

Overall, from the comprehensive analysis of the corner-peak function, we can rank the importance of the following three factors.

$$\text{logarithm tranformation} > \text{sample size} > \text{design class}.$$

4.3 Comparison of Projection Designs

In this section, we consider the two projection-based designs: m2LHD and OA-based LHD with strength 2. The two designs are evaluated together here mainly because they both have certain projection properties: m2LHD is a variant of mLHD, where optimization is conducted based on (4.3) implying that the 2-d projection properties are the focus. Moreover, a strength 2 OA-based LHD means any two dimensions will have all combinations of levels appear exactly λ times, where λ is the index of an OA. Thus, both designs have their 2-d projection properties controlled or optimized to a certain extent. In addition, we also include mLHD as a reference.

The sample size is mainly determined by OA-based LHD as it has a strict restriction on it, while mLHD and m2LHD are flexible. In general, the maximum sample size of a function is chosen such that the majority of designs considered attain acceptable accuracies: usually the normalized RMSE should decrease to or below 0.1.

4.3.1 Borehole Function

We now consider two OA-based LHDs with $n = 32$ (index 2, level 4, strength 2), $n = 64$ (index 1, level 8, strength 2), respectively. The strength

is kept at 2 for all OA-based LHDs. The original OA designs are downloaded from the online catalog at `http://neilsloane.com/oadir/index.html` maintained by Neil J. A. Sloane and we convert them to OA-based LHDs using the first 8 columns only. m2LHDs with the same sample sizes are generated using R package *MaxPro* (Joseph et al., 2015). The mLHD is also included as the reference. The hold-out set is the same 10,000 points generated from a random LHD. We generate equivalent designs by randomly permuting the columns of the base design as before. The results for normalized RMSE are reported in Figure 4.9 and results for the normalized maximum absolute error are in Figure C.6 in the appendix.

We observe from Figure 4.9 that difference between m2LHD and mLHD is actually very small for both sample sizes. The performance of the OA-based LHD is worse than the two other designs and has a larger variation. When the sample size is 64, all of the designs considered have achieved very good accuracy: the normalized RMSE values are about 0.01. The maximum error results in Figure C.6 show a similar pattern.

4.3.2 PTW Function

The second function we consider in this section is the PTW function. With $n = 128$ and 256 and the same settings as above, the results for normalized RMSE and the normalized maximum absolute error are reported in Figures 4.10 and C.7, respectively. They show same pattern as that observed for the Borehole function in section 4.3.1.

4.3.3 Weighted Franke's Function

With the same setting as for the PTW function, the results for the normalized RMSE and the normalized maximum absolute error are reported in Figures 4.11 and C.8 in the appendix for the weighted Franke's function defined in (4.8).

We observe from Figure 4.11 that the performance of the OA-based LHD is still the worst although we have expanded the Franke function to have 2-dimensional interactions only and have presumably used designs with

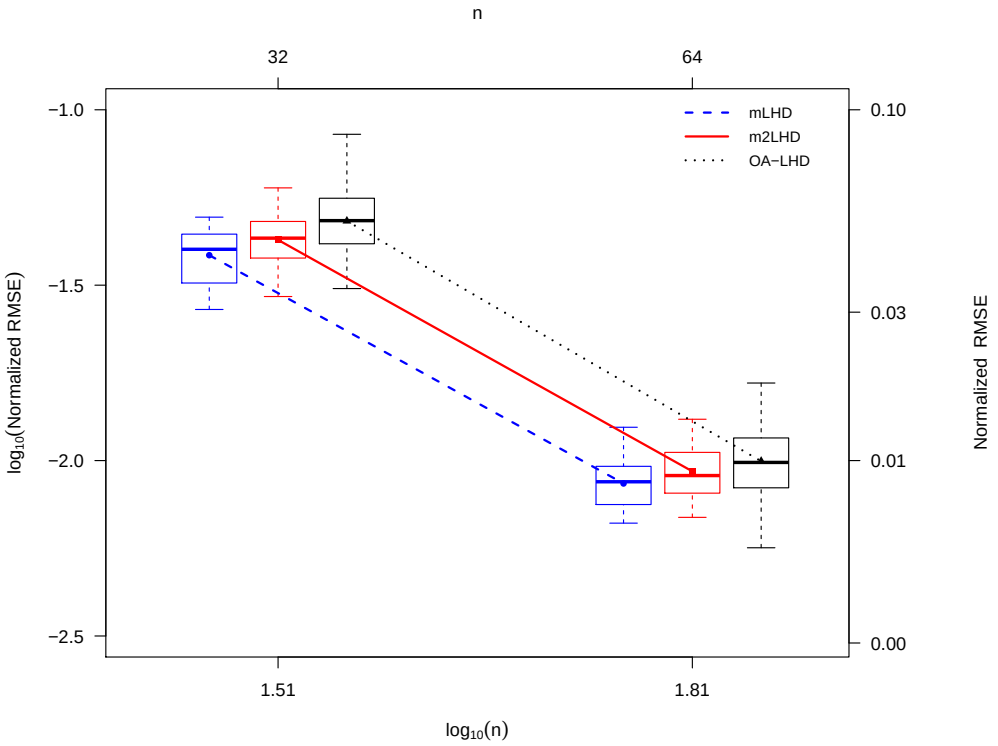

Figure 4.9: Borehole function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 32, 64$ for m2LHD, OA-based LHD designs and mLHD. The two OA-based LHDs are $n = 32$ (index 2, level 4, strength 2) and $n = 64$ (index 1, level 8, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

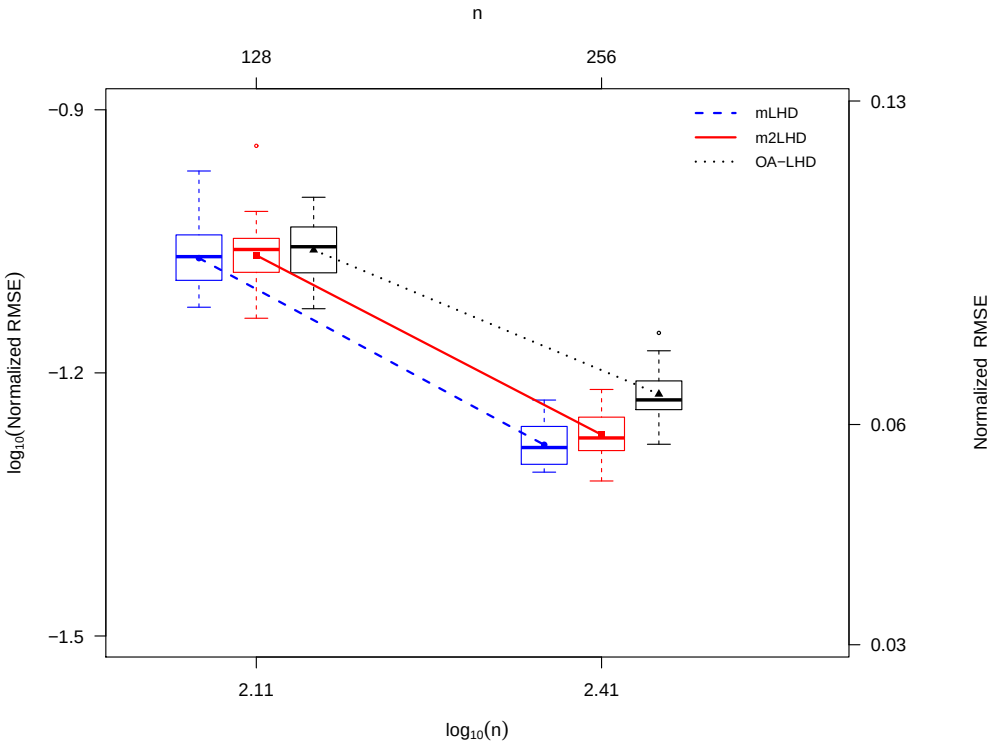

Figure 4.10: PTW function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 128, 256$ for m2LHD, OA-based LHD designs and mLHD. The two OA-based LHDs are $n = 128$ (index 2, level 8, strength 2) and $n = 256$ (index 1, level 16, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

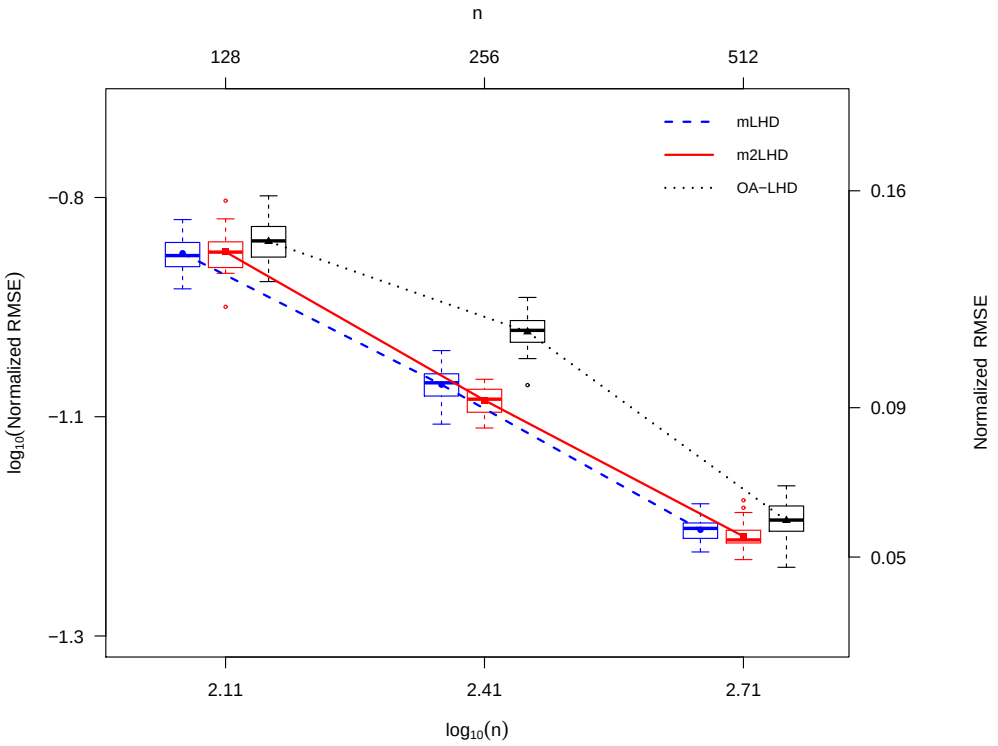

Figure 4.11: Weighted Franke function: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 128, 256, 512$ for m2LHD, OA-based LHD and mLHD. The three OA-based LHDs are $n = 128$ (index 2, level 8, strength 2), $n = 256$ (index 1, level 16, strength 2) and $n = 512$ (index 2, level 16, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

good 2-dimensional projection properties. Relatively, the OA-based LHD is much worse when the sample size increases to 256, but is comparable to mLHD and m2LHD when $n = 512$. mLHD and m2LHD have the best overall performance and the differences between the two designs is actually very small. This agrees with what we observed in previous examples. The results for the normalized max absolute error in Figure C.8 show similar trends.

4.3.4 Corner-peak Function: Logarithmic Scale

In this section, we consider the corner-peak function with the logarithmic transformation directly, since none of the designs can attain an acceptable prediction accuracy without a transformation of y within a reasonable sample size. The same $10,000$ points are used as the hold-out set and 25 equivalent designs are generated by permuting the columns of each base design. Unless mentioned otherwise, all the other settings are kept same as in section 4.3.1. The trLHD is added as another benchmark, as we observed in Figure 4.8 that trLHD has the best performance for the corner-peak function analyzed on the logarithmic scale. With $n = 64$ and 128, the results for normalized RMSE are reported in Figure 4.12 and the results for normalized maximum absolute error are report in Figure C.9

We observe in Figure 4.12 that on a relative scale, trLHD still has the best performance. OA-based LHD is as good as trLHD when the sample size is 64, however, it deteriorates when the sample size increases to 128. mLHD and m2LHD have similar performance as seen in previous examples. The trends are consistent with Figure C.9 for the normalized maximum absolute error in the appendix.

4.4 Non Space Filling Design – SGD

The SGD reviewed in section 4.1 is less space-filling. Because of its special structure, the primary goal of the SGD is to facilitate reasonably fast computations of predictions when the sample size becomes big enough

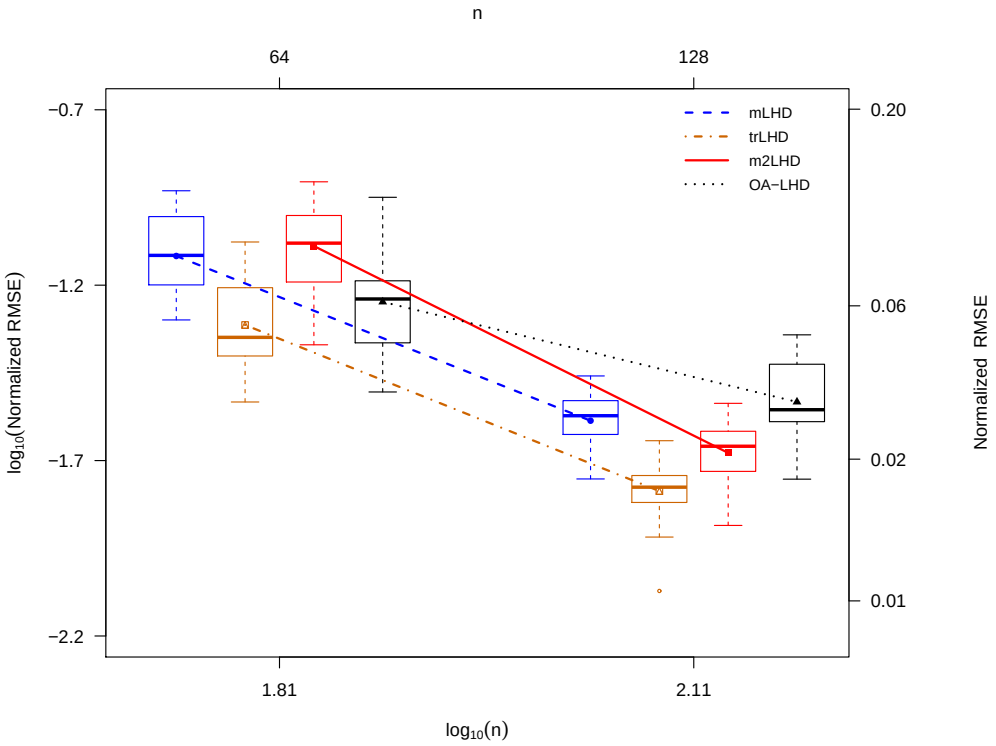

Figure 4.12: Corner-peak function analyzed at the logarithmic scale: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 64, 128$ for m2LHD, OA-based LHD and mLHD. The two OA-based LHDs are $n = 64$ (index 4, level 4, strength 2) and $n = 128$ (index 2, level 8, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Box plots are joined by their means.

that other designs would not be able to yield predictions within a feasible time frame. That being said, we observe in this section that the prediction accuracy of SGD is usually much worse than that of an mLHD when the sample size is not large. We argue that when sample size is small, for example $n < 1000$, and prediction accuracy is a concern, we would not recommend SGD.

Figure 4.13 shows two SGD designs jittered by a small amount of noise to show replicates with $d = 8, \eta = 10$ and $d = 8, \eta = 11$. We only plot the first two input variables. It is clear that SGD is no longer a space-filling design as we can see large holes from Figure 4.13.

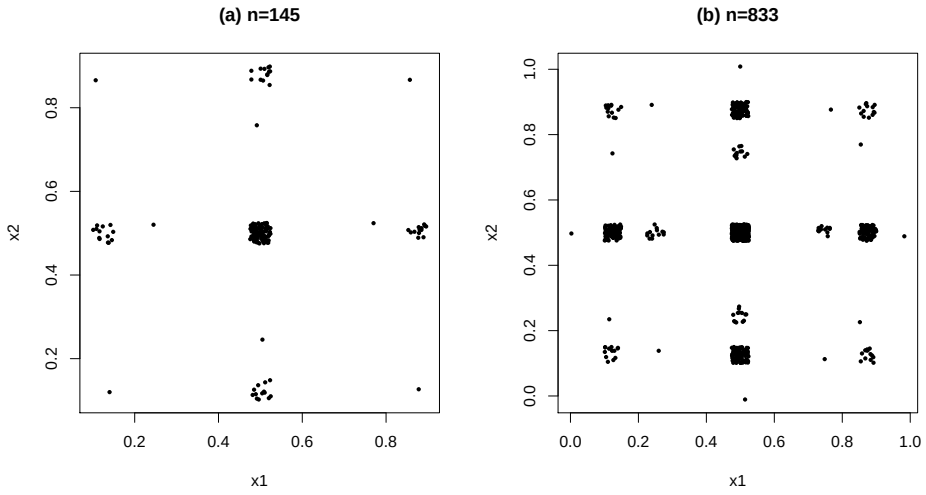

Figure 4.13: Two SGD designs with $d = 8, \eta = 10$ and $d = 8, \eta = 11$, respectively. The designs are jittered by a small amount of noise to show replicates and the sample sizes are $n = 145, 833$, repectively. Only the x_1, x_2 are plotted.

Results of two examples are reported as follows. The first example we consider is the Borehole function. Two SGDs are generated: one is n=145 with $\eta = 10$ and the other is $n = 833$ with $\eta = 11$. Those are the designs plotted in Figure 4.13. The component designs used for constructing the two

SGDs are suggested by Plumlee (2014). Moreover, these are two reasonable sample sizes to make a comparison for the Borehole model with mLHD, for example. Given an SGD, it is impossible to make equivalent designs by permuting its columns. Therefore, the SGD will only report one number for its normalized RMSE. We include mLHD with the same sample size as a competing design, for which we permute the columns to generated 3 equivalent designs. In addition to the two criteria: normalized RMSE and normalized maximum absolute error, we also report the median absolute prediction error (MAPE). MAPE is one of the criteria considered in the original paper (Plumlee, 2014). It is the median of the absolute prediction errors of all points in the hold-out set (not standardized), which is robust to extreme prediction errors. The hold-out set is the same $10,000$ points generated from a rLHD as before. The results are reported in Table 4.1.

	n=145		n=833	
	SGD	mLHD	SGD	mLHD
Normalized RMSE	0.0610	0.0022	0.0441	0.0002
Normalized Max Absolute Error	0.1701	0.0067	0.1691	0.0003
Median Absolute Prediction Error	0.6516	0.0401	0.3247	0.0034

Table 4.1: Borehole function: prediction results for SGD and mLHD, $n = 145,833$. Three predcition criteria are considered. There is one design for SGD. For mLHD, 3 random permutations of its columns give 3 values of each criterion and the mean of the 3 numbers is reported.

From Table 4.1, the performance difference is clear: given $n = 145$, the normalized RMSE of SGD is 0.061, which is about 30 times bigger than that of the mLHD. The story persists when we talk about MAPE, the criterion used by Plumlee (2014): the MAPE of SGD is 0.65 and is 16 times bigger than that of the mLHD. When sample size increases to $n = 833$, the difference is even larger: the MAPE of SGD is 0.3247, which is now 95 times bigger than that of the mLHD.

Even though an SGD, which has computational advantages, is useful when the size is really large, the prediction performance for the Borehole model, where the sample size (n=145, 833) is relatively affordable and the

dimension is high (d=8), is not comparable to mLHD. The example suggests that an SGD design could potentially have a huge negative impact on the prediction accuracy.

The second example on which we report is the $d = 10$ corner-peak function both on the original scale and on the logarithmic scale. The sample size for SGD is $n = 221$ with the same component designs. An mLHD with the same size is generated as the benchmark. Other settings are kept the same. The results are reported in Table 4.2. Again for all three criteria considered, the performance of SGD is much worse than that of mLHD, except in that the MAPE of SGD is comparable to that of the mLHD for the corner-peak function at its original scale. The MAPE is largely determined by prediction accuracy away from the localized peak, where the function is easy to model by any method.

		SGD	mLHD
CP	Normalized RMSE	0.6626	0.3095
	Normalized Max Absolute Error	0.8794	0.4068
	Median Absolute Prediction Error	2.28×10^{-4}	2.03×10^{-4}
CP log	Normalized RMSE	0.0675	0.0091
	Normalized Max Absolute Error	0.1403	0.0233
	Median Absolute Prediction Error	3.01×10^{-6}	4.64×10^{-7}

Table 4.2: Corner-peak function and Corner-peak function analyzed on the logarithmic scale: Prediction results for SGD and mLHD design, $n = 221$. Three prediction criteria are considered. There is one design for SGD. For mLHD, 3 random permutations of its columns give 3 values of each criterion and the mean of the 3 numbers is reported.

4.5 Discussion

4.5.1 The Effect of Regression Component

We have already discussed the effect of the method of analysis in chapter 2 of the thesis. As interaction between the method and design may exist, which will certainly further complicate the evaluation. We consider using

a full linear GP model to refit the corner-peak function analyzed on the logarithmic scale. The corner-peak function is an example where switching from a constant mean GP to a full linear GP might bring some benefits to the prediction accuracy. In fact, by taking a logarithmic transformation when analyzing the corner-peak function, we have already included the effect of the analysis, whose impact on prediction accuracy is astonishingly large as the performances of all designs have dramatically improved on the logarithmic scale.

With the same settings, the results for normalized RMSE are reported in Figure 4.14 and the results for normalized maximum absolute error are reported in Figure C.10 in the appendix. The vertical axis limits are kept the same as those in Figures 4.8 and C.5 to facilitate direct comparison.

From the two plots, it is very clear that for small sample size (n=50), results for the full linear GP are worse, which indicates the FL model actually hinders the prediction performance. For larger sample sizes, the performances are similar for the two GP models. The observations are consistent with those made in chapter 2.

4.5.2 Grid Generation

Grid generation is the way the points are generated. To be more specific, suppose we want to generate a rLHD with $n = 10$ in one of the dimensions. There exists at least three choices of points:

- Type 1: A random point each in $(0, 0.1), (0.1, 0.2), \ldots, (0.9, 1)$.
- Type 2: $0/9, 1/9, 2/9, \ldots, 9/9$.
- Type 3: $1/20, 3/20, \ldots, 19/20$.

