Emulation of Random Output Simulators

ALEXIS BOUKOUVALAS

Doctor Of Philosophy

– ASTON UNIVERSITY –

September 2010

ASTON UNIVERSITY

Emulation of Random Output Simulators

ALEXIS BOUKOUVALAS

Doctor Of Philosophy, 2010

Abstract

Computer models, or simulators, are widely used in a range of scientific fields to aid understanding of the processes involved and make predictions. Such simulators are often computationally demanding and are thus not amenable to statistical analysis. Emulators provide a statistical approximation, or surrogate, for the simulators accounting for the additional approximation uncertainty. This thesis develops a novel sequential screening method to reduce the set of simulator variables considered during emulation. This screening method is shown to require fewer simulator evaluations than existing approaches. Utilising the lower dimensional active variable set simplifies subsequent emulation analysis. For random output, or stochastic, simulators the output dispersion, and thus variance, is typically a function of the inputs. This work extends the emulator framework to account for such heteroscedasticity by constructing two new heteroscedastic Gaussian process representations and proposes an experimental design technique to optimally learn the model parameters. The design criterion is an extension of Fisher information to heteroscedastic variance models. Replicated observations are efficiently handled in both the design and model inference stages. Through a series of simulation experiments on both synthetic and real world simulators, the emulators inferred on optimal designs with replicated observations are shown to outperform equivalent models inferred on space-filling replicate-free designs in terms of both model parameter uncertainty and predictive variance.

Keywords: Gaussian Process, Fisher Information, Optimal Design, Input-dependent variance, Heteroscedasticity.

To Laurence and my parents, Στέφανος and Joyce.

Acknowledgements

I thank my supervisor Dan Cornford for his guidance and help throughout my PhD. The work presented in Chapter 3 was done in collaboration with J.P. Gosling and H. Maruri-Aguilar with whom I had many thought-provoking discussions. I also wish to thank all the members on the MUCM team for the opportunity to discuss my research and the insightful feedback they provided. I also thank Milan Stehlík for his help to understand the theory of optimal design. I appreciate the support of the NCRG department whose staff and students create such a wonderful environment to do research in. In particular thanks go to Diar Nashiev, Michail Vrettas, Harry Goldingay, Remi Barillec and all the other PhD students with whom I shared the PhD experience. Lastly, I thank my two PhD examiners, professors Ian Nabney and Chris Williams for their invaluable feedback.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

CONTENTS

Computer simulators of real world processes are of great importance in many scientific fields. Often, these simulators are both computationally expensive and require many inputs. Examples include climate projections (Hargreaves et al., 2004), estimation of national carbon balances (Kennedy et al., 2008), epidemiology (Singer et al., 2009) and systems biology (Henderson et al., 2009a), where biochemical reactions of cell processes are modelled. The problem of the computational expense of simulators can be handled using emulation technology where the simulator is approximated by a statistical probabilistic model known as an emulator. Emulation of deterministic computer simulators is a well established methodology that allows for the statistical analysis of complex computationally expensive simulators. Using these simulators for predicting outcomes for a limited number of input scenarios has been common practice but in order to quantify the uncertainty of these predictions a large number of simulator runs is required which can be prohibitively expensive for computationally demanding simulators.

The emulator is very fast to evaluate and allows subsequent analysis to be performed by leveraging the emulator as a surrogate of the simulator. In emulation of deterministic models, the probabilistic model most commonly used is the Gaussian Process (GP) which allows for the specification of a wide range of prior beliefs on the properties of the simulator response such as its smoothness and variability.

The majority of the emulator literature deals with deterministic simulators where the output is invariant to repeated executions of the simulator at the same input setting. In this thesis, the focus is on emulation of stochastic, or random output, simulators. Random output simulators typically arise where the simulator has some internal source of randomness, common examples of which are chemical and biological reaction models (described in Chapter 6) and agent-based models. In terms of analysis, stochastic simulators typically require more evaluations than deterministic systems as the additional intrinsic variability of the simulator needs to be captured. For this reason we believe emulation to be useful even for medium complexity stochastic simulators.

In Section 1.1 the contribution of this thesis is discussed, followed by an outline providing a summary of each chapter in Section 1.2. Finally Section 1.3 lists publications stemming from this work.

1.1 Contribution

High-dimensional input spaces can make the calculations required for emulation challenging. By identifying the active variables of a simulator, known as screening, subsequent tasks in the emulation framework are simplified and fewer simulator models runs are required for the analysis to proceed. A sequential screening technique is developed that is simple to implement. The tech-

nique acts in a sequential way in order to keep the number of simulator runs down to a minimum, whilst identifying the inputs that have non-linear effects. Our proposal is built upon the method proposed by Morris (1991) for screening and therefore keeps that technique's simplicity. The method utilises a threshold to separate non-linear from linear effects. As direct elicitation of this quantity can be challenging, an alternate approach is developed that allows the elicitation to be conducted on the simulator output space. The sequential method is successfully applied to the output of a 13-dimensional stochastic rabies model and it is shown to require fewer simulator runs than the batch Morris method to identify the same set of factors as having non-linear effects.

Following the screening procedure, the parameters of the Gaussian Process (GP) emulator are inferred on the reduced active variable design space. As the variance of stochastic simulators is often a function of the input variables, two novel methods of performing GP regression on heteroscedastic datasets with replicated observations are developed. The Coupled GP method builds on the work of Kersting et al. (2007) by explicitly considering replicate observations and applying corrections due to finite sample size effects. The resulting model is flexible and inference is efficient for designs with replicate observations as the moments of the replicates are used rather than repeating the observations. When the simulator variance response is sufficiently simple or expert judgements are available, a simpler parametric variance model can be utilised. For such cases the Joint Likelihood model is introduced, where a deterministic functional form is used for the variance response.

The process of inferring the parameters of the GP model requires the selection of a set of input points, an experimental design, at which to evaluate the simulator. A model-based experimental design method is developed that is shown to reduce the variance of parameter estimation under both Maximum Likelihood (ML) and fully Bayesian inference. The method is based on the utilisation of the Fisher information to select the maximally informative set of points with respect to parameter estimation. Using the Joint Likelihood model allows for the analytic derivation of the Fisher information for designs with replicate observations. An extensive simulation study is presented to examine the impact of the model-based optimal designs on both parameter estimation and predictive variance using the Joint Likelihood GP model.

1.2 Outline

In Chapter 2 an overview of the framework of emulation for computer models is presented to set the context for the thesis. In particular this chapter does not provide an extensive literature review but rather an overview of the main methods and techniques in the emulation of computer simulators. An in-depth review of screening, heteroscedastic GPs and optimal design is provided

in the corresponding chapters.

The identification of the simulator active variables, known as screening, is discussed in Chapter 3. Following a review of existing screening methods, the sequential Morris method is presented, which is a novel approach to help identify factors with non-linear effects on the simulator response using a smaller number of simulator runs than a batch approach.

In Chapter 4 the GP framework is extended to admit the case of input-dependent noise, known as heteroscedastic regression. Two novel methods of performing heteroscedastic GP regression on complex datasets with replicated observations are presented. These are applied to a one dimensional function and their performance discussed.

Optimal experimental design is discussed in Chapter 5. The aim of the methodology is to maximise the information provided by a set of input locations, known as an experiment design, with regards to a specified criterion. In this thesis we use the Fisher Information as the criterion in order to minimise the generalised variance of the parameter estimation. An extensive set of simulation results is presented to examine the impact of optimal designs on emulation under both Maximum Likelihood (ML) and fully Bayesian inference.

In Chapter 6 the screening, emulation and optimal design frameworks discussed in the previous chapters are applied to real world stochastic models. The sequential Morris method is applied to a 13-dimensional stochastic rabies model to identify the factors that are most relevant in determining the probability of disease extinction within five years. Two stochastic models simulating biological reactions within a cell are utilised in Section 6.3 with the aim of demonstrating the heteroscedastic emulation and optimal design methods presented in Chapters 4 and 5 respectively.

Finally in Chapter 7 the thesis is summarised and we conclude with a discussion of the research outcomes and possible directions for future research.

1.3 Disclaimer

This thesis is submitted for the degree of Doctor of Philosophy (Ph.D). The work presented here is original and has not been submitted previously for a degree, diploma or qualification anywhere else. However, parts of the work have been published and presented in the following papers, conferences and seminars:

1. Boukouvalas, A., Cornford, D., Singer, A., Managing Uncertainty in Complex Stochastic Models: Design and Emulation of a Rabies Model. Accepted to St. Petersburg Workshop on Simulation (2009).

2. Boukouvalas, A. and Cornford, D., Experimental Design for Heteroscedastic Gaussian Process emulators. Accepted for poster presentation at the Machine Learning Summer School,

Cambridge, (2009).

3. Boukouvalas, A., Cornford, D., Maniyar, D. M. and A. Singer, Gaussian process emulation of stochastic models: developments and application to rabies modelling. Accepted for poster presentation at the Royal Statistical Society Conference, Nottingham, (2008).

2 Overview of Emulation for Computer Models

CONTENTS

This chapter provides a brief introduction to emulation methodology. An overview of the main methods and techniques involved in the emulation of simulators is provided. The discussion is not intended as an in-depth review of each stage of emulation. An extended review of the screening, heteroscedastic emulation and optimal design aspects of emulation is provided in the chapters 3, 4 and 5 respectively.

Definitions of terms commonly used in this thesis are provided in Section 2.1. In Section 2.2 an overview of the emulation methodology is presented, followed by a discussion of each stage. An overview of experimental design is given in Section 2.3. In Section 2.4 the GP formalism is presented which forms the basis of the statistical approximation to the simulator. The validation of the emulator is discussed in Section 2.5. We conclude with a summary in Section 2.6.

2.1 Definitions

For clarity in the discussion that follows in subsequent chapters, we define some key terms:

- *Experimental Design*: A set of input combinations at which to evaluate the simulator.
- *Optimal Experimental Design*: The use of mathematical and statistical methods to select the minimum number of experiments for optimal coverage of descriptor or variable space. In the context of this thesis we are considering a design over the input space of the simulator model.
- *Latin Hypercube*: A square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. A *Latin Hypercube* is the generalisation of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it.
- *Simulator*: A simulation is an imitation of some real thing, state of affairs, or process. The act of simulating something generally entails representing certain key characteristics or behaviours of a selected physical or abstract system. In the context of this thesis the simulator is typically a piece of computer code (or function), with a set of inputs and outputs.
- *Input variables* or *Inputs* or *Factors*: the set of variables required to determine the output of the simulator. These might include both parameters of the system being modelled and the initial (time) state of the system.
- *Emulator*: The Gaussian Process emulator is a statistical approximation to the simulator which is faster to run and allows a variety of subsequent analyses to be carried out. The statistical approximator need not be a GP but we do not consider such cases here.

- *Replicated observations*. Observations obtained through repeated evaluations of the simulator at a fixed input value.

2.2 Emulation Process

In Figure 2.1 a diagrammatic overview of emulation methodology is provided. Given a description of *reality*, the *simulator* is developed. The simulator might then be validated using a set of *observations*. The description of the system typically involves physical laws and mechanisms or other known structural information regarding the system. Simulators vary greatly in their complexity. For highly complex systems such as the climate, the resulting simulators can have high computational requirements. For certain analysis such as *sensitivity analysis* (described in Chapter 3) where the effect of inputs on output uncertainty is examined, a prohibitively large number of simulator runs would be required. In such cases an *emulator* is constructed which acts as a statistical surrogate of the simulator for subsequent analysis.

The emulator is constructed by first identifying the most relevant inputs by employing a *screening* technique. The aim of the screening procedure is to identify the most relevant factors in terms of their effect on the simulator output. The least relevant factors can be fixed to their nominal values or discarded entirely for the subsequent steps which can greatly simplify the analysis. We discuss existing methods and propose a sequential screening method in Chapter 3.

Experimental design is the process of selecting input points at which to evaluate the simulator. In this thesis we focus on *optimal* experimental design where the design is selected such that a criterion function is maximised. In most instances screening precedes optimal experimental design as the latter usually requires a numerical optimisation of the criterion function which can be more easily accomplished in a lower dimensional space. In the case of adaptive experimental design, the process can be iterated whereby new sets of points at which to evaluate the simulator are proposed at each stage. Experimental design is discussed further in Section 2.3.

The next step of the methodology involves the construction of the emulator. This proceeds in two stages. Firstly a prior specification of the functional form of the simulator is used to construct a *Gaussian Process* prior model. The unknown parameters of the prior model are *inferred* using the experimental design and the corresponding simulator evaluations. This process is described in more detail in Section 2.4.

Prior to utilising the emulator, *validation* methods are employed to check the correctness of the emulator. This procedure aims to uncover incorrect prior specifications or inference issues which would lead to a poor fit of the emulator to the simulator. Typically a separate experimental design to the training set used in emulation inference is used for validation. Validation methods

are described in Section 2.5.

If the emulator is shown to be an acceptable surrogate to the simulator, statistical analysis techniques can be employed using the emulator rather than the slower simulator. Examples of such analysis are *uncertainty analysis* (O'Hagan et al., 1998), where the effect of input uncertainty on the simulator output is calculated, and *calibration* (Kennedy and O'Hagan, 2001) where given a set of observations of *reality*, the model parameters are inferred and possibly the discrepancy of the simulator to *reality* is estimated.

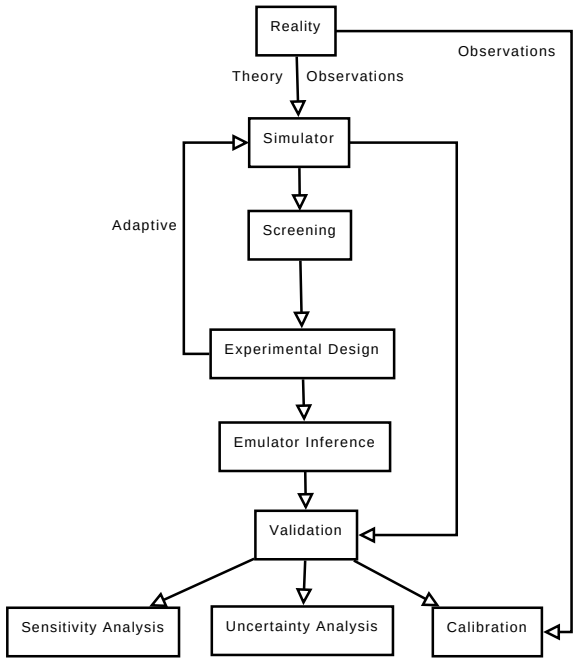


Figure 2.1: Diagrammatic view of emulation methodology.

2.3 Experimental Design

In this section, we briefly review existing approaches to experimental design for computer experiments. The discussion is based on MUCM Toolkit (World Wide Web electronic publication, Release 6, 2010), page `ThreadTopicExperimentalDesign`, to which the reader is referred for a more extensive discussion.

Two main classes of design are developed in the literature: general purpose designs that can be used for a variety of simulators or model-based designs that are optimal, in some sense, for a particular model.

General purpose designs are developed utilising geometric criteria. A frequently used type of general purpose design are space-filling designs. Such designs place points so that they are well separated and cover the input space well. The rationale is that for deterministic simulators points very close to each other carry little information due to the process correlation. This does

not hold for stochastic simulators which we consider in this thesis as even replicated observations are informative. Also for deterministic simulators discovering the range of correlation requires a range of inter-point distances in the design.

A variety of space-filling designs have been used for computer experiments:

- *Fully Factorial Design*. A set of predetermined p values, called levels, is assigned to each factor. The design is the combination of all possible levels of all factors. However even if only 2 levels are assigned to each factor, the number of required runs for a d dimensional design space is 2^d. This is usually prohibitively high for most simulators and Fractional Factorial designs have been developed that consist of subsets of the Full Factorial design.

- *Optimised Latin Hypercube Designs*. A Latin Hypercube (LH) is a random set of points subject to the constraint that for each input factor the points are evenly spread in the design domain. LH designs are not guaranteed to be space-filling in the entire design domain but rather just in each dimension separately. The most commonly applied approach to enhance the space-filling property of LH designs is to generate a large number of them and select the LH where the minimum distance between points is maximised. This is known as the *Maximin Latin Hypercube* design and is employed extensively in our simulation studies in this thesis. An example is presented in Figure 2.2.

- *Pseudo-Random Sequences*. A specific set of pseudo-random generating functions have been shown to generate space-filling designs. The Sobol' sequence in particular is an example of a low discrepancy sequence where discrepancy is a measure of departure of a set of points from a uniform spread. The benefit of using such sequences is that they are very fast to generate and can be employed in a sequential setting where more points may be generated as needed. However, especially for small design sizes, clusters and ridges of points may be generated by such a sequence. For more information on the construction of the Sobol' sequence see Kuipers and Niederreiter (2005).

In some instances a more sophisticated design approach is needed. Applying space-filling designs, points will not be placed in close proximity. However to estimate certain kernel parameters in the GP such as the length-scale parameters, it is beneficial to have points close to each other. Further, geometrical designs cannot easily be altered to accommodate prior information on the model parameters. Model-based design allows the specification of a model and a criterion function with respect to which the design is optimised.

Two main types of criterion functions have been explored in the literature:

- Minimise average/maximum predictive uncertainty.

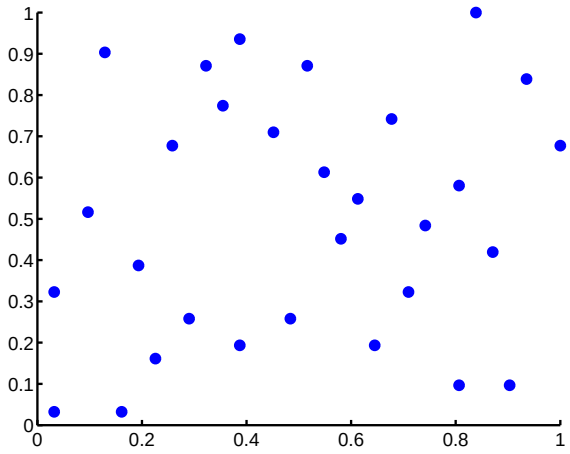


Figure 2.2: Example of 30-point Maximin Latin Hypercube.

- Minimise generalised variance of parameters.

Criteria used for computer experiments are typically based on minimising predictive variance because the quality of prediction is of critical importance in the usefulness of the emulator. However all such methods that we are aware of require an initial parameter estimate and therefore could benefit from an initial design that minimises parameter uncertainty. For example in Krause et al. (2008), where the Mutual Information criterion is used to minimise predictive variance at unsampled locations, the GP kernel parameters are assumed to be known. In Krause and Guestrin (2007) a hybrid approach of switching between exploration, where the design is optimised for parameter estimation, and exploitation, where the parameters are fixed, is developed. In Youssef (2010) a Karhunen Loeve expansion is used to linearise the GP correlation function. An initial Latin Hypercube design is used to estimate the parameters prior to the expansion. We therefore believe a pragmatic approach to design should incorporate explicit minimisation of parameter uncertainty as robust parameter estimation would allow for more robust prediction. Optimal and Hybrid design approaches are discussed more extensively in Chapter 5.

2.4 Gaussian Processes

Formally a Gaussian Process is defined as (Rasmussen and Williams, 2006):

Definition 2.4.1. *A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

For the discussion that follows we use the observational model:

$$t = f(x) + \varepsilon,$$

where $f(x)$ is the noise-free unknown function we wish to interpolate, ε is a Gaussian noise variable and t are the observed noisy functional values.

GPs are an example of a non-parametric method as they characterise a prior over functions directly instead of requiring an explicit parametrisation of the unknown function f (Mackay, 1998). In this thesis we assume a zero-mean GP prior but we note that all results can be readily extended to include a non-constant mean function. GP models are closely related to Kriging in the geostatistical literature (Stein, 1999b).

The GP framework is derived in Section 2.4.1 and a list of covariance functions used in this thesis is given in Section 2.4.2. How GPs are used for prediction is described in Section 2.4.3 and how parameters are inferred is discussed in Section 2.4.4. An example of GP inference and prediction is given in Section 2.4.5 and issues and extensions of the GP framework are reviewed in Section 2.4.6.

2.4.1 Derivation of a GP

A GP can be understood by considering a finite linear-in-the-parameters model $t = \Phi(\mathbf{X})w + \varepsilon$ where $\Phi(\mathbf{X})$ is a $n \times M$ matrix of M fixed basis functions applied on n points, w an M-dimensional parameter vector, ε a Gaussian distribution $N(0, \sigma^2\mathbf{I})$ and $\mathbf{I}$ the identity matrix.

By placing a Gaussian prior on the parameters,

$$p(w) = N(0, a^{-1}\mathbf{I}),$$

the posterior of the noise-free function $f = \Phi(\mathbf{X})w$ is Gaussian with mean and covariance

$$E[f] = \Phi(X)\, E(w) = 0,$$
$$\Sigma(f) = \Phi(X)\, E[ww']\, \Phi(X)' = a^{-1}\, \Phi(X)\Phi(X)'.$$

The noisy observations t can be described by a GP:

$$p(t) = N(0, a^{-1}\Phi(X)\Phi(X)' + \sigma^2\mathbf{I}) = N(0, K + \sigma^2\mathbf{I}).$$

where $K = [k(x,x')]$ the matrix obtained via the evaluation of the kernel function at all pairs of training points.

If an algorithm depends solely on inner products in input space it can be lifted into higher dimensional spaces by replacing the inner products with a kernel function $k(x,x')$ (Rasmussen and Williams, 2006). This is known as the kernel trick and allows the GP to operate even in infinite

dimensional spaces of basis functions ($M \to \infty$).

2.4.2 Covariance functions

A GP is defined by a *mean* and a *covariance* function, the specification of which allows the incorporation of prior knowledge in the emulation analysis such as the smoothness and differentiability of the approximated function.

In this thesis we use the following covariance functions (Rasmussen and Williams, 2006):

- Squared Exponential:

$$k_\theta^{SE}(r) = \sigma_p^2 \exp\left(-\frac{r^2}{2\lambda^2}\right),$$

- Exponential, also known as Ornstein-Uhlenbeck (OU):

$$k_\theta^{OU}(r) = \sigma_p^2 \exp\left(-\frac{r}{\lambda}\right),$$

- Matérn with fixed order 5/2:

$$k_\theta^{Mat}(r) = \sigma_p^2 \left(1+\frac{r\sqrt{5}}{\lambda}+\frac{5r^2}{3\lambda^2}\right)\exp\left(-\frac{r\sqrt{5}}{\lambda}\right),$$

where $r = ||x_i - x_j||$ the Euclidean distance between support points. The kernel parameters are $\theta = (\sigma_p^2, \lambda)$. The σ_p^2 is known as the process-variance term and controls the amplitude of the kernel response. λ, commonly referred to as the length-scale parameter, has the effect of rescaling the inputs and can be used to infer the relative importance of an input - see Section 3.1.1.

2.4.3 Prediction

Assuming the GP covariance parameters are known, prediction of the output t_* at a new site x_* given the training data $\{\mathbf{x}, \mathbf{t}\}$ can be calculated using the conditioning property of Gaussian distributions. Specifically the joint distribution $p(\mathbf{t}, t_*|\mathbf{x}, x_*)$ is:

$$p(\mathbf{t}, t_*|\mathbf{x}, x_*) = \mathcal{N}\left(\begin{bmatrix} t_* \\ \mathbf{t} \end{bmatrix} | \mathbf{0}, \begin{bmatrix} K(x_*, x_*) + \sigma^2\mathbf{I} & K(x_*, x) \\ K(x_*, x)^T & K(x, x) + +\sigma^2\mathbf{I} \end{bmatrix}\right),$$

and by conditioning on the training set (Appendix A.3) the predictive distribution $p(t_*|\mathbf{x}, x_*, \mathbf{t})$ is also Gaussian and has mean and covariance:

$$E[t_*] = K(x_*, \mathbf{x})\left[K(\mathbf{x}, \mathbf{x}) + \sigma^2\mathbf{I}\right]^{-1}\mathbf{t}, \tag{2.1}$$

$$\mathrm{Cov}[t_*, t_*] = K(x_*, x_*) + \sigma^2\mathbf{I} - K(x_*, \mathbf{x})\left[K(\mathbf{x}, \mathbf{x}) + \sigma^2\mathbf{I}\right]^{-1}K(\mathbf{x}, x_*). \tag{2.2}$$

2.4.4 Inference

Given a training set $\{\mathbf{x}, \mathbf{t}\}$ there exist a range of methods to infer the kernel parameters θ.

The simplest approach, known as *Maximum Likelihood* (ML), is to maximise the marginal log likelihood of the GP:

$$\log p(\mathbf{t}|\mathbf{x}, \theta) = -\frac{1}{2}\log|K + \sigma^2\mathbf{I}| - \frac{1}{2}\mathbf{t}'(K + \sigma^2\mathbf{I})^{-1}\mathbf{t} - \frac{N}{2}\log(2\pi), \tag{2.3}$$

where N the number of training points and $|\ldots|$ denotes the determinant. Derivatives of Equation (2.3) with respect to the kernel parameters θ can then be used to optimise the likelihood using a non-linear numerical optimisation method such as scaled conjugate gradient (Nabney, 2001). Kernel parameters required to be non-negative may still be optimised using general numerical methods by optimising their logarithm, i.e. optimising with respect to $z = \log(\theta)$.

If informative priors are available for the kernel parameters, a *Maximum-A-Posteriori* (MAP) estimation is obtained by maximising the logarithm of the parameter posterior:

$$\log p(\theta|\mathbf{t}, \mathbf{x}) \propto \log p(\mathbf{t}|\mathbf{x}, \theta) + \log p(\theta),$$

where $p(\theta)$ is the parameter prior.

A method known as *Restricted Maximum Likelihood* (REML) (Neumaier and Groeneveld, 1998) arises when a non-zero linear-in-the-parameters mean function is used in the GP prior. Under such a setup it is possible to assign an uninformative improper prior on the mean function parameters and analytically integrate them out of the likelihood. This method is not examined further in this thesis as we do not utilise non-constant mean functions in the GP prior.

In a fully Bayesian approach, the conditioning of the likelihood on the kernel parameters can be integrated out $p(\mathbf{t}|\mathbf{x}) = \int p(\mathbf{t}|\mathbf{x}, \theta)p(\theta)$. However this integral is highly intractable and *Markov Chain Monte Carlo* methods have been employed to perform the integration numerically (Neal, 1997). In Section 5.7 we utilise a Markov Chain method to examine the effect of optimal designs on inference for parameter posteriors.

2.4.5 Example

In this section a simple example of GP inference and prediction is presented. A zero-mean GP Prior with a squared exponential kernel and a constant nugget σ_n^2 is placed on the unknown simu-

lator function f:

$$E[f(\mathbf{x})] = 0,$$
$$Cov(\mathbf{x}_i, \mathbf{x}_j) = \sigma_p^2 \, e^{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{\lambda^2}} + \delta_{ij}\sigma_n^2.$$

The nugget parameter σ_n^2 is constant across the entire input domain. This variance model is known as homoscedastic and will be extended in Chapter 4 to allow for the modelling of input dependent variance. Even in the case of deterministic simulators, a nugget is included in the GP covariance specification as it helps with numerical stability issues.

The parameters are inferred using the ML method on a training set of six points. The predictive distribution is shown in Figure 2.3. As the training set is concentrated on the first half of the design space, the predictive GP model reverts to the mean (0) away from the training points. This is a consistent feature of stationary GPs when extrapolating - the model reverts to the mean and the predictive variance reaches a maximum value equal to the sum of the process-variance σ_p^2 and the nugget σ_n^2 (known as the sill). These features can be understood by examining the predictive equations (2.1)-(2.2) and setting the training-test point correlation to zero.

Another feature of the GP fit is that variance does not collapse to zero at the training points but is equal to the nugget variance σ_n^2. If no nugget term was included, the GP mean would interpolate exactly at the training points with zero variance at those points.

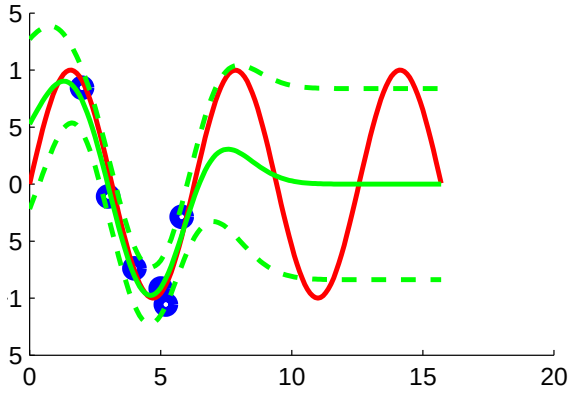


Figure 2.3: An example of a Gaussian Process inference and prediction. The blue dots denote the training points, the red line the simulator, the green solid and dashed lines the GP mean and variance prediction respectively.

2.4.6 Extensions

The GP formalism described previously has been extended in a variety of ways to extend its applicability. In this section we describe some of the extensions relevant to the field of computer experiments. A more complete discussion of GP extensions is given in Rasmussen and Williams

(2006).

GP methods are limited to small designs due to the inversion of the training covariance matrix $K+\sigma^2\mathbf{I}$ which appears in the likelihood (Equation (2.3)) and requires order $O(N^3)$ computations where N the number of training points. A range of sparse approximation methods have been developed to overcome this limitation and a general theoretical framework to describe them is presented in Quinonero-Candela and Rasmussen (2005). In Chapter 4 we review one such method, the Sparse Pseudo-Input GP, where the effect of the N training points is projected to a smaller set of M basis points to reduce the computational load to $O(NM^2)$.

Another approach to extend GPs to larger datasets is to distribute the training set into a set of disjoint sets and perform inference and prediction separately. In Latouche (2007), a method of combining the individual joint predictions using the Bayesian Committee Machine (Tresp, 2000) is presented. In addition, by factorising the GP parameter posterior and employing the Laplace propagation algorithm (Smola et al., 2004), a joint optimisation across nodes is also proposed.

GPs can also be extended to multivariate outputs. The simplest approach where each output is treated independently ignores correlations between outputs, that can be utilised for more accurate inference and prediction. In the separable model (Conti and O'Hagan, 2007; Bonilla et al., 2008) the GP covariance is represented by a Kronecker product $\Sigma_O \otimes \Sigma_I$ where Σ_O is the between-output covariance and Σ_I the input correlation. The drawback of this approach is that although output correlations are explicitly modelled, the smoothness of all outputs is assumed to be identical as a common set of input correlations length-scales is used. The Linear Model of Coregionalisation (Goulard and Voltz, 1992) allows the modelling of each output through a linear combination of kernels thus removing the input-output separability assumption. However a larger number of parameters needs to be inferred.

Lastly, when no nugget parameter is included in the GP covariance function, as is the case for the emulation of deterministic simulators, and a Gaussian prior or an improper uninformative prior is used for the process-variance parameter, the integration of the parameter uncertainty can proceed analytically leading to the Student-t process (Kennedy and O'Hagan, 2001). However when adding a nugget parameter to the GP covariance the integration can no longer proceed unless the nugget is entangled with the process-variance parameter (Rasmussen and Williams, 2006, Section 9.9). The latter refers to a reparametrisation of the covariance as $\tau(k(.,.)+\nu)$ where τ captures both the nugget and process-variance effects on the output. The nugget in this case is entangled, $\tau \times \nu$, and interpretation becomes more difficult as the observations cannot be written as the sum of independent signal and noise contributions (Rasmussen and Williams, 2006).

2.5 Validation

Various diagnostics are used in the literature to validate emulators. The Mean Squared Error (MSE) is used to assess the predictive accuracy of the GP with regards to the mean only. We utilise a standardised form (divided by the sample variance of the observations):

$$\text{MSE} = \frac{1}{N\text{Var}[\mathbf{y}]} \sum_{i=1}^{N} (E[\mathbf{t}_i] - \mathbf{y}_i)^2,$$

where $E[\mathbf{t}_i]$ the GP predictive mean defined in Equation (2.1) for test point $i \in \{1,\ldots,N\}$, $\mathbf{y}_i$ the observation at that point and $\text{Var}[\mathbf{y}]$ the sample variance of the observations. This is referred to as the Standardised MSE in Rasmussen and Williams (2006), page 23. For a trivial model which predicts using the mean of the training targets, the value of the MSE will be close to 1. Smaller values can be interpreted as the model doing better than this trivial model.

The Negative Likelihood Predictive Distribution (NLPD) weighs the errors on the mean by the predictive variance, therefore penalising incorrect variance estimates (Rasmussen and Williams, 2006, page 23):

$$-\log p(\mathbf{y}_*|D,\mathbf{X}_*) = \frac{1}{2N_v} \sum_{i=1}^{N_v} \left(\log(2\pi\sigma_i^2) + \frac{(\mathbf{y}_i - \mathbf{t}_i)^2}{\sigma_i^2} \right),$$

where the likelihood is evaluated at the test set $\mathbf{X}_*$. However this is a univariate measure which ignores the correlation structure between test points and takes a simple average across all test points. For this reason we utilise the Dawid score, a multivariate extension of the NLPD, which is defined as a loss:

$$\text{Dawid} = \log|\Sigma| + (\mathbf{y}-\mathbf{t})^T \Sigma^{-1} (\mathbf{y}-\mathbf{t})$$

where $|\ldots|$ denotes the determinant and Σ the covariance matrix of the joint predictive distribution at the set of test points. Bastos (2010) notes that the difference between the Dawid scores of two competing models can be seen as a numerical approximation to the log Bayes factor.

Finally the Mahalanobis error (D_{MD}) is a more precise error measure than the NLPD since the full predictive covariance is utilised without assuming the errors are uncorrelated. Unlike the Dawid score, the Mahalanobis error sampling distribution can be derived analytically and allows for various decompositions to help identify the sources of error.

$$D_{MD} = (\mathbf{y} - E[\mathbf{t}_*])^T \text{Cov}[\mathbf{t}_*,\mathbf{t}_*]^{-1} (\mathbf{y} - E[\mathbf{t}_*]),$$

where $\text{Cov}[\mathbf{t}_*,\mathbf{t}_*]$ is the predictive GP covariance defined in Equation (2.2). Sampling theory given in Bastos and O'Hagan (2009) allows interpretation and further analysis of the Mahalanobis error. In the case of GPs the asymptotic distribution of the Mahalanobis distance is proven to be a χ^2

distribution with n degrees of freedom where n is the size of the test set. In particular the theoretical mean value of D_{MD} for GPs is the number of validation points. Lower values than the theoretical mean can signify an underconfident GP where the predictive variance it too high. Higher values on the other hand typically occur when the GP predictions are overconfident.

The Mahalanobis error may be decomposed using the Cholesky decomposition to $D_{MD} = \mathbf{v}^T\mathbf{v}$ where the components of the $\mathbf{v}$ vector are termed uncorrelated errors and allow for the identification of the contribution to the total error of each validation input point. As the uncorrelated errors have a theoretical distribution of $N(0,1)$, errors larger than two standard deviations may be indicative of an issue with the emulator and can be further investigated by examining the emulator behaviour at the corresponding input locations. Lastly, Bastos and O'Hagan (2009) propose the use of the Pivoted Choleksy Decomposition (PCD) to decompose the Mahalanobis error. In PCD the data is permuted such that the first element is the one with the largest variance, the second element is the one with the largest predictive variance conditioned on the first element and so on. The benefit of this decomposition is that the ordering of the errors aids in the identification of possible causes. For instance, errors early in sequence are typically on test points far from training data where the predictive variance is high and possible causes include non-stationarity of the function output and misidentification of the process-variance/nugget terms. Errors at the end of the sequence are typically from test points close to training points or test points close to other test points and point to a problem in the identification of the correlation structure, i.e. the covariance parameters (Bastos and O'Hagan, 2009). The PCD decomposition is utilised in Section 5.6.6 to validate emulators trained on different designs.

When comparing optimal designs in Chapter 5, the emulators will be structurally identical with the only difference being the training set used in each case. We utilise both the Mahalanobis error and Dawid score to evaluate the impact of the designs on emulator performance. In the case of invalid emulators where the Mahalanobis error is found to be larger than the expected range from its theoretical distribution, the PCD of the Mahalanobis error allows for the identification of the likely cause of the error. The difference of the Dawid scores of two emulators is equivalent to the log likelihood ratio of the two models evaluated at the same test set. In the case of Bayesian inference (Section 5.7) as the same priors are used for the hyperparameters of competing emulators, the difference in Dawid score is proportional to the Bayes factor allowing for direct model comparison.

2.6 Summary

In this chapter, an introduction to the concepts behind experimental design and emulation was offered. In Section 2.1 definitions for terms that are frequently used in this thesis have been provided.

The emulation framework was introduced in Section 2.2. The role of emulation within the context of simulation of systems was discussed and each stage of emulation briefly described. An overview of experimental design for computer experiments was provided in Section 2.3. The distinction of geometric designs, that can be used for a wide range of simulators, to model-based optimal design theory, where the design stems from the optimisation of a functional criterion of a probabilistic model, was discussed. Classes of geometric designs such as the Latin Hypercube were reviewed and will be contrasted to optimal designs in chapters 5 and 6 through a set of simulation experiments. A more extensive discussion of optimal design, which is the focus of this thesis, is given in Chapter 5.

In Section 2.4 the GP framework was presented. The framework was derived (Section 2.4.1) by considering a finite linear-in-the-parameters fixed basis model where the kernel trick was applied to arrive to the full non parametric GP model. The GP framework allows for flexibility in the modelling through the specification of different covariance structures reflecting different beliefs of simulator behaviour. The list of covariance functions used in this thesis was given in Section 2.4.2. The predictive equations of the GP were derived in Section 2.4.3 and a multitude of methods on how to learn the GP parameters, a process known as inference, were presented in Section 2.4.4. An example of inference and prediction was given in Section 2.4.5 and relevant extensions to the GP framework were discussed in Section 2.4.6.

Finally, methods of validating the GP emulator approximation to the simulator were presented in Section 2.5. In particular the Mahalanobis diagnostic was described which is extensively used in the subsequent chapters to validate the emulator fit under different designs.

3 Screening

CONTENTS

In this chapter we discuss the issues of screening within context of emulating of stochastic simulators. The material presented is based on Boukouvalas et al. (2010).

Screening involves identifying the relevant input factors that drive a simulator's behaviour (Saltelli et al., 2000). Screening, also known as variable selection in the machine learning literature, is a research area with a long history. Traditionally, screening has been applied to physical experiments where a number of observations of reality are taken. One of the primary aims is to remove, or reduce, the requirement to measure inconsequential quantities (inputs) thus decreasing the time and expense required for future experiments. More recently, screening methods have been developed for computer experiments where a simulator is developed to model the behaviour of a physical, or other, system. In this context, the quantities represent the input variables and the benefit of reducing the dimension of the input space is on the emulator model complexity and training efficiency rather than on the cost of actually obtaining the input values themselves.

With the increasing usage of ever more complex models in science and engineering, dimensionality reduction of both input and output spaces of models has grown in importance. It is typical, for example in complex models, to have several tens or hundreds of input (and potentially output) variables. In such high-dimensional spaces, efficient algorithms for dimensionality reduction are of paramount importance to allow effective probabilistic analysis. For very high (say over 1000) sizes of input and/or output spaces open questions remain as to what can be achieved. Even in simpler models, efficient application of screening methods can reduce the computational cost and permit a focused investigation of the relevant factors for a given model.

Screening is a constrained version of dimensionality reduction where a subset of the original variables is retained. In the general dimensionality reduction case, the variables may be transformed before being used in the emulator, typically using a projection. The transformation, or projection, may be linear as is the case for the commonly used Principal Components Analysis method (PCA) or non-linear as is the case in the Neuroscale algorithm, where a radial basis function network is used to perform the mapping (Lowe and Tipping, 1997). An overview of dimensionality reduction methods for emulation is given in Boukouvalas and Cornford (2008).

Both screening and sensitivity analysis may be utilised to identify variables with negligible total effects on the output variables. They can provide results at various levels of granularity from a simple qualitative ranking of the importance of the input variables through to more exact quantitative results of the percentage of output variance explained by each factor. Sensitivity analysis methods provide more accurate variable selection results but require larger number of simulator evaluations, and thus entail higher computational cost as we show empirically in Section 3.1.3.2.

Screening methods can be seen as a form of preprocessing and the simulator evaluations used

in the screening activity can also be used to construct the emulator.

The benefits of screening are many fold:

1. Emulators are simpler; the reduced input space typically results in simpler models with fewer (hyper)parameters that are more efficient, both to estimate and use.

2. Experimental design is more efficient, in a sequential setting; the initial expense of applying screening is typically more than recouped since a lower dimensional input space can be filled with fewer design points.

3. Interpretability is improved; the input variables are not transformed in any way and thus the practitioner can immediately infer that the quantities represented in the discarded variables need not be estimated or measured in the future.

Screening can be employed as part of the emulator construction and in practice is often applied prior to any statistical analysis.

Single-output simulators are the focus of this chapter. The methods presented may be extended to the case of multiple outputs by treating each output independently and active inputs for each output identified separately. In Section 3.1 an overview of existing screening methods is given. Examples of the most commonly used methods are provided and their performance compared. In Section 3.2 a novel sequential screening approach based on the Morris screening method is presented. A summary is provided and possible future research directions are discussed in Section 3.3.

3.1 Overview of existing methods

Screening methods can be placed in two broad categories. Unsupervised methods operate solely on the inputs. An example of such a method is Principal Variables (McCabe, 1984) which is closely related to Principal Components where the factors are ranked according to a variance measure. Supervised methods, where input factors are ranked according to their effect on a response variable, are the main focus of this thesis.

Supervised screening methods have been broadly categorised in the following categories (Guyon and Elisseeff, 2003):

1. Screening Design methods. An experimental design is constructed with the express aim of identifying active factors. This approach is the classical statistical method, and is typically associated with the Morris method (Section 3.1.3). Other methods are available (Saltelli et al., 2000) but the Morris method has been found to be the most effective in practice (Saltelli et al., 2006).

2. Ranking methods. Input variables are ranked according to some measure of association between the simulator inputs and outputs. Typical measures considered are correlation, or partial correlation coefficients between simulator inputs and the simulator output. Other non-linear measures of association are possible, but these methods tend not to be widely used due to overly restrictive assumptions such as output monotonicity.

3. Wrapper methods. A model is used to assess the predictive power of subsets of variables. Wrapper methods can use a variety of search strategies:

   (a) Forward selection where variables are progressively incorporated in larger and larger subsets.

   (b) Backward elimination where variables are sequentially deleted from the set of active inputs, according to some scoring method, where the score is typically the root mean square prediction error of the simulator output (or some modification of this such as the Bayesian information criterion).

   (c) Efroymson's algorithm, also known as stepwise selection, proceeds as forward selection but after each variable is added, the algorithm checks if any of the selected variables can be deleted without significantly affecting the Residual Sum of Squares (RSS).

   (d) Exhaustive search where all possible subsets are considered.

   (e) Branch and Bound strategies eliminate subset choices as early as possible by assuming the performance criterion is monotonic, i.e. the score improves as more variables are added.

4. Embedded methods. For both variable ranking and wrapper methods, the model is considered a perfect black box. In embedded methods, the variable selection is integrated as part of the training of the model, although this might proceed in a sequential manner, to allow some benefits of the reduction in input variables to be considered. The ARD approach discussed in Section 3.1.1 is an example of this class of methods.

In this chapter we focus on methods most appropriate for computer experiments that are the most general, i.e. the assumptions made are not overly restrictive to a particular class of models. For a more general discussion of all screening methods see Boukouvalas and Cornford (2007).

If the simulator is available, the Morris method (see Section 3.1.3) can be effective where a one factor at a time (OAT) design is used to identify active inputs. The Morris method is a simple process, which can be understood as the construction of a design to estimate the expected value and variance (over the input space) of the partial derivatives of the simulator output with respect

to the simulator inputs. The method creates efficient designs to estimate these; to use the method it will be necessary to evaluate the simulator over the Morris design and the method cannot be reliably applied to data from other designs.

If the simulator is not easily evaluated (maybe simply because we don't have direct access to the code), or the training design has already been created, then design based approaches to screening are not possible and the alternative methods described above need to be considered. If the critical features of the simulator output can be captured by a small set of fixed basis functions (often simply linear or low order polynomials) then a regression (wrapper) analysis can be used to identify the active inputs. An example of a commonly used wrapper method is Least Angle Regression (Efron et al., 2002) which is a less greedy version of traditional forward selection and chooses a linear model from a large collection of possible covariates. We do not consider these methods general enough, however, as their performance is directly related to the specification of an appropriate list of fixed basis functions and thus are suitable for only relatively simple input-output mappings or where strong prior information is available on the mapping.

An alternative to the wrapper methods above is to employ an embedded method, such as Automatic Relevance Determination (ARD) which is described in Section 3.1.1. ARD essentially uses the estimates of the input variable length scale hyperparameters in the emulator covariance function to assess the relevance of each input to the overall emulator model. The method has the advantage that the relatively flexible Gaussian Process model is employed to estimate the impact of each input, as opposed to a finite parametric linear in parameters regression model, but the cost is increased computational complexity.

3.1.1 Automatic relevance determination

We describe here the method of Automatic Relevance Determination (ARD) where the correlation length scales δ_i in a covariance function can be used to determine the input relevance. This is also known as the application of independent priors over the length scales in the covariance models. The relevance of the input factors is determined by optimising the model marginal likelihood, described in Section 2.4. When the number of input factors is significantly high in relation to the number of training points, Qi et al. (2004) note that the ARD method can overfit due to the large number of parameters that need to be estimated. They suggest optimising the parameters by maximising the leave-one-out cross-validation score estimated using the expectation propagation algorithm (Minka, 2001). In the methodology presented in this section, we suggest that by proper validation of the emulator such overfitting can be detected. The purpose of the procedure is to perform screening on the simulator inputs, identifying the active inputs.