Therefore, even if researchers call it a random LHD, different people may choose different ways to generate such a rLHD. In short, variants do exist though they all bear the same name. The major difference among the above three types depends on whether to include the boundaries or not. We can see that a type 2 grid does include 0 and 1, while type 1 and type 3 tend to

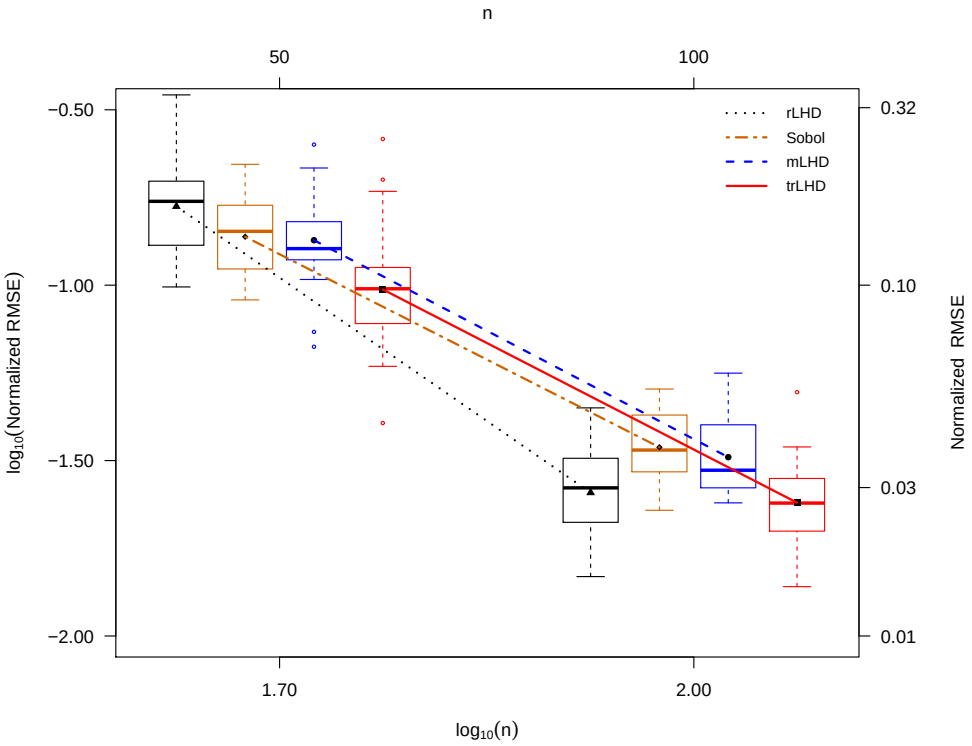

Figure 4.14: Corner-peak function analyzed at the logarithmic scale with a full linear GP model: Normalized RMSE of prediction, $e_{\text{rmse, ho}}$; $n = 50, 100$ for rLHD, Sobol, mLHD and trLHD designs.. For each base design, 25 random permutations of its columns give 25 values of $e_{\text{rmse, ho}}$, displayed as a box plot. Boxplots are joined by their means. The vertical axis limits are kept the same as in Figure 4.8 to faculitate direct comparison.

generate "middle" points. In this chapter, we use the R packages *MaxPro*, *SLHD* to generate m2LHD and mLHD, respectively. It turns out that both R packages take type 3 grids. Hence, in order to be consistent, we use type 3 grids as well when generating rLHDs, OA-based LHDs. For example, we convert an OA to an OA-based LHD by $\frac{i-0.5}{n}$, where n is the sample size, i is an integer in the original OA. In summary, all the designs considered in this chapter do not include the boundaries. By pushing points towards the boundaries, the trLHD designs have enhanced sampling near the boundary, however.

4.6 Some Concluding Remarks

In this chapter, we first reviewed several different designs in section 4.1 and used some examples to assess the prediction performance of the designs in section 4.2, 4.3 and 4.4 . Some important issues are discussed in section 4.5. The major contributions are summarized as follows.

1. Given the same sample size, we find mLHD has the best overall prediction accuracy. Therefore, mLHD is the default design we suggest if there is no prior information available. Actually, the prediction performances of different designs are not as large as we might expect. Designs, such as rLHD and Sobol, which are trivial to generate, have shown acceptable performances. For the two projection designs, m2LHD has a performance similar to that of mLHD. However, the OA-based LHD is notably worse than mLHD. The performance of SGD is worse than mLHD for all the examples considered. However, it is not necessarily interpreted as a drawback as its primary objective is fast prediction when sample size is big enough to disable other designs. Finally, we want to emphasize again that it could be risky to equate attractive theoretical results of a design with a good prediction accuracy.

2. Sample size and transformation of y have a much bigger effect on the prediction performance. In general, increasing sample size can greatly improve prediction accuracy. Transforming y requires knowing the nature of the function. For instance, we know a logarithmic transformation can

help since we are aware that the function grows rapidly in part of the input space. In practice, one can first conduct a pilot experiment on the computer model to see if the function varies by order of magnitude.

Recall that for the corner peak function, the challenge is that y grows rapidly when $\mathbf{x}$ is close to the origin. Without a logarithmic transformation of y, none of the designs is able to yield a good RMSE. Since the designs considered in this chapter are all "fixed" designs in the sense that the experimental points are fixed once generated, poor prediction performance on the original scale also suggests the inability of fixed designs in analyzing and predicting functions showing non-stationarity in a certain region. This motivates the use of a sequential design, which is the topic of the next chapter.

Chapter 5

Sequential Computer Experimental Design for Estimating an Extreme Probability or Quantile

5.1 Introduction

Consider the problem of estimating an extreme tail probability or quantile of the output of a computer model. Let the inputs, $\mathbf{X} = \{X_1, \ldots, X_d\}$ be d random variables, the output $Y = y(\mathbf{X})$, as a function of $\mathbf{X}$, is a random variable as well. Taking probability estimation as an example, given y_f as a critical value of y of interest, a research problem of engineering interest is to provide an estimate of $p_f = P(y(\mathbf{X}) > y_f)$. If y_f is large enough rendering the said probability close to 0, p_f is a right tail probability. A left tail probability is analogous. In addition, the tail quantile estimation is an inverse problem of the probability estimation. It searches for the $1 - p_f$ quantile y_f, such that $P(y(\mathbf{X}) < y_f) = 1 - p_f$, where p_f is given. In a nutshell, both require the accurate depiction of the tail of the distribution $Y = y(\mathbf{X})$. The focus of the chapter is to provide a good estimate of the extreme tail probability or quantile when the computer model is expensive to run.

The computer model of interest in the chapter is a model of a floor system. The inputs are d values of the Modulus of Elasticity (MOE) of the d joists and the output is the maximum deflection under a fixed static load. If the maximum deflection exceeds a cut-off deflection, the system

will fail. The details of this particular computer model will be introduced in section 5.2. Moreover, the general research objective discussed in the previous paragraph can be stated in the specific context of the computer model of a floor system as follows: given the cut-off deflection y_f, it is of interest to estimate the failure probability $p_f = P(y(\mathbf{X}) > y_f)$. Conversely, given the failure probability p_f, it is of interest to estimate the quantile, y_f such that $p_f = P(y(\mathbf{X}) > y_f)$.

If it were feasible to evaluate the computer model many times, the Monte Carlo (MC) method can be used directly to provide an estimate of the tail probability or quantile. Suppose there is a discrete MC set, $\mathcal{X}_{\text{MC}} = \{\mathbf{x}_{\text{MC}}^{(1)}, \ldots, \mathbf{x}_{\text{MC}}^{(N)}\}^T$ generated from the distribution of $\mathbf{X}$. How to appropriately choose the input distribution is a topic that will be addressed in section 5.5. Plugging $\mathcal{X}_{\text{MC}} = \{\mathbf{x}_{\text{MC}}^{(1)}, \ldots, \mathbf{x}_{\text{MC}}^{(N)}\}^T$ into the computer model, one can obtain the N true outputs $\mathbf{y}(\mathcal{X}_{\text{MC}}) = \{y(\mathbf{x}_{\text{MC}}^{(1)}), \ldots, y(\mathbf{x}_{\text{MC}}^{(N)})\}^T$. Then the empirical distribution of the output Y is given by

$$\hat{F}(w) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y(\mathbf{x}_{\text{MC}}^{(i)}) < w), \tag{5.1}$$

where $\mathbb{1}(E) = 1$ if event E is true and 0 otherwise. The failure probability or quantile are then the (approximate) solutions of $p_f = 1 - \hat{F}(y_f)$. According to the famous Glivenko-Cantelli theorem (Cantelli, 1933, Glivenko, 1933), the supremum of the difference between the empirical distribution and the true CDF converges to 0 almost surely under regularity conditions as N increases. However, the novel part of the research is that the computer code is expensive and so it is not affordable to evaluate the computer code many times to obtain a large sample. Consider an expensive computer model with limited experimental budget, for example $N = 40$. It would be extremely inaccurate if we estimate the empirical distribution of Y based on the $N = 40$ real outputs directly.

The solution here is to use the Gaussian process (GP) model (Sacks et al., 1989, Santner et al., 2003) as a cheap statistical proxy for the expensive computer model. The primary idea is as follows: first generate a discrete train-

ing set $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}^T$ and obtain $\mathbf{y}(\mathcal{X}) = \{y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)})\}^T$ through the computer model. The available data enables one to build a GP model. Then, obtain the predicted values from the trained GP model: $\hat{\mathbf{y}}(\mathcal{X}_{\text{MC}}) = \{\hat{y}(\mathbf{x}_{\text{MC}}^{(1)}), \ldots, \hat{y}(\mathbf{x}_{\text{MC}}^{(N)})\}^T$. The failure probability p_f is estimated by $1 - \hat{F}(y_f)$, where the true outputs $\mathbf{y}(\mathcal{X}_{\text{MC}})$ are replaced by $\hat{\mathbf{y}}(\mathcal{X}_{\text{MC}})$. The estimate of the quantile can be obtained in an analogous way.

There are several ways to choose the training set for fitting the GP. The first way, which we call a fixed design strategy, is to use up all of the design budget n, to train one statistical surrogate, which is employed to predict the outputs of the points in the MC set. The fixed strategy is simple and fast, but the estimation accuracy may not be good enough, especially for estimating an extreme tail probability or quantile. The second way, a sequential design strategy, is to use part of the design budget at the beginning to train an initial GP and sequentially adds points into the design space (one at a time), guided by a search criterion until the budget has been exhausted. Each time a new point is added, both the surrogate model and the probability/quantile estimate get updated. Compared with the fixed design strategy, the second strategy is a dynamic process that allows new information being added "on the fly", thus should intuitively, provide more accurate estimation. Better estimation outcomes using sequential methodology is well recognized; see Jones et al. (1998) and Ranjan et al. (2008) for examples.

The core of a sequential method is the search criterion. Jones et al. (1998) proposed the expected improvement criterion (EI) based on an improvement function for global optimization and the EI criterion has been popular during the past 20 years. In general, there are different improvement functions for different statistical objectives. Ranjan et al. (2008) introduced an improvement function for contour estimation, i.e., search $\mathbf{x} = \{x_1, \ldots, x_d\}$ such that $y(\mathbf{x}) = a$, where a is the targeted level. Roy and Notz (2014) used two different criteria for estimating a quantile where $p_f = 0.2$: the first one is also based on EI and used nearly the same improvement function as Ranjan et al. (2008) for estimating a quantile. The second criterion of Roy and Notz (2014) is the so-called hypothesis testing-based criterion, which opti-

mizes a "discrepancy" between the current prediction and the response at any untried input set. The details will be reviewed in section 5.3.

The work we propose in this chapter takes such a sequential design approach. The focus is to provide answers and recommendations to some practical questions an engineer asks when applying the sequential design strategy. They are as follows.

- There are two search criteria: EI and the hypothesis testing-based criterion for quantile and probability estimation. Which yields a more accurate result, especially for extreme probability and quantile estimation? Roy and Notz (2014) compared the performance of sequential designs and fixed designs and showed sequential designs produced more accurate quantile estimates. However, they did not consider probability estimation. None of their examples covered extreme quantile estimation. In this chapter, we explicitly contrast the performances of EI and the hypothesis testing-based criterion for quantile and probability estimation. We conclude that the hypothesis testing-based criterion has a faster convergence to its target and hence is preferred.

- How to select the input distribution for $\mathbf{X}$? It is noted that the distribution of Y is fully determined by the distribution of $\mathbf{X} = \{X_1, \ldots, X_d\}$ through the deterministic computer model. We explore several different ways of modelling the input distribution.

- How to generate the MC set? One obvious option is to take a simple random sample from the input distribution. But the required sample size may be computationally inefficient even with GP prediction. In this chapter, we adopt a stratified sampling scheme to generate a smaller MC set such that the extreme probability is well estimated.

- How to select the initial design for training? We explore two different choices and compare their performances in an engineering model.

- Based on the hypothesis testing-based criterion, we generate diagnostic plots to check if the sequence has converged or not. It is critically

important to know when to stop in practice when the "true" probability and quantile are unknown.

- When estimating the unknown parameters of a GP, we use the Bayesian method proposed in chapter 3 instead of the MLE paradigm. Both the EI and the hypothesis testing-based criteria contain the standard error of the response at an untried point. Bayesian methods are able to fully quantify the parameter estimation uncertainty. Therefore, it is preferred to use Bayesian methods to train a GP model and predict in sequential design.

Those are the topics we will address in this chapter, which are very practice-oriented. It is clear that this chapter involves several details that did not draw much attention in the literature. In summary, the major contribution we make here is to fill in the details which close the gap between the theory of sequential design and its practical use.

The rest of the chapter is organized as follows: In section 5.2, we introduce the wood computer model used to illustrate the methodology. In section 5.3, the sequential algorithm and the aforementioned criteria are reviewed. A study based on an engineering model is reported in section 5.4. The wood computer model is revisited in section 5.5. Some concluding remarks are made in section 5.6.

5.2 Computer Model of a Floor System

The application that motivates this research is a numerical model of a floor system (McCutcheon, 1984). The performance characteristic of interest is the floor's maximum deflection under a static load. The d inputs $x_1, \ldots,$ x_d to the model are the modulus of elasticity (MOE) values in units of psi of d supporting joists. For instance, Figure 5.1 shows a system with $d = 8$ joists. The load could be a further input, but it will be kept constant in the analysis of section 5.5 because we are more interested in the relationship between MOEs and the floor's maximum deflection. Given the joist MOE inputs, the computer model outputs the d deflections of the d joists, and

the maximum deflection is taken as the output of interest, y (units in); see Figure 5.2. The computer code will be treated as a black-box, i.e., we do not

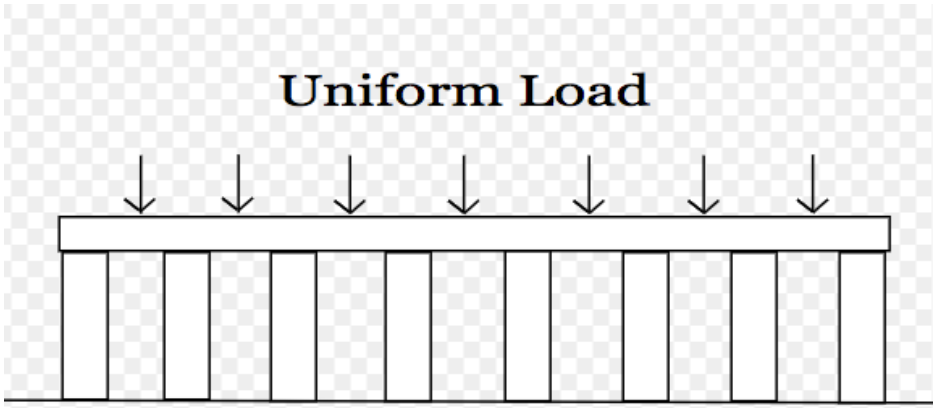

Figure 5.1: Floor with $d = 8$ joists (supporting beams). The 8 joists act like springs, i.e., they deflect under a load.

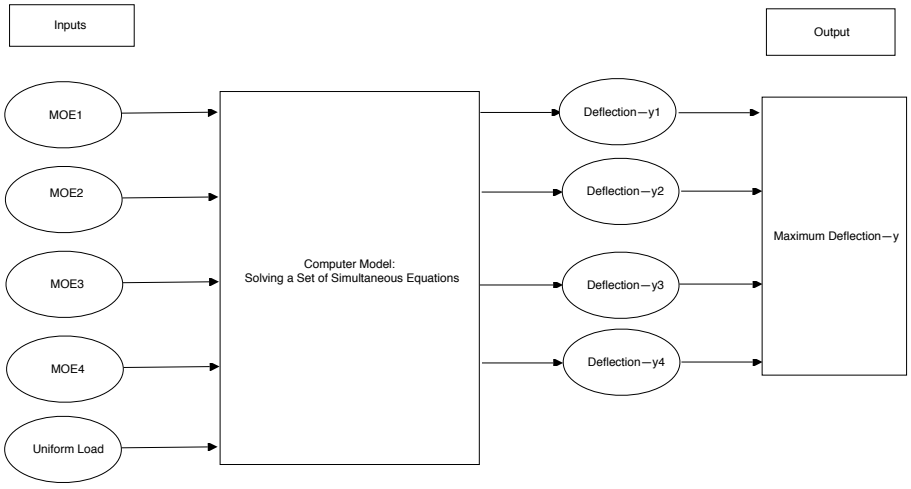

Figure 5.2: Computer model of a floor system with $d = 4$ joists.

take account of the form of the equations in the mathematical representation.

The code is deterministic, but if we think of the inputs $x_1, \ldots, x_d$ as realizations of random variables $X_1, \ldots, X_d$, then the output deflection y is a realization of a random variable Y. The reliability problem is to estimate either:

- The failure probability $p_f = P(Y > y_f)$, for a given maximum allowable deflection y_f; or
- The quantile y_f such that $P(Y > y_f) = p_f$, where p_f is a given small failure probability.

5.3 Sequential Experimental Design

Here we describe the sequential design strategies for estimation of a tail probability or quantile. The heart of a sequential algorithm is the criterion for adding new evaluations to the search, and we contrast existing criteria.

5.3.1 Sequential Algorithms

The basic idea is to apply MC using the GP predictions of $y(\mathbf{x}_{\mathrm{MC}}^{(i)})$ for all points in the MC set $\mathcal{X}_{\mathrm{MC}}$ rather than evaluating the computer model. Given initial training data of n_0 evaluations, $y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n_0)})$, and hence a trained GP, Algorithms 2 and 3 compute tail probability and quantile estimates, respectively. They add n_+ points sequentially according to the search criteria in sections 5.3.2 and 5.3.3.

Note that in section 5.5, the probability estimate in step 6 of algorithm 2 will be replaced by the stratified random sampling estimate in (5.10). Step 6 in algorithm 3 for quantile estimation will be replaced in an analogous way. Both algorithms are based on the method of Ranjan et al. (2008) for contour estimation up to the search criterion. Mapping out a given contour where $y(\mathbf{x}) = y_f$ is equivalent to separating the MC points into those where $y(\mathbf{x}_{\mathrm{MC}}^{(i)}) > y_f$ versus those where $y(\mathbf{x}_{\mathrm{MC}}^{(i)}) \leq y_f$. In Algorithm 2 it is straightforward to estimate p_f based on the $\hat{y}(\mathbf{x}_{\mathrm{MC}}^{(i)})$ (Bichon et al., 2009). For quantile estimation, Algorithm 3 estimates the unknown quantile

Algorithm 2 Return an estimate of $p_f = P(Y > y_f)$ for a given y_f

function ProbabilityEstimate($n_0, n_+, \mathcal{X}, \mathbf{y}, y_f$)
  $n = n_0$
  for $i = 1$ to n_+ **do**
    Use the current training data, $\mathcal{X}$ and $\mathbf{y}$, to fit the GP.
    Compute predictions $\hat{y}(\mathbf{x}_{\text{MC}}^{(1)}), \ldots, \hat{y}(\mathbf{x}_{\text{MC}}^{(N)})$ for the MC set
    $\hat{p}_f = (1/N) \sum_{i=1}^{N} \mathbb{1}(\hat{y}(\mathbf{x}_{\text{MC}}^{(i)}) > y_f)$
    if $i < n_+$ **then**
      Choose $\mathbf{x}_{(n+1)}$ from $\mathcal{X}_{\text{cand}}$ based on a search criterion
      Add $\mathbf{x}_{(n+1)}$ to $\mathcal{X}$
      Evaluate $y(\mathbf{x}_{(n+1)})$ and add it to $\mathbf{y}$
      n is replaced by $n + 1$
  return $\hat{p}_f$

Algorithm 3 Return an estimate of y_f, where $P(Y > y_f) = p_f$ for a given p_f

function QuantileEstimate($n_0, n_+, \mathcal{X}, \mathbf{y}, p_f$)
  $n = n_0$
  for $i = 1$ to n_+ **do**
    Use the current training data, $\mathcal{X}$ and $\mathbf{y}$, to fit the GP.
    Compute predictions $\hat{y}(\mathbf{x}_{\text{MC}}^{(1)}), \ldots, \hat{y}(\mathbf{x}_{\text{MC}}^{(N)})$ for the MC set
    Compute $\hat{y}_f$ such that $p_f \simeq (1/N) \sum_{i=1}^{N} \mathbb{1}(\hat{y}(\mathbf{x}_{\text{MC}}^{(i)}) > \hat{y}_f)$
    if $i < n_+$ **then**
      Choose $\mathbf{x}_{(n+1)}$ from $\mathcal{X}_{\text{cand}}$ based on a search criterion
      Add $\mathbf{x}_{(n+1)}$ to $\mathcal{X}$
      Evaluate $y(\mathbf{x}_{(n+1)})$ and add it to $\mathbf{y}$
      n is replaced by $n + 1$
  return $\hat{y}_f$

at each iteration and then chooses the next point as if the estimated quantile is the known contour of interest (Roy and Notz, 2014)

5.3.2 Expected Improvement Criterion

The improvement function proposed by Ranjan et al. (2008) for mapping out a contour where $y(\mathbf{x})$ equals the constant y_f can be written as

$$I(\mathbf{x}) = \begin{cases} \alpha^2 v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x}) - (y(\mathbf{x}) - y_f)^2 & \text{if } |y(\mathbf{x}) - y_f| < \alpha\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})} \\ 0 & \text{otherwise,} \end{cases} \tag{5.2}$$

where $v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})$ is obtained from (1.9). The two subscripts emphasize the predictive variance is conditional on the GP parameters, and α determines the level of confidence. Thus, a large improvement would result from a new evaluation at $\mathbf{x}$ where the uncertainty measured by $v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})$ is currently large and $y(\mathbf{x})$ turns out to be close to the target y_f. Taking the expectation of $I(\mathbf{x})$ with respect to the predictive distribution of $y(\mathbf{x})$ gives the expected improvement,

$$\begin{aligned} E(I(\mathbf{x})) &= \left(\alpha^2 v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x}) - (\hat{y}(\mathbf{x}) - y_f)^2\right)(\Phi(u_2) - \Phi(u_1)) \\ &+ v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})\left((u_2\phi(u_2) - u_1\phi(u_1)) - (\Phi(u_2) - \Phi(u_1))\right) \\ &+ 2(\hat{y}(\mathbf{x}) - y_f)\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})}(\phi(u_2) - \phi(u_1)), \end{aligned}$$

where $\hat{y}_(\mathbf{x})$ is the predictive mean in (1.8), $u_1 = (y_f - \hat{y}_(\mathbf{x}))/\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})} - \alpha$, $u_2 = (y_f - \hat{y}_(\mathbf{x}))/\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})} + \alpha$, and $\phi(\cdot)$ and $\Phi(\cdot)$ are the PDF and CDF of the standard normal distribution, respectively. The sequential-design criterion for selecting the next point, $\mathbf{x}^*$, is to maximize the expected improvement:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}_{\text{cand}}} E[I(\mathbf{x})], \tag{5.3}$$

where $\mathcal{X}_{\text{cand}}$ can be either a discrete or continuous candidate set.