ARD is typically applied using a zero mean GP emulator. Provided the inputs have been stan-

dardised, the correlation length scales may be directly used as importance measures. Another case where ARD may be used is with a non-zero mean function GP where we wish to identify factor effects in the residual process. For example with a linear mean, correlation length scales indicate non-linear and interaction effects. If the effect of a factor is strictly linear with no interaction with other factors, it can still be screened out by subtracting from the simulator output prior to emulation.

To implement the ARD method, a range of covariance functions can be used (see Section 2.4.2). In fact any covariance function that has a length scale vector included can be used for ARD; for example the commonly used squared exponential covariance. Another example of an ARD covariance is the Rational Quadratic (RQ) (Rasmussen and Williams, 2006):

$$v(x_p, x_q) = \sigma^2[1 + (x_p - x_q)^\tau P^{-1}(x_p - x_q)/(2\alpha)]^{-\alpha},$$

where σ is the scale parameter and $P = \text{diag}(\delta_i)^2$ a diagonal matrix of correlation length scale parameters. Taking the limit $a \to \infty$, we obtain the squared exponential kernel.

Assuming p input variables, each hyperparameter δ_i is associated with a single input factor. The δ_i hyperparameters are referred to as characteristic length scales and can be interpreted as the distance required to move along a particular axis for the function values to become uncorrelated (Rasmussen and Williams, 2006). If the length-scale has a very large value the covariance becomes almost independent of that input, effectively removing that input from the model. Thus length scales can be viewed as a total effect measure and used to determine the relevance of a particular input.

Lastly, if the simulator produces random outputs the emulator should no longer exactly interpolate the observations. In this case, a nugget term σ_n^2 should be added to the covariance function to capture the response uncertainty.

Given a set of simulator runs, the ARD procedure can be implemented in the following order:

1. *Standardisation*. It is important to first standardise the input data so all input factors operate on the same scale. If rescaling is not done prior to the inference stage, length scale parameters will generally have larger values for input factors operating on larger scales. Standardisation methods are described in Appendix B.2.

2. *Inference*. The Maximum-A-Posteriori values of the length scale hyper-parameters are typically obtained by iterative non-linear optimisation using standard algorithms such as scaled conjugate gradients, although in a fully Bayesian treatment posterior distributions could be approximated using Monte Carlo methods. Maximum-A-Posteriori is the process of identifying the mode of the posterior distribution of the hyperparameter and is described in more

detail in Section 2.4.4. One difficulty using ARD stems from the use of an optimisation process since the optimisation is not guaranteed to converge to a global minimum and thus ensure robustness. The algorithm can be run multiple times from different starting points to assess robustness at the cost of increasing the computational resources required. In case of a very high-dimensional input space, maximum likelihood may be too costly or intractable due to the high number of free parameters (one length scale for each dimension). In this case Welch et al. (1992) propose a constrained version of maximum likelihood where initially all inputs are assumed to have the same length scale and iteratively, some inputs are assigned separate length scales based on the improvement in the likelihood score.

3. *Validation*. To ensure robustness of the screening results, prior to utilising the length scales as importance measures the emulator should be validated as described in Section 2.5.

3.1.1.1 ARD example on synthetic data

We demonstrate the implementation of the ARD method on a simple 1D synthetic example. The simulator function is $f(x_1,x_2) = \sin(x_1/10) + 0 \times x_2$, i.e. a two variable function which ignores the second input altogether. A 7 point design was used to train a emulator with a squared exponential function:

x_1	0.10	0.23	0.36	0.50	0.63	0.76	0.90
x_2	0.24	0.37	0.91	0.64	0.11	0.51	0.77
$f(x_1,x_2)$	0.84	0.72	-0.50	-0.95	0.05	0.98	0.41

Note that both input factors are operating on the same scale so no standardisation is needed in this case. The inference is done by using a scaled conjugate gradient algorithm to maximise the log likelihood of the Gaussian Process emulator (that is in this case no priors are placed over the length scales). To check the fit of the emulator a grid test set of 1000 points is used. We can clearly see from Figure 3.1 that both the simulator and emulator responses are insensitive to the value of x_2.

To further validate the emulator and examine the output predictive variance, we plot in Figure 3.2 a profile of the simulator function at $x_2 = 1$ and x_1 a grid design of 1000 points. The emulator fits the simulator function well and the uncertainty captures the prediction error away from the training points.

The length scales obtained through maximum likelihood are $\delta_1 = 0.16$ and $\delta_2 = 48.6$ which can be interpreted as the emulator using the first variable and ignoring the second. The ARD method therefore correctly identifies the second variable as redundant.

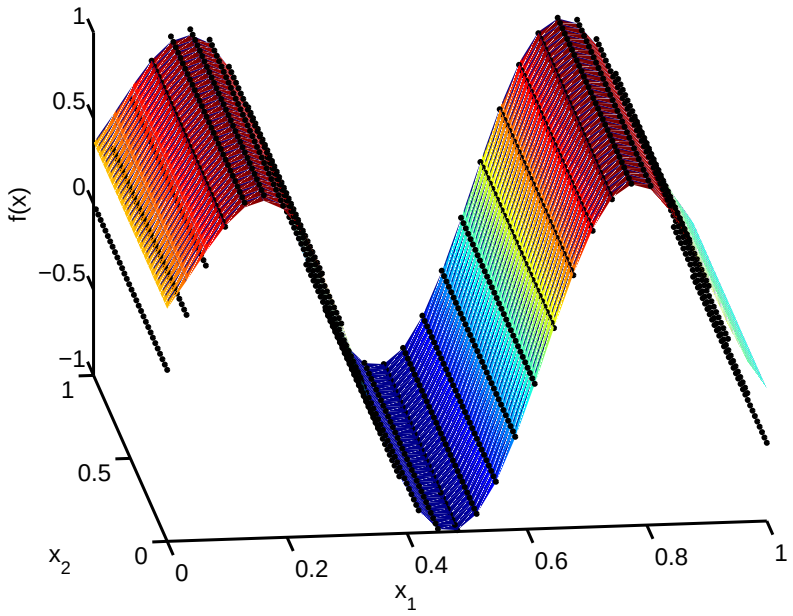


Figure 3.1: Validation of ARD Emulator. The simulator values are plotted in black dots and the emulation mean prediction is the smooth coloured surface.

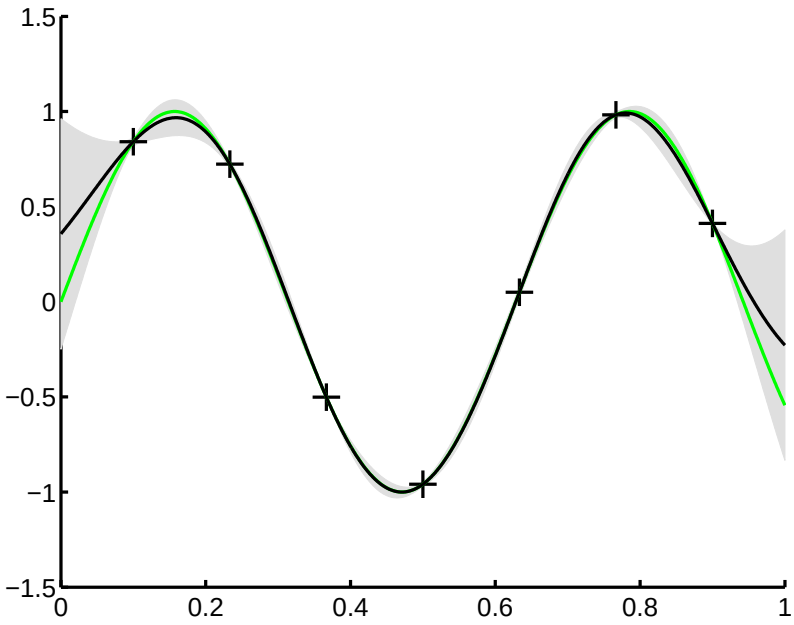


Figure 3.2: Profile of emulator and simulator along the x_1 factor. The simulator function is shown in green against the emulator prediction in black with the predictive variance in grey. The training data are shown as crosses (although the x_2 coordinate varies in the training data but this clearly has no effect on the output value from the simulator).

3.1.2 Variance-Based Methods

The method of Sobol' (Sobol, 1993) is a variance decomposition method where Monte Carlo integration yields sensitivity indices. The presentation in this section is based on Saltelli et al. (2000) to which the reader is referred for further details. Variance-based methods estimate the variance of the conditional expectation (VCE) of each input factor X_i in relation to the function output Y. The importance of factors is then calculated with the correlation ratio:

$$\eta_i^2 = \frac{\mathrm{Var}_{X_i}\left[E(Y|X_i)\right]}{\mathrm{Var}[Y]},$$

where $\mathrm{Var}_{X_i}\left[E(Y|X_i)\right] = \int \left[E(Y|X_i) - E(Y)\right]^2 p(X_i)\mathrm{d}X_i$ is the VCE and $\mathrm{Var}[Y]$ the variance of Y. The Sobol' method offers an effective approach to estimating the VCE.

Let $\mathbf{x}$ be the k-dimensional input vector $(x_1,\ldots,x_k)$ and $\Omega^k = (\mathbf{x}|0 \le x_i \le 1; i = 1,\ldots,k)$ the design region. The Sobol' method relies on a decomposition of the simulator function $f(\mathbf{x})$ into summands of increasing dimensionality:

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^{k} f_i(x_i) + \sum_{1\le i<j\le k} f_{ij}(x_i,x_j) + \cdots + f_{1,2,\ldots,k}(x_1,\ldots,x_k), \tag{3.1}$$

where $f_0 = \int_{\Omega^k} f(\mathbf{x})\mathrm{d}\mathbf{x}$.

For Equation (3.1) to hold the integrals of every summand over any of its own variables must be zero:

$$\int_0^1 f_{i_1,\ldots,i_s}(x_{i_1},\ldots,x_{i_s})\mathrm{d}x_{i_k} = 0 \qquad \text{if } 1 \le k \le s.$$

Given this condition, Sobol' proved that all summands in Equation (3.1) are orthogonal and all terms can be evaluated via multidimensional integrals:

$$f_i(x_i) = -f_0 + \int_0^1 \ldots \int_0^1 f(\mathbf{x})\mathrm{d}\mathbf{x}_{\sim i},$$
$$f_{ij}(x_i,x_j) = -f_0 - f_i(x_i) - f_j(x_j) + \int_0^1 \ldots \int_0^1 f(\mathbf{x})\mathrm{d}\mathbf{x}_{\sim (ij)}.$$

where $\mathrm{d}\mathbf{x}_{\sim(ij)}$ denotes the integration over all variables except x_i and x_j. The variance-based sensitivity indices can now be derived. The total variance D is defined as:

$$D = \int_{\Omega^k} f^2(\mathbf{x})\mathrm{d}\mathbf{x} - f_0^2$$

and the partial variances are computed for each term in Equation (3.1):

$$D_{i_1,\ldots,i_s} = \int_0^1 \ldots \int_0^1 f^2_{i_1,\ldots,i_s}(x_{i_1},\ldots,x_{i_s})\mathrm{d}x_{i_1}\ldots\mathrm{d}x_{i_s}$$

where $1 \leq i_1 < \cdots < i_s \leq k$ and $s = 1, \ldots, k$. By squaring and integrating Equation (3.1) over Ω^k and by the orthogonality constraint we obtain:

$$D = \sum_{i=1}^{k} D_i + \sum_{1 \leq i < j \leq k} D_{ij} + \cdots + D_{1,2,\ldots,k}.$$

The sensitivity indices used to measure the factor effects are given by:

$$S_{i_1,\ldots,i_s} = \frac{D_{i_1,\ldots,i_s}}{D} \qquad \text{for} 1 \leq i_1 < \cdots < i_s \leq k.$$

The indices are identified by an *order* which is the number of input factors for that effect, e.g. an index of order one measures the effect of a single input factor on the response, while an index of order 2 measures interaction effect between two input factors. If k is the number of input factors, there are $\binom{k}{1}, \ldots, \binom{k}{k}$ Sobol' indices of order $1, 2, \ldots, k$. Specifically S_i is called the first order sensitivity index for factor x_i and measures the main effect of x_i on the output, i.e. the fractional contribution of x_i to the variance of $f(\mathbf{x})$. S_{ij}, where $i \neq j$, is known as the second order sensitivity index and measures the interaction effect, that is the part of the variation of $f(\mathbf{x})$ due to factors x_i and x_j that cannot be explained by the sum of the individual effects of x_i and x_j. We also note that a consequence of these definitions is the sum of the sensitivity indices is 1 which helps interpreting the magnitude of each sensitivity index.

The total effect index is defined as the sum of all Sobol' indices for a specified input factor. In particular, by partitioning $\mathbf{x}$ into $\mathbf{x}_{\sim i}$ and x_i one can compute with a single Monte Carlo integral the total effect for factor x_i:

$$TS_i = 1 - S_{\sim i},$$

where $S_{\sim i}$ is the sum of all $S_{i_1,\ldots,i_s}$ terms that do not include the index i. The computation of the total effect index TS_i does not fully characterise the effect of the factor x_i on the system but is much more reliable that the first order index S_i while avoiding the computation of all $2^k - 1$ sensitivity indices that involve the factor x_i. In practice both the first order and total order indices are computed for each factor as part of the sensitivity analysis of a simulator.

The Sobol' formulation of sensitivity indices is very general and includes as special cases most other sensitivity analysis methods (Archer et al., 1997).

3.1.3 Morris Method

The Morris method (Morris, 1991) is a popular and simple methodology for the sensitivity analysis of computer simulators. The method, which is also known as the Elementary Effect (EE) method, is predicated upon global approximations to the simulator partial derivatives.

The method works as follows. Let k be the number of input variables for the simulator. The design region for these factors is assumed to be linearly normalised to $[0,1]^k$. The simulator $Y(\cdot)$ is assumed to be a smooth real-valued function with domain containing the design region. The elementary effect for the i-th input variable at $\mathbf{x} \in [0,1]^k$ is the classic approximation to the derivative of $Y(\cdot)$ with respect to x_i evaluated at point $\mathbf{x}$:

$$EE_i(\mathbf{x}) = \frac{Y(\mathbf{x}+\Delta\mathbf{e}_i) - Y(\mathbf{x})}{\Delta}. \tag{3.2}$$

The divisor Δ is a fixed step size, and $\mathbf{e}_i$ is the unit vector in the direction of the i-th axis for $i = 1,\ldots,k$. Each elementary effect is computed with observations at the pair of points $\mathbf{x}$, $\mathbf{x}+\Delta\mathbf{e}_i$ that differ in the i-th input variable by the fixed step size Δ.

The classic approach for computing elementary effects is to start from a point $\mathbf{x}$, from which a trajectory is constructed with k random moves of size Δ, each movement in the direction of a coordinate axe, to end in the point $\mathbf{x}+\Delta(\mathbf{e}_1+\cdots+\mathbf{e}_k)$. In this form, $k+1$ evaluations of simulator $Y(\cdot)$ are performed, ending with elementary effects $EE_1(\mathbf{x}),\ldots,EE_k(\mathbf{x})$, see Morris (1991).

Now consider a set of R points $\mathbf{x}_1,\ldots,\mathbf{x}_R$ in the input space. At each point $\mathbf{x}_r$, $r = 1,\ldots,R$, we perform k one-at-a-time (OAT) runs and compute elementary effects $EE_i(\mathbf{x}_r)$ for every input factor. The following sample moments are computed for each input factor:

$$\mu_i = \frac{1}{R}\sum_{r=1}^{R} EE_i(\mathbf{x}_r), \mu_i^* = \frac{1}{R}\sum_{r=1}^{R} |EE_i(\mathbf{x}_r)| \text{ and } \sigma_i = \sqrt{\sum_{r=1}^{R} \frac{(EE_i(\mathbf{x}_r)-\mu_i)^2}{R-1}}. \tag{3.3}$$

The sample moment μ_i is an average-effect measure, and a high value suggests a dominant contribution of the i-th input factor in positive or negative response values. The sample moment μ_i^* is a main-effect measure; a high value indicates large influence of the corresponding input factor. The moment μ_i^* was proposed in Campolongo et al. (2004) since μ_i may prove misleading due to cancellation of effects. Non-linear and interaction effects are estimated with σ_i. The total number of model runs needed in Morris's method is $(k+1)\times R$.

An effects plot can be constructed by plotting μ_i or μ_i^* against σ_i. This plot is a visual tool to detect and rank effects. Factor effects close to the origin are the least influential. region

There is interest in doing input screening with as few runs as possible but as the number of input factors k is fixed, the size of the experiment is controlled by R. Usually small values of R are used; for instance, Morris (1991) used $R = 3$ and $R = 4$ in his examples. A value of R between 10 and 50 is mentioned in the more recent literature, see (Campolongo et al., 2004, 2007). A larger value of R will improve the quality of the estimations, but at the price of extra runs.

The step size Δ is selected in such a way that all the simulator runs lie in the input space and

the elementary effects are computed within reasonable precision. The usual choice of Δ in the literature is determined by the input space considered for experimentation, which is a k dimensional grid constructed with p uniformly spaced values for each input. The number p is recommended to be even and Δ to be a multiple of $1/(p-1)$, for example $\Delta = p/(2(p-1))$, see (Morris, 1991; Campolongo et al., 2004). The step Δ is usually kept at the same value for all the inputs, but the method can be generalised to instead use different values of Δ and p for every input.

In Morris's original proposal, the points $\mathbf{x}_1, \ldots, \mathbf{x}_R$ were taken at random from the input space grid. Campolongo et al. (2007) proposed spreading runs over the design space by generating a large number of trajectory designs and selecting a subset by maximising the minimum distance between them.

In the case of deterministic systems, a potential drawback of the OAT designs used in the EE method, is that design points fall on top of each other when projected into lower dimensions. This disadvantage becomes more apparent when the design runs are to be used in further modelling after discarding unimportant factors. An alternative is to construct a randomly rotated simplex at every point $\mathbf{x}_r$, from which elementary effects are computed (Pujol, 2009). The computation of distribution moments μ_i, μ_i^*, σ_i and further analysis is similar to the EE method, with the advantage that projections of the resulting design do not fall on top of existing points, and all observations can be reused in a later stage. A potential disadvantage of this approach is the loss of efficiency in the computation of elementary effects, i.e. computing effects from a rotated simplex is suboptimal when compared with the Equation (3.2) which is optimal for computing elementary effects.

3.1.3.1 Morris Example

An example of a two factor Morris design with the discretisation level set to $p = 10$, $\Delta = p/(2(p-1)) = 0.55$ and $R = 5$ trajectories is used to identify the active factors of the following function:

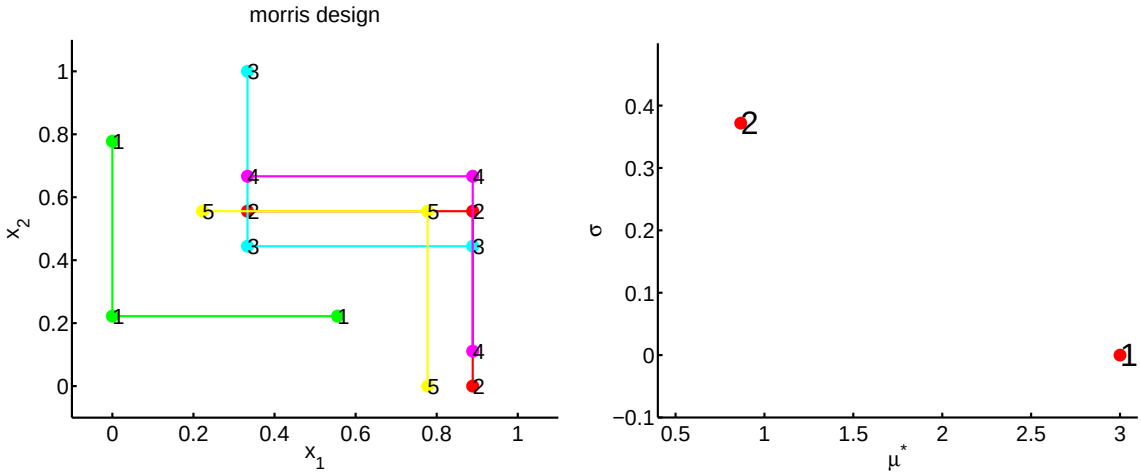
$$f(x) = 3x_1 + x_2^2. \tag{3.4}$$

As evidenced by Figure 3.3(a) the ensemble of the trajectory designs cover the input space reasonably well. The distributional moments of the Elementary Effects are plotted in Figure 3.3(b) and are tabulated below:

Factor	μ_*	μ	σ
x_1	3	3	0
x_2	0.86	0.86	0.37

The linear effect of factor x_1 is evident as the EE deviation σ is zero while for x_2 σ is 0.37 pointing to the non-linear effect of the factor. For factor x_1 we note the high μ and low σ values

signify a linear effect. For factor x_2 the large σ value demonstrates the non-linear/interaction effect. The agreement of μ to μ_* for all factors shows a lack of cancellation effects, due to the monotonic nature of the input-output response in this simple example. In general this will not be the case, particularly for models with non-linear responses.



(a) Design with five trajectories marked 1-5.

(b) Moments of Elementary Effects

Figure 3.3: Morris design and first two moments (μ^*, σ) of Elementary Effects for the synthetic simulator function given in Equation (3.4).

3.1.3.2 Comparison To Sobol'

We demonstrate the efficiency of the Morris method compared to more traditional variance-based sensitivity analysis methods described in Section 3.1.2. Campolongo et al. (2004) compared the efficacy of the Morris method to the variance-based methods for relatively low-dimensional problems.

A simple yet highly multidimensional function is used to demonstrate the efficacy of the Morris screening technique compared to the Sobol' method. Details of the function are given in the Appendix B.4. The function has 99 inputs, and each of these inputs has one of five effects on the function's response: linear, periodic, polynomial of order 2 or greater, near-linear and step-linear (Figure 3.4). In particular, gradient-based methods such as Morris's will fail to identify step-linear effects without sufficient coverage of the input space.

We note that since the function used to generate the synthetic data set is not monotonic, methods such as the partial rank correlation coefficient (Saltelli et al., 2000) which make such an assumption are not appropriate and performed poorly as expected. We therefore do not include these results here.

In Figure 3.5, we show the Morris variable ranking computed from 10^3 and 10^4 simulator evaluations. The results are qualitatively stable suggesting the lower sample-size effect estimates

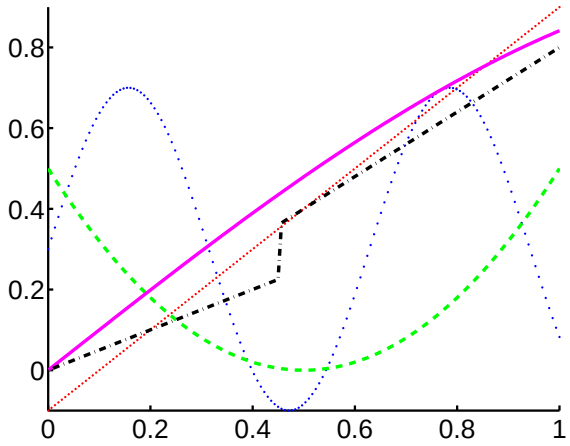


Figure 3.4: Five types of functions represented in the multidimensional example: linear (dotted), periodic (dotted), polynomial (dashed), near-linear (solid) and step-linear (dot-dashed).