5.3.3 Hypothesis Testing-Based Criterion (Distance-Based Criterion)

Roy and Notz (2014) defined the "discrepancy" between y_f and $y(\mathbf{x})$ at any untried input set to be

$$D(\mathbf{x}) = \begin{cases} \frac{(y(\mathbf{x})-y_f)^2+\epsilon}{v_{\boldsymbol{\psi},\sigma^2}} & \text{if } v_{\boldsymbol{\psi},\sigma^2} \neq 0 \\ \infty & \text{otherwise,} \end{cases} \tag{5.4}$$

where $\epsilon > 0$. If $\epsilon = 0$, the expression looks like an F-statistic. Roy and Notz (2014) proposed to choose the next point that minimizes the expectation of D($\mathbf{x}$). The expectation with respect to the predictive distribution of $y(\mathbf{x})$ is

$$E(D(\mathbf{x})) = \begin{cases} \left(\frac{y(\mathbf{x})-y_f}{\sqrt{v_{\boldsymbol{\psi},\sigma^2}}}\right)^2 + \frac{\epsilon}{v_{\boldsymbol{\psi},\sigma^2}} + 1, & \text{if } v_{\boldsymbol{\psi},\sigma^2} \neq 0 \\ \infty & \text{otherwise.} \end{cases} \tag{5.5}$$

In this chapter, we work with $\epsilon = 0$. Minimizing $E(D(\mathbf{x}))$ is equivalent to the following expression when $v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x}) \neq 0$

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}_{\text{cand}}}{\text{argmin}} \left(\sqrt{\frac{(\hat{y}(\mathbf{x}) - y_f)^2}{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})}} \right) = \underset{\mathbf{x}\in\mathcal{X}_{\text{cand}}}{\text{argmin}} \left(\frac{|\hat{y}_{\boldsymbol{\psi}}(\mathbf{x}) - y_f|}{\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})}} \right). \tag{5.6}$$

Minimizing (5.6) is also equivalent to maximizing the following probability

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}_{\text{cand}}}{\text{argmax}}\, P(\mathbf{x}) = \underset{\mathbf{x}\in\mathcal{X}_{\text{cand}}}{\text{argmax}}\, P\left(z < -\frac{|\hat{y}_{\boldsymbol{\psi}}(\mathbf{x}) - y_f|}{\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})}} \right), \tag{5.7}$$

where $z \sim N(0,1)$. We call $\frac{|\hat{y}_{\boldsymbol{\psi}}(\mathbf{x})-y_f|}{\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})}}$ the **distance** between the predicted response and the quantile adjusted by the prediction standard error. Therefore, we call it the **distance-based criterion**. It is similar in spirit to the hypothesis testing-based criterion. The magnitude of the probability is governed by two terms: (1) How close is $\hat{y}_{\boldsymbol{\psi}}(\mathbf{x})$ to y_f. (2) How large $v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})$ is. The criterion tends to select the following two types of points: (1) given the same prediction variance, the point that has the closest predictive mean to

y_f, which is in favour of a local search; (2) given the same predictive mean, the point has the largest predictive variance, which leads to a global search. Thus the algorithm combines local and global search. The criterion will not select a point that is either in the initial training set or has been selected as its information has already been incorporated.

5.4 An Example—Short Column Function

The short column function models a short column with uncertain material properties and subject to uncertain loads. It was used by Kuschel and Rackwitz (1997) to study the trade-off between structural reliability and cost. We consider only the reliability component of the function. For a given input $\mathbf{x} = \{z, m, p\}$ it is

$$f(\mathbf{x}) = 1 - \frac{4m}{bh^2z} - \frac{p^2}{b^2h^2z^2}, \tag{5.8}$$

where the output $f(\mathbf{x})$ is the limit state, i.e., the difference between resistance and load, and $f(\mathbf{x}) < 0$ means the system fails. The three inputs (z, m, p) and their explicit distributions (Kuschel and Rackwitz, 1997) are:

- $Z \sim$ Lognormal(mean $= 5, \text{sd} = 0.5$). Y is the yield stress.
- $M \sim N(\mu = 2000, \sigma = 400)$. M is the bending moment.
- $P \sim N(\mu = 500, \sigma = 100)$. P is the axial force.

The parameters b (width of the cross-section, in mm) and h (depth of the cross-section, in mm) are meaningfully chosen as $b = 3, h = 10$ such that a simulation based on 10 million points generated from their respective input distributions yield $P(f(\mathbf{x}) < 0) = 0.0025$ with negligible binomial standard error. Therefore, the true probability of system failure is assumed to be 0.0025. This is the probability that we aim to approximate using sequential methodology.

Note that the probability of interest in the short column function is actually a lower-tail probability. Hence step 6 of algorithm 2 will change to

$\hat{p}_f = (1/N)\sum_{i=1}^{N} \mathbb{1}(\hat{y}(\mathbf{x}_{\text{MC}}^{(i)}) < 0)$. The quantile estimation in algorithm 3 will change in a similar way.

5.4.1 Probability Approximation

Suppose the experimental budget is $n = 40$. The size of initial design is kept as $n_0 = 20$. We consider two different choices for the initial design: the first option is to independently generate from the respective input distributions. The second is to first generate a random LHD from 0 to 1 and then map the three variables into mean ± 3 standard deviations of their respective distributions. We label the first option as "random" and the second as "uniform". In order to estimate the failure probability well, we can see from (5.8) that large m and h, small z in the training set tend to lead to failure. Hence, we speculate that the uniform initial design will have a better performance as it over-samples the tails of each distribution.

The MC set is $100,000$ points independently generated from the respective input distributions. The optimization in (5.3) or (5.7) is done by additionally generating $10,000$ points from the respective input distributions to form a finite candidate set and adding the point that maximizes (5.2) or (5.7) in the candidate set. The whole process is repeated 10 times.

The above sample sizes are meaningfully chosen such that they keep a balance between the approximation accuracy and the computational time. A constant mean power exponential GP model is used as the surrogate statistical model and inference on the unknown GP parameters is conducted using the full Bayesian method implemented in chapter 3. The results are shown in Figure 5.3, Figure 5.4 and Table 5.1.

Table 5.1: RMSE based on the final estimates for probability estimation.

Initial design	EI	Distance-based
Random	0.00439	0.00014
Uniform	0.00094	0.00011

From Figure 5.3, after additionally adding 10 points, the distance-based criterion converges to the "true" probability. It stays there and does not

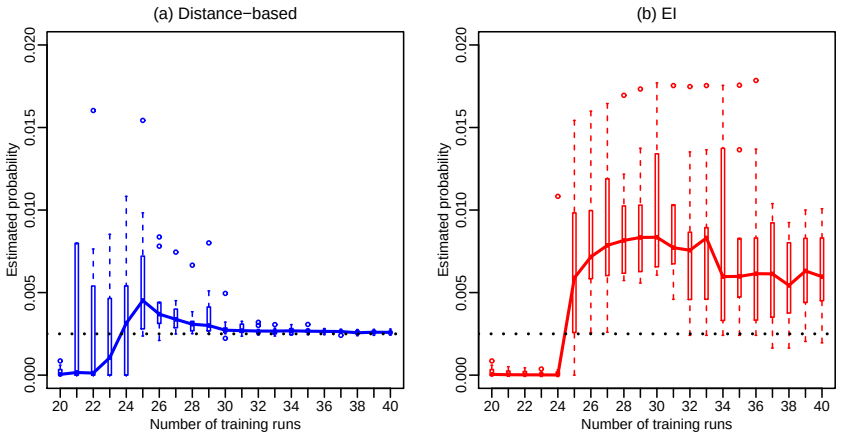

Figure 5.3: Probability estimation and the initial design is random. (a) estimates from the distance-based criterion. (b) estimates from the EI criterion. The medians over 10 repeat experiments are joined by solid lines. The dotted line is the true probability, 0.0025.

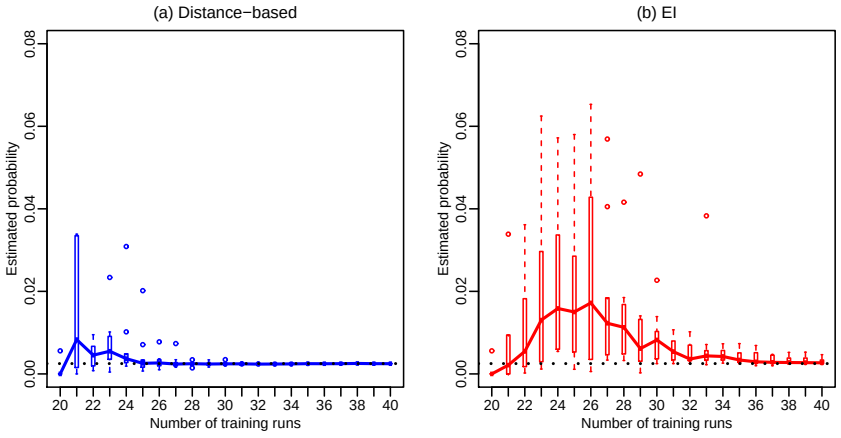

Figure 5.4: Probability estimation and the initial design is uniform. (a) estimates from the distance-based criterion. (b) estimates from the EI criterion. The medians over 10 repeat experiments are joined by solid lines. The dotted line is the true probability, 0.0025.

jump to any other places. However, the EI criterion barely shows any sign of convergence. Table 5.1 suggests the distance-based criterion performs better than the EI for the random initial design.

Figure 5.4 tells the same story for the uniform initial design: the distance-based criterion outperforms the EI. The second row of Table 5.1 confirms this. In addition, if we compare Figure 5.4 to Figure 5.3, the uniform initial designs perform better than the random initial design, in agreement with our expectation. It is also interesting that the uniform initial design "helps" more for the EI criterion than for the distance-based criterion.

Moreover, we further increase the sample size for the EI criterion with the random initial design only, and it is observed (though not reported here for brevity) that the EI method eventually converges by adding 30 more points after the 20-point initial design, making the total sample size $n = 50$. This clearly indicates that compared with the distance-based criterion, the EI criterion converges more slowly. But, why is it the case? We explore the performance difference below. Let us define $T_1 = M/Y$ and $T_2 = (P/Y)^2$. Therefore, (5.8) can be rewritten as

$$f(\mathbf{x}) = 1 - \frac{4}{bh^2}t_1 - \frac{1}{b^2h^2}t_2,$$

where f is a linear function of t_1 and t_2. We concentrate on one repeat and show the initial points, the contour of interest and the 20 points added in Figures 5.5 (a) for EI and the 20 points chosen in Figure 5.5 (b) by the distance-based method.

From Figure 5.5, we observe that the initial points are far from the contour of interest and the sequential method manages to explore the unknown space and to settle down to the place of interest. However, among the 20 points added, the distance-based criterion places 13 points on the contour at level 0,which is the search target. But from Figure 5.5 (b), the EI method only adds 3 point close to the contour, i.e., it wastes many points in uninteresting regions of the input space. This can explain why it does not converge to the "true" probability with 20 points added.

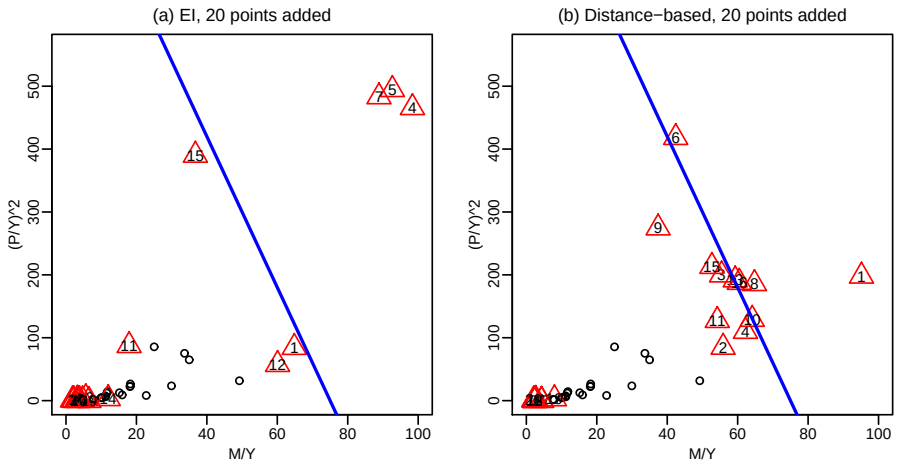

Figure 5.5: The dots are the 20-point initial design. The solid line is the contour $f(\mathbf{x}) = 0$ of interest. The triangles are points added with the order indicated within each triangle.

5.4.2 Quantile Estimation

Following the mode of analysis used above, we investigate extreme quantile estimation with the same settings as in section 5.4.1. For two different initial designs and two search criteria, results are reported in Figure 5.6, Figure 5.7 and Table 5.2.

Table 5.2: RMSE based on the final estimates for quantile estimation.

Initial design	EI	Distance-based
Random	0.07592	0.01287
Uniform	0.04999	0.00991

The results are similar to what we observe in section 5.4.1. The distance-based criterion leads to estimates that converge faster than those for the EI method for both initial designs. The uniform initial design performs better than the random initial design. Based on those observations, we recommend the uniform initial design and the distance-based criterion.

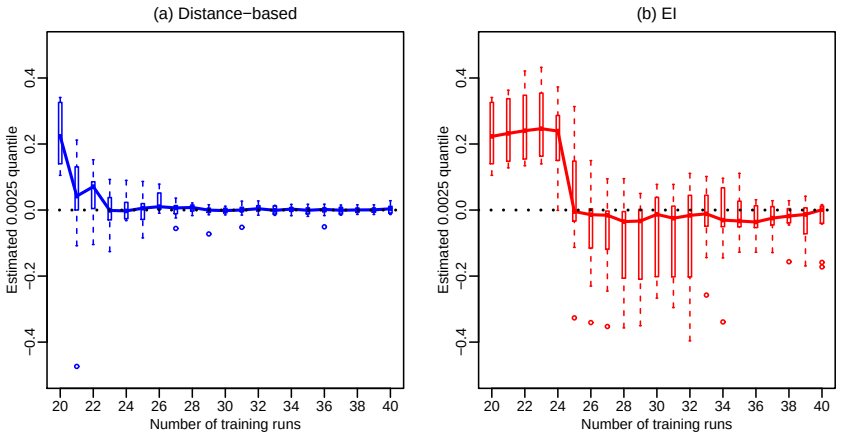

Figure 5.6: Quantile estimation and the initial design is random. (a) estimates from the distance-based criterion. (b) estimates from the EI criterion. The medians are joined by solid lines. The dotted line is the true 0.0025 quantile, 0.

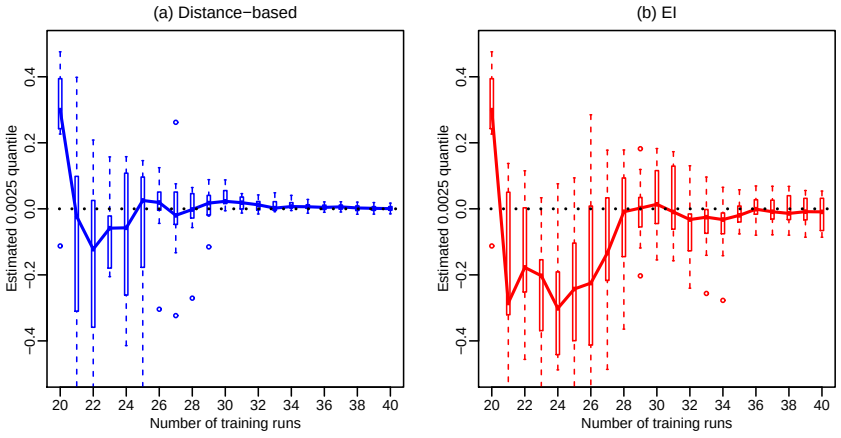

Figure 5.7: Quantile estimation and the initial design is uniform. (a) estimates from the distance-based criterion. (b) estimates from the EI criterion. The medians are joined by solid lines. The dotted line is the true 0.0025 quantile, 0.

5.5 Application to the Computer Model of the Floor System

5.5.1 Preliminary Analysis

Consider the computer model introduced in section 5.2 with $d = 8$. Before doing any formal modelling, we carry out a preliminary sensitivity analysis by fitting a constant Gaussian process with an $n = 20$ random Latin Hypercube design (LHD) (McKay et al., 1979). We use functional analysis of variance (ANOVA) which takes each variable's main effect as well as all of the two level interaction effects into account (Schonlau and Welch, 2005). All higher order interactions are ignored. The ANOVA decomposition results are reported in Table 5.3.

Table 5.3: ANOVA contribution analysis. The 8 main effects explain 99.23% of the total variance. Interactions between two input factors are not shown as none of them contributes more than 0.5%.

Input (MOE)	% Variation
x_1	0.01
x_2	5.33
x_3	12.83
x_4	26.21
x_5	37.51
x_6	16.14
x_7	0.97
x_8	0.23

From Table 5.3, we observe that the four middle beams (x_3, x_4, x_5, x_6) have the most important effects on the response, y. The relationship between them and the response is illustrated in Figure 5.8 again using the methods of Schonlau and Welch (2005). The relationship between the four side beams and y are similar to those of the four middle beams but weaker. It is clear from Figure 5.8 that a small MOE value results in a larger deflection. Therefore, the lower tail of the input distribution, H is important

when estimating the upper probability. This is valuable prior information obtained from the preliminary analysis.

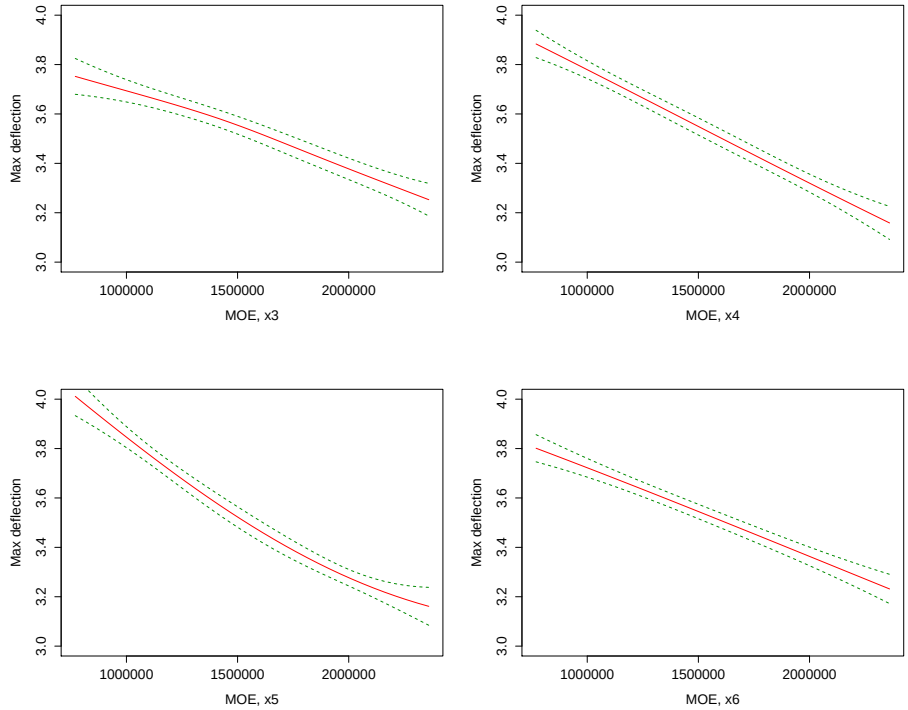

Figure 5.8: The relationship between the MOE of the four middle beams and the response. The solid lines are the estimated main effects with 95% pointwise confidence intervals as dotted lines. Note that the inputs are between 0.77×10^6 and 2.36×10^6 (pounds per square inch), which will be explained in section 5.5.2.

Note that in section 5.4 with the short column function, where the function is explicitly specified in (5.8), we can determine immediately the critical part of the input space. In practice, however, when the functional relationship between inputs and output is unclear, we can carry out a preliminary study as in this section to examine the effects of the inputs.

5.5.2 Modelling the Input Distribution

We need to specify the input distribution. From the analysis in section 5.5.1, we know that a small MOE leads to a large deflection. Therefore, the lower tail of the input distribution is critical. A dataset of 580 MOE values measured from 580 boards collected from a mill is available to us. One way to proceed is to fit a parametric input distribution, for example the Weibull. However, training a parametric distribution based on all of the 580 MOE data values does not emphasize the importance of the lower tail. Therefore, at the beginning of the research, we considered the following two semi-parametric input distributions:

- a mixture distribution, with $p_1 = 0.1$ to sample from a two parameter censored Weibull distribution and $p_2 = 0.9$ to sample with replacement from the upper 90% of the available data. The two parameter censored Weibull distribution is trained by the lower 10% of the available data with the rest (90% data) treated as right censored (Liu et al., 2018).
- a mixture distribution, with $p_1 = 0.1$ to sample from a three parameter censored Weibull distribution and $p_2 = 0.9$ to sample with replacement from the upper 90% of the available data. The three parameter censored Weibull distribution is trained by the lower 10% of the available data with the rest (90% data) treated as right censored (Liu et al., 2018).

The unknown parameters were estimated by the maximum likelihood method and their density curves are added to the empirical histogram of the lower 10% data shown in Figure 5.9. It is clear that there is a non-negligible bump (3 boards) in the lower tail of the dataset, which neither of the semi-parametric distributions is able to characterize. Hence, we propose to use the following non-parametric input distribution, $H(x)$:

$$H(x) = p_1 \times G(x) + p_2 \times S(x), \tag{5.9}$$

where $G(x)$ is the empirical distribution of the lower 10% data of the dataset. $S(x)$ is the empirical distribution of the upper 90% data of the dataset.

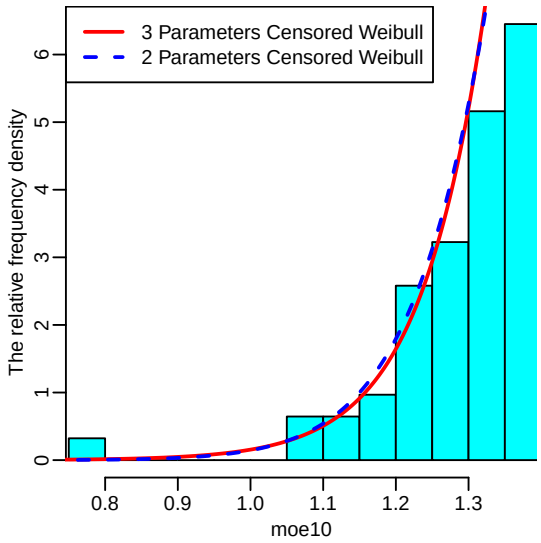

Figure 5.9: Modelling the lower 10% data: Histogram of the lower 10% data, the two parameter censored Weibull distribution and the three parameter censored Weibull distribution.

$p_1 = p_2 = 0.5$ is used to deliberately over-sample the lower tail of the input distribution. It will later be corrected using methods for weighted stratified random sampling when computing the probability/quantile estimates. In a nutshell, we use the empirical distribution of the 580 MOE data values as the input distribution, but deliberately over-sample the lower tail when generating the MC set and the finite candidate set.

5.5.3 MC Set

The MC set is comprised of $12,800$ points that are generated as follows. Each dimension has 2 strata for the mixture distribution: either sampling from $G(x)$ or sampling from $S(x)$. Hence, in total there are $2^8 = 256$ strata. For each stratum, we generate 50 different points from the relevant combination of $G(x)$ and $S(x)$ distributions and thus we have $256 \times 50 = 12800$ points in total to form the MC set. The above sample sizes are meaningfully chosen such that they keep a balance between the approximation accuracy

and the computational time.

Working with this MC set, step 6 in algorithm 2 for probability estimation will change to

$$\hat{p}_f = \sum_{h=1}^{256} \left\{ w_h \left((1/n_h) \sum_{i=1}^{n_h} \mathbb{1}(\hat{y}(\mathbf{x}_{\text{MC}}^{(i)}) > y_f) \right) \right\}, \tag{5.10}$$

where $h = 1, 2, \ldots, 256$ and w_h with $\sum_{h=1}^{256} w_h = 1$. For instance, the combination with all dimensions coming from $G(x)$ has weight $w_h = 0.1^8$ as the actual stratum weight. For quantile estimation, step 6 in algorithm 3 also changes to finding $\hat{y}_f$ such that $\hat{p}_f$ equals a pre-specified probability, a. In practice, any trial value of y_f gives a $\hat{p}_f$ in (5.10). We minimize $|\hat{p}_f - a|$ numerically with respect to y_f using the optimize() function in the MASS package of R to find $\hat{y}_f$ with 10^{-6} tolerance.

5.5.4 True Probability and Quantile

With $p_f = 0.001$, we simulated 10 different MC sets from the mixture distribution in (5.9) and ran the computer model. With 10 repeats the mean of the estimated 0.999 quantiles is 3.88057 inches and the standard error of the mean is 0.00701. Therefore, the "true" 99.9^{th} percentile is taken to be 3.88057 inches.

5.5.5 Application Results

Suppose the design budget is $n = 30$. The number of points for the initial design is $n_0 = 20$ and 10 additional points are chosen by optimizing a search criterion and are added to the design space sequentially. The whole process is repeated $K = 10$ times with different initial designs. The MC set contains 12800 points simulated as described in section 5.5.3. The finite candidate set is the same as the MC set. For the initial design with only $n_0 = 20$ points, however, we take a random uniform LHD to cover the input space globally, following the recommendations from the short column model in section 5.4. A constant mean GP model with the power exponential correlation structure

is used as the surrogate statistical model and inference about unknown GP parameters is conducted using the full Bayesian method implemented in chapter 3. Results based on the distance criterion are reported in Figure 5.10.

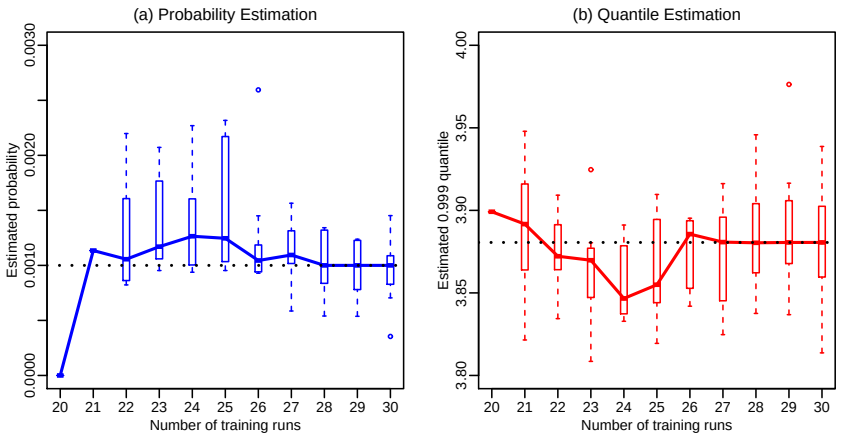

Figure 5.10: The computer model of the floor system and the initial design is uniform with distance based criterion. (a) probability estimates. (b) quantile estimates. The medians over 10 repeat experiments are joined by solid lines. The dotted line is the true probability 0.999 in (a) and the true quantile 3.88057 in (b).

We can see from Figure 5.10 that after adding 10 additional points, both the probability estimate and quantile estimate are very close to the true values, which demonstrates the effectiveness of the sequential design. Moreover, we estimate the extreme probability and quantile well using merely the carefully designed 12,800 points in the MC and candidate sets with a reweighted input distribution. This is a result of the preliminary analysis by which we showed a negative association between MOE of joists and the associated deflection.

5.6 Comments and Discussion

5.6.1 Diagnostics

We next discuss convergence diagnostics for sequences based on the distance based criterion. From the formulation in (5.3), we know that as the sample size of the training set increases to infinity, the distance, $|\hat{y}_{\boldsymbol{\psi}}(\mathbf{x}) - y_f|/\sqrt{v_{\boldsymbol{\psi},\sigma^2}(\mathbf{x})}$ multiplied by -1 as in (5.7) will go to negative infinity. At each step, after a point is added, we compute the distance criterion point-wise for the MC set and draw a boxplot against the sample size of the training set. We would expect a declining trend as the sample size increases. Based on one repeat, the diagnostic plots for the short column function with the uniform initial design and the floor system model are reported in Figures 5.11 and 5.12.

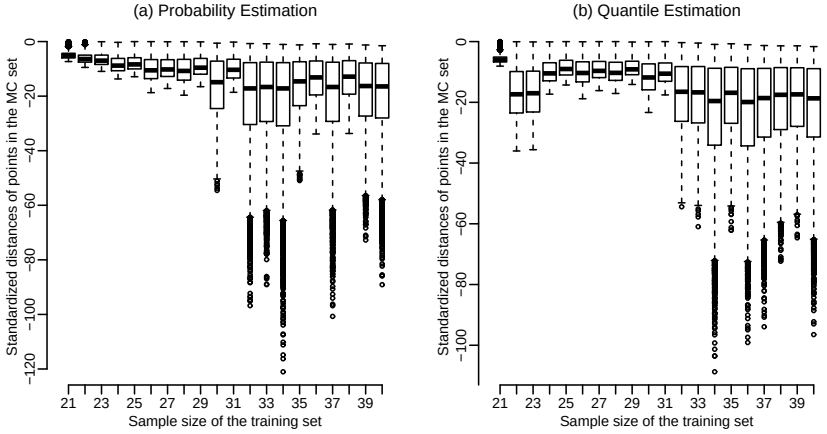

Figure 5.11: Diagnostic plots for the short column function with the uniform initial design. (a) probability estimates. (b) quantile estimates.

From Figures 5.11 and 5.12, the observations are consistent with our expectation as all plots exhibit a decreasing trend. After adding 20 points to the initial design for the short column function, most of the distances multiplied by -1 as in (5.7) in the MC set are less than -20, i.e., a huge standardized distance between the predictive mean and the contour of interest. The pattern carries over to the computer model for the floor system,

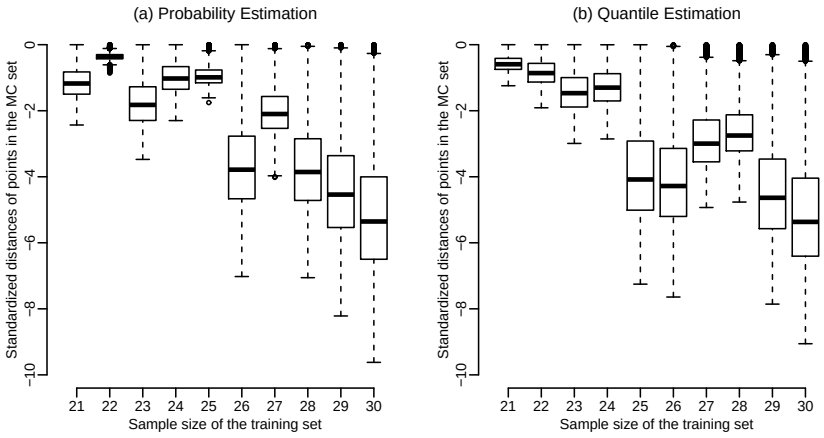

Figure 5.12: Diagnostic plots for the floor system computer model. (a) probability estimates. (b) quantile estimates.

which indicates the algorithm is converging. This information is important, when one works with an expensive computer code, for which the "true" probability/quantile would not be known.

5.6.2 Final Remarks

This chapter illustrates the usefulness of a sequential method in estimating an extreme probability or its associated quantile. Throughout the chapter, we have addressed some very practical issues an engineer faces when using a sequential design. From all of the analyses, we have the following recommendations:

- Use the distance-based criterion, which is more straightforward and converges faster than EI in our study.
- Use the uniform initial design, which over-samples one or both tails (for the floor model only one tail is over-sampled).
- The stratified sampling type MC set is very helpful and is preferred over randomly generating a huge MC set from the input distribution.

- It can be important to do a preliminary analysis to gain some insights on the relationship between inputs and output.
- How to model the input distribution is an important step when estimating an extreme probability/quantile.
- The search criteria considered in this chapter uses the standard error of prediction to guide local/global search. As Bayesian methods can provide more realistic uncertainty estimates (chapter 3), their use is recommended for sequential search.

Chapter 6

Conclusions and Future Work

The Gaussian process model has been widely used as a standard statistical model to analyze complex black-box type computer models. However, there are several fundamental aspects about GPs requiring in-depth analyses. The aim of this thesis is not only to provide a state-of-the-art assessment of the important aspects that suggest solutions of outstanding issues, but also to extend existing methods to overcome the drawbacks revealed from the analyses. The scope of the thesis is rather broad: chapters 2 and 3 discuss the analysis of computer models; chapters 4 and 5 focus on the design of computer experiments. We provide a summary of each chapter in section 6.1. Some future work is discussed in section 6.2.

6.1 Summary of the Thesis

Chapter 2 provides a comprehensive study of the analysis of computer experiments. The primary aim of that chapter is to answer the following two important questions: when faced with a computer code and a set of runs and the primary goal is to make predictions for untried sets of inputs, (1) how to choose the regression terms of a GP; (2) how to choose the correlation structure of a GP? Through intensive evaluation of several real computer models, we conclude that a constant mean regression model is adequate for all the examples we have considered and the PowExp and the Matérn correlation functions are recommended. Moreover, several other conclusions are

1. Design matters but cannot be controlled completely. Variability of performance from equivalent designs can be uncomfortably large.

2. The Matérn correlation function optimized over a few levels of smoothness is a reasonable alternative to PowExp.

3. Full Bayes methods relying on the SqExp structure were seen to be sometimes inferior, and extensions to accommodate less smooth correlations are needed.

4. There is not enough evidence to conclude that composite GaSP methods are ever better than a constant mean GP model with the PowExp structure in practical settings where the output exhibits non-stationary behaviour.

In chapter 3, Bayesian implementations with SqExp correlation structure were extended to allow PowExp or Matérn structure. To estimate the additional smoothness parameters α_j in PowExp, hybrid and fully Bayesian approaches were proposed and applied. The main contribution are

1. We combine Bayesian inference with the flexible correlation structures to propose new methods. Bayesian inference can quantify all sources of parameter estimation uncertainty and the flexible correlation structures can estimate the needed smoothness from the available data. We essentially combine the good properties of each component.

2. The methods we proposed not only have better uncertainty quantification, but also have better prediction accuracy than methods with the SqExp structure in examples where roughness is needed.

Chapter 4 conducts an extensive analysis of the design of computer experiments. We first reviewed several popular design classes and applied different examples to assess the prediction performance of the designs. Our important findings are

1. Given the same sample size, we find mLHD has the best overall prediction accuracy. Therefore, mLHD is the default design we suggest if there is no prior information available.

2. Sample size and transformation of y have much bigger effects on the prediction performance than that of design class.

Chapter 5 provides some insights in how to use the sequential design to solve a real engineering problem. In chapter 4, we evaluate the performances of different design classes and those designs are fixed in the sense that once generated they will not change. Chapter 5 discusses sequential design. It is an adaptive process that the design updates "on the fly" as new points are sequentially added. With the purpose of estimating a failure probability and its associated quantile, we contrast the performances of two sequential search criteria. Moreover, several other practical issues have also been carefully studied. Some important recommendations are made.

6.2 Future Work

The thesis suggests a number of future projects. First of all, in chapter 2, we show that there is a lack of information to conclude the CGP has better performance than a GP with the PowExp in practical settings where the output exhibits nonstationary behaviours. There are several methods for dealing with nonstationary outputs in the literature, such as treed GP (TGP) (Gramacy and Lee, 2008) and local GP (Gramacy and Apley, 2015). It would be interesting to contrast the performances of those methods with that of a GP with the PowExp structure when nonstationarity exists.

In chapter 3, we discussed the PowExp and Matérn structure for Bayesian GPs. There exist other correlation families in the literature, for example the non stationary covariance function used by Paciorek and Schervish (2004), worth trying. In other words, we would follow similar steps for Bayesian inference of a GP in chapter 3, but with a non stationary covariance structure instead of the power exponential structure and the Matérn structure.

In chapter 4, we have expanded Franke's function to have 2-dimensional interactions only and have presumably follow designs with good 2-dimensional projection properties. But the performances of the projection-based designs are not good for Franke's function. Are there other types of functions, for

which there would be substantial improvement for the projection-based designs? This remains unknown.

When optimizing the sequential criterion, we use a fixed finite candidate set in chapter 5. However, it would be desirable to have a dynamic candidate set, which can update automatically based on the point that is being added to the training set. Moreover, we acknowledge that both examples used in chapter 5 are relatively simple examples. We will seek other engineering models to compare the performance of the existing sequential methods. Those other models may well suggest new research objectives. Therefore, it is possible that those existing sequential methods needs to be modified or we may have to propose a completely different criterion to fit the new objective. Whatever the engineering objectives, a sequential method is likely to be advantageous for a complex application.

Bibliography

Abt, M. (1999), "Estimating the Prediction Mean Squared Error in Gaussian Stochastic Processes with Exponential Correlation Structure," *Scandinavian Journal of Statistics*, 26, 563–578.

Andrianakis, I. and Challenor, P. G. (2012), "The Effect of the Nugget on Gaussian Process Emulators of Computer Models," *Computational Statistics & Data Analysis*, 56, 4215–4228.

Ba, S. and Joseph, R. (2012), "Composite Gaussian Process Models for Emulating Expensive Functions," *Annals of Applied Statistics*, 6, 1838–1860.

Ba, S., Myers, W. R., and Brenneman, W. A. (2015), "Optimal Sliced Latin Hypercube Designs," *Technometrics*, 57, 479–487.

Barthelmann, V., Novak, E., and Ritter, K. (2000), "High Dimensional Polynomial Interpolation on Sparse Grids," *Advances in Computational Mathematics*, 12, 273–288.

Bastos, L. and O'Hagan, A. (2009), "Diagnostics for Gaussian Process Emulators," *Technometrics*, 51, 425–438.

Bayarri, M., Berger, J., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J., Lin, C.-H., and Tu, J. (2007), "A Framework for Validation of Computer Models," *Technometrics*, 49, 138–154.

Bayarri, M. J., Berger, J. O., Calder, E. S., Dalbey, K., Lunagomez, S., Patra, A. K., Pitman, E. B., Spiller, E. T., and Wolpert, R. L. (2009), "Using Statistical and Computer Models to Quantify Volcanic Hazards," *Technometrics*, 51, 402–413.

Bichon, B. B., Mahadevan, S., and Eldred, M. S. (2009), "Reliability-Based Design Optimization Using Efficient Global Reliability Analysis," Palm Springs, California, 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, pp. 1–12.

Bingham, D., Ranjan, P., and Welch, W. (2014), "Design of Computer Experiments for Optimization, Estimation of Function Contours, and Related Objectives," in *Statistics in Action: A Canadian Outlook*, ed. Lawless, J. F., Boca Raton, Florida: CRC Press, chap. 7, pp. 110–123.

Cantelli, F. P. (1933), "Sulla Determinazione Empirica Delle Leggi di Probabilita," *Giornale Istituto Italiano Attuari*, 4, 221–424.

Caselton, W. F. and Zidek, J. V. (1984), "Optimal Monitoring Network Designs," *Statistics & Probability Letters*, 2, 223–227.

Chapman, W. L., Welch, W. J., Bowman, K. P., Sacks, J., and Walsh, J. E. (1994), "Arctic Sea Ice Variability: Model Sensitivities and a Multidecadal Simulation," *Journal of Geophysical Research: Oceans*, 99, 919–935.

Chen, H. (2013), "Bayesian Prediction and Inference in Analysis of Computer Experiments," Master's thesis, University of British Columbia, Vancouver.

Dette, H. and Pepelyshev, A. (2010), "Generalized Latin Hypercube Design for Computer Experiments," *Technometrics*, 52, 421–429.

Fang, K.-T., Lin, D. K., Winker, P., and Zhang, Y. (2000), "Uniform Design: Theory and Application," *Technometrics*, 42, 237–248.

Franke, R. (1979), "A Critical Comparison of Some Methods for Interpolation of Scattered Data," Tech. rep., Naval Postgraduate School Monterey CA.

Glivenko, V. (1933), "Sulla Determinazione Empirica Della Legge di Probabilita," *Giornale Istituto Italiano Attuari*, 4, 92–99.

Gramacy, R. and Lee, H. (2008), "Bayesian Treed Gaussian Process Models With an Application to Computer Modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

Gramacy, R. B. and Apley, D. W. (2015), "Local Gaussian Process Approximation for Large Computer Experiments," *Journal of Computational and Graphical Statistics*, 24, 561–578.

Gramacy, R. B. and Lee, H. K. (2012), "Cases for the Nugget in Modeling Computer Experiments," *Statistics and Computing*, 22, 713–722.

Handcock, M. S. and Stein, M. L. (1993), "A Bayesian Analysis of Kriging," *Technometrics*, 35, 403–410.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, 103, 570–583.

Johnson, M., Moore, L., and Ylvisaker, D. (1990), "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26, 131–148.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998), "Efficient Global Optimization of Expensive Black-box Functions," *Journal of Global Optimization*, 13, 455–492.

Joseph, V. R., Gul, E., and Ba, S. (2015), "Maximum Projection Designs for Computer Experiments," *Biometrika*, 102, 371–380.

Joseph, V. R., Hung, Y., and Sudjianto, A. (2008), "Blind Kriging: a New Method for Developing Metamodels," *Journal of Mechanical Design*, 130, 031102–1–8.

Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011), "Efficient Emulators of Computer Experiments Using Compactly Supported Correlation Functions With an Application to Cosmology," *Annals of Applied Statistics*, 2470–2492.

Kennedy, M. (2004), "Description of the Gaussian Process Model Used in GEM-SA," Tech. rep., University of Sheffield, available online at `http://www.tonyohagan.co.uk/academic/GEM/`.

Kennedy, M. and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models," *Journal of Royal Statistical Association, Series B*, 63, 425–464.

Kuschel, N. and Rackwitz, R. (1997), "Two Basic Problems in Reliability-Based Structural Optimization," *Mathematical Methods of Operation Research*, 46, 309–333.

Liu, Y., Salibián-Barrera, M., Zamar, R. H., and Zidek, J. V. (2018), "Using Artificial Censoring to Improve Extreme Tail Quantile Estimates," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 1–22. DOI: 10.1111/rssc.12262.

Loeppky, J., Sacks, J., and Welch, W. (2009), "Choosing the Sample Size of a Computer Experiments: A Practical Guide," *Technometrics*, 51, 366–376.

McCutcheon, W. J. (1984), "Deflections of Uniformly Loaded Floors: a Beam-Spring Analog," Tech. Rep. Research Paper FPL449, United States Department of Agriculture, Madison, WI.

McKay, M., Beckman, R., and Conover, W. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 21, 239–245.

Morris, M., Mitchell, T., and Ylvisaker, D. (1993), "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, 35, 243–255.

Niederreiter, H. (1988), "Low Discrepancy and Low Dispersion Sequences," *Journal of Number Theory*, 30, 51–70.

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, 1st ed.

Nilson, T. and Kuusk, A. (1989), "A Reflectance Model for the Homogeneous Plant Canopy and its Inversion," *Remote Sensing of Environment*, 27, 157–167.

Paciorek, C. J. and Schervish, M. J. (2004), "Nonstationary Covariance Functions for Gaussian Process Regression," in *Advances in Neural Information Processing Systems*, pp. 273–280.

Paulo, R. (2005), "Default Priors for Gaussian Processes," *Annals of Statistics*, 33, 556–582.

Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. (2013), "Quantile-Based Optimization of Noisy Computer Experiments With Tunable Precision," *Technometrics*, 55, 2–13.

Plumlee, M. (2014), "Fast Prediction of Deterministic Functions Using Sparse Grid Experimental Designs," *Journal of the American Statistical Association*, 109, 1581–1591.

Preston, D., Tonks, D., and Wallace, D. (2003), "Model of Plastic Deformation for Extreme Loading Conditions," *Journal of Applied Physics*, 93, 211–220.