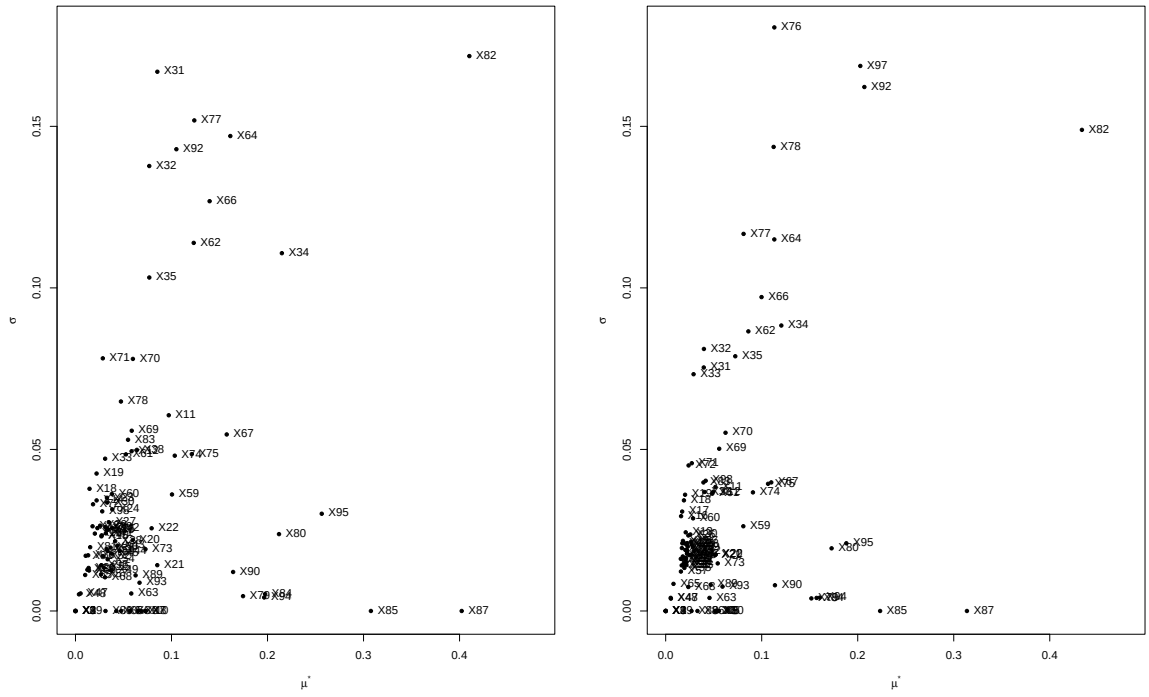
are informative.

As Campolongo et al. (2007) point out, the sample sizes required for the Sobol' method to obtain reliable estimates of the main and total effect indices are higher than those used in the Morris method. Our experiments confirm this conclusion. Using 10^3 model evaluations, the uncertainty of the Sobol' indices is very high and no conclusions can be drawn with respect to factor relevance (Figure 3.6(a)). The Sobol' method provides satisfactory results when the sample size is increased to 10^5(Figures 3.6(b) and 3.6(c)). As we would expect, the Morris method provides reliable qualitative results that are useful in the early stages of model analysis. In subsequent analysis, many more simulator runs can be obtained in the reduced input space enabling the usage of the more accurate variance-based methods. Our synthetic experiments empirically demonstrate the efficiency of the Morris method in a high-dimensional setting.

3.2 Sequential Morris

Computer simulators are often expensive to run, sometimes taking between several minutes to hours in order to compute a single run. In such a case, screening across a large number of inputs with Morris's method requires a relatively large number of computer simulations, which may turn into a very expensive computation.

We propose a sequential screening method. Such a method allows the experimenter to perform a initial number of runs, and, depending on the results obtained, continue with extra runs if required. The methodology aims to separate between factors with linear effect and with non-linear effect. The rationale behind it is that if σ_i is small for a given factor, then we should investigate whether σ_i remains small over other areas of the design region. At the end of experimentation, those input factors for which σ_i remained small are considered to have linear effect, and factors for



(a) 10^3*Observations* (b) 10^4*Observations*

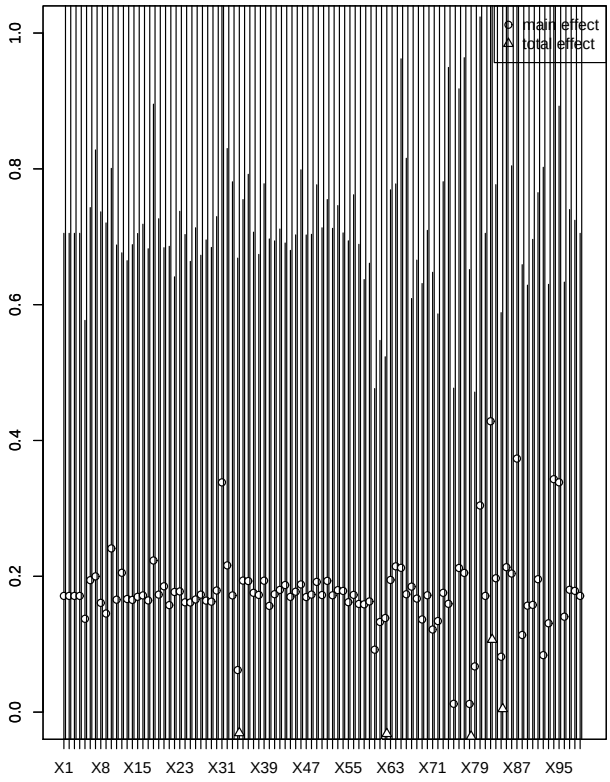
Figure 3.5: Morris method applied to the 99-dimensional synthetic data set. The x-axis indicates main effects (μ_*) and the y-axis non-linear and interaction effects (σ). Results are shown for 10^3 and 10^4 simulator evaluations.

which σ_i was bigger than a threshold have a non-linear effect on the output. A method of eliciting the choice of threshold is presented in Section 3.2.1.

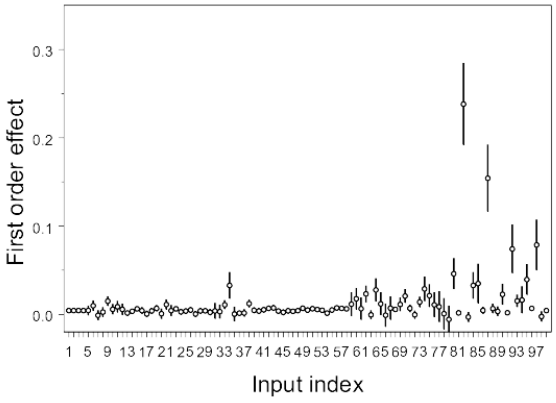
The justification of thresholding solely on the variance of the elementary effects σ_i is that independent linear effects of factors may be removed from the simulator output at a preprocessing stage or during the emulation phase. The emulator GP function may be parametrised such that factors with linear effects are incorporated in the mean function while omitted from the covariance specification. If we denote by A the subset of $\{1,\ldots,k\}$ which indexes factors with linear effects, the GP prior may be written as $Y(x) = \beta + \sum_{i\in A} a_i X_i + Z^*$ where Z^* is a stochastic process whose covariance structure depends only on the variables with non linear effects, i.e. those x_i with $i \in \{1,\ldots,k\} \setminus A$. The residual process Z^* is therefore placed in a reduced dimensionality space simplifying the design and inference tasks.

For our algorithm to run, a space filling design with M points is created. This design provides the sequence of points at which the Morris OAT runs will be tested. Initially we select a good space filling design, such as a Maximin Latin Hypercube (LH) (Morris and Mitchell, 1995). The value of M is selected such that $(k+1)M$ is the maximum number of runs that can be performed during the whole screening process.

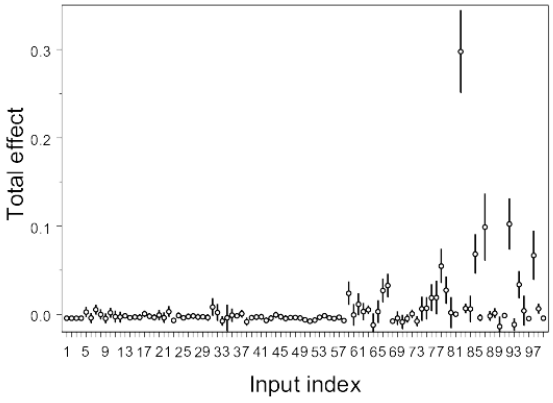
A preprocessing stage orders the design points according to the biggest distance between



(a) 10^3 Observations



(b) 10^5 Observations Main Effects



(c) 10^5 Observations Total Effects

Figure 3.6: Sobol' method applied to the 99-dimensional synthetic data set. The x-axis shows the input index and the y-axis the mean and 95% confidence intervals. The confidence intervals were obtained using 8000 bootstrap samples (Archer et al., 1997).

points. The first two points are those whose Euclidean distance is largest; then the third point maximises the minimum distance between itself and the first two points, then a fourth point is ordered in the same way, and so on. This procedure of ordering points mirrors nearest-neighbour clustering, but acts in an opposite manner as points are ordered from those farthest apart to end up with closest points.

Example 1. For $k = 3$ input factors and $M = 6$ runs, consider a Maximin LH design in $[0,1]^3$ with point coordinates $x_1 = (4/5, 1, 3/5)$, $x_2 = (1/5, 0, 2/5)$, $x_3 = (2/5, 3/5, 0)$, $x_4 = (3/5, 2/5, 1)$, $x_5 = (0, 4/5, 4/5)$ and $x_6 = (1, 1/5, 1/5)$. The preprocessing stage first selects the points x_5 and x_6, which are furthest apart. The next point, x_1, maximises the distance between those remaining points and the first two points chosen. The procedure continues by selecting x_2, then x_3 and finishes with x_4. In summary, the preprocessing stage produces the ordered sequence of points $x_5, x_6, x_1, x_2, x_3, x_4$, which are relabelled as $x_{(1)}, \ldots, x_{(6)}$.

The screening algorithm starts with the computation of elementary effects for all input factors at the first two points. OAT runs are created at those two points and elementary effects are computed. With this initial data, a poor estimation of the moments μ_i, μ_i^* and σ_i is available. If for a given input factor, its sample moment σ_i is larger than a specified threshold σ_0 then we say that this output is responding non-linearly to the corresponding input. We declare that input as active and remove it from the list of current input factors. The technique continues by adding OAT runs at the next point, but only for those factors not active. Elementary effects are computed and moments are updated for each added point. Factors are then removed if the condition for σ_i is met. The methodology ends when all input factors have been removed, or after computing elementary effects for all M points. On ending, the input factors are separated into two groups: those having non-linear effect and those with linear or no effect on the output. Algorithm 3.1 sets out the procedure in pseudo-code form, and a proposal for the thresholding value σ_0 is presented in Section 3.2.1.

Example 2. To show how the proposed sequential algorithm works, consider $Y(x_1, x_2, x_3) = \cos(x_3/5)(x_2 + 1/2)^4/(x_1 + 1/2)^2$ on the design region $[0,1]^3$. The function Y is treated as a simulator, from which the only information we require are its values at design points. We use the same pre-ordered LH design of Example 1; set $p = 10$ for step size $\Delta = 5/9$ and threshold $\sigma_0 = 0.15$. See Section 3.2.1 for details on the construction of the threshold σ_0.

Random trajectories are constructed with the first two ordered points, giving the following moments of elementary effects $(\mu_1, \mu_2, \mu_3) = (-6.37, -1.16, 0.02)$ and $(\sigma_1, \sigma_2, \sigma_3) = (13.25, 4.10, 0.02)$. The values of σ_1, σ_2 are greater than the threshold σ_0 and thus x_1 and x_2 are separated as having non-linear effects. As $\sigma_3 < \sigma_0$, further investigation is required for x_3. At the third design

Algorithm 3.1 The procedure for completing our screening technique.

Screening algorithm

Input: Simulator $Y(\cdot)$ with k inputs; total number of one-at-a-time experiments M; step size Δ; threshold σ_0.
Output: Moments μ_i, σ_i, μ_i^*; lists of factors with linear (C) and with non-linear effect (A).

A. *Preprocessing stage*

1. Set design region to $[0,1]^k$ and create space filling design with M points $\mathbf{x}_1, \ldots, \mathbf{x}_M$.
2. Order the design points using maximum distance between points. Label the ordered points as $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(M)}$.

B. *Calculating the elementary effects*

1. Set $R := 2$ and the initial design to be $D := \{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}\}$. Set list of current factors to $C := \{1, \ldots, k\}$ and list of active effects $A := \emptyset$.
2. For every point in D, create one-at-a-time runs only for those input factors indexed by C. Run the simulator at those points. This totals $|C| + 1$ experiments for every point in D.
3. Using simulator runs from B2 and Equation (3.3), compute elementary effects $\{EE_i(\mathbf{x}) : \mathbf{x} \in D, i \in C\}$.
4. If $R = 2$, compute moments μ_i, μ_i^* and σ_i using elementary effects for all factors. If $R > 2$, only update moments for the current list of input factors, indexed by C.
5. For $i \in C$, if $\sigma_i > \sigma_0$ then update $C := C \setminus \{i\}$ and $A := A \cup \{i\}$.
6. If $C = \emptyset$, then all the inputs were identified active. Algorithm ends.
7. If $R = M$, then all the design points available are exhausted. Algorithm ends.

C. *Producing the next design point*

1. Update R:=R+1; set $D = \{\mathbf{x}_{(R)}\}$.
2. Goto B2.

point, OAT experimentation only for the factor x_3, produces updated moments $\mu_3 = 0.05$ and $\sigma_3 = 0.05 < \sigma_0$, that is, x_3 is still under investigation. The sequential methodology continues for x_3 until finishing with all the design points. At this final step, updated moments for x_3 are $\mu_3 = 0.03$ and $\sigma_3 = 0.05$, i.e. the linearity of the response in terms of x_3 over the design region could not be rejected. In fact, in the design region, the factor x_3 has a near-linear effect, as defined in Section 3.2.1.

The total experimental effort was 16 runs, from which the first 8 runs involved trajectories for all factors, while further 8 runs were required for the linear factor under investigation. This is a 33% reduction from the $(3+1)*6 = 24$ runs needed to perform the complete EE method.

The moment estimates obtained for the non-linear factors are only rough approximations of the true moment values, but the moment estimates for the linear factor were computed with more information. This asymmetry is apparent when comparing with exact analytic sensitivity results $\mu = (-5.34, 6.62, -0.04)$ and $\sigma = (8.88, 7.42, 0.06)$.

An alternative to using (and preprocessing) a design with a fixed number of runs M is to instead consider points from an infinite sequence, from which points can be taken sequentially as required. For example, points can be generated from a *low discrepancy* space filling sequence, such as Sobol's or Niederreiter's sequences (Niederreiter, 1992). The only change required in the pseudo-code of Algorithm 3.1 is to remove step A2. Sampling from low discrepancy sequences has the advantage of sequential generation of points. However, for small sample sizes the spread of points of a low discrepancy sequence may not be as good as a space filling design with fixed size.

3.2.1 Selection of variance threshold

In the sequential pseudo-code given in Algorithm 3.1, the elementary effect variance threshold σ_0 is an input. However it may be quite hard in certain cases to elicit. In this section an approach to estimate σ_0 indirectly by eliciting the expected divergence from linear of the factor effect is presented. An application of the sequential procedure is presented on a synthetic test function in Section 3.2.1.1 and on a real world simulator in Section 6.2. In Section 3.2.1.2 a simulation study is used to empirically demonstrate the effectiveness of the threshold calculation even in cases where the factor effect deviates slightly from linear.

A linear (or near-linear) effect of the variable x_i is represented by an additive noise model:

$$Y(x_i) = ax_i + b + \varepsilon, \tag{3.5}$$

where ε is a normal random variable with zero mean and variance γ and observations of ε are

assumed to be independent. In other words, the marginal effect due to the factor x_i is modelled with a simple regression line. We will assume that the variance γ is known. In practice, this variance γ will be elicited prior to the screening experiment and it can take several meanings:

1. We believe the factor x_i has a linear effect but the simulator runs contain a numerical error. In this case we expect γ to be set to small value, such as a multiple of machine precision.

2. We believe that small non-linear effects will not have an appreciable impact on the model output. Here γ should be chosen to reflect the level of variation from a straight line that we will tolerate.

The capturing of information about the variance parameter is in sharp contrast to (Kadane et al., 1980) and (Garthwaite and Dickey, 1988) where full probability distributions are elicited that reflect beliefs about the parameters of the linear model. In the present application, we do not wish to prejudge the behaviour of the model: we want a point estimate of how far from being linear we can tolerate.

Given the variance γ, the sampling distribution of the variance of the elementary effects can be calculated according to the following lemma, whose proof is given in Appendix B.3.

Lemma 3.2.1. *Let $x_1, \ldots, x_R$ be univariate design points, at each of which trajectories are constructed. Assume that observations taken at design points and trajectories follow the model given in Equation (3.5). Let elementary effects and moments be defined as in Equations (3.2) and (3.3) and let $\sigma_\Phi^2 = \frac{2\gamma}{\Delta^2}$. Then*

$$\sigma^2 \sim \frac{\sigma_\Phi^2}{R-1}\chi^2_{R-1}. \tag{3.6}$$

where χ^2_{R-1} denotes a chi-square distribution with $R-1$ degrees of freedom.

Since the sampling distribution of the EE variance is now known we propose to use the 99% quantile of the cumulative distribution function of the chi-square distribution to derive the EE variance threshold σ_0. The following equation

$$P\left(\sigma^2 \leq \sigma_0\right) = P\left(\frac{\sigma_\Phi^2}{R-1}\chi^2_{R-1} \leq \sigma_0\right) = 0.99,$$

which inverted yields the threshold

$$\sigma_0 = \chi^2_{0.99,R-1}\sigma_\Phi^2/(R-1), \tag{3.7}$$

where $\chi^2_{0.99,R-1}$ is the 99% quantile of a chi-squared distribution with $R-1$ degrees of freedom.

In other words, σ_0 defines a threshold over which the effect is considered non-linear, i.e. if

$\sigma^2 > \sigma_0$ then the input variable will be retained. Note that Lemma 3.2.1 applies directly in a multivariate setting, in which case the comparison is performed separately for each input variable.

In Example 2 we used a single threshold σ_0 for all variables. The values $R = 6$, $\Delta = 5/9$, $\sqrt{\gamma} = 8.7 \times 10^{-2}$ and quantile $\chi^2_{0.99,5} = 15.08$ were used to obtain $\sigma_0 = 0.15$.

The method we propose might be thought of as a sequential hypothesis test, where the null hypothesis is that data follows linear model described in Equation (3.5). To simplify the algorithm the threshold σ_0 may be kept fixed for all computations rather than adapting σ_0 to the actual number of trajectories involved. The main difference is in the degrees of freedom for the scaled chi-square distribution in Equation (3.7). The adaptive approach, which is utilised in the simulation experiments presented, involves recomputing σ_0 with updated degrees of freedom prior to step B5 in Algorithm 3.1. If the simplified approach, i.e. using only a single value σ_0 is used, the method becomes more conservative, i.e. the rejection rate of a simple, linear model is higher with fixed threshold than with a variable one.

3.2.1.1 Simulated high-dimensional example

In this section we illustrate the sequential screening method on the synthetic test function introduced in Morris (1991). The function is defined on 20 inputs $\mathbf{x} \in [0,1]^{20}$ as follows:

$$y = \beta_0 + \sum_{i=1}^{20} \beta_i w_i + \sum_{i<j}^{20} \beta_{ij} w_i w_j + \sum_{i<j<l}^{20} \beta_{ijl} w_i w_j w_l + \sum_{i<j<l<s}^{20} \beta_{ijls} w_i w_j w_l w_s, \tag{3.8}$$

where $w_i = 2(x_i - \frac{1}{2})$ except for $i = 3,5,7$ where $w_i = 2\left(1.1x_i/(x_i + 0.1) - \frac{1}{2}\right)$. The coefficients are set to $\beta_i = 20$ for $i = 1,\ldots,10$, $\beta_{ij} = -15$ for $i,j = 1,\ldots,6$, $\beta_{ijl} = -10$ for $i,j,l = 1,\ldots,5$ and $\beta_{ijls} = 5$ for $i,j,l,s = 1,\ldots,4$. The remaining first and second order coefficients are generated independently from a zero mean unit variance normal distribution and the remainder third and fourth order coefficients are set to zero.

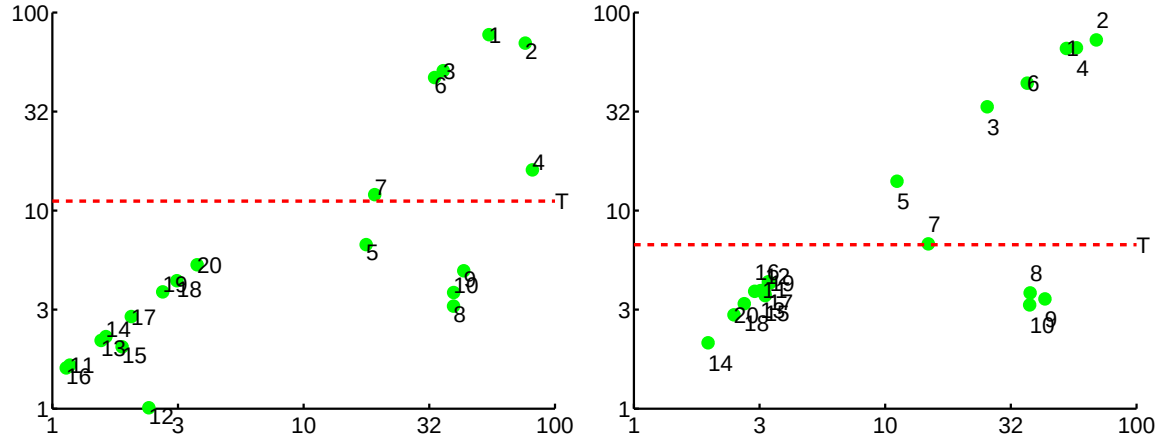
Given the range of the function defined in Equation (3.8) is approximately $y \in [-225,139]$, the threshold value is set to $\gamma = 2.6$ corresponding to an approximate standard deviation from linear of 0.005%.

As both Morris (1991) and Pujol (2009) show, factors $x_1,\ldots,x_7$ have a non-linear effect on the function output while factors x_8,x_9,x_{10} have a linear effect and factors $x_{11},\ldots,x_{20}$ have negligible effect.

The screening experiment was performed under the configuration used in (Pujol, 2009) for 100 realisations. As in Pujol (2009) the discretisation level has been set to $p = 20$ and the number of trajectories to $R = 10$. A total of 210 function evaluations are required for the batch EE procedure while for the sequential EE procedure on average 150 are required with a standard deviation of 13.

Factors $x_1,\ldots,x_7$ are correctly identified as having non-linear effect 99 out of the 100 realisations. Factors $x_8,\ldots,x_{20}$ are found to have linear effects in 92% of the realisations. The full batch EE screening results and the first step of one realisation of the sequential algorithm is shown in Figure 3.7.

We conclude that the sequential approach results in significant computational savings compared to the batch EE method as factors with clear non-linear effects can be eliminated in the early screening stages with high confidence.



(a) After the first step (b) Average batch results.

Figure 3.7: Applying the batch and sequential EE screening method on the 20 input factor Morris test function. X axis is μ_* and Y axis σ of Elementary Effects. Horizontal dashed red line denotes the σ_0 threshold value for the given step.

3.2.1.2 Simulation Results

We test the previous distributional results on two functions:

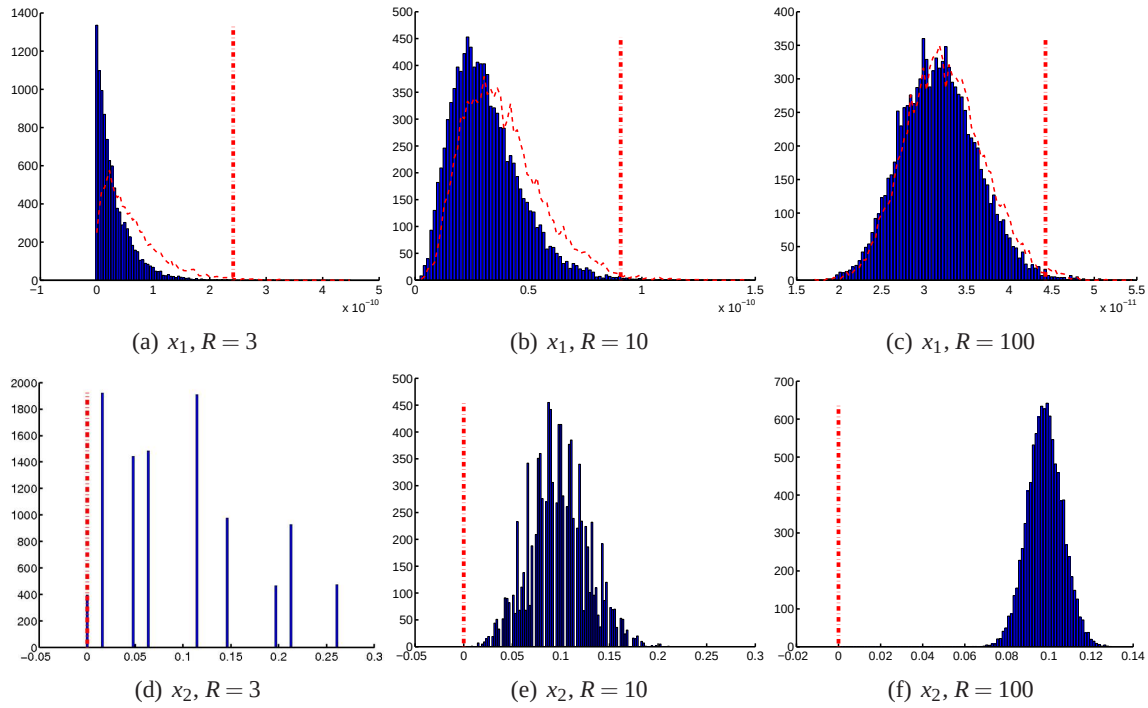
$$f(x) = 3x_1 + x_2^2 + N(0,\gamma) \tag{3.9}$$

$$g(x) = 3x_1 + \sin(x_2) + x_3^2 \tag{3.10}$$

For $f(x)$ we examine the threshold under a vary small prior variance γ simulating the numerical error scenario. The prior variance is set to $\sqrt{\gamma} = 10^{10}\varepsilon \approx 2^{-6}$. The Morris design was constructed with $p = 10$ and $\Delta = p/(2(p-1))$.

We perform 10^4 realisations of a simulation experiment with $R = 3$, $R = 10$ and $R = 100$ trajectories and plot the sampling distributions for both the linear and non-linear factor in Figure 3.8. For $R = 3$ the quadratic effect factor x_2 is incorrectly identified as near-linear 446 out of the 10^4 Morris experiments. In these cases, we would incorrectly classify the effect of x_2 as near-linear. For the higher number of trajectories, no such errors occur. As the assumption of independent and

Gaussian noise is satisfied for this scenario the theoretical distribution of the EE variance matches very well the empirical distribution.



(a) x_1, $R = 3$ (b) x_1, $R = 10$ (c) x_1, $R = 100$

(d) x_2, $R = 3$ (e) x_2, $R = 10$ (f) x_2, $R = 100$

Figure 3.8: Sampling distributions for EE variance for each factor using 10^4 realisations of the experiment. The $f(x)$ function in Equation (3.9) is used. x_1 has a noisy linear effect while x_2 has a noisy quadratic effect. Dashed line is theoretical sampling distribution. Vertical dashed line is the 99% threshold.

For function $g(x)$ we use a truncated Taylor series to estimate the variance γ which stems from the linear approximation. We wish to consider $\sin(x)$ as having a near-linear effect and aim to derive an appropriate value for γ. The first two terms of the Maclaurin series are $x - \frac{x^3}{3!}$. Hence we can approximate $\sin(x)$ with the linear function $y = x + N(0,\gamma)$. We get an estimate of the variance γ by examining the approximation error $\varepsilon = \sup_{x\in[0,1]} |x|^3/3!$. We treat the approximation error bound as three standard deviations, i.e. $3 \times \sqrt{\gamma} = \varepsilon$. Hence $\gamma = (\varepsilon/3)^2$. The approximation is shown in Figure 3.9.

The distribution of the EE variance is given in Figure 3.10 for $R = 10$ and $R = 100$. The variance has been set to $\gamma = 0.0031$ following the Taylor approximation. We can see clearly a mismatch of the theoretical distribution to the empirical due to the non-Gaussianity and heteroscedasticity of the noise. This does not improve as R is increased although the separation of the non-linear term x_3 becomes clearer. Therefore, despite the approximation error the non-linear term is correctly identified.

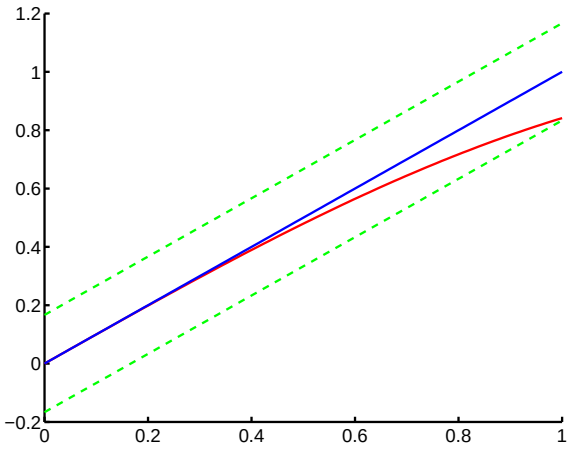
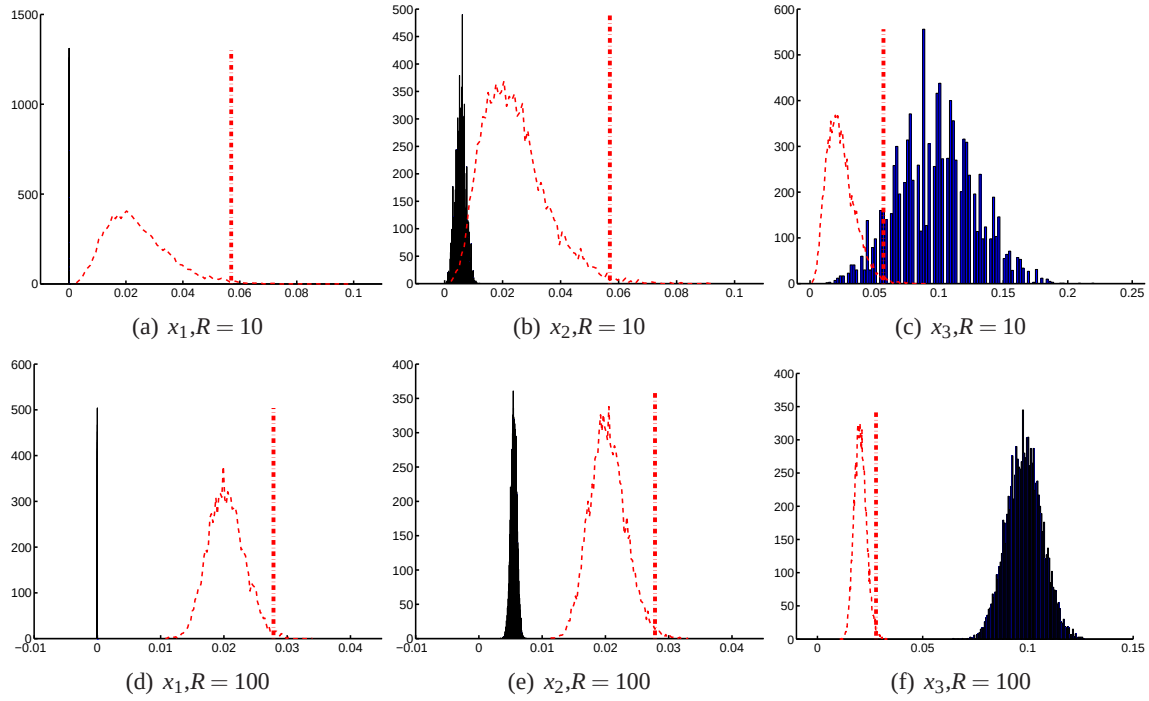


Figure 3.9: Approximation of $\sin(x)$ with a linear function and an appropriate variance γ to capture the discrepancy.



(a) x_1,$R = 10$ (b) x_2,$R = 10$ (c) x_3,$R = 10$

(d) x_1,$R = 100$ (e) x_2,$R = 100$ (f) x_3,$R = 100$

Figure 3.10: Sampling distributions for EE variance for each factor using 10^4 realisations of the experiment. The $g(x)$ function in Equation (3.10) is used. Dashed line is the theoretical sampling distribution. Vertical dashed line is the 99% threshold.

3.3 Conclusions

In this chapter a theoretical overview of existing screening methods that are used in practice when dealing with high-dimensional spaces, has been presented. The focus of the experiments has been on the input space of single-output models where inputs with non-linear and interaction effects as well as strong and weak effects on the response variable have been simulated.

If the simulator is quick to execute and a more detailed analysis of the effect of input variance on the output variance is required, standard analysis of variance methods such as Sobol' indices (Section 3.1.2) can be used. They typically require many more runs than the Morris method as was empirically demonstrated in Section 3.1.3.2 which precludes their usage on complex high-dimensional problems but for simple simulators they can provide quite accurate results.

For complex high-dimensional systems even the standard Morris method (Section 3.1.3) can require a prohibitively large number of simulator runs. In Section 3.2 a sequential version of the Morris method is proposed where factors with non-linear effects are removed from subsequent stages of the screening experiment. Factors that are shown to have a near-linear or no effect on the output can be discarded from further analysis as their effect can be removed during a preprocessing stage through an appropriately specified mean function as discussed in Section 3.2. Factors identified as having linear effects by the screening procedure can be treated independently.

The resulting screening procedure requires fewer simulator runs than the standard batch Morris method. In order to apply the sequential Morris method, the analyst must make a number of choices. To create the ordered design of OAT-experiment start-points, the maximum number of trajectories M must be specified. M is recommended to be chosen with respect to the effort needed to run the simulator; at worst, the simulator will need to be run $(k+1) \times M$ times. The step size Δ should be chosen so that the screening method will cover a large portion of the input space. The threshold value, σ_0, for discarding an input from subsequent OAT experiments can be set to zero so that only true linear- and no-effect inputs are investigated. Even in the case of deterministic simulators however, it is suggested to use a threshold $\sigma_0 > 0$ due to the computational errors in simulators.

The threshold may be set using the elicitation method described in Section 3.2.1. This method allows the prior specification of the divergence from linear of the factor effect in terms of a variance term. More restrictive forms may be desired that allow deviation from linear only in certain regions of the input domain and this is a direction for future research. In Section 6.2.4 an application of the sequential Morris method to a stochastic model is presented and contrasted to the batch Morris method.

A direction for future research would be to utilise higher-economy trajectory designs in the

Morris method and to theoretically prove for a given number of simulator evaluations the optimality of a design with respect to economy. Let K be the number of variables, with R trajectories in a Morris design, λ the total number of elementary effects calculated and M the total number of model executions. Then as defined by Morris (1991) the economy is:

$$E = \frac{\lambda}{M}.$$

For the standard Morris trajectory design, a single point is shared for two elementary effect calculations. The starting point is random and then shifted in each coordinate in sequence. The economy then is $E = \frac{K}{K+1}$.

As noted by Morris (1991) higher-economy designs can be achieved if the number of points in each trajectory is increased beyond $K+1$. The higher-economy designs modify the sampling strategy so the samples for each set of elementary effects per variable are no longer taken independently. As shown in Morris (1991) this is equivalent to cluster sampling and valid inferences about the population can still be made. In Section 5.1 of Morris (1991) a class of designs similar in structure to the standard Morris design but with higher-economy is proposed but no proof is offered that for a given number of model evaluations the proposed design offers the maximum possible economy. In Boukouvalas and Cornford (2007) we prove that for a design size of 2^K the maximum economy design is a hypercube. However a more general result for any design size would be needed in practice.

4 Heteroscedastic Emulation

CONTENTS

4.1 Introduction

Gaussian Processes (GPs) offer a principled way to perform many tasks including non-linear regression. They are applied in a multitude of problem domains and recent developments have shown how they can be extended to handle large datasets (see Quinonero-Candela and Rasmussen (2005) for a review). In this chapter we are specifically interested in large datasets which contain replicate observations of outputs for given inputs. Examples of such datasets, which arise where the underlying process truly behaves as a *stochastic process* include WiFi network signal strengths and stochastic computer (simulation) models.

In this chapter we present two novel methods of performing GP regression on complex datasets with replicated observations under heteroscedastic, i.e. input dependent, noise. An overview of existing work on heteroscedastic GPs is given in Section 4.2.

The Coupled Model presented in Section 4.4 extends the work of Kersting et al. (2007) by considering replicate observations and applying corrections due to finite sample size effects. The method of Kersting et al. (2007) is first discussed in Section 4.3 and a new interpretation of its working is offered that offers insight into the implicit approximations made in the method. We also describe how to correct a systematic bias error in the method which results in significations gains in predictive accuracy.

We introduce the issue of experimental design, that is the placement of input points, and provide empirical evidence on the effectiveness of utilising replicate observations compared to a space-filling design. The Coupled Model however is too complex to be utilised in the model-based design approach we develop in Chapter 5. The Joint Likelihood model, in which a simpler parametric variance model is used, retains tractability for the design framework and is discussed in Section 4.5.

A comparison of the Coupled Model and two variations of the Joint Likelihood model on a one-dimensional synthetic data set is given in Section 4.5.4. Finally in Section 4.6 we discuss possible model extensions.

4.2 Relation to Existing Work

One approach to modelling heteroscedastic noise within a GP framework is to use a system of coupled GPs modelling the mean and variance functions respectively. In Goldberg et al. (1998) a Monte Carlo approach was utilised to incorporate the uncertainty of the variance GP into the overall predictive uncertainty. The computational expense of this method however motivated an approximation whereby only the most likely value of the variance is utilised and the associated uncertainty around this estimate is discarded (Kersting et al., 2007). In both methods, the logarithm

of the variance is modelled using an independent GP on the transformed space $z(x) = \log(r(x))$ where $r(x)$ is the variance. A different training set may in principle be used for the log variance GP although in practice the same training set as for the mean GP is used.

Specifically, the predictive distribution at a new point x_* given a training set $\mathcal{D} = (x_i, t_i)_{i=1}^N$ is:

$$P(t_*|x_*, \mathcal{D}) = \int\int P(t_*|x_*, z, z_*, \mathcal{D})\, P(z, z_*|x_*, \mathcal{D}) \mathrm{d}z\, \mathrm{d}z_*,$$

where $z = \log(r(x_1), r(x_2), \ldots, r(x_N))$ the vector of variance predictions at the training points and $z_* = \log(r(x_*))$ the predictive noise level at the new test point x_*. Goldberg et al. (1998) use Monte Carlo to evaluate this integral by sampling from $P(z, z_*|x_*, \mathcal{D})$. Kersting et al. (2007) propose to use only the most likely values for the noise levels and approximate the predictive distribution $P(t_*|x_*, \mathcal{D}) \approx P(t_*|x_*, \hat{z}, \hat{z_*}, \mathcal{D})$ where $\hat{z}, \hat{z_*}$ the most likely values for the noise levels estimated using the mean value of the log variance GP. As pointed out by Kersting et al. (2007) this approximation is reasonable when the predictive variance of the log variance GP is sufficiently small so ignoring it will not have a significant impact on prediction accuracy. The Kersting et al. (2007) is extensively discussed in Section 4.3 as it forms the basis for the Coupled model we proposed in Section 4.4.

The Goldberg et al. (1998) and Kersting et al. (2007) type of approach which we follow for the Coupled Model in Section 4.4 allows for the specification of different GP priors for the mean and variance response. It is quite straightforward to incorporate most of the sparse approximations (Quinonero-Candela and Rasmussen, 2005) to handle very large datasets in these methods.

Snelson and Ghahramani (2005) proposed the Sparse Pseudo-Input GP (SPGP) as a sparse representation of a GP. The GP N^2 dimensional covariance, where N the number of training points, is approximated by a lower dimensional projection of size M. The M support points, known as *pseudo-points*, need not be a subset of the original training set and are treated as model parameters optimised through the maximisation of the likelihood. The construction allows for the implicit modelling of heteroscedastic variance through the location and density of the support points.

Snelson and Ghahramani (2006) propose a modification to SPGP (hereafter SPGP+HS) where an uncertainty parameter is associated with each pseudo-point and results in more accurate heteroscedastic prediction. The extra set of model parameters control the influence of each pseudo-input on the predictive distribution. The SPGP+HS predictive distribution is:

$$E(t_*|x_*, \mathcal{D}) = Q_{*N}\Sigma^{-1}\mathbf{t},$$
$$\mathrm{Var}(t_*|x_*, \mathcal{D}) = K_* - Q_{*N}\Sigma^{-1}Q_{N*} + \sigma_n^2 I,$$

where σ_n^2 is a nugget parameter, the training data matrix $\Sigma = Q_N + \text{diag}(K_N - Q_N) + \sigma_n^2 I$ and $\mathbf{t}$ the training data observations. The full GP covariance is denoted as K_N and K_* for the training and test data respectively. The sparsity is achieved through the Q matrix where $Q_N = K_{NM}(K_M + \text{diag}(h))^{-1}K_{MN}$ for training data and $Q_{*N} = K_{*M}(K_M + \text{diag}(h))^{-1}K_{M*}$ for the test data. h denotes the vector $h = (h_1, h_2, \ldots, h_M)$ of M parameters introduced in the SPGP+HS model to control the influence of each support point. Examining the equations, we see that the approximation is exact for the diagonal of the training data matrix Σ. The h vector affects the predictions through the Q_{*N} matrix where the correlation of the test to support points is calculated.

However no functional form of the variance is available so incorporating prior beliefs on the smoothness of the variance response as well as certain analyses such as variable selection for the variance of the output are not handled naturally in this framework.

Also as was noted in Snelson and Ghahramani (2006), this method does not perform well when small numbers of observations are available due to the flexibility of the model. Large training set sizes are uncommon in the emulation context where simulator runs are typically expensive to obtain – where the simulator is very cheap, its direct use might be preferred. The SPGP+HS method could be used in our design framework discussed in Chapter 5 as the method is equivalent to the specification of a non-stationary kernel for the GP and the calculations remain tractable. However the large number of free parameters would be problematic for small design sizes and for larger designs experimental design has less impact on inference efficiency. Walder et al. (2008) extend the SPGP method so that each basis function can have its own length scale. This improves predictive performance in some scenarios but requires the optimisation of twice the number of parameters.

Kleijnen and van Beers (2005) consider transformations of the output to remove the heteroscedasticity of the variance but are quite limited in their application. A "Studentising" transformation is suggested to transform the simulator output at each design point x:

$$\tilde{Z}(x) = \frac{\bar{Z}(x) - \hat{S}(x)}{\hat{\sigma}(x)/\sqrt{m}},$$

where $\bar{Z}(x)$ the mean value of the simulator output, $S(x)$ the "signal function" used to detrend the data, $\hat{\sigma}(x)$ a variance model and m the number of replicate observations. The signal function is specified *a priori* such that the data are of zero mean in the transformed space. Using the same number of replicates at each design point, a zero-mean GP with a single nugget parameter can be used in the transformed space as the distribution of the transformed variables is:

$$\tilde{Z}(x) \sim N(0, \frac{m}{m-2}).$$

For the variance function $\hat{\sigma}(x)$ the authors recommend to use the empirical sample variance if x is a training point and otherwise use piecewise linear interpolation between the variances of the two neighbouring training points. This method of interpolation avoids predicting negative variances.

4.3 The Kersting method

An extensive overview of the Kersting method is first provided in Section 4.3.1. We subsequently correct a systematic bias in the method in Section 4.3.3 and offer a new interpretation of the method in Section 4.3.4 which allows for a fuller understanding of the implicit approximations made. In Section 4.4 we extend the Kersting model to allow for efficient inference when the training data contain replicated observations.

4.3.1 Overview of the Kersting method

In this section we describe the Kersting approach (Kersting et al., 2007) referring to it when we say "the authors". We thank the authors for providing code to replicate most of the simulation experiments presented in their paper.

As in Goldberg et al. (1998) the noise variance is modelled using a second GP in addition to the GP governing the noise-free output value. In contrast to Goldberg et al. (1998), rather than using a Monte Carlo approach to approximate the posterior noise variance, a most likely approach is adopted, i.e. the uncertainty of the variance GP is not utilised.

The authors describe an iterative optimisation scheme for learning both the hidden noise variances z and the kernel hyperparameters $\theta = \{\theta_y, \theta_z\}$. Unlike Goldberg et al. (1998), the noise free y values are not explicitly represented. In fact Kersting alter the Goldberg et al. (1998) notation and write down the observation model as $t_i = f(x_i) + \mathcal{N}(0, r(x_i))$ where $r(x_i)$ the input dependent variance noise and the noise free values are denoted as f rather than y. However here we will use y to keep the notation consistent with Goldberg et al. (1998). The noisy observed output value at location x_i is denoted by t_i. As in Goldberg et al. (1998) the authors place a GP prior on y and conditional on the noise levels $R = \text{diag}[\exp(z_i)]$ the predictive distribution $p(t^*|\mathbf{t}, R, \theta_y)$ is:

$$E[t^*] = K_y^*(K_y + R)^{-1}t \tag{4.1}$$

$$Var[t^*] = K_y^{**} + R^* - K_y^*(K_y + R)^{-1}K_y^{*T}. \tag{4.2}$$

In Kersting et al. (2007) the squared exponential or a Matérn type covariance function is used for K_y and K_z. In the code an estimated nugget is used on both the y and z processes, i.e. $R = \text{diag}[\exp(z_i)] + \sigma_n^2 I$ where σ_n^2 the nugget variance.