Pronzato, L. and Müller, W. G. (2012), "Design of Computer Experiments: Space Filling and Beyond," *Statistics and Computing*, 22, 681–701.

Ranjan, P., Bingham, D., and Michailidis, G. (2008), "Sequential Experimental Design for Contour Estimation from Complex Computer Codes," *Technometrics*, 50, 527–541.

Ranjan, P., Haynes, R., and Karsten, R. (2011), "A Computationally Stable Approach to Gaussian Process Interpolation of Deterministic Computer Simulation Data," *Technometrics*, 53, 366–378.

Roberts, G. O. and Rosenthal, J. S. (2009), "Examples of Adaptive MCMC," *Journal of Computational and Graphical Statistics*, 18, 349–367.

Rosenthal, J. S. (2014), "Optimizing and Adapting the Metropolis Algorithm," in *Statistics in Action: A Canadian Outlook*, ed. Lawless, J. F., Boca Raton, Florida: CRC Press, chap. 6, pp. 93–108.

Roy, S. and Notz, W. I. (2014), "Estimating Percentiles in Computer Experiments: a Comparison of Sequential-adaptive Designs and Fixed Designs," *Journal of Statistical Theory and Practice*, 8, 12–29.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–435.

Santner, T., Williams, B., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, Springer Series in Statistics, Springer Press, 1st ed.

Schonlau, M. and Welch, W. J. (2005), "Screening the Input Variables to a Computer Model via Analysis of Variance and Visualization," in *Screening Methods for Experimentation in Industry, Drug Discovery and Genetics*, eds. Dean, A. and Lewis, S., New York: Springer, chap. 10, pp. 308–327.

Sheather, S. J. and Jones, M. C. (1991), "A Reliable Data-based Bandwidth Selection Method for Kernel Density Estimation." *Journal of the Royal Statistical Society Series B*, 53, 683–690.

Shewry, M. C. and Wynn, H. P. (1987), "Maximum Entropy Sampling," *Journal of Applied Statistics*, 14, 165–170.

Sobol, I. (1967), "On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals," *USSR Computational Mathematics and Mathematical Physics*, 7, 86–112.

Spiller, E. T., Bayarri, M. J., Berger, J. O., Calder, E. S., Patra, A. K., Pitman, E. B., and Wolpert, R. L. (2014), "Automating Emulator Construction for Geophysical Hazard Maps," *SIAM/ASA Journal on Uncertainty Quantification*, 2, 126–152.

Stein, M. (1987), "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, 29, 143–151.

Stein, M. (1988), "Asymptotically Efficient Prediction of a Random Field With a Misspecified Covariance Function," *Annals of Statistics*, 16, 55–63.

Stein, M. (1999), *Interpolation of Spatial Data: Some Theory for Kriging*, New York: Springer Press.

Tang, B. (1993), "Orthogonal Array-Based Latin Hypercubes," *Journal of the American Statistical Association*, 88, 1392–1397.

Welch, W., Buck, R., Sacks, J., Wynn, H., and Toby Mitchell, M. M. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Morris, M. D., and Schonlau, M. (1996), "Response to James M. Lucas," *Technometrics*, 38, 199–203.

West, O. R., Siegrist, R. I., Wynn, T. J., and Jenkins, R. A. (1995), "Measurement Error and Spatial Variability Effects on Characterization of Volatile Organics in the Subsurface," *Environmental Science and Technology*, 29, 647–656.

Appendix A

Supplemental Materials for Chapter 2

A.1 Test Functions in Chapters 2

A.1.1 Borehole Model

The output y is generated by

$$y = \frac{2\pi T_u(H_u - H_l)}{\log(r/r_w)\left(1 + \frac{2LT_u}{\log(r/r_w)r_w^2 K_w} + T_u/T_l\right)},$$

where the 8 input variables and their respective ranges of interest are as in Table A.1.

Variable	Description (units)	Range
r_w	radius of borehole (m)	$[0.05, 0.15]$
r	radius of influence (m)	$[100, 5000]$
T_u	transmissivity of upper aquifer (m^2/yr)	$[63070, 115600]$
H_u	potentiometric head of upper aquifer (m)	$[990, 1110]$
T_l	transmissivity of lower aquifer (m^2 / yr)	$[63.1, 116]$
H_l	potentiometric head of lower aquifer (m)	$[700, 820]$
L	length of borehole (m)	$[1120, 1680]$
K_w	hydraulic conductivity of borehole (m/yr)	$[9855, 12045]$

Table A.1: Borehole function input variables, units, and ranges. All ranges are converted to $[0, 1]$ for statistical modeling.

A.1.2 Gprotein Model

The differential equations generating the output y of the G-protein system dynamics are

$$\begin{aligned}
\dot{\eta}_1 &= -u_1\eta_1 x + u_2\eta_2 - u_3\eta_1 + u_5, \\
\dot{\eta}_2 &= u_1\eta_1 x - u_2\eta_2 - u_4\eta_2, \\
\dot{\eta}_3 &= -u_6\eta_2\eta_3 + u_8(G_{\text{tot}} - \eta_3 - \eta_4)(G_{\text{tot}} - \eta_3), \\
\dot{\eta}_4 &= u_6\eta_2\eta_3 - u_7\eta_4, \\
y &= (G_{\text{tot}} - \eta_3)/G_{\text{tot}},
\end{aligned}$$

where $\eta_1, \ldots, \eta_4$ are concentrations of 4 chemical species, $\dot{\eta}_1 = \frac{\delta\eta_1}{\delta t}$, etc, and $G_{\text{tot}} = 10000$ is the (fixed) total concentration of G-protein complex after 30 seconds. The input variables in this system are described in Table A.2. Only $d = 4$ inputs are varied: we model y as a function of $\log(x)$, $\log(u_1)$, $\log(u_6)$, $\log(u_7)$.

Variable	Description	Range
u_1	rate constant	$[2 \times 10^6, 2 \times 10^7]$
u_2	rate constant	5×10^{-3} (fixed)
u_3	rate constant	1×10^{-3} (fixed)
u_4	rate constant	2×10^{-3} (fixed)
u_5	rate constant	8 (fixed)
u_6	rate constant	$[3 \times 10^{-5}, 3 \times 10^{-4}]$
u_7	rate constant	$[0.3, 3]$
u_8	rate constant	1 (fixed)
x	initial concentration	$[1.0 \times 10^{-9}, 1.0 \times 10^{-6}]$

Table A.2: G-protein code input variables and ranges. All variables are transformed to log scales on $[0, 1]$ for statistical modeling.

A.2 Results of Normalized maximum absolute errors in Chapter 2

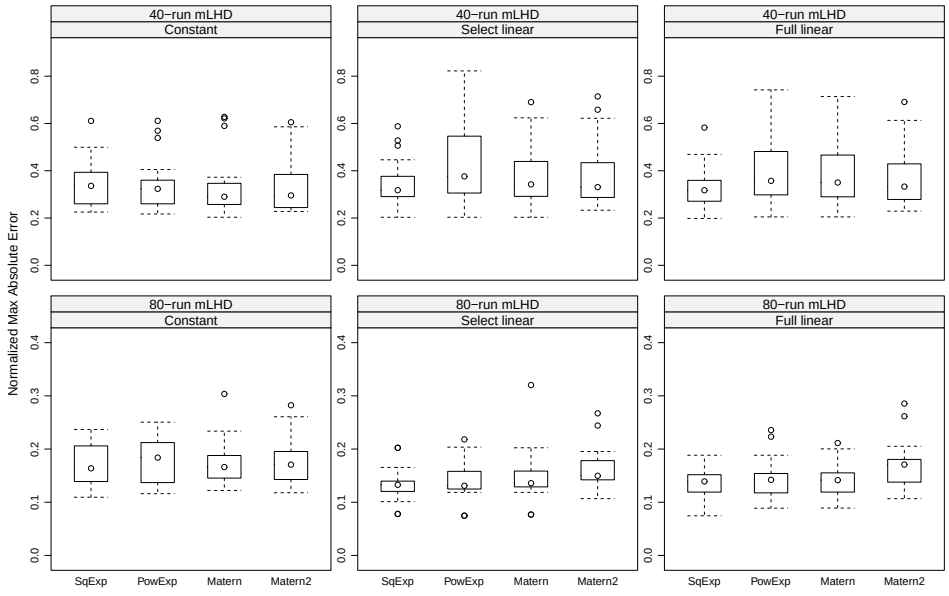

Figure A.1: G-protein model: Normalized maximum absolute error of prediction, $e_{\max,\text{ho}}$, for GaSP with all combinations of three regression models and four correlation functions. There are three base designs: a 40-run mLHD (top row); and a 80-run mLHD (bottom row). For each base design, 24 random permutations of its columns give the 24 values of $e_{\max,\text{ ho}}$ in a boxplot.

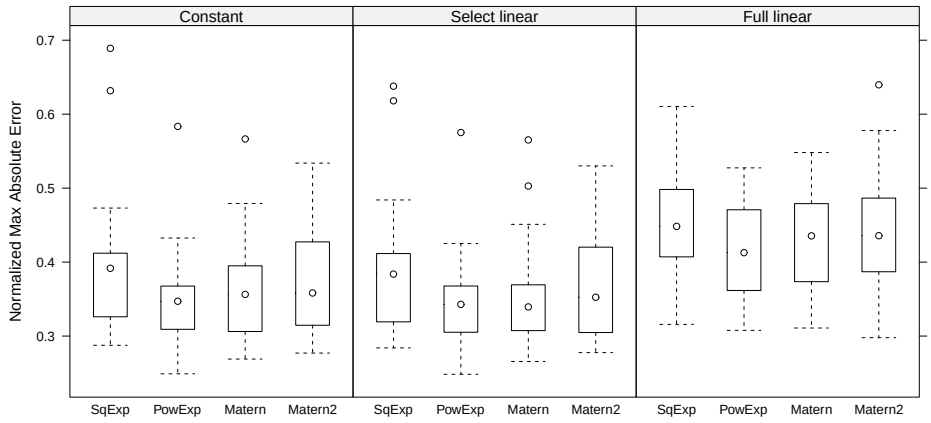

Figure A.2: PTW model: Normalized maximum absolute error of prediction, $e_{\max,\ \mathrm{ho}}$, for GaSP with all combinations of three regression models and four correlation functions. There is one base design: a 110 mLHD. 25 random permutations of its columns give the 25 values of $e_{\max,\ \mathrm{ho}}$ in a boxplot.

	$e_{\max,\ \mathrm{ho}}$			
Regression	Correlation function			
Model	SqExp	PowExp	Matern2	Matern
Constant	0.35	0.27	0.28	0.28
Select linear	0.35	0.26	0.27	0.24
Full linear	0.30	0.27	0.28	0.28
Quartic	0.29	0.28	0.29	0.31

Table A.3: Nilson-Kuusk model: Normalized maximum absolute error of prediction, e_{maxho}, for four regression models and four correlation functions. The experimental data are from a 100-run LHD, and the hold-out set is from a 150-run LHD.

	$e_{\text{max, ho}}$			
Regression	Correlation function			
Model	SqExp	PowExp	Matern2	Matern
Constant	0.21	0.16	0.18	0.17
Select linear	0.25	0.16	0.19	0.18
Full linear	0.23	0.16	0.21	0.17
Quartic	0.23	0.16	0.21	0.17

Table A.4: Nilson-Kuusk model: Normalized maximum absolute error of prediction, e_{maxho}, for four regression models and four correlation functions. The experimental data are from a 150-run LHD, and the hold-out set is from a 100-run LHD.

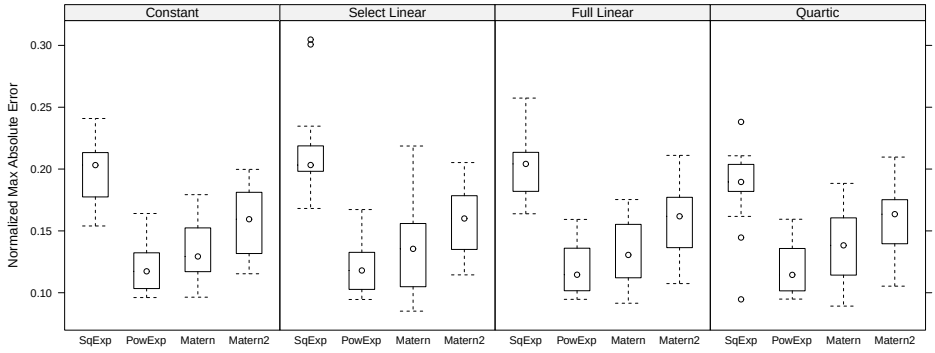

Figure A.3: Nilson-Kuusk model: Normalized maximum absolute error of prediction, $e_{\text{max,ho}}$, for four regression models and four correlation functions. Twenty-five designs are created from a 150-run LHD base plus 50 random points from a 100-run LHD. The remaining 50 points in the 100-run LHD form the holdout set for each repeat.

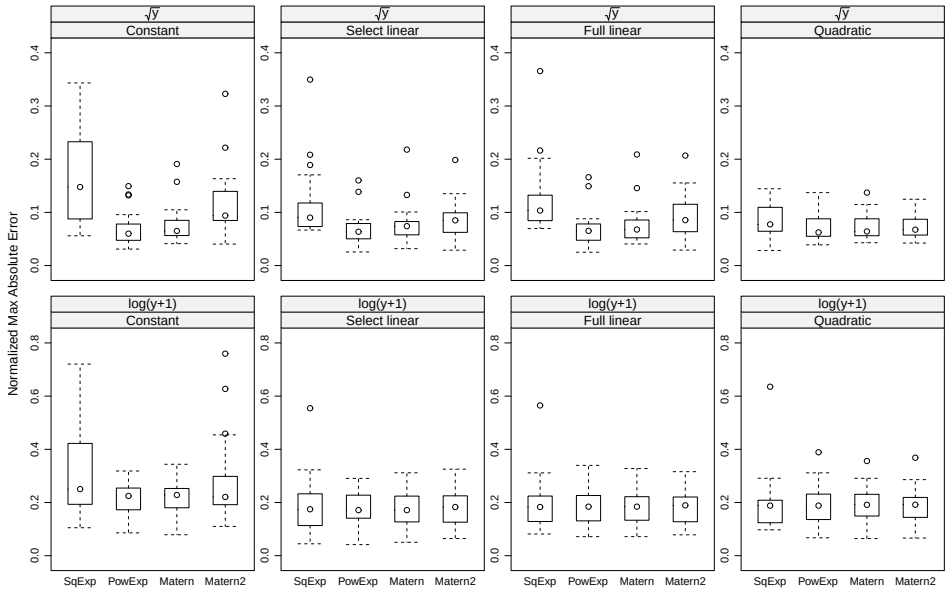

Figure A.4: Volcano model: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$, for three regression models and two correlation functions.

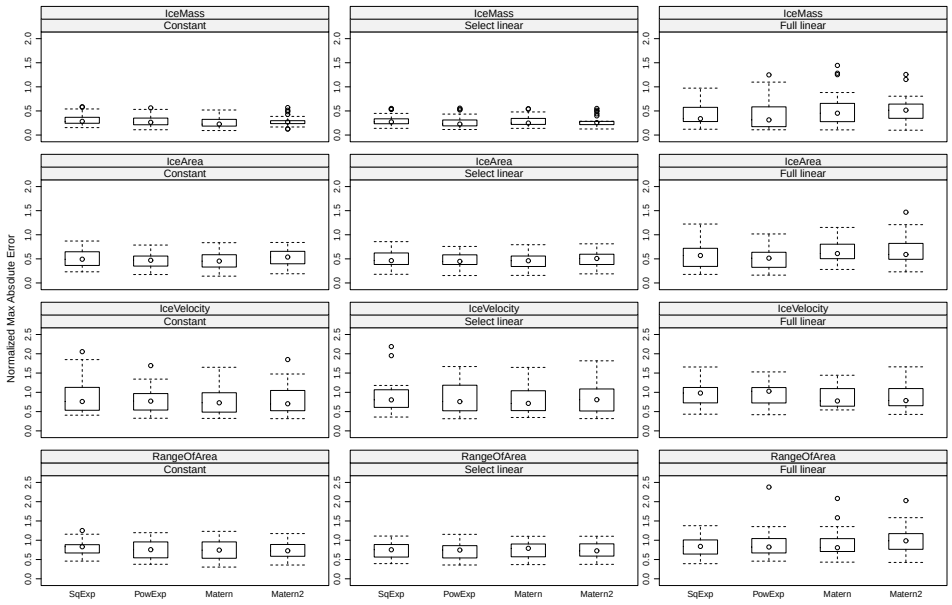

Figure A.5: Seaice model: Normalized maximum absolute error of prediction, $e_{\max,\mathrm{ho}}$, for three regression models and four correlation functions. The outputs are: IceMass, IceArea, IceVelocity and RangeOfArea, which are modelled independently.

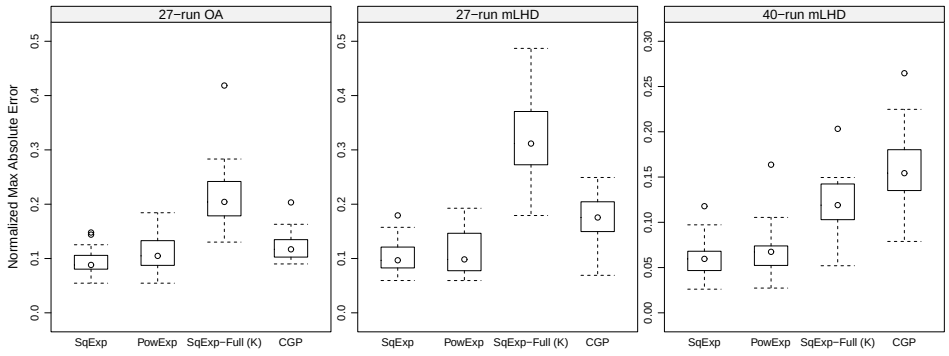

Figure A.6: Borehole function: Normalized maximum absolute error of prediction, $e_{\max,\mathrm{ho}}$, for GaSP(Const, Gauss), GaSP(Const, PowerExp), SqExp-Full (K), and CGP. There are three base designs: a 27-run OA (left), a 27-run mLHD (middle); and a 40-run mLHD (right). For each base design, 25 random permutations of its columns give the 25 values of $e_{\max,\ \mathrm{ho}}$ in a boxplot.

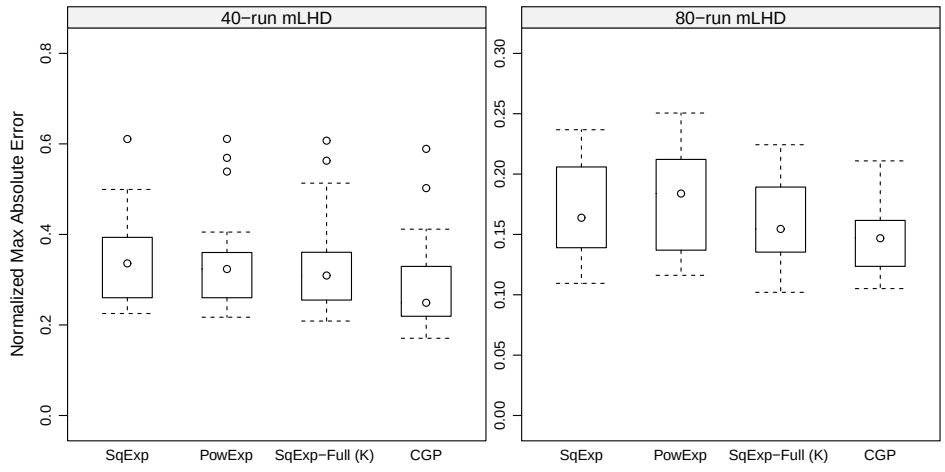

Figure A.7: G-protein model: Normalized maximum absolute error of prediction, $e_{\max,\mathrm{ho}}$, for GaSP(Const, Gauss), GaSP(Const, PowerExp), SqExp-Full (K), and CGP. There are two base designs: a 40-run mLHD (left); and an 80-run mLHD (right). For each base design, all 24 permutations of its columns give the 24 values of $e_{\max,\ \mathrm{ho}}$ in a boxplot.

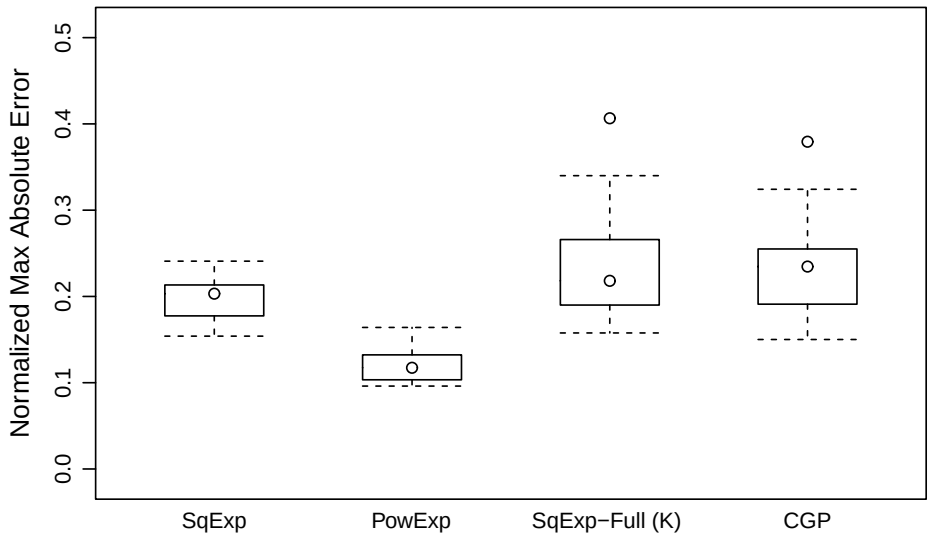

Figure A.8: Nilson-Kuusk model: Normalized maximum absolute error of prediction, $e_{\max,\mathrm{ho}}$, for GaSP(Const, Gauss), GaSP(Const, PowerExp), SqExp-Full (K), and CGP.

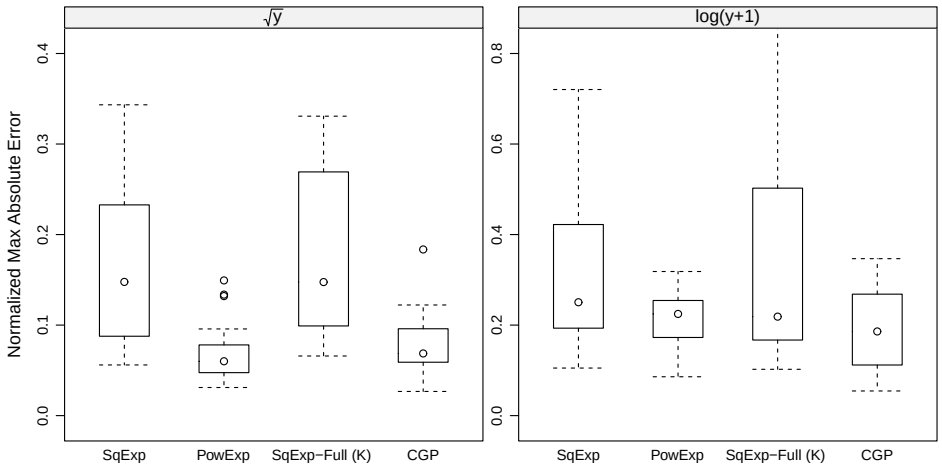

Figure A.9: Volcano model: Normalized maximum absolute error of prediction, $e_{\max,\mathrm{ho}}$, for GaSP(Const, Gauss), GaSP(Const, PowerExp), SqExp-Full (K), and CGP.

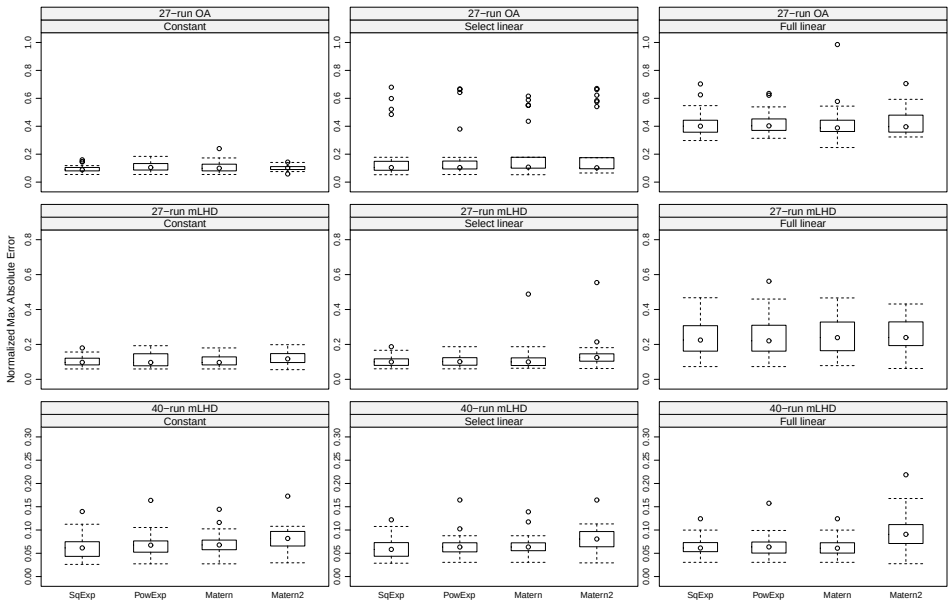

Figure A.10: Borehole model: Normalized maximum absolute error of prediction, $e_{\max,\mathrm{ho}}$, for GaSP adding a nugget estimated by its MLEs, and with all combinations of three regression models and four correlation functions. There are three base designs: a 27-run OA (top row); a 27-run mLHD (middle row); and a 40-run mLHD (bottom row). For each base design, 25 random permutations of its columns give the 25 values of $e_{\max,\ \mathrm{ho}}$ in a boxplot.

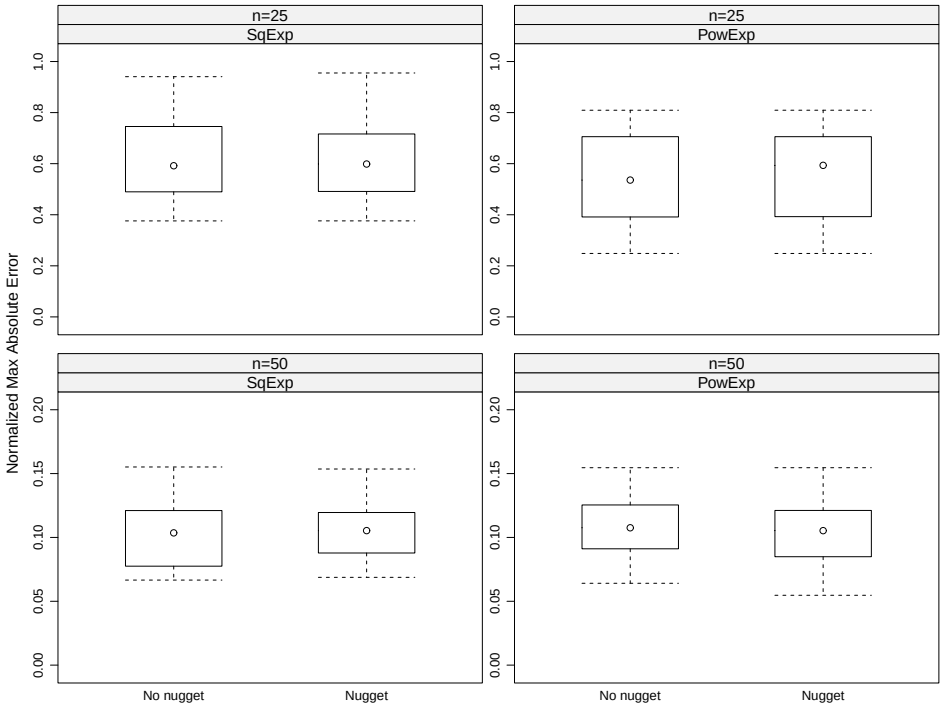

Figure A.11: Friedman function: Normalized maximum absolute error of prediction, $e_{\max,\text{ho}}$,, for GaSP(Const, SqExp) and GaSP(Const, PowExp) models with no nugget term versus the same models with a nugget. There are two base designs: a 25-run mLHD (top row); and a 50-run mLHD (bottom row). For each base design, 25 random permutations between columns give the 25 values of $e_{\max,\text{ ho}}$ in a boxplot.

Appendix B

Supplemental Materials for Chapter 3

B.1 Results of Normalized Maximum Absolute Errors

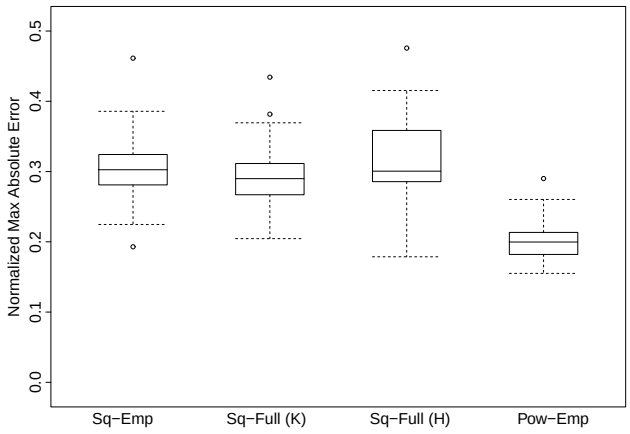

Figure B.1: Normalized maximum absolute errors for the Nilson-Kuusk (motivating example) from four methods: SqExp-Emp, SqExp-Full (K), SqExp-Full (H) and PowExp-Emp. Each method has 25 normalized maximum absolute errors from 25 random training-test data splits.

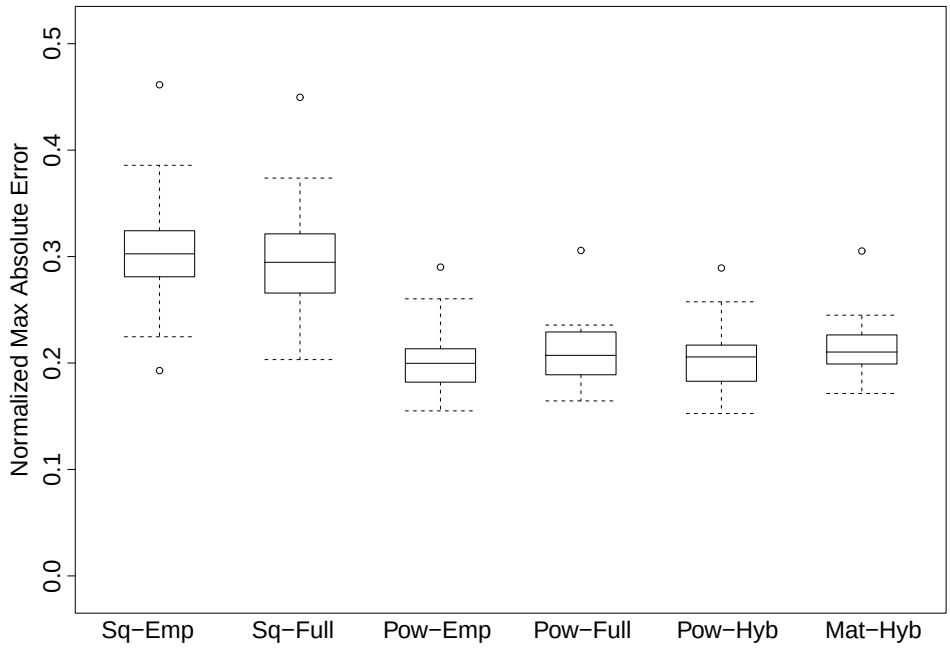

Figure B.2: Normalized maximum absolute error for the Nilson-Kuusk model from six methods: SqExp-Emp, SqExp-Full, PowExp-Emp, PowExp-Full, PowExp-Hybrid and Matérn-Hybrid. Each method has 25 normalized maximum absolute error values from 25 random training-test data splits.

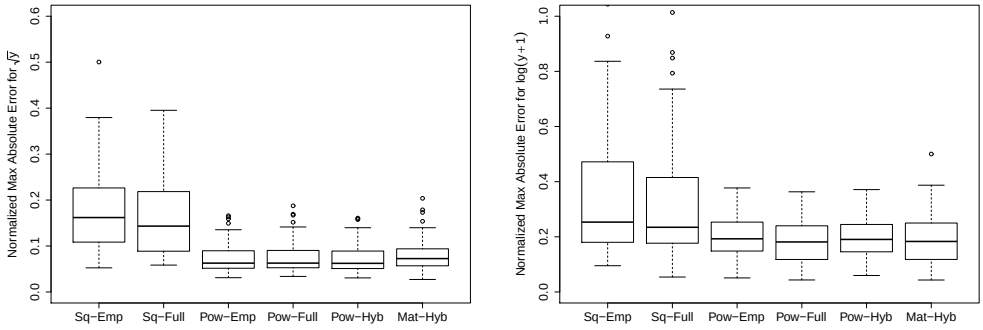

Figure B.3: Normalized maximum absolute error for the volcano model from six methods: SqExp-Emp, SqExp-Full, PowExp-Emp, PowExp-Full, PowExp-Hybrid and Matérn-Hybrid. Each method has 100 normalized maximum absolute error values from 100 random training-test data splits. Two transformations of y are considered: $\sqrt{y}$ (left panel) and $\log(y+1)$ (right panel).

B.2 Marginal Posterior Distribution of the Correlation Parameters

Here we derive the marginal posterior distribution of $\boldsymbol{\psi}_B$, the correlation parameters treated in a fully Bayesian way and hence sampled by MCMC. Any remaining correlation parameters are estimated by empirical Bayes, and the expressions below are conditional on their plugged-in MLEs. We also take priors $\pi(\mu) \propto 1$ and $\pi(\sigma^2) = IG(\phi_1, \phi_2)$.

We need to integrate out μ and σ^2 from the joint posterior of μ, σ^2, and

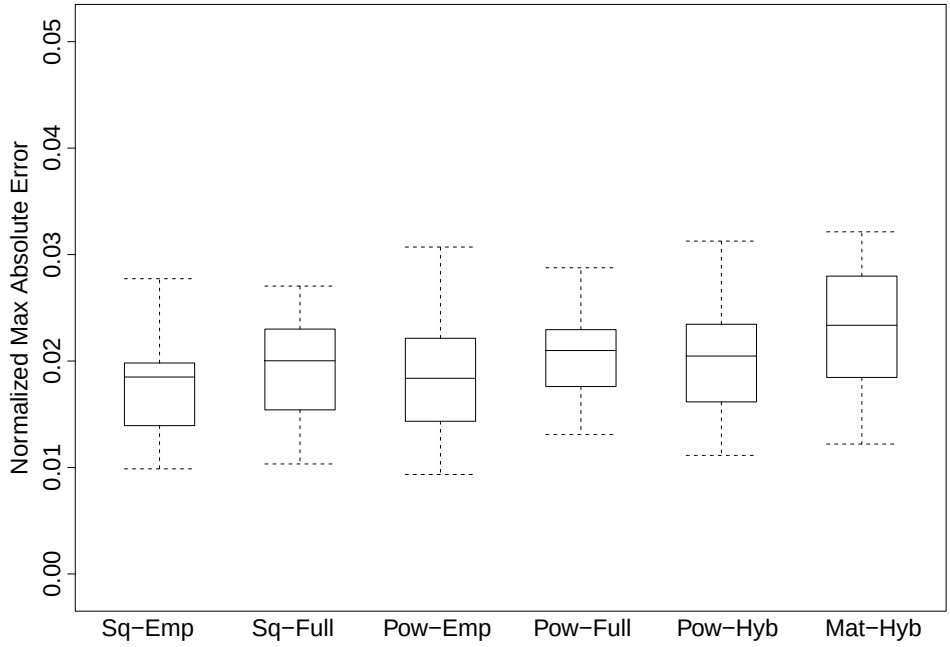

Figure B.4: Normalized maximum absolute error for the Borehole function and a 80-point mLHD base design. Each method has 25 normalized maximum absolute error values from repeat experiments permuting the columns of the base design.

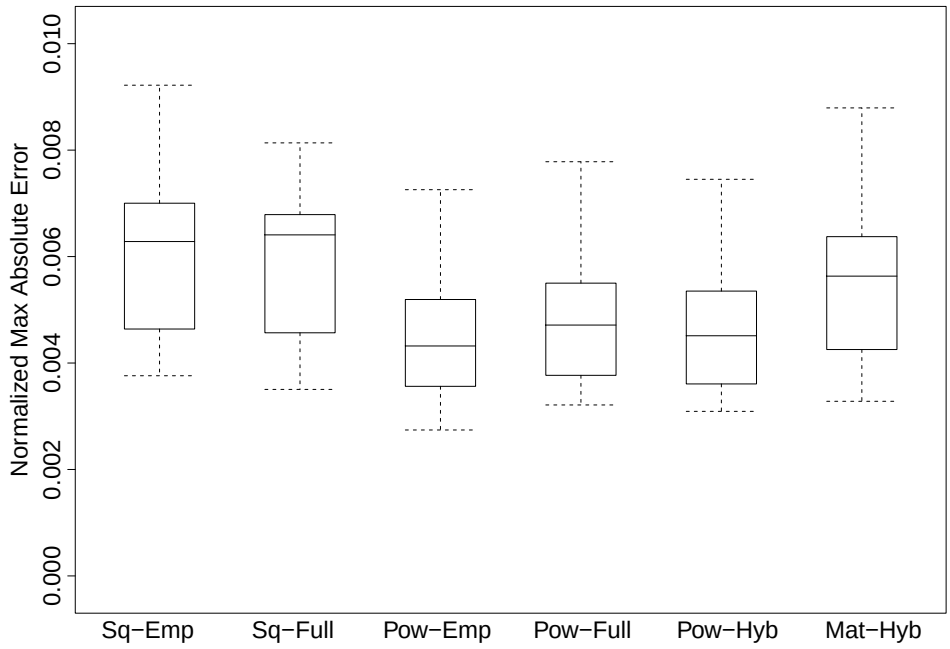

Figure B.5: Normalized maximum absolute error for the Borehole function and a 200-point mLHD base design. Each method has 25 normalized maximum absolute error values from repeat experiments permuting the columns of the base design.

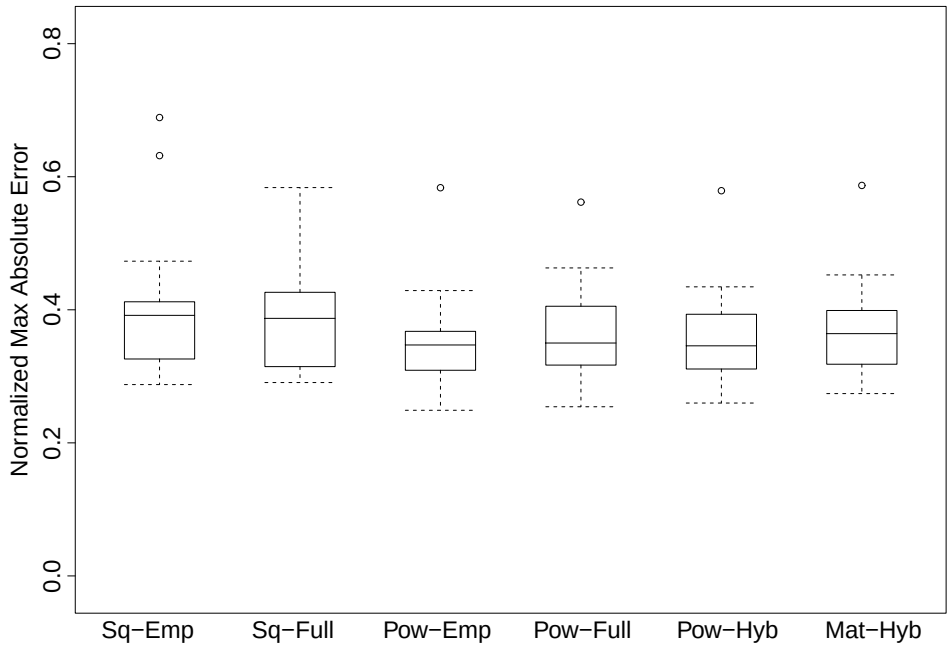

Figure B.6: Normalized maximum absolute error for the PTW model with an mLHD base design. Each method has 25 normalized maximum absolute error values from repeat experiments permuting the columns of the base design.

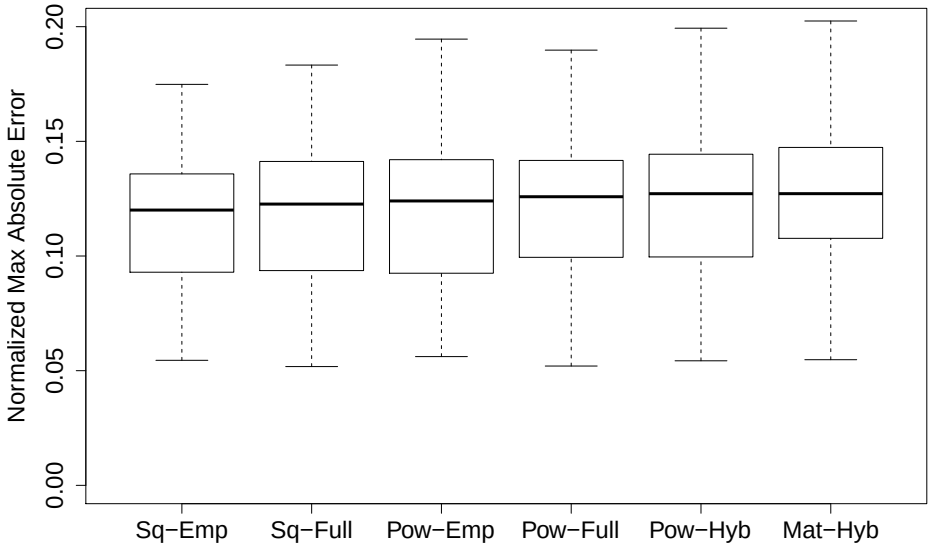

Figure B.7: Normalized maximum absolute error with $d = 10$ inputs and output simulated from a GP with SqExp correlation. The boxplots show the results from 25 random realizations of the GP.

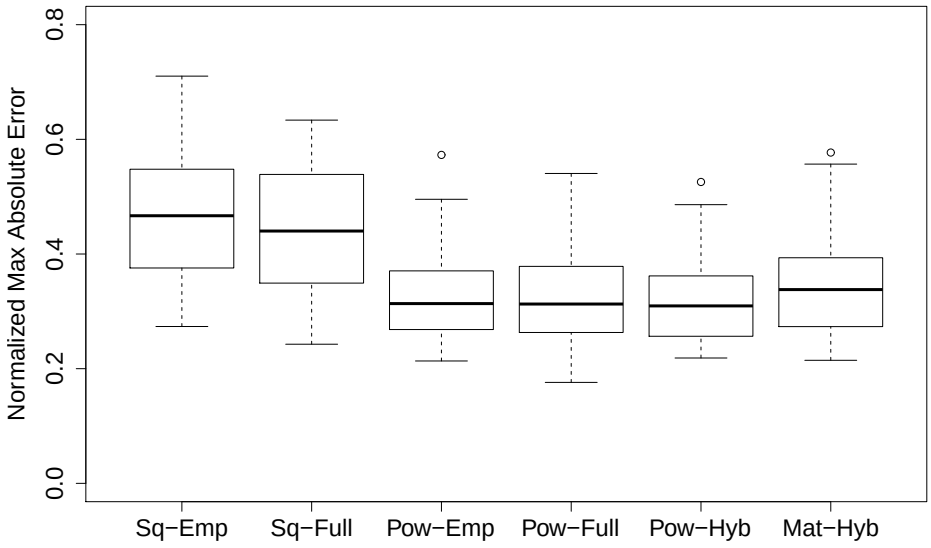

Figure B.8: Normalized maximum absolute error with $d = 10$ inputs and output simulated from a GP with PowExp correlation and all $\alpha_j = 1.8$. The boxplots show the results from 25 random realizations of the GP.