To ensure the predicted variances are always positive, the variance GP prior is placed on the logarithms of the noise levels, denoted by $z(x) = \log(r(x))$. The authors state that in principle the training set locations $\mathbf{X}$ for the z-process could be different than for the y process but for notational convenience they are taken to be the same.

The authors state that as the noise rates z_i are independent latent variables in the combined regression model, the predictive distribution is:

$$p(\mathbf{t}^*|\mathbf{t}) = \int\int p(\mathbf{t}^*|\mathbf{z},\mathbf{z}^*,\mathbf{t})p(\mathbf{z},\mathbf{z}^*|\mathbf{t})\mathrm{d}\mathbf{z}\mathrm{d}\mathbf{z}^*. \tag{4.3}$$

This is equation (4) in the Kersting paper where we have changed the notation slightly to remove the explicit conditional on $\mathbf{X}$ and have replaced D with $\mathbf{t}$ to be consistent with the Goldberg et al. (1998) notation. The explicit inclusion of $\mathbf{z}^*$ in Equation (4.3) is not important and the equation can be understood by considering only the training points as Goldberg et al. (1998) do (see first equation in Section 2.1 of the paper).

The first term $p(\mathbf{t}^*|\mathbf{z},\mathbf{z}^*,\mathbf{t})$ is a Gaussian prediction with mean and variance given by Equations (4.1)-(4.2). As the authors note the problematic term in Equation (4.3) is $p(\mathbf{z},\mathbf{z}^*|\mathbf{t})$. In Goldberg et al. (1998) a set of samples $\{(\mathbf{z}_1,\mathbf{z}_1^*),(\mathbf{z}_2,\mathbf{z}_2^*),\ldots,(z_\mathbf{z},\mathbf{z}_k^*)\}$ is generated and the integrals in Equation (4.3) are approximated by:

$$p(\mathbf{t}^*|\mathbf{t}) = \frac{1}{k}\sum_{k=1}^{k} p(\mathbf{t}^*|\mathbf{z}_j,\mathbf{z}_j^*,\mathbf{t}).$$

This sampling procedure is computationally demanding so the authors propose to approximate the integral by the most likely values:

$$p(\mathbf{t}^*|\mathbf{t}) = p(\mathbf{t}^*|\tilde{\mathbf{z}},\tilde{\mathbf{z}^*},\mathbf{t}). \tag{4.4}$$

where $(\tilde{\mathbf{z}},\tilde{\mathbf{z}^*})$ the most likely values, that is:

$$(\tilde{\mathbf{z}},\tilde{\mathbf{z}^*}) = \arg\max_{\mathbf{z},\mathbf{z}^*} p(\mathbf{z},\mathbf{z}^*|\mathbf{t}). \tag{4.5}$$

This will be a good approximation if most of the probability mass of $p(\mathbf{z},\mathbf{z}^*|\mathbf{t})$ is concentrated around the most likely values.

The authors state that now computing the most likely noise levels (Equation (4.5)) and the predictive density $p(\mathbf{t}^*|\mathbf{t})$ (Equation (4.4)) requires only standard GP inference. For the latter this is clearly the case as computing the predictive density given the most likely noise values is straightforward. However the former is not clear, since a maximisation over $p(\mathbf{z},\mathbf{z}^*|\mathbf{t})$ is required.

The approach taken by the authors is described in the following section.

4.3.2 Optimisation

In this section we discuss how Kersting et al. (2007) propose to solve the maximisation problem in Equation (4.5). In fact as we shall see the authors break up the problem by estimating the empirical noise levels first without direct reference to the variance GP and subsequently utilising the variance GP to smooth the estimates.

In particular the iterative optimisation algorithm proposed separates the estimation of the noise levels and the parameters $\theta = \{\theta_y, \theta_z\}$. The authors state that learning would be easy if the noise level values were known for all data points.

The algorithm involves the following steps:

1. Given the observed data $\mathbf{t}$, we estimate the parameters θ_y of a standard homoscedastic GP, $\mathbf{G_1}$ by maximum likelihood. Specifically the optimisation problem is $\arg\max_{\theta_y,\sigma_n^2} p(\mathbf{t}|\theta_y,\sigma_n^2)$ where σ_n^2 an input-independent nugget parameter. After this step we have a density estimate for the noise-free values, i.e. $p(\mathbf{y}|\mathbf{t},\sigma_n^2)$.

2. Given $\mathbf{G_1}$, the empirical noise levels $\hat{\mathbf{z}}$ for the training data are estimated, i.e. $\log(\mathrm{var}[t_i, \mathbf{G_1}(x_i,\mathbf{t})])$. This is a crucial step in the algorithm and is discussed below. Essentially this is a smoothing step across the (very noisy) empirical noise estimates using another GP, $\mathbf{G_2}$. In this step θ_z is estimated by maximum likelihood.

3. The combined heteroscedastic GP $\mathbf{G_3}$ is estimated using $\mathbf{G_2}$ to predict the logarithmic noise levels. In this step θ_y is re-estimated. In the Kersting code a nugget term is also estimated. Hence the optimisation problem solved is:

$$(\theta_y, \sigma_n^2) = \arg\max_{\theta_y,\sigma_n^2} p(\mathbf{t}|\sigma_n^2, \theta_y, \mathbf{z}), \tag{4.6}$$

   where $\mathbf{z}$ the smoothed logarithmic noise levels estimated using the most likely value of $\mathbf{G_2}$.

4. If not converged, set $\mathbf{G_1} = \mathbf{G_3}$ and go to step 2. In the code the number of iterations is actually set a priori and no convergence criterion is used. Alternatively a metric of the difference of the current parameter estimates to the previous step estimates could be utilised as a convergence criterion.

The authors note that the algorithm is not guaranteed to improve the likelihood at each step (as it is not strictly speaking EM) and may oscillate as it considers only most-likely completions of the data.

The authors identify the estimation of the empirical noise levels (step 2) as the crucial step in the algorithm. The authors describe the problem thus: *Given* the observations t and the predictive distribution of the current GP estimates ($\mathbf{G_1}$), *find* an estimate of the noise levels $\text{var}[t_i, \mathbf{G_1}(x_i, \mathbf{t})]$, i.e. the variance of the observations at site i with respect to the GP prediction at that site. The GP predictive density $p(\mathbf{t}_*|\mathbf{t}, \theta_y, \mathbf{z})$ utilises the most likely prediction of the smoothed noise levels $\mathbf{z}$ obtained at the previous iteration. In the first iteration they are set to the input-independent nugget of the homoscedastic GP, i.e. $\mathbf{z} = \log(\sigma_n^2)$.

A set of s samples is obtained from the GP predictive density and are denoted t_i^j, $j \in \{1, \ldots, s\}$ for training point i. The authors state that viewing the observation t_i and each sample t_i^j as two independent observations of the same noise-free, unknown target, their arithmetic mean $(t_i - t_i^j)^2/2$ is a natural estimate of the empirical noise level at site i. The usage of the arithmetic mean is further discussed in Section 4.3.4. Finally they take the expectation of the arithmetic mean with respect to all s samples:

$$\text{var}[t_i, \mathbf{G_1}(x_i, \mathbf{t})] \approx \frac{1}{s} \sum_{j=1}^{s} \frac{1}{2}(t_i - t_i^j)^2. \tag{4.7}$$

The authors conclude by stating that this calculation minimises the average distance between the predictive distribution and the observation t_i and hence for a large enough number of samples ($s >$ 100), will be a good estimate for the empirical noise levels. We note the authors took a different optimisation approach for a similar modelling scenario in Plagemann et al. (2008) where an outer cross-validation loop is used to infer the GP hyperparameters whereby within each iteration a numerical minimisation of the model likelihood is used to infer the noisy observations for the latent GP.

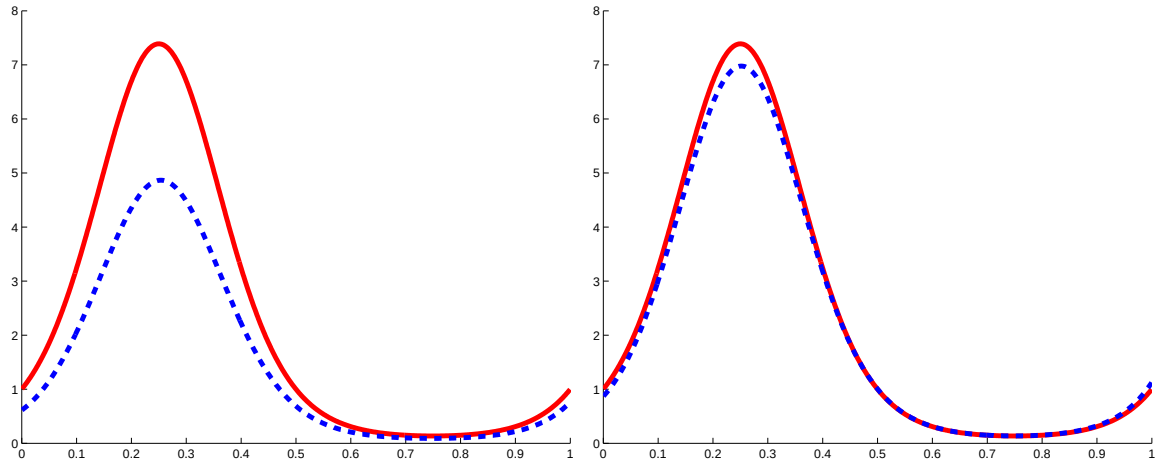
4.3.3 Correcting systematic bias

As we saw in Section 4.3 only the most likely prediction of the noise levels is used in the Kersting framework to keep the calculations tractable. However in Kersting et al. (2007) the mean value of the prediction from the variance GP is directly exponentiated. This results in under-predicting the true noise levels by introducing a bias from the log transformation. The correct way to account for the transformation can be found in the description of Warped GPs (Snelson, 2007):

$$E[r_*] = \int \exp(z_*) N(z_*|\mu_*, \sigma_*^2) dz_*, \tag{4.8}$$

where $N(z_*|\mu_*, \sigma_*)$ the posterior variance GP prediction. This integral can be analytically solved and corresponds to the mean value for the Log Normal distribution, i.e. $E[r_*] = \exp(\mu_* + \sigma_*^2/2)$. A simulation experiment demonstrating the effectiveness of the correction is shown in Figure 4.1. It can be clearly seen that without the correction, the Kersting method does not recover the true func-

tion even using an very dense training data set whereas utilising the log correction, the prediction is accurate with no systematic error apparent.



(a) Variance No Correction (b) Variance with Correction

Figure 4.1: Correcting the bias in the Kersting method due to the log transformation. Synthetic experiment using 1080 training points, showing the variance prediction for the Yuan and Wahba test function (Equation (4.17)). Red solid line denotes the true variance and blue dashed lines denote the prediction obtained using the heteroscedastic GP framework proposed by Kersting with and without the log correction described Section 4.3.3. The systematic bias due to the log transformation is evident when not corrected as the variance is underestimated everywhere in the design region.

4.3.4 A new interpretation

The crucial step in the Kersting algorithm is the sampling from the GP of the previous step (initially a homoscedastic GP) to create variance observations for the variance GP inference (see Equation (4.7)).

Examining the Monte Carlo sampling described by equation (4.7) we realise it is approximating an integral. Denoting $\hat{r}_i^{\tau+1}$ the estimated variance observation at iteration step $\tau+1$, the integral is:

$$\hat{r}_i^{\tau+1} = \frac{1}{s}\sum_{j=1}^{s}\frac{1}{2}(t_i^{obs} - t_i^j)^2 \approx \frac{1}{2}\int (t_i^{obs} - t_i)^2 p(t_i|\mathbf{t}, \mathbf{r}^{\tau}, \theta_y) dt_i, \tag{4.9}$$

where t_i^{obs} the observation at point x_i, $\mathbf{t}$ the training data observations and $\mathbf{r}^{\tau}$ the estimated noise levels obtained at the previous iteration τ. Initially $\mathbf{r}^{\tau}$ is obtained from the homoscedastic GP and is equal to the nugget term, i.e. $\mathbf{r}^1 = \sigma_n^2$. The hyperparameters for the heteroscedastic GP θ_y are fixed during this step. The conditioning on the training inputs $\mathbf{x}$ and x_i is omitted for brevity. The reason for the $\frac{1}{2}$ term will be explained later in this section.

The distribution $p(t_i|\mathbf{t},\mathbf{r}^\tau,\theta_y)$ is simply the predictive GP distribution for training point i:

$$E[t_i] = K_y^*(K_y + R^\tau)^{-1}\mathbf{t}$$
$$Var[t_i] = K_y^{**} + r_i^\tau - K_y^*(K_y + R^\tau)^{-1}K_y^{*T},$$

where $R^\tau = \text{diag}(r_1^\tau,\ldots,r_N^\tau)$ is the diagonal matrix of variances obtained at the previous step τ from the variance GP prediction.

Note that the observation t_i^{obs} appears both in the conditioning of $p(t_i|\mathbf{t},\mathbf{r}^\tau,\theta_y)$ and in the variance expression $(t_i^{obs}-t_i)^2$. The resulting double counting may be rectified by conditioning on all the sites except i, i.e. $p(t_i|\mathbf{t}_{-i},\mathbf{r}^\tau,\theta_y)$ where $\mathbf{t}_{-i}$ denotes the set of all training observations except the i^{th}. This density differs in that the training data matrix R^τ does not include the variance at point i, r_i^τ - see Equations (4.13)-(4.14).

We can reformulate the above expressions in terms of the distribution of the latent noise-free variables $\mathbf{y}$:

$$E[t_i] = E[y_i]$$
$$Var[t_i] = r_i^\tau + \sigma_{y_i}^2,$$

where $y_i \sim N(E[y_i],\sigma_{y_i}^2)$ the predictive distribution of the noise-free latent variable at site i. Note than both the predictive mean $E[y_i]$ and variance $\sigma_{y_i}^2$ depend on the estimated variance levels obtained at the previous iteration τ through the R^τ matrix.

The integral in Equation (4.9) can be solved analytically:

$$\hat{r}_i^{\tau+1} = \frac{1}{2}\int (t_i^{obs}-t_i)^2\, p(t_i|\mathbf{t}_{-i},\mathbf{r}^\tau,\theta_y)\mathrm{d}t_i = \frac{1}{2}\left[(t_i^{obs}-E[t_i])^2 + r_i^\tau + \sigma_{y_i}^2\right]. \tag{4.10}$$

If we treat the predictive density $p(t_i|\mathbf{t}_{-i},\mathbf{r}^\tau,\theta_y)$ as a likelihood and maximise with respect to r_i we obtain the maximum likelihood solution:

$$r_i^{ML} + \sigma_{y_i}^2 = (t_i - E[t_i])^2. \tag{4.11}$$

If we fix the observed value $t_i = t_i^{obs}$ we can re-express Equation (4.10):

$$\hat{r}_i^{\tau+1} = \frac{r_i^{ML} + r_i^\tau}{2} + \sigma_{y_i}^2. \tag{4.12}$$

We can interpret this expression as the variance estimate at iteration step $\tau+1$ which is taken to be the average of the maximum likelihood noise estimation r_i^{ML} and the previous smoothed estimate

of the noise level r_i^{τ}. The term $\sigma_{y_i}^2$ expresses our uncertainty of the noise free function value y_i.

The inclusion of the noise-free variance $\sigma_{y_i}^2$ in the estimation of the noise level r_i stems from our inability to disambiguate the two sources of uncertainty, the intrinsic function variance and our uncertainty due to not knowing the true noise-free model values $\mathbf{y}$. When a large amount of training data is available, $\sigma_{y_i}^2 \to 0$ and can be ignored. In sparse training data scenarios however it acts as a regulariser by ensuring the variance estimate is greater than a minimum threshold. Examining the maximum likelihood expression for the noise level in Equation (4.11), we note if $\left(t_i^{obs} - E[t_i]\right)^2 < \sigma_{y_i}^2$ the sample t_i^{obs} is too small and no useful estimate of r_i^{ML} is available. In such cases the inclusion of the noise-free uncertainty $\sigma_{y_i}^2$ in Equation (4.12) prevents the algorithm from considering very small or zero empirical variance estimates which would lead to overfitting the observed data by the variance GP. An alternative viewpoint is to state that when $(t_i^{obs} - E[t_i])^2 < \sigma_{y_i}^2$, the uncertainty on the noise free value y_i does not allow for a direct estimation of the variance r_i at that point.

In summary the Kersting method may be directly implemented without need for sampling by directly evaluating:

$$\hat{r}_i^{\tau+1} = \frac{1}{2}\int (t_i^{obs} - t_i)^2 p(t_i|\mathbf{t}_{-i}, \mathbf{r}^{\tau}, \theta_y) dt_i = \frac{1}{2}\left((t_i^{obs} - E[t_i])^2 + Var[t_i]\right),$$

where

$$E[t_i] = K_y^*(K_y + R_{-i}^{\tau})^{-1}\mathbf{t}_{-i} \tag{4.13}$$

$$Var[t_i] = K_y^{**} + r_i^{\tau} - K_y^*(K_y + R_{-i}^{\tau})^{-1}K_y^{*T}, \tag{4.14}$$

where $R_{-i}^{\tau} = \mathrm{diag}(r_1^{\tau}, \ldots, r_{t-i}^{\tau}, r_{t+i}^{\tau}, \ldots, r_N^{\tau})$ the diagonal matrix of variances obtained at the previous step τ for all training points except x_i.

The use of the previous estimate of the noise variance, i.e. the r_i^{τ} term in Equation 4.12, which stems from a GP regression step on the log variances, allows the algorithm to take into account the correlation between the variances of neighbouring points. Other approaches such as doing direct maximum likelihood of the multivariate likelihood $p(\mathbf{t}|\mathbf{r})$ lead to overfitting as the correlation between variances is not considered. By placing a GP prior on $\mathbf{r}$ direct optimisation is challenging and a sampling type approach may be preferable such as the Metropolis algorithm originally proposed by Goldberg et al. (1998), which is guaranteed to converge to the optimal solution. We therefore see that the iterative nature of the Kersting algorithm allows for the variance correlation to be considered without the need for computationally expensive sampling or high dimensional non-linear optimisation. The heuristic nature of the approach however implies that no conver-

gence guarantees are available although as both Kersting and we have observed, in numerical experiments it performs reasonably well.

In Appendix A.1 we provide an alternate interpretation of the Kersting method by explicitly deriving the empirical variance estimation step (Equation (4.9)) from the posterior distribution of the noise process (Equation (4.5)). The derivation allows us to better understand the nature of the approximations implicit in the Kersting method:

- *Univariate optimisation*: The optimisation of the empirical noise levels is performed one point at-a-time rather than jointly. Further, a batch optimisation is used where the new estimates for each variance level are not used in the estimation of the other noise levels until the subsequent iteration.

- *Noise-free targets*: The noise-free latent variables $\mathbf{y}$ are assumed to be known, i.e. the variance $\lambda_y = 0$. This is a reasonable assumption only under strong prior knowledge on the noise-free process or for very dense training data where the variance λ_y is negligible.

- *Equal-weighing*: The maximum likelihood estimate of the noise level at each iteration is averaged with the smoothed noise level from the previous iteration (see Equation (4.12)). The two terms are weighed equally by taking their arithmetic mean.

- *Variance of variance GP ignored*: At no point in the algorithm is the variance of the variance GP taken into account. Therefore in scenarios where the uncertainty of the variance GP varies significantly (for e.g. under a clustered training set), the estimates of the variance GP are all treated equally despite the differing amount of uncertainty associated with each prediction.

In our opinion therefore the method can only be justifiably used in scenarios where either dense training data are available or strong prior knowledge can be used to justify some of the approximations.

4.4 Coupled Model

We show how to extend the most likely heteroscedastic GP framework of Kersting et al. (2007) to use replicate observations which permits more accurate and efficient learning of heteroscedastic GPs. This section is an extension of Boukouvalas et al. (2009).

As in Kersting et al. (2007) we use a coupled system of GPs to predict the mean and heteroscedastic variance. Our framework can learn the GP using a mixture of single and replicate observations, utilising the first two moments of the latter. The variance GP operates on log space to

ensure the predicted variance is always non-negative. The log transformation however introduces a bias whose effect can be significant since we expect relatively few replicates at each input point. For this reason we introduce a correction to the sample log variance described in Section 4.4.1.

The modifications to the Kersting model and optimisation method used to infer the parameters is described in Section 4.4.2.

Another issue commonly occurring in the context of complex datasets is that of experimental design, i.e. where to obtain the observations in input space, and the related sequential problem of active learning. Using our framework we assess in Section 4.4.3 the efficiency of different designs, comparing the use of replicates against single observations, which better cover the input space. A more principled approach to design is presented in Chapter 5.

Lastly, in Section 4.5.4 we demonstrate the Coupled Model method on a known test function and compare it with the Joint Likelihood model described in Section 4.5.

4.4.1 Log sample variance bias correction

When computing the logarithm of the sample variance a bias is introduced in the estimation due to the non-linear transformation. The bias can be significant especially when using relatively few observations. Standard theory (Cox and Solomon, 2003) allows us to estimate the bias and variance of the log sample variance estimator:

$$z = \log(\mathbf{S}^2) - \psi\left(\frac{n-1}{2}\right) - \log 2 + \log(n-1) \tag{4.15}$$

where z is the true log variance, $\mathbf{S}^2$ is the sample variance estimate and Ψ the digamma function. The uncertainty of the estimate of the log variance can also be computed (Cox and Solomon, 2003):

$$\sigma^2_{\mathbf{S}^2} = \Psi_2((n-1)/2), \tag{4.16}$$

where Ψ_2 is the trigamma function. A proof of these results is given in Appendix A.2.

These corrections can be applied directly to the estimation of $\mathbf{G_2}$ by using Equation (4.15) to correct the sample log variance for each design point. The corresponding uncertainty of the log variance estimates can be included in the likelihood of $\mathbf{G_2}$ using Equation (4.16). The main rationale for suggesting these improvements is to make the method more robust to smaller sample sizes where the bias due to the log transformation can be significant.

4.4.2 Utilising repeated observations

Explicitly considering replicated observations requires only small modifications to the Kersting model. In particular for design points where only single observations are available, the inference proceeds as in the original Kersting method described in Section 4.3.

We split the observations to two sets, *r*=replicate observations and *s*=single observations. For replicated observations, the sample mean output $\bar{y_i}$ and corrected sample variance, r_i is calculated for each input point i in the training set.