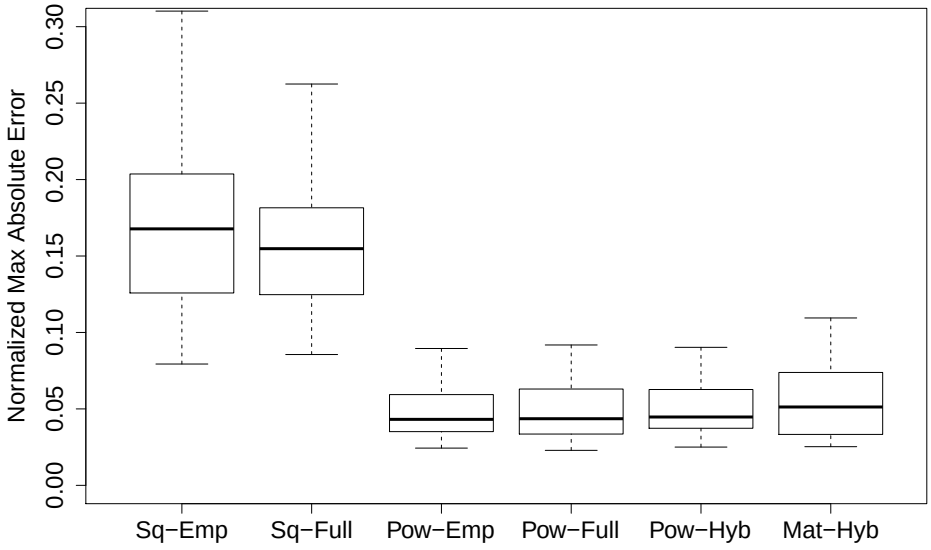

Figure B.9: Normalized maximum absolute error with $d = 10$ inputs and output simulated from a GP with Matérn correlation and all $\delta_j = 1$. The boxplots show the results from 25 random realizations of the GP.

$\boldsymbol{\psi}_B$:

$$\begin{aligned}
\pi(\boldsymbol{\psi}_B|\mathbf{y}) &\propto \int\int \pi(\mu)\pi(\sigma^2)\pi(\boldsymbol{\psi}_B)L(\mathbf{y}|\mu,\sigma^2,\boldsymbol{\psi}_B)\,d\mu\,d\sigma^2 \\
&\propto \int\int \pi(\boldsymbol{\psi}_B)\frac{\phi_2^{\phi_1}}{\Gamma(\phi_1)}(\sigma^2)^{-\phi_1-1}\exp\left(-\frac{\phi_2}{\sigma^2}\right) \\
&\qquad \times \frac{1}{(\sigma^2)^{n/2}\,|\mathbf{R}|^{1/2}}\exp\left(-\frac{(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right)d\mu\,d\sigma^2 \\
&= \frac{\pi(\boldsymbol{\psi}_B)}{|\mathbf{R}|^{1/2}}\int \frac{\phi_2^{\phi_1}}{\Gamma(\phi_1)}(\sigma^2)^{-\phi_1-1}\exp\left(-\frac{\phi_2}{\sigma^2}\right) \\
&\qquad \times \frac{1}{(\sigma^2)^{n/2}}\int \exp\left(-\frac{(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right)d\mu\,d\sigma^2. \quad \text{(B.1)}
\end{aligned}$$

First, we evaluate the integral with respect to μ. Rewrite the quadratic form in the integrand as

$$\begin{aligned}
(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu) &= (\mathbf{y}-\mathbf{1}\hat{\mu}-\mathbf{1}(\mu-\hat{\mu}))^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu}-\mathbf{1}(\mu-\hat{\mu})) \\
&= (\mathbf{y}-\mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu}) + (\mu-\hat{\mu})\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}(\mu-\hat{\mu}). \quad \text{(B.2)}
\end{aligned}$$

The cross-product term disappears because

$$(\mathbf{y}-\mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}\mathbf{1} = \mathbf{y}^T\mathbf{R}^{-1}\mathbf{1}-\hat{\mu}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1} = \mathbf{y}^T\mathbf{R}^{-1}\mathbf{1}-(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1} = 0$$

after substituting $\hat{\mu}$ from (1.6). Further simplification of (B.2) follows by noting that

$$(\mathbf{y}-\mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu}) = (2\phi_1+n-1)\hat{\sigma}^2_{\boldsymbol{\psi}_B}.$$

Also note that

$$\int \frac{1}{\sqrt{2\pi}}\left(\frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}{\sigma^2}\right)^{1/2}\exp\left(-\frac{(\mu-\hat{\mu})^T\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}(\mu-\hat{\mu})}{2\sigma^2}\right)d\mu$$

is the integral of the probability density of $\mu \sim N(\hat{\mu}, (\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}/\sigma^2)^{-1})$ and

hence integrates to 1. Thus, in (B.1),

$$\int \exp\left(-\frac{(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right) d\mu \propto \exp\left(-\frac{(2\phi_1+n-1)\hat{\sigma}^2_{\boldsymbol{\psi}_B}}{2\sigma^2}\right)\left(\frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}{\sigma^2}\right)^{-1/2}. \tag{B.3}$$

Substituting (B.3) into (B.1) leaves just the integral with respect to σ^2:

$$\begin{aligned}
&\pi(\boldsymbol{\psi}_B|\mathbf{y}) \\
&\propto \frac{\pi(\boldsymbol{\psi}_B)}{|\mathbf{R}|^{1/2}} \int \frac{\phi_2^{\phi_1}}{\Gamma(\phi_1)}(\sigma^2)^{-\phi_1-1}\exp\left(-\frac{\phi_2}{\sigma^2}\right) \\
&\times \frac{1}{(\sigma^2)^{n/2}}\exp\left(-\frac{(2\phi_1+n-1)\hat{\sigma}^2_{\boldsymbol{\psi}_B}}{2\sigma^2}\right)\left(\frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}{\sigma^2}\right)^{-1/2} d\sigma^2 \\
&\propto \frac{\pi(\boldsymbol{\psi}_B)}{|\mathbf{R}|^{1/2}\left(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}\right)^{1/2}} \int (\sigma^2)^{-(\phi_1+\frac{n-1}{2}+1)}\exp\left(-\frac{\phi_2+\frac{(n-1)}{2}\hat{\sigma}^2_{\boldsymbol{\psi}_B}}{\sigma^2}\right) d\sigma^2 \\
&= \frac{\pi(\boldsymbol{\psi}_B)}{|\mathbf{R}|^{1/2}\left(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}\right)^{1/2}} \frac{\Gamma(\phi_1+\frac{n-1}{2})}{(\phi_2+\frac{n-1}{2}\hat{\sigma}^2_{\boldsymbol{\psi}_B})^{(\phi_1+\frac{n-1}{2})}},
\end{aligned}$$

which is the posterior distribution of $\boldsymbol{\psi}_B$ given in (3.6) up to constants of proportionality. The last line follows since

$$\int \frac{(\phi_2+\frac{n-1}{2}\hat{\sigma}^2_{\boldsymbol{\psi}_B})^{(\phi_1+\frac{n-1}{2})}}{\Gamma(\phi_1+\frac{n-1}{2})}(\sigma^2)^{-(\phi_1+\frac{n-1}{2}+1)}\exp\left(-\frac{\phi_2+\frac{n-1}{2}\hat{\sigma}^2_{\boldsymbol{\psi}_B}}{\sigma^2}\right) d\sigma^2$$

is the integral of an inverse-gamma probability density for σ^2 with parameters $\phi_1+\frac{n-1}{2}$ and $\phi_2+\frac{n-1}{2}\hat{\sigma}^2_{\boldsymbol{\psi}_B}$.

B.3 Posterior Distributions and Acceptance Rates

We report posterior distributions for the correlation parameters and their acceptance rates for the Nilson-Kuusk example with a 100-point LHD to train a GP model. The number of MCMC iterations is 10,000 with the first 4,000 as burn-in, and the thinning parameter is set as 10.

For PowExp, Figures B.10 and B.11 show the empirical posterior density plots of the θ_j and α_j for the PowExp-Full method. The Metropolis-

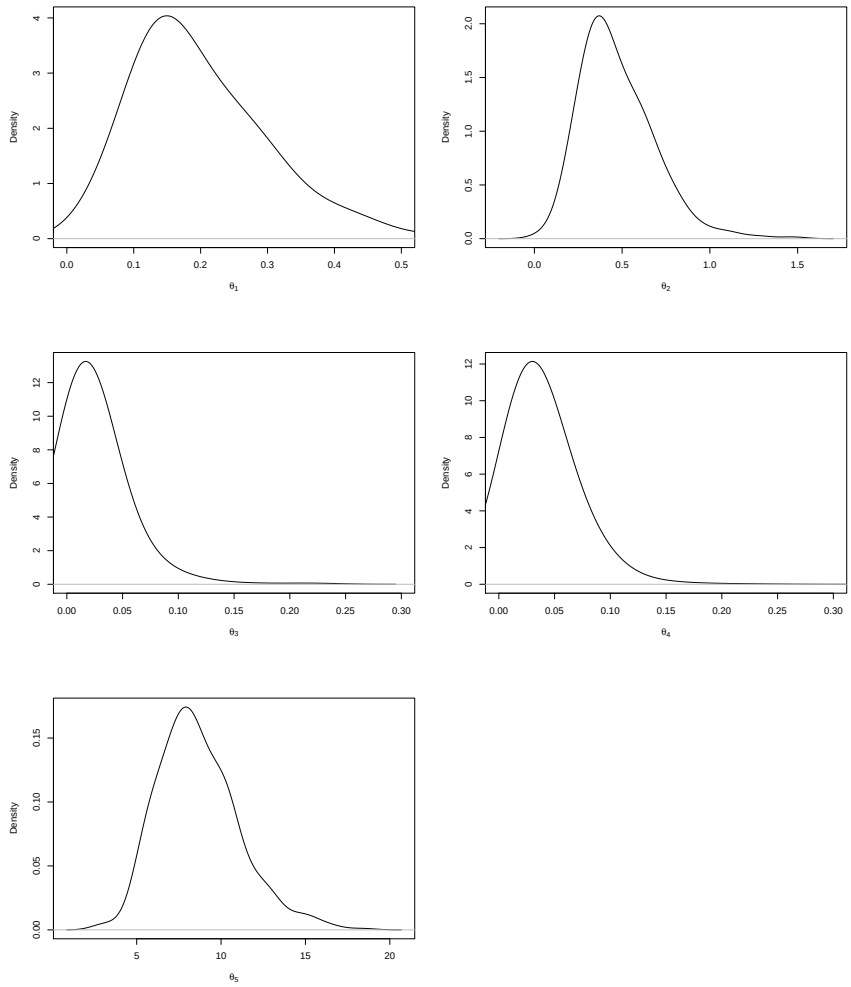

Figure B.10: Empirical posterior density plots of $\theta_1, \ldots, \theta_5$ for the Nilson-Kuusk model with the PowExp-Full method.

Hasting algorithm works on the λ_j and γ_j scales, but the parameters are transformed back to the θ_j and α_j scales in the plots. The method proposed by Sheather and Jones (Sheather and Jones, 1991) is used to select the bandwidth of a Gaussian kernel density estimator, which is implemented

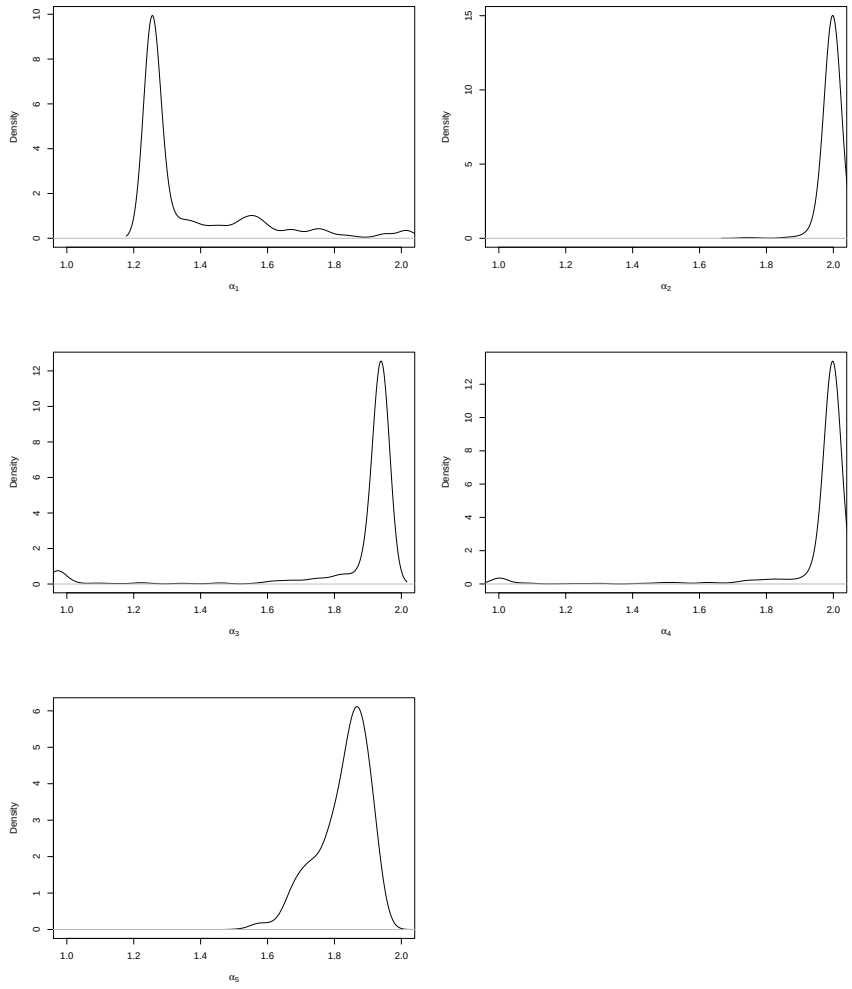

Figure B.11: Empirical posterior density plots of $\alpha_1, \ldots, \alpha_5$ for the Nilson-Kuusk model with the PowExp-Full method.

by the `width.SJ` function in the `MASS` library in R. Similarly, Figure B.12 shows the empirical posterior density plots of the θ_j for the PowExp-Hybrid method. These posterior distributions may be compared with maximum

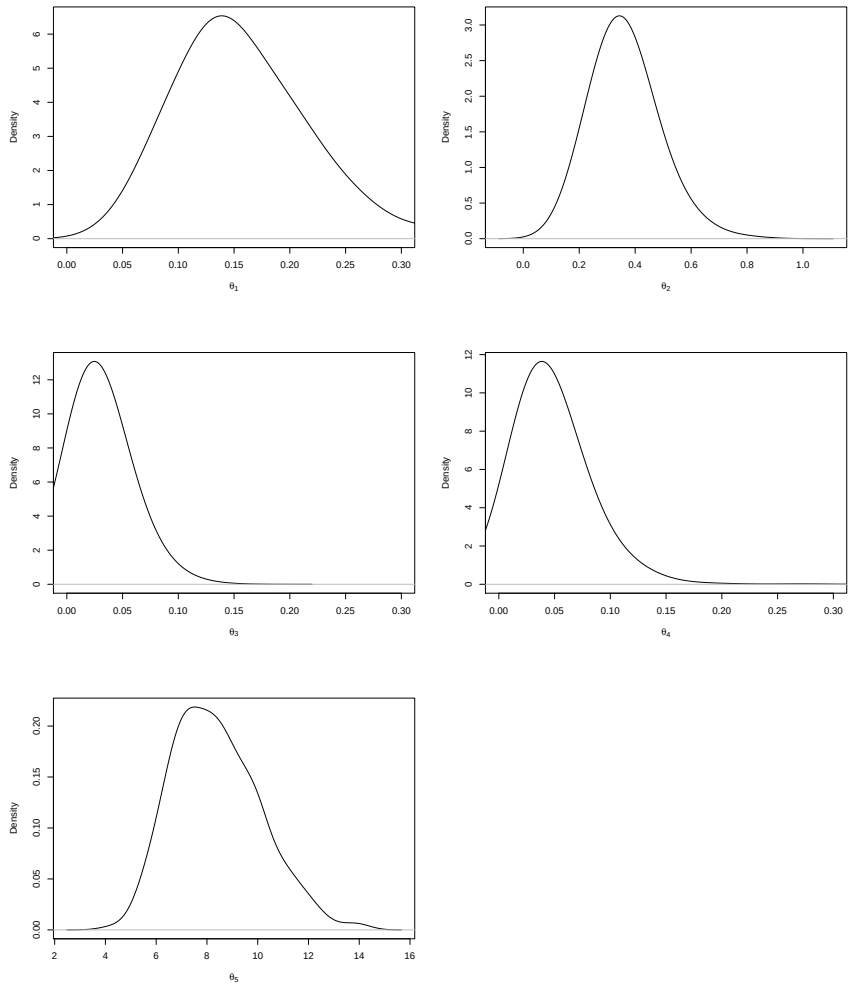

Figure B.12: Empirical posterior density plots of $\theta_1, \ldots, \theta_5$ for the Nilson-Kuusk model with the PowExp-Hybrid method.

likelihood estimates (also on the θ_j and α_j scales) from empirical Bayes, as shown in Table B.1.

Figure B.13 shows the empirical posterior density plots of the θ_j for the

$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$	$\hat{\theta}_5$	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$
0.16	0.35	0.02	0.04	8.89	1.26	2.00	1.90	2.00	1.89

Table B.1: Maximum likelihood estimates of θ_j and α_j from PowExp-Emp for the Nilson-Kuusk model.

Matérn-Hybrid method. These posterior distributions may be compared

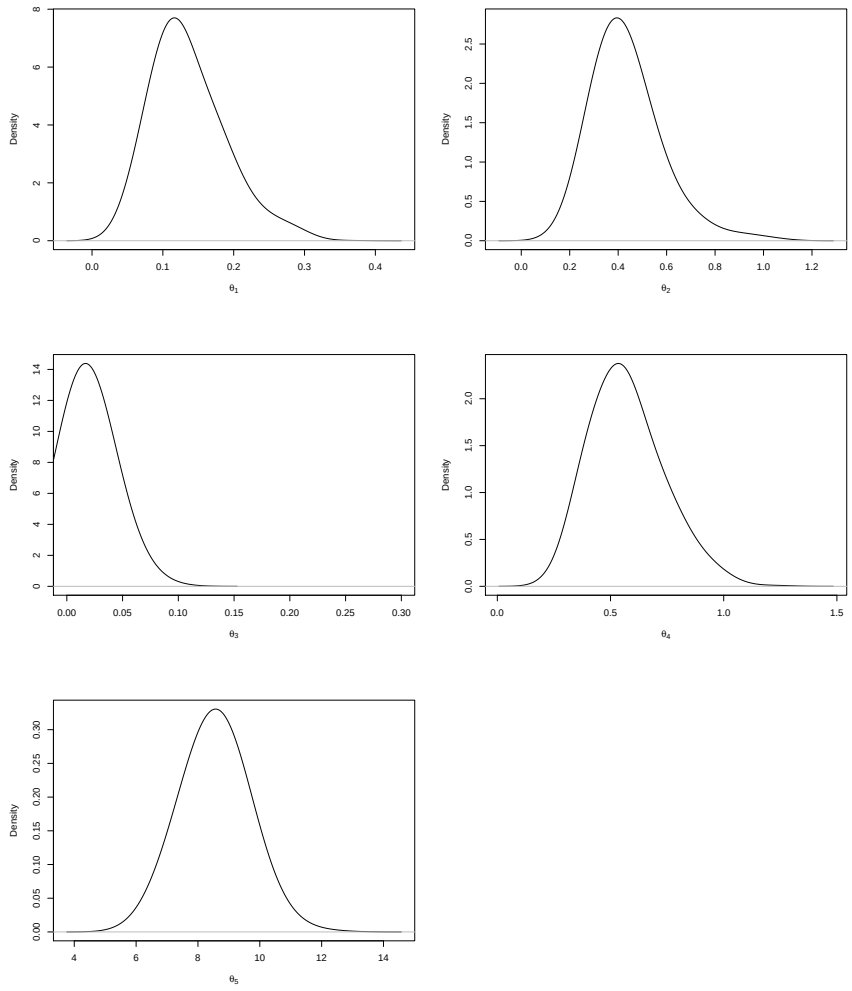

Figure B.13: Empirical posterior density plots of $\theta_1, \ldots, \theta_5$ for the Nilson-Kuusk model with the Matérn-Hybrid method.

with maximum likelihood estimates of the θ_j and δ_j from empirical Bayes, as shown in Table B.2.

$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$	$\hat{\theta}_5$	$\hat{\delta}_1$	$\hat{\delta}_2$	$\hat{\delta}_3$	$\hat{\delta}_4$	$\hat{\delta}_5$
0.11	0.38	0.02	0.50	8.37	0	$\to \infty$	$\to \infty$	2	2

Table B.2: Maximum likelihood estimates of the θ_j and δ_j from empirical Bayes and the Matérn correlation for the Nilson-Kuusk model.

The Metropolis-Hastings acceptance rates for the λ_j (PowExp-Full, PowExp-Hybrid, and Matérn-Hybrid) are reported in Table B.3 and for the γ_j (PowExp-Full) in Table B.4.

Method	λ_1	λ_2	λ_3	λ_4	λ_5
PowExp-Full	30.0%	42.6%	45.1%	43.4%	31.8%
PowExp-Hybrid	41.8%	45.6%	47.6%	43.3%	41.3%
Matérn-Hybrid	18.9%	27.2%	32.0%	23.4%	31.6%

Table B.3: Metroplis-Hastings acceptance rates for the λ_j for the Nilson-Kuusk model.

Method	γ_1	γ_2	γ_3	γ_4	γ_5
PowExp-Full	38.4%	49.4%	45.8%	43.9%	33.2%

Table B.4: Metroplis-Hastings acceptance rates for the γ_j for the Nilson-Kuusk model.

Appendix C

Supplemental Materials for Chapter 4

C.1 Results of Normalized Max Absolute Errors for the Main Comparison

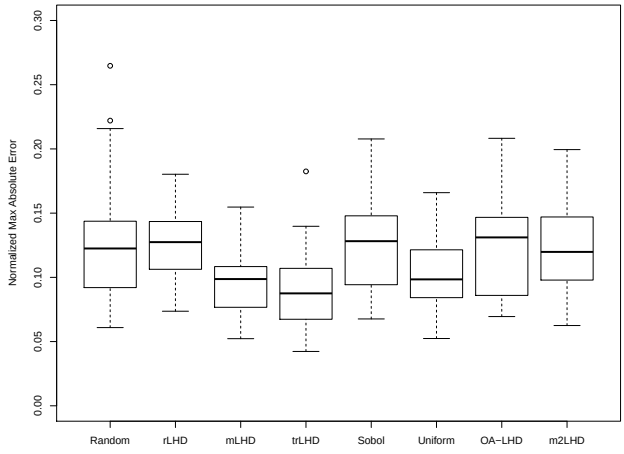

Figure C.1: Borehole function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$, for all 8 base designs, $n = 27$. For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

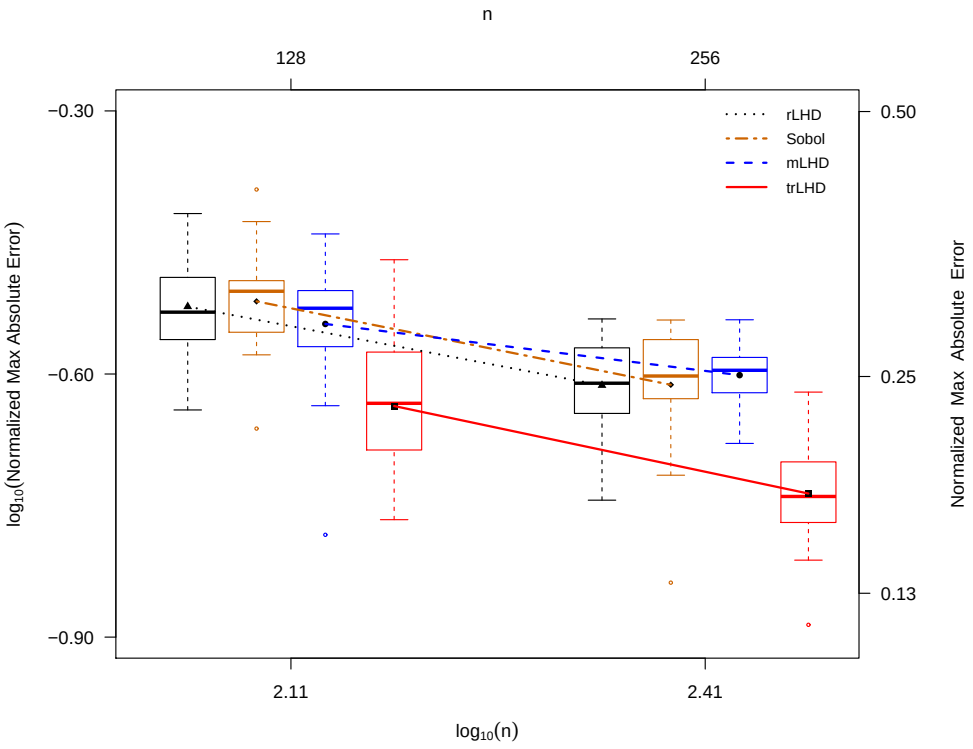

Figure C.2: PTW function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 128, 256$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

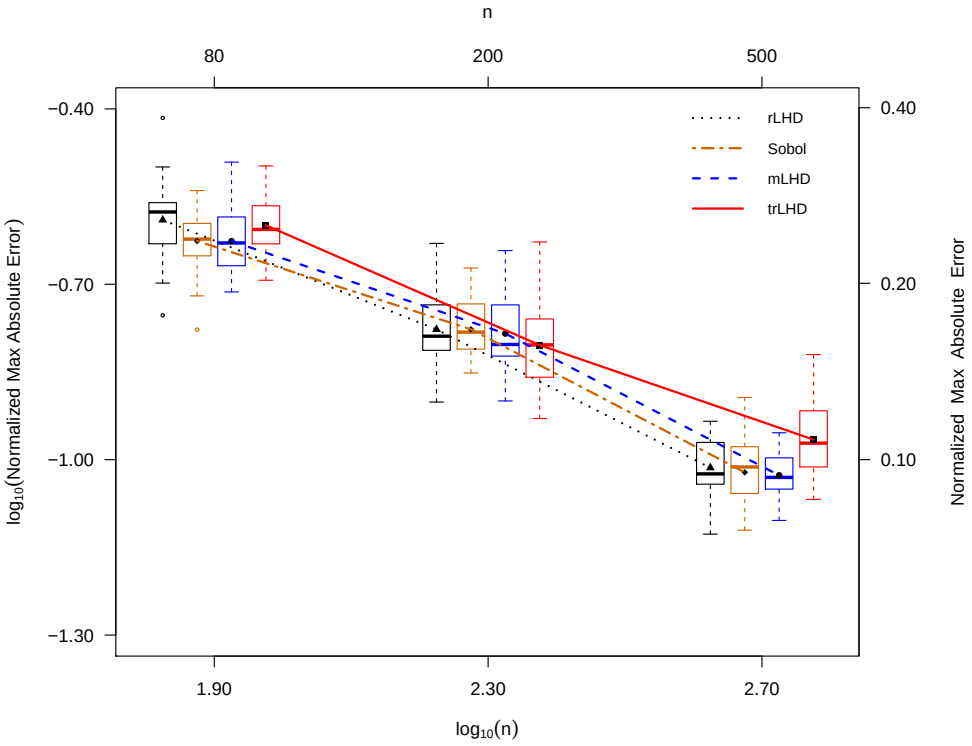

Figure C.3: Weighted Franke's function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 80, 200$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

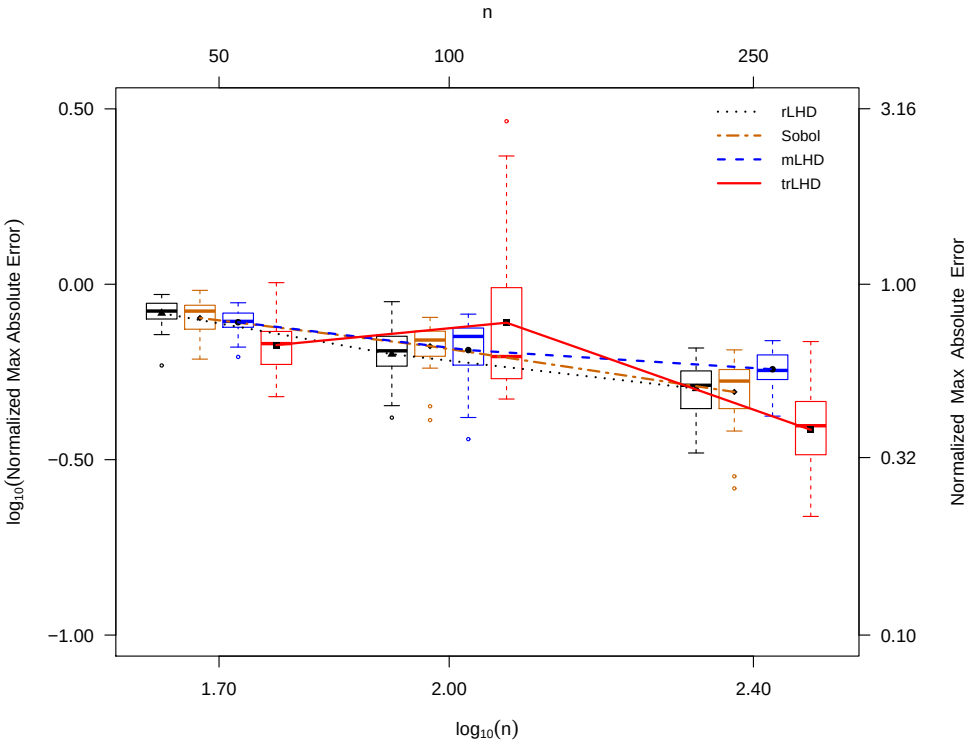

Figure C.4: Corner-peak function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 50, 100, 250$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

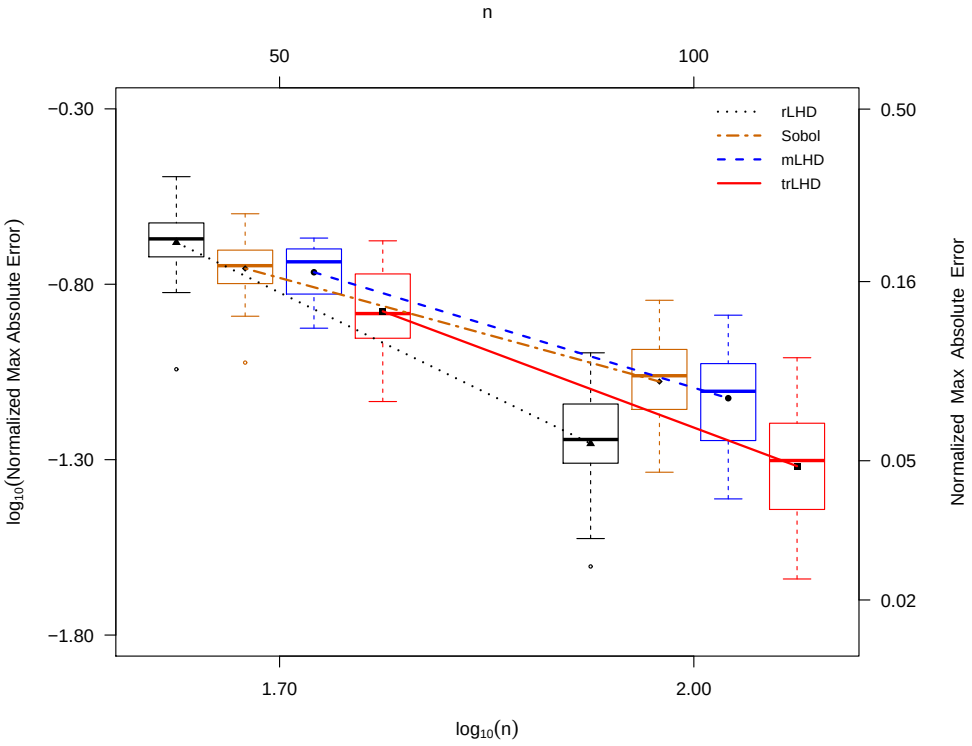

Figure C.5: Corner-peak function analyzed at the logarithmic scale: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 50, 100$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

C.2 Results of Normalized Max Absolute Errors for the Comparison of Projection Designs

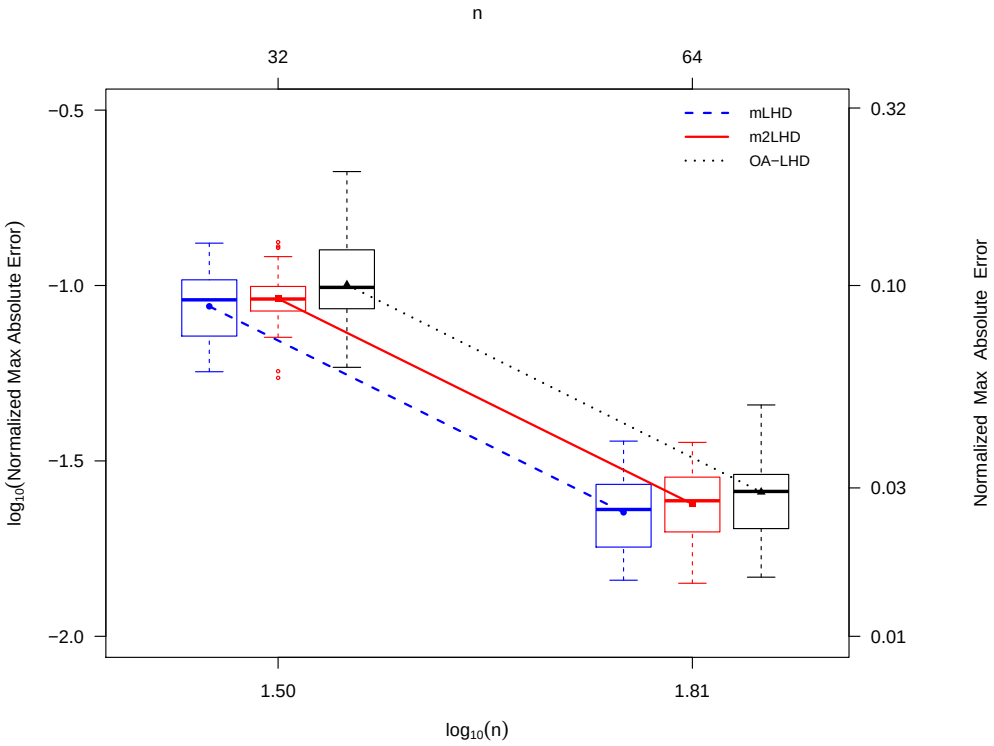

Figure C.6: Borehole function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 32, 64$ for m2LHD, OA-based LHD designs and mLHD. The two OA-based LHDs are $n = 32$ (index 2, level 4, strength 2) and $n = 64$ (index 1, level 8, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

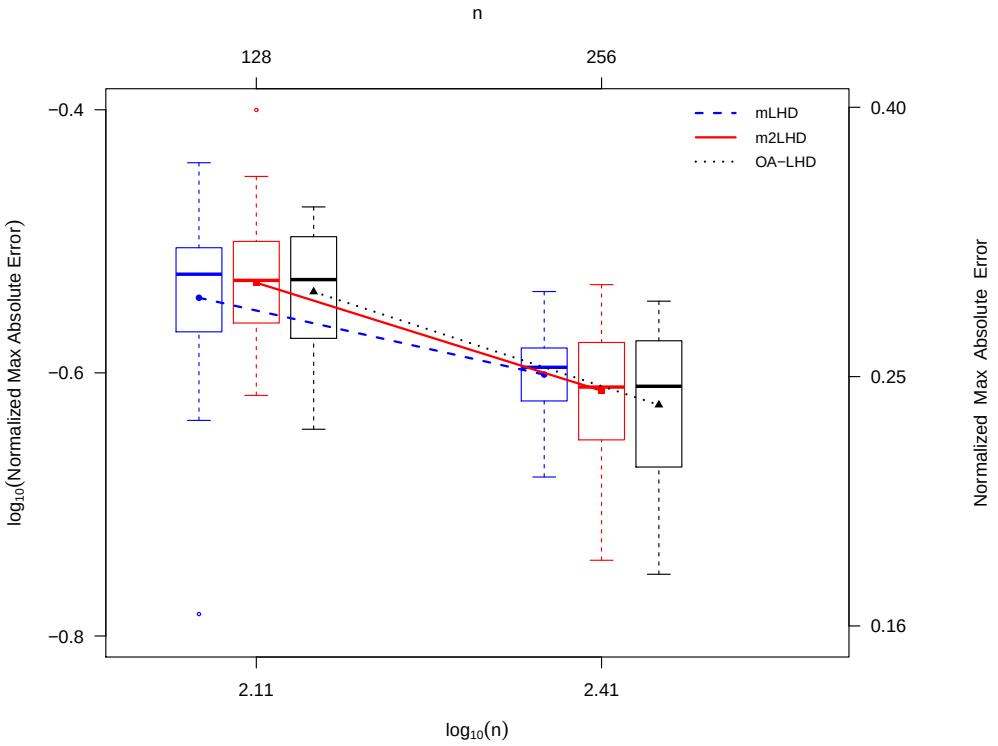

Figure C.7: PTW function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 128, 256$ for m2LHD, OA-based LHD designs and mLHD. The two OA-based LHDs are $n = 128$ (index 2, level 8, strength 2) and $n = 256$ (index 1, level 16, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

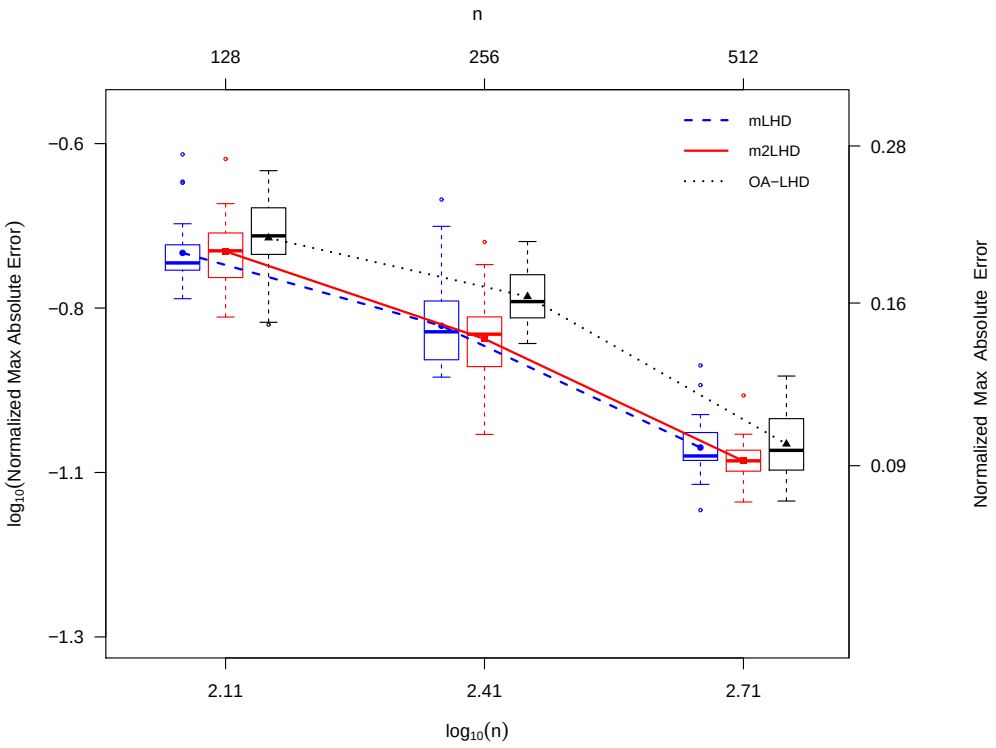

Figure C.8: Weighted Franke's function: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 128, 256, 512$ for m2LHD, OA-based LHD designs and mLHD. The three OA-based LHDs are $n = 128$ (index 2, level 8, strength 2), $n = 256$ (index 1, level 16, strength 2) and $n = 512$ (index 2, level 16, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

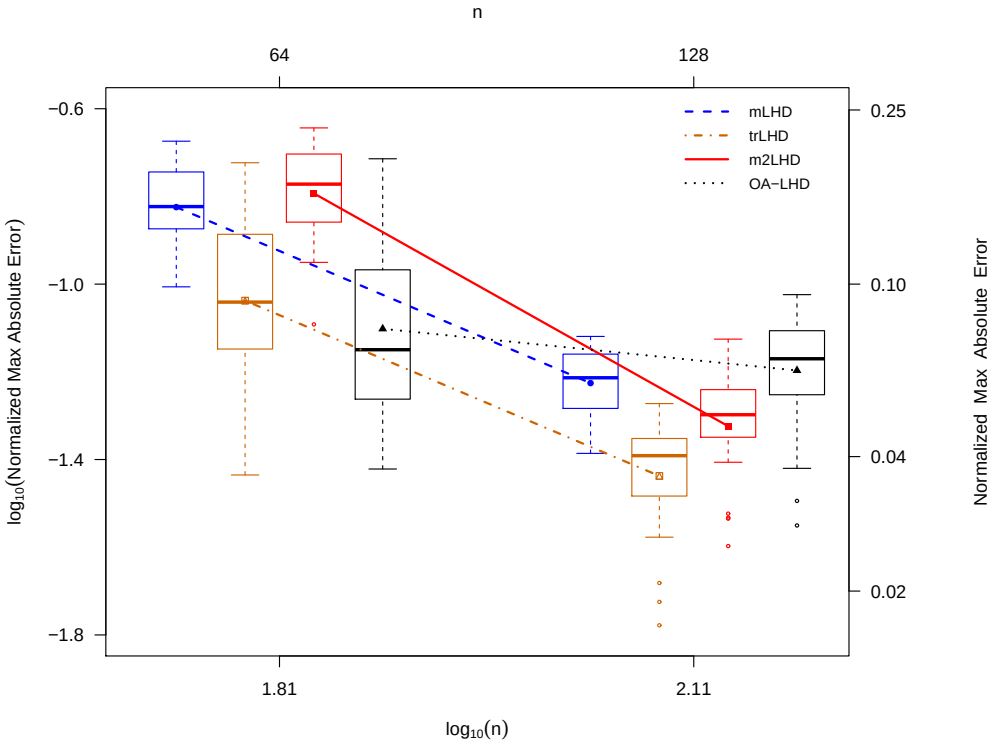

Figure C.9: Corner-peak function analyzed at the logarithmic scale: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 64, 128$ for m2LHD, OA-based LHD designs and mLHD. The two OA-based LHDs are $n = 64$ (index 4, level 4, strength 2) and $n = 128$ (index 2, level 8, strength 2). For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot. Boxplots are joined by their means.

C.3 Results of Normalized Maximum Absolute Errors for the Discussion Section

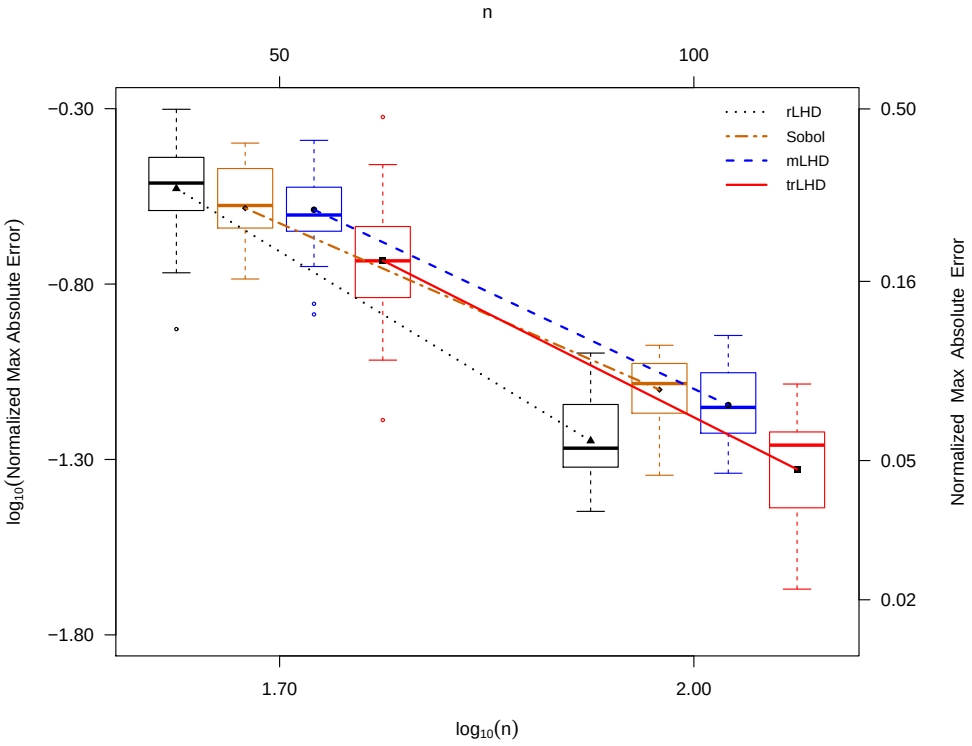

Figure C.10: Corner-peak function analyzed at the logarithmic scale with a full linear GP model: Normalized maximum absolute error of prediction, $e_{\max,\,\mathrm{ho}}$; $n = 50, 100$ for rLHD, Sobol, mLHD and trLHD designs. For each base design, 25 random permutations of its columns give 25 values of $e_{\max,\,\mathrm{ho}}$, displayed as a box plot and are joined by the means. The vertical axis limits are kept same as in Figure C.5 to faculitate direct comparison.