To initialise the algorithm, a standard homoscedastic GP ($\mathbf{G_1}$) is estimated by by maximum likelihood on the two sets of observations $\mathbf{t}_\mu = \{\mathbf{t}_s, \mathbf{t}_r\}$ where $\mathbf{t}_s = (y_1, \ldots y_s)$ the vector of single observations and $\mathbf{t}_r = (\bar{y_1}, \ldots \bar{y_r})$ the vector of empirical means for the replicated observations. For the set $\mathbf{t}_r$ the observation error can be estimated as the distribution of the sample mean for Gaussian variables is $\bar{y_i} \sim \mathcal{N}\left(\mu_i, \frac{\sigma_i^2}{n_i}\right)$ where μ_i the true mean, σ_i^2 the true variance and n_i the number of replicate observations. By using the sample variance s^2 as an estimate of the true variance, a fixed nugget of size $\frac{\sigma^2}{n}$ can be used in the covariance of $\mathbf{G_1}$.

As in the original Kersting algorithm, $\mathbf{G_1}$ is used to provide an initial estimate of the variance at design points where only single observations are available (see Step 2 in Section 4.3.2). For replicate observations, the empirical variance of the training data at x_i is computed. To correct for the biased estimate due to the log transformation Equations (4.15) and (4.16) are used.

The combined set of estimated empirical variances for single observations and corrected log samples variance for replicated observations is used in the training of the variance GP, $\mathbf{G_2}$. We note here that typically the number of replicated observations is much smaller than the samples obtained from $\mathbf{G_1}$ when no replicate observations are available at the design point. Thus the training of $\mathbf{G_2}$ takes into consideration the noise on the variance, computed using Equation (4.16), which is particularly important in the small sample case where the second moment estimates can be quite noisy. This allows $\mathbf{G_2}$ to smooth the variance estimates based on the prior GP specified, and produces more reliable estimates of the underlying noise variance. Specifically the predictive distribution equations for $\mathbf{G_2}$ are:

$$\begin{aligned} \mu_{\mathbf{G_2}*} &= K^*(K + R_{\mathbf{G_2}})^{-1}\mathbf{t}_{\mathbf{S}^2}, \\ \Sigma_{\mathbf{G_2}*} &= K^{**} + R^*_{\mathbf{G_2}} - K^{*^T}(K + R_{\mathbf{G_2}})^{-1}K^*, \end{aligned}$$

where the target values $\mathbf{t}_{\mathbf{S}^2}$ are the sample log variances either estimated in the previous step in the case of single observations or computed directly from the samples in the case of replicated observations. K is the training point covariance, K^{**} the test point covariance and K^* the training-

test point covariance. The matrix $R_{\mathbf{G_2}}$ is defined as:

$$R_{\mathbf{G_2}} = \sigma^2_{\mathbf{G_2}} I + \begin{pmatrix} V_s & 0 \\ 0 & V_r \end{pmatrix},$$

where $\sigma^2_{\mathbf{G_2}}$ is the noise hyperparameter (nugget). $V_s = \text{diag}(\sigma^2_{v1}, \ldots \sigma^2_{vS})$ is the variance of the log variance from Equation (4.16) for the single observations where typically the number of samples is high since sampling from a GP is cheap. Hence V_S will be nearly zero in most cases. $V_r = \text{diag}(\sigma^2_{\mathbf{S}_1^2}, \ldots, \sigma^2_{\mathbf{S}_r^2})$ on the other hand contains the variance of the log variance for repeated observations where the assumption is that there will be few replicates per training point and hence the variance calculated in Equation (4.16) will be higher. The variance at the replicate points r should in fact be lower than for the non-replicate set s but we are not aware of any suitable uncertainty estimate of the variances for the latter. Heuristics such as setting $V_s = \max V_r$ are possible but have not been used in the simulation experiments. $R^*_{\mathbf{G_2}} = \sigma^2_{\mathbf{G_2}} I$ is the predicted noise level at the test points.

As in the Kersting method, the hyperparameters of the heteroscedastic GP,$\mathbf{G_3}$, are then inferred to jointly predict the mean and variance. The equations for $\mathbf{G_3}$ are slightly more involved since the most likely value of the variance from $\mathbf{G_2}$ is included and the effect of utilising moments of replicated observations must be considered. The derivation of $\mathbf{G_3}$ is given in Appendix A.3. The predictive distribution equations for $\mathbf{G_3}$ are:

$$\begin{aligned} \mu_{\mathbf{G_3}*} &= K^* \left(K + R_{\mathbf{G_3}} P^{-1}\right)^{-1} \mathbf{t}_\mu, \\ \Sigma_{\mathbf{G_3}*} &= K^{**} + R^*_{\mathbf{G_3}} - K^{*^T} \left(K + R_{\mathbf{G_3}} P^{-1}\right)^{-1} K^*, \end{aligned}$$

where $P = \text{diag}(n_1, \ldots, n_N)$ the number of samples at each training point, $R_{\mathbf{G_3}} = \text{diag}[r(x_1), \ldots, r(x_N)]$ the variance estimate from $\mathbf{G_2}$ at the training points and $R^*_{\mathbf{G_3}}$ the $\mathbf{G_2}$ variance estimate at the test points. Note that the training target values $\mathbf{t}_r$ within $\mathbf{t}_\mu$ are the sample means and not individual observations of the underlying random process. Since the variance of the empirical mean is inversely proportional to the number of replicated observations (see Appendix A.3) the variance prediction from $\mathbf{G_2}$ has to be divided by the number of replicates n_i.

The algorithm is iterated until a suitably defined convergence criterion is satisfied (see the discussion in Section 4.3.2 for a discussion of convergence).

4.4.3 Experimental Design Simulation Study

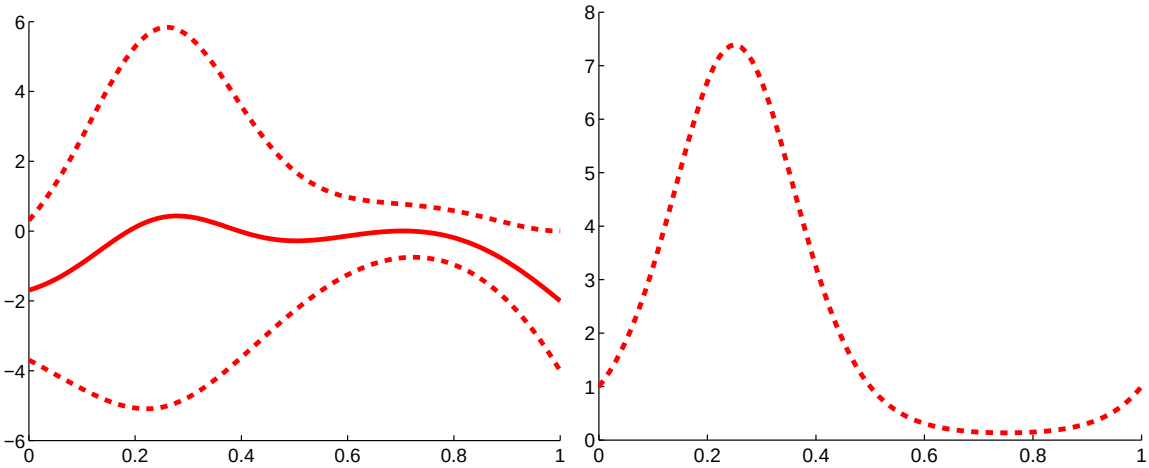
In this section we compare the approach of Kersting et al. (2007) with the replicate approach presented in Section 4.4.2 on a variety of designs to examine the effect of replication on predictive

performance.

The synthetic dataset $\mathbf{Y}$ originally used by Yuan and Wahba (2004) is utilised in this section:

$$y = 2(e^{-30(x-0.25)^2} + \sin(\pi x^2)) - 2 + e^{sin(2\pi x)}\mathcal{N}(0,1), \tag{4.17}$$

where $\mathcal{N}(0,1)$ is the standard normal distribution. In Figure 4.2 a visualisation of the function is provided. The validation measures used are the Mean Squared Error (MSE) and the Dawid score described in Section 2.5 to assess the goodness of the mean and covariance prediction. A 2000 point single observation random design is used for validation.



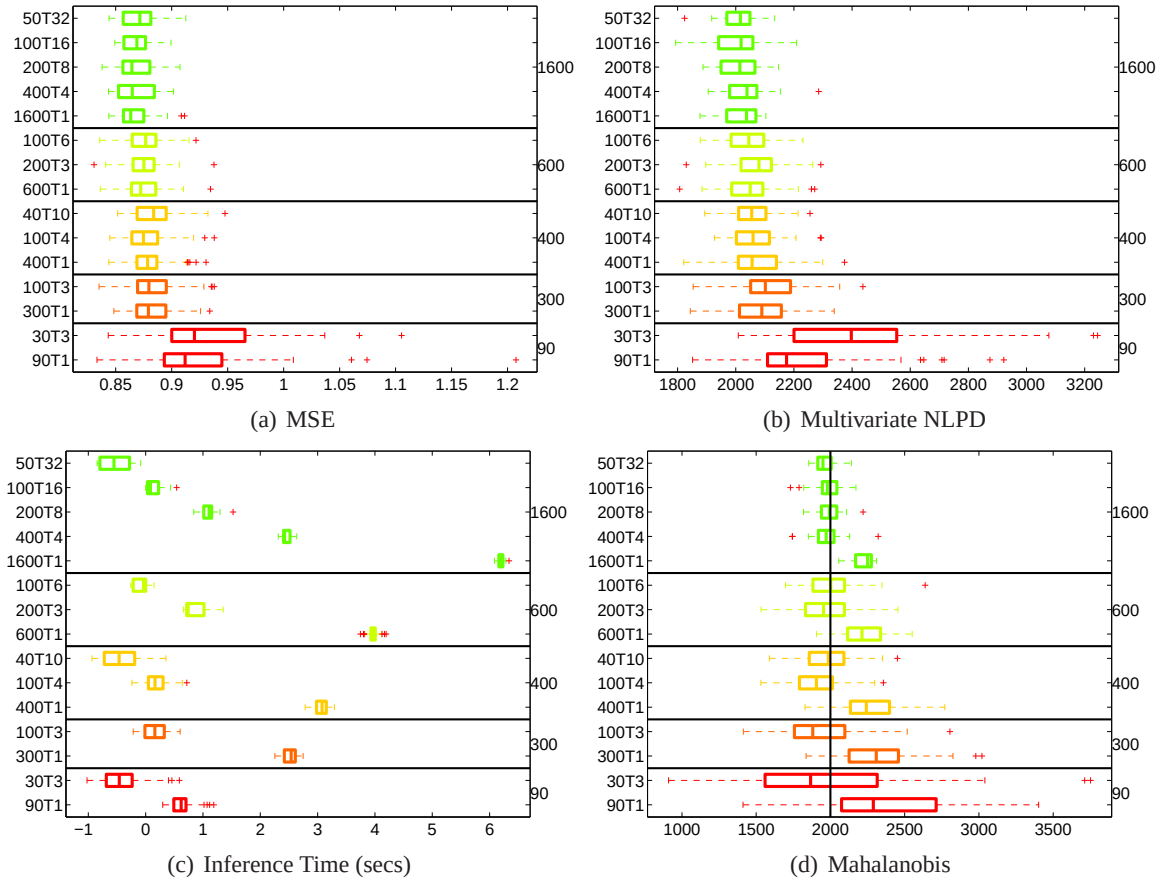
(a) Mean and 2 standard deviations (b) Variance

Figure 4.2: Visualisation of the Yuan and Wahba function (Equation (4.17)) used in the simulation study in Section 4.4.3. (a) Mean (solid line) and two standard-deviation error bars (dashed line). (b) Variance.

The training design we have used is space-filling where we either have single observations only or the same number of replicate observations across all design points. For each realisation of the experiment, 1000 realisations of random designs were generated and the design with the maximum minimum distance is selected as the training set. Clearly in our framework more complex designs are allowed with different number of replicates per training point but we have focused on these two extremes to highlight the effect of replicate observations without making unrealistic assumptions of prior knowledge on the shape of the true function mean and variance. Note that for the case where only single observations are made for all design points, i.e. no replicate observations are made, our method reduces to that of Kersting et al. (2007).

In Figure 4.3 the predictive performance of the Coupled Model on a progressively sparser set of designs with more replicated observations is examined. The total number of simulator evaluations is kept fixed at 90, 300, 400, 600 and 1600. The benefit of a completely space-filling design where only single observations are used is contrasted with a sparser training design with more

replicate observations per design point. For example, the bottom box corresponds to a total of 90 observations, being either a training set 90 observations or 30 training points of sample means and variances computed using 3 replicate observations.



(a) MSE (b) Multivariate NLPD

(c) Inference Time (secs) (d) Mahalanobis

Figure 4.3: Performance of replicate and non-replicate designs with the total number of observations fixed. Notation 30T3 = 30 training points each with 3 replicates. Results shown for a total of 90, 300, 400, 600 and 1600 observations used in the training set. A test set of 2000 points is used for validation. For each input configuration 100 realisations of the experiment were performed except for the 1600 simulator evaluation designs where only 20 realisations have been used.

Overall there is little difference in terms of MSE and Multivariate NLPD signifying similar performance with regards to the accuracy across all designs as can be seen in Figure 4.3. The differences in Mahalanobis error are discussed below. For the smallest training size examined where only 90 model observations are available, the highly clustered 30×3 design performs worse than the space-filling 90×1 design in terms of both validation measures. In fact as will be confirmed in Chapter 5, highly clustered designs generally achieve worse MSE compared to space-filling designs as the latter allow for more accurate interpolation due to the better coverage of the space. For the larger training sizes examined, the input space is sufficiently covered so all designs achieve similar errors.

The replicate designs are however substantially faster to use from a computational perspective, i.e. inference time, as the number of replicates increases. This can easily be understood since as

the number of replicates increases, the number of training points decreases. The latter of course determines the size of the GP training covariance matrix that needs to be inverted during inference. We have replicated these results on the synthetic dataset originally used in Goldberg et al. (1998) and obtained similar results.

We note here that the results presented here differ from Boukouvalas et al. (2009) where the replicate designs were shown to achieve lower Mahalanobis errors and the conclusion drawn was that they more accurately capture the variance response. However the large errors observed were due to the bias error in the most likely variance prediction (Section 4.3.3) which has a larger impact on single observation designs rather than replicate designs. Correcting the bias error, the Mahalanobis error is smaller for all designs and especially so for the single replicate cases as can be seen in Figure 4.3(d). Furthermore as noted by Bastos (2010) the Multivariate NLPD is more appropriate for emulator comparison where different training sets are used. We will see however in Chapter 5 that in higher-dimensional scenarios with sparse designs, replicate designs do in fact capture the variance response more accurately than single observation designs.

We conclude from these experiments that using the first two moments of replicate observations proves beneficial in terms of inference time without significantly affecting predictive accuracy given sufficient coverage of the design space. In Chapter 5 we further investigate the effect of replicated observations through a more rigorous experimental design approach.

4.5 Joint Likelihood Model

The model we develop in this section is similar to the SPGP+HS described in Section 4.2 but allows different mean and variance response structures.

In the most likely heteroscedastic framework of Kersting et al. (2007) and the extension presented in Section 4.4 only the predictive mean value of the variance GP is used whereas its predictive variance is discarded. The complexity of explicitly optimising the variance GP therefore seems unnecessary and a simpler interpolation model could suffice. We introduce a new heteroscedastic model, which simplifies previously proposed models, making the optimal experimental design problem more tractable. In addition, we believe that for some systems the variance response will be less complex than the mean response allowing for the adoption of a simpler model for the former while retaining the full non-parameter probabilistic GP for the latter.

In this section we present the Joint Likelihood model where the optimisation of the mean and variance model parameters proceed jointly. The crucial simplification is the consideration of only deterministic variance models. The stochastic process for the variance GP is discarded and replaced with a variance model of the form $f_{\sigma^2}(x, \beta)$ with unknown parameters β. The het-

eroscedastic GP prior is thus:

$$p(\mu|\theta,\mathbf{x}) = N\left(0, K_\mu + \text{diag}(\exp(f_{\sigma^2}(x,\beta))P^{-1})\right),$$

where diag denotes the diagonal matrix of the input vector, K_μ is the usual covariance matrix which depends on parameters θ_μ representing process variance and length scales, β the variance model parameters and P a diagonal matrix containing the number of replicated observations at each training point site. The set of free parameters for this model is $\theta = \{\theta_\mu, \beta\}$.

The likelihood for the model when considering replicated observations is derived in Section 4.5.1. The two forms for f_{σ^2} used in this thesis, the Fixed Basis and Latent-Kernel models, are described in Sections 4.5.2 and 4.5.3 respectively.

4.5.1 Derivation of Likelihood

Assuming normality, the sample variance is distributed as a scaled $\mathcal{X}^2$ distribution with $n_i - 1$ degrees of freedom:

$$s_i^2 \sim \frac{f_{\sigma^2}(x,\beta)}{n_i - 1}\mathcal{X}^2_{n_i-1}$$

where n_i the number of replicates at location x_i. This can also be expressed as a Gamma distribution:

$$p(s_i^2|\beta, x_i, n_i) \sim \Gamma\left(\frac{n_i-1}{2}, \frac{2f_{\sigma^2}(x,\beta)}{n_i-1}\right),$$

A zero-mean GP prior is placed on the mean:

$$p(\mu|\theta) = GP(0, K_\theta), \tag{4.18}$$

where K_θ is the input dependent correlation and θ the kernel hyperparameters.

The joint log likelihood of the sample mean $\hat{\mu}$ and variance s^2 for N observations can then be derived:

$$\log p(\hat{\mu}, s^2|\mathbf{X},\theta,\beta) = \left(\sum_{i=1}^{N} \log p(s_i^2|\beta, x_i, n_i)\right) + \log N(\hat{\mu}|0, K_\theta + RP^{-1}), \tag{4.19}$$

where R the diagonal matrix with elements $\exp(f_{\sigma^2}(x_i,\beta))$ and P the diagonal matrix of the number of replicated observations. The derivation is given in Appendix A.4.

4.5.2 Fixed Basis

In the Fixed Basis variance model, the log variance function is modelled as a linear-in-parameters regression using a set of fixed basis functions:

$$f_{\sigma^2}(x,\beta) = \exp\left(H(x)^T\beta\right), \tag{4.20}$$

where $H(x)$ is the set of fixed basis functions with known parameters. A simple example in 2D space is a linear variance model: $f_{\sigma^2}(x,\beta) = \exp(\beta_0 + x_1\beta_1 + x_2\beta_2)$.

Two types of basis functions have been considered in this thesis, local (e.g. radial basis functions) and global (e.g. polynomial) to provide the input dependent nugget term. An advantage of local basis functions is the interpretability of priors on the β coefficients as they relate to a particular region of input space. However the number of local basis functions required for domain coverage grows exponentially with the input dimension. Polynomial and other global basis are therefore better suited for higher-dimensional spaces but imply a relatively simple variance response.

4.5.3 Latent-Kernel

In high-dimensional cases a non-parametric method could be considered using an additional 'variance kernel'. For the Coupled Model, the variance prediction of $\mathbf{G_2}$ is not utilised in the prediction of $\mathbf{G_3}$. We further simplify this model by explicitly incorporating the mean prediction of $\mathbf{G_2}$ as a deterministic function into $\mathbf{G_3}$:

$$f_{\sigma^2}(x,z) = k_\Sigma^T(K_\Sigma + \sigma_n^2)^{-1}z,$$

where $K_\Sigma = k(X_z, X_z)$ and $k_\Sigma = k(X_z, X_t)$ are the variance kernel functions, depending on parameters θ_Σ and σ_n^2 a nugget term. In this case z is a variance 'pseudo observation' vector. In principle the latent points X_z could be set to the entire training data set X_t of the GP K_μ but for quicker inference it can be set to a much smaller set without the need to be a subset of X_t.

Note that sparse approaches to this parametrisation, similar to Snelson and Ghahramani (2006), are likely to be more computationally attractive. The main difference of this model from the model of Snelson and Ghahramani (2006) is that we do not entangle the mean and variance response, allowing separate kernels for each. This will be important where the complexity of the mean and variance response is different. This model also bears resemblance to the Kersting et al. (2007) model, however here we directly represent the log variance function as a non-parametric kernel regression rather than employing a Gaussian process model and then using the most likely value. This enables us to write down a simpler model, with the same flexibility as Kersting et al. (2007), for which we can evaluate the design criterion in Chapter 5.

The parameters of the model are X_z, z and θ_Σ the parameters of the kernel function k_Σ. Although all could in principle be optimised, in the experiments presented we simplify the optimisa-

tion task by fixing X_z to a Latin Hypercube design and fixing θ_Σ to constant values. For example for a squared exponential kernel (Section 2.4.2), the length scale is replaced by the location of the latent variables X_z and the process variance by the optimised coefficients z.

This model is of intermediate complexity. It is more flexible than the Fixed Basis model (Section 4.5.2) allowing the specification of any kernel function for the variance response. However it is more limited than the Coupled Model as the simulator variance is no longer treated as a random variable but rather the variance responses are interpolated deterministically. An example of the Latent-Kernel model is provided in Section 4.5.4.

4.5.4 Example of all three variance models

In this section a comparison of the Coupled (Section 4.4), Latent-Kernel (Section 4.5.3) and Fixed Basis (Section 4.5.2) variance models via a simple one-dimensional example is provided.

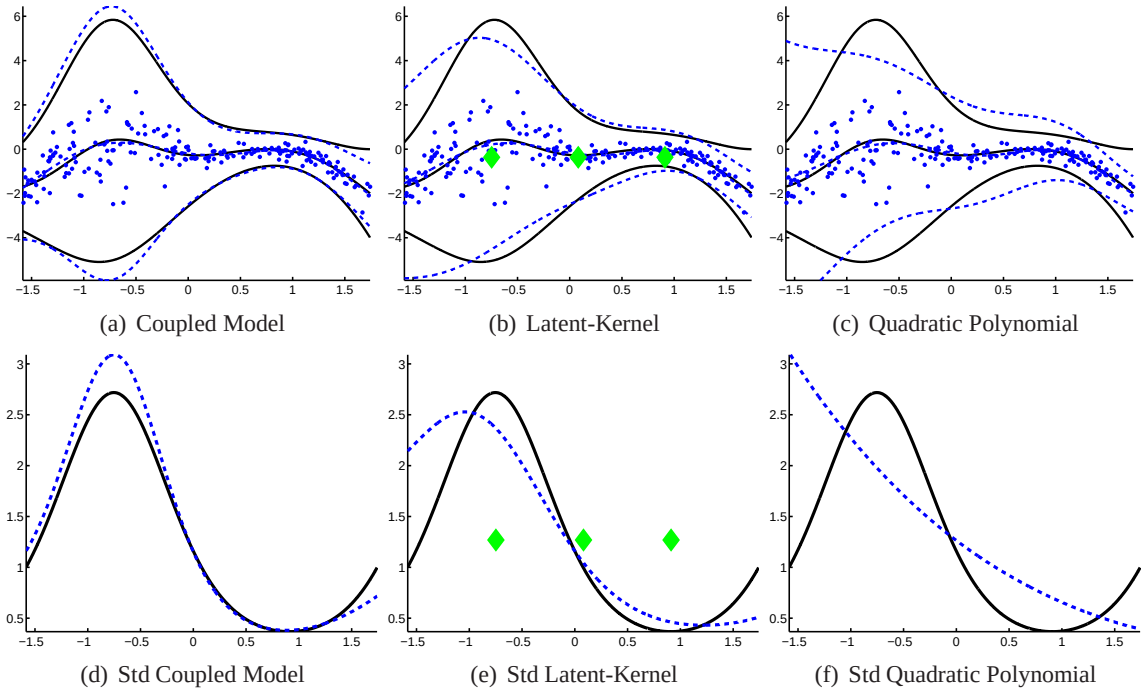
The test function used in Section 4.4.3 is utilised as the stochastic simulator and a Latin Hypercube 200 point design with 4 replicates at each point is used as the training design. A squared exponential kernel is used in all models, including for the variance GP in the Coupled Model. The Coupled Model also includes a nugget parameter in the variance GP kernel specification. The Latent-Kernel model consists of three latent points X_z chosen to be equally spaced in the design space. Finally a quadratic function is used for the Fixed Basis variance model. All models therefore have a total of five free parameters.

In Figure 4.4 the mean and variance prediction for all models is given. In terms of predictive performance, the Coupled Model offers the best match to the simulator output. The Latent-Kernel overestimates the variance in the high variance region of the simulator output when the distance from the closest latent point increases. The variance prediction for the quadratic model is poor due to the inherent inflexibility of the model. Due to the large training set size, the Coupled Model, being the most flexible of the three, does best in this example.

4.6 Conclusions

The investigation of the Kersting method in Section 4.3 has allowed for a clear understanding of the theoretical underpinnings of the method and the approximations implicit in its original formulation. Furthermore the correction of the bias due to the non-linear transformation of the most likely variance (Section 4.3.3) has significantly improved the accuracy of the method.

In Section 4.4 we have introduced the Coupled model which further extends the Kersting model to the case of designs with a mixture of single and replicated observations. The introduction of finite sample size corrections to the variance estimator, in conjunction with the corresponding



(a) Coupled Model (b) Latent-Kernel (c) Quadratic Polynomial

(d) Std Coupled Model (e) Std Latent-Kernel (f) Std Quadratic Polynomial

Figure 4.4: Comparison of the Coupled, Latent-Kernel and Quadratic polynomial variance models. (a)-(b)-(c) combined plots of mean and two standard deviation prediction. (d)-(e)-(f) standard deviation prediction. Training set consists of using 200 point design with 4 replicate observations at each site. Dots are the empirical means of the samples. The black solid lines are the true function mean and standard deviation and the blue dashed lines the GP predictions.

uncertainty estimates allowed us to create coupled mean and variance GPs using only small numbers of replicates per design point. Through a set of simulation experiments discussed in Section 4.4.3, the inference speed up when using replicated observations was clearly demonstrated. In Chapter 5 we will develop a model-based design methodology that explicitly considers replicated observations.

The Coupled Model however is still too complex to allow for tractability in the design calculations described in Chapter 5. We have therefore presented a simpler class of variance models in Section 4.5 where a deterministic function explicitly models the variance. Unlike existing methods which are either not tractable (Kersting et al., 2007; Goldberg et al., 1998) or do not allow for a straightforward specification of the variance model (Snelson and Ghahramani, 2006; Walder et al., 2008), the Fixed Basis class of models allow for both tractability and convenient elicitation of prior beliefs about the simulator variance response. In the future, alternative parametrisations of the Latent-Kernel model (Section 4.5.3) can be investigated rather than fixing the variance kernel parameters or using a Latin design for the latent points.

In Section 4.5.4 we compared the Coupled and Fixed Basis models on a simple one-dimensional toy example. Using a dense training set the Coupled Model performed best. However we envisage that for sparser data sets the constrained nature of the Fixed Basis models will prove beneficial.

All models presented in this chapter can be combined with existing GP extensions such as online learning, dependent outputs, non-stationary covariance functions and sparse approximations. In the original paper Kersting et al. (2007) had also extended the heteroscedastic model to include a projected process approximation (Rasmussen and Williams, 2006) and more sophisticated sparsity approximations (e.g. Csato (2002); Snelson and Ghahramani (2005)) can easily be incorporated to help deal with larger data sets.

Our framework allows further analysis to be carried out in a straight-forward and efficient manner using the emulator as a proxy for the simulator. Furthermore the computer model parameter space can be explored without the necessity of a large number of (computationally demanding) simulator runs. In combination with a discrepancy model and real-world observations, these methods could facilitate the efficient statistical calibration of stochastic models.

5 Experimental Design for Parameter Estimation

CONTENTS

5.1 Introduction

Experimental design plays a crucial role in the building of an emulator (Sacks et al., 1989), where unlike data-driven learning we are able to choose the inputs at which the simulator is evaluated with almost complete freedom. The simulator is typically expensive to run, thus it is beneficial to optimise the input points at which the simulator is run given the available *a priori* knowledge. The GP emulator is then trained on the selected design set and corresponding simulator evaluations.

Section 5.2 reviews the current literature on experimental design with a particular focus on optimal design for parameter estimation. Our optimal design approach is presented in Section 5.3 which allows the calculation of optimal designs under heteroscedastic models with replicated observations. The Bayesian formulation of optimal design where the design criterion is integrated over the parameter prior is review in Section 5.4. The issue of optimisation in optimal design is discussed in Section 5.5. The properties of our design approach are examined through a series of simulation studies for Maximum Likelihood estimation in Section 5.6 and for Bayesian inference using Hybrid Monte Carlo in Section 5.7. Conclusions and directions for future research are discussed in Section 5.8.

5.2 Optimal Design For Parameter Estimation

In this section a brief overview of the theory of optimal design is presented. We begin with an overview of traditional optimal design theory (Section 5.2.1) followed by a discussion of the asymptotic results motivating the usage of the finite sample Fisher information in the presence of correlated errors in Section 5.2.2. A description of the issues that arise in the correlated error setup are given in Section 5.2.3. A discussion of extensions to the heteroscedastic, multiple objective and adaptive setups and a review of criticisms of optimal designs is given in Section 5.2.4.

5.2.1 Optimal Design For Linear Models

Optimal design theory was first developed for linear models where several analytical results are known on the properties of optimal designs. The relevant theory is reviewed and the most important theorems described. The theory described here is only applicable to linear models. Where an extension to a specific class of non-linear models has been proven, it is explicitly stated. In optimal design theory for linear models a function of the information matrix of the model parameters is optimised. Formally the design criterion is defined in terms of the Fisher Information Matrix (FIM), a $p \times p$ symmetric matrix, where p is the number of unknown parameters θ. The FIM for a design ξ is defined as:

$$\mathcal{F}(\xi,\theta) = -\int \left(\frac{\partial^2}{\partial\theta^2}\ln[L(X|\theta,\xi)]\right) L(X|\theta,\xi)\,\mathrm{d}X,$$

where $L(X|\theta,\xi)$ is the likelihood function. The most common design criteria are D-optimality where the negative log determinant of the FIM is minimised, i.e. $-\log|\mathcal{F}(\xi,\theta)|$ and A-optimality where the trace of the inverse is minimised, i.e. $\mathrm{tr}(\mathcal{F}(\xi,\theta)^{-1})$ (Atkinson and Donev, 1992).

A design ξ of size M is typically denoted as :

$$\xi = \left\{ \begin{array}{cccc} x_1, & x_2, & \cdots, & x_M \\ \omega_1, & \omega_2, & \cdots, & \omega_M \end{array} \right\}$$

where $x_1, x_2, \ldots, x_M$ are the M design points. Denote by $N = \sum_{i=1}^{M} N_i$ the total number of simulator evaluations. The weights associated with each design point $\omega_i = N_i/N$ usually represent or are proportional to the number of replicate observations. The weights ω_i can also be regarded as precision or duration of the measurements (Müller, 2005). When N_i and N are positive integers the design is known as exact but a large part of optimal design theory aims to approximate the discrete ξ with a continuous-in-ω design measure $\xi(x)$ such that the optimisation problem is reduced in complexity (Müller, 2005). The design in the latter case is termed approximate.

For approximate optimal designs, the General Equivalence Theorem (GET) can be used to check the optimum design is minimally supported, i.e. designs with fewer points are not optimal. Also for approximate designs, G-optimal design where the predictive variance is minimised is equivalent to D-optimality where we minimise the generalised variance of the parameter estimates or equivalently maximise the information gain of the parameter likelihood (Atkinson and Donev, 1992, p57).

Chaloner and Larntz (1989) derive a generalisation of the GET for concave design criteria for non-linear models and present simulation results on a logistic regression example for Bayesian D and A optimality. In Bayesian optimal design, the criterion function is the integral of the corresponding criterion function over a parameter prior, i.e. $\int \mathcal{F}(\xi,\theta)p(\theta)$. In their simulation experiments they demonstrate that as the prior becomes less informative, the number of support points required for a minimally supported optimal design grows without bound. This intuitive result was later proven formally by Braess and Dette (2007).

Formally the GET is defined below. If the design criterion is *concave* then a ϕ-optimal ξ^* design can be equivalently characterised by any of the three conditions:

1. ξ^* maximises $\phi(\xi)$.
2. ξ^* minimises $\sup_{x\in X} \mathrm{d}(\xi,x)$ where $\mathrm{d}(\xi,x)$ the directional derivative of $\phi(\xi)$ at ξ in the direction of x.

3. $\sup_{x \in X} \mathrm{d}(\xi^*, x) = 0$

where X is the candidate set. For the theorem to apply X must be a compact set, the derivatives for ϕ must exist and be continuous in x. The directional derivative $\mathrm{d}(\xi, x)$ is also known in the literature as the sensitivity function. The GET allows one to check the optimality of a given design by examining the sensitivity function.

The GET can be used to construct an efficient optimisation routine for optimal designs known as the exchange algorithm which proceeds sequentially by adding the point x that maximises the sensitivity function and removes the point that has the least impact on the sensitivity function (Müller, 2005).

For linear models the number of support points can be bounded due to additivity of information matrices to $p(p+1)/2$. Additivity does not hold for non-linear models so no bound is known. Bayesian or composite designs for linear models are not bounded since the criterion is a linear function of multiple information matrices (Atkinson and Donev, 1992, p165). The GET however still holds for the Bayesian design of linear models (Atkinson and Donev, 1992, p165).

Optimal designs for linear models place points on the edges of the design space (Fedorov, 1972). MacKay (1992) proves this results also holds for Bayesian optimal design of linear models with homescedastic noise. This can also hold for non-linear models where a Gaussian approximation used in the proof is valid, i.e. the second derivative of the interpolator function with respect to the parameters is neglected in the expansion of the interpolator around its most probable value. In particular MacKay (1992) finds that to obtain maximal information about the parameter posterior of the interpolant, the next observation should be sampled where the variance of the interpolant is largest. For many interpolators, including linear models, the variance is largest beyond the most extreme points where data has been gathered. This approach would therefore place the design points on the edges of the design space.

5.2.2 Justification for FIM under Correlated Errors

When considering correlated processes, the majority of the results described in the previous section do not apply. The theoretical basis on which the usage of Fisher information for design under correlated errors relies is described here.

In GP regression, a parametric covariance function is used to model the variance and correlation of the unknown function. The parameters of the covariance are usually estimated using Maximum Likelihood (ML) or sampling. By utilising asymptotic results of parameter estimators, useful approximations to finite sample properties can be constructed. Two asymptotic frameworks are used in the literature (Zhang and Zimmerman, 2005; Stein, 1999a):

- *Increasing domain.* The minimum distance δ between neighbouring points does not collapse to 0, i.e. $\delta > 0$, as the number of design points goes to infinity. The domain of the design space is unbounded.

- *Infill domain.* The design domain is bounded and as the number of points goes to infinity, $\delta \to 0$.

Zhang and Zimmerman (2005) have found that for certain consistently estimable parameters of exponential kernels with and without a nugget under ML estimation, approximations corresponding to these two asymptotical frameworks perform about equally well. A parameter is consistently estimable under a given estimator (e.g. ML) if the sequence of estimators converge in probability to the quantity being estimated as the sample size grows without bound. Mathematically, a sequence of estimators $t_n; n \geq 0$ is a consistent estimator of θ if and only if, for all $\varepsilon > 0$, we have:

$$\lim_{n\to\infty} \Pr\{|t_n - \theta|\} < \varepsilon\} = 1.$$

For parameters that are not consistently estimable however, the infill asymptotic framework is preferable. The finite sample Fisher information is found to be a compromise between the two frameworks.

Mardia and Marshall (1984) showed under increasing domain asymptotics that the Maximum Likelihood (ML) estimator $\hat{\theta}$ converges in probability to the true parameter θ, $\hat{\theta} \to N(\theta, I^{-1}(\theta))$ where $I(\theta)$ is the Fisher information matrix. Unfortunately no such general results exist under infill asymptotics. Abt and Welch (1998) show that under infill asymptotics for the triangular, exponential and Gaussian kernels, the variance of the ML estimator still asymptotically converges to the inverse of the Fisher information matrix. The Gaussian kernel result was demonstrated using simulation and not a formal proof.

Pázman (2007) provides justification of the FIM for small noise levels without using asymptotics in the proof but rather using a truncated function expansion which is only valid for very low process noise levels. No bound is given on how small the process variance has to be for the proof to be valid. Furthermore no nugget is included in the model.

An experimental justification for the use of the FIM under homoscedastic noise was given in Zhu and Stein (2005) where simulations from Matérn covariance-function based GPs were used to study whether the inverse Fisher information matrix is a reasonable approximation to the empirical covariance matrix of ML estimators.

5.2.3 Issues in Optimal Design for Correlated Processes

When non-linear models with non-concave design criteria are considered the GET and additivity of the information matrices do not hold. The focus of this thesis is on optimal design when a correlation structure is present and the aim of the design is the estimation of the parameters of the correlation function. Unfortunately the nice features of well established design theory, such as additivity of the information matrix and concavity of design criteria, do not carry over to this setting (Müller and Stehlík, 2009). In the particular case of computer experiments, Müller and Stehlík (2009) identify the following issues:

- Asymptotic unidentifiability. As Zhang and Zimmerman (2005) show the finite sample Fisher information matrix approximation breaks down when certain kernel parameters are not consistently identifiable under infill asymptotics.

- Non-replicability. In the field of computer experiments, replication in experimental design is not desirable in the case of deterministic simulators. Therefore, some authors (Bursztyn and Steinberg, 2006) use other approaches to design of computer experiments such as space-filling designs, discussed in Section 2.3. Another approach is via optimisation to consider only replication-free designs altough the resulting design is no longer optimal. When considering stochastic simulators as in this thesis, this is not an issue as replicated observations are informative.

- Non-additivity of the information matrix. The information from different design points cannot be separated as in standard theory.

- Choice of correlation structure and design robustness. For certain kernels such as the exponential, the optimal design without a nugget collapses to a single point which for other choices of kernel would carry little information. Thus misspecification of the covariance function can lead to very ineffective designs.

- Nugget effect. For certain kernels such as the exponential, the optimal design has been shown to collapse to a single point for two point designs. Such behaviour is avoided by the introduction of a nugget parameter.

5.2.4 Extensions and Criticisms of Optimal Design

Most of the literature on optimal experimental design assumes homoscedastic noise. Tack et al. (2002) examine optimal design under a fixed basis linear-in-the parameters model. Although stochastic processes are not considered, the variance model used is similar to the fixed basis model

utilised in this work. They follow a Bayesian approach to design and demonstrate that informative priors lead to more efficient designs.

In certain cases there may exist multiple objective functions which depend upon different information matrices. Compound optimal design provides a general approach of combining multiple such objective functions such as model discrimination (T-Optimality) and parameter estimation (A- or D-optimality) via a weighted average of their information matrices (McGree et al., 2008). Compound designs may also be used to generate designs with non-equal emphasis on the trend and covariance parameters (Müller and Stehlík, 2010).

Hybrid criteria that explicitly combine prediction and parameter estimation also have been developed (Zimmerman, 2006; Zhu and Stein, 2006). Zimmerman (2006) proposes local EK-optimality, a linear combination of the maximum predictive variance and a scalarisation of the covariance of the ML parameter estimate. While this criterion selects observations which reduce parameter uncertainty and predictive uncertainty given the current parameter, it does not take into account the effect of parameter uncertainty on prediction error (Krause et al., 2008). To address this issue, Zhu and Stein (2006) propose an amended criterion, which they term Estimation Adjusted, and derive an iterative algorithm which alternates between optimising the design for covariance estimation and spatial prediction. Krause et al. (2008) note that the hybrid design criterion is not a submodular function and no theoretical bounds are available on its optimisation. Optimisation issues are discussed further in Section 5.5.

Seeger (2008) presents a sequential adaptive optimal design method using a linear model with a sparsity prior on the model parameters. The expectation propagation (EP) algorithm is used for inference. For optimal design the information gain of the parameter posterior is used as the design criterion. Seeger (2008) does not contrast the proposed design to other designs such as space-filling (see Section 2.3).

In the case of GPs, Krause and Guestrin (2007) present an exploration-exploitation approach, where initially the parameter uncertainty is reduced followed by the near-optimal selection of observations where the parameters are assumed known. They derive a bound on the benefit of continuing the exploration phase which is used as a stopping rule to decide to switch to the exploitation phase. Intuitively, if the parameter posterior is highly peaked, very little benefit can be gained from further exploration steps as opposed to an *a priori* batch exploitation design.

Several papers criticise traditional optimal design for linear models as the points are placed on the boundary of the design region. O'Hagan and Kingman (1978) suggest such a design strategy is not robust to model error and specifying the design region can in practice be very difficult. To protect against model error, the authors argue the resulting design should cover the space well and use as many distinct points as possible in order to detect every form of deviation from the proposed

model.

MacKay (1992) states that placing points on the edges of the input space might be considered non-ideal behaviour because in fact the practitioner is not interested in the interpolant behaviour in those regions. He proposes a transductive approach where an entropy criterion is maximised with respect to future unsampled design locations. A transductive approach is also taken by Yu and Bi (2006) where the predictive variance of a linear model is minimised. The authors apply reproducing kernels to kernelise the criterion and address the scalability issue with an EM-like iterative two step optimisation algorithm. The non-linear kernel criterion is $\max_X \text{tr}[K_{VX}(K_{XX} + \mu I)^{-1}]K_{XV}]$ subject to $X \subset V, |X| = m$ where V is the candidate and X the selected set of size m, μ a positive regulariser and K the specified kernel matrix with kernel function $k(.,.)$. This criterion can be interpreted as maximising the trace of the test-train point correlation in the variance prediction of a GP.

5.3 Fisher information for Replicated Observations

In this section we derive the Fisher information for the Joint Likelihood model (Section 4.5).

In the case of multivariate normal distributions the FIM can be computed analytically. The parameters may also appear in the mean function of the prior GP but the focus of this work is on identifying covariance function parameters and we will assume the mean function parameters are known or of no interest. Therefore let $\mathbf{X}$ be distributed as $N(0, \Sigma(\theta))$, the j,pth element of the $\mathcal{F}$ is:

$$\mathcal{F}_N^{jp} = \frac{1}{2}\text{tr}\left(\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_j}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_p}\right), \tag{5.1}$$

where tr denotes the trace. For a proof see Pázman (2004).

The (j,p)th element of the FIM for parameters θ_j, θ_p of the heteroscedastic model with replicate observations is:

$$\boxed{\mathcal{F}^{jp} = \sum_{i=1}^{M} F_{si}^{jp} + F_N^{jp},} \tag{5.2}$$

where

- M is the number of design points.
- $F_{si}^{jp} = \frac{n_i - 1}{2}\frac{\partial f_{\sigma^2}}{\partial\theta_j}\frac{\partial f_{\sigma^2}}{\partial\theta_p}$ where n_i is the number of replicate observations at design point i and $\frac{\partial f_{\sigma^2}}{\partial\theta_j}$ the derivative of the variance model $f_{\sigma^2}(\theta)$ (Section 4.5) with respect to parameter θ_j. In the case of the fixed basis model (Section 4.5.2) $f_{\sigma^2}(x,\beta) = \exp(H(x)^T\beta)$ and $F_{si}^{jp} = \frac{1}{2}(n_i - 1)H(x_i)^T J_j H(x_i)^T J_p$ where J_j the zero vector with j^{th} element 1 and $H(x)$ the basis

function matrix.

F_{si}^{jp} reflects the contribution of the sample variance to the parameter uncertainty. If we examine the formula, $F_{si}^{jp} = 0$ unless both θ_j and θ_p are parameters of the variance model f and the number of replicates is at least 2, i.e. $n_i > 1$.

- $F_N^{jp} = \frac{1}{2}\text{tr}(\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_j}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_p})$ as defined in Equation (5.1).

A complete proof is given in Appendix A.5.

For illustrative purposes, the FIM for a fixed basis variance model is derived. The GP prior is $\text{N}(0, K + RP^{-1})$ and $R = \text{diag}(\exp(\beta x))$ the Log-Linear fixed basis variance model for a one-dimensional input space with zero nugget. Also $P = \text{diag}(n_i)$ the diagonal matrix of the number of replicate observations. The model specification is completed by specifying the kernel K with a single parameter, the length-scale λ. For this model, the FIM is:

$\downarrow \theta_i, \theta_j \rightarrow$	λ	β
λ	$\frac{1}{2}\text{tr}\left(\Sigma^{-1}\frac{\partial K}{\partial\lambda}\right)^2$	$\frac{1}{2}\text{tr}(\Sigma^{-1}\frac{\partial K}{\partial\lambda}\Sigma^{-1}\frac{\partial R}{\partial\beta}P^{-1})$
β	$\frac{1}{2}\text{tr}(\Sigma^{-1}\frac{\partial R}{\partial\beta}P^{-1}\Sigma^{-1}\frac{\partial K}{\partial\lambda})$	$\frac{1}{2}\text{tr}\left(\Sigma^{-1}\frac{\partial R}{\partial\beta}P^{-1}\right)^2 + \sum_{m=1}^{M}\frac{n_i-1}{2}\beta^2$

where $\Sigma = K + RP^{-1}$, $\frac{\partial R}{\partial\beta} = R \odot x$, and $\odot$ denotes the Hadamard element-wise matrix multiplication.

5.4 Bayesian Design

The calculation of the FIM is defined for a given parameter value vector, θ_0. If a point estimate for θ is used the design is termed locally optimal, in the sense that an optimal design is obtained for that specific parameter value θ_0. In practice θ will not be known *a priori* so a Bayesian approach is preferred. In full generality, a Bayesian design criterion (Chaloner and Verdinelli, 1995) is specified as:

$$U(\xi) = \int\int U(\theta,\xi,Z)\,p(\theta|Z,\xi)\,p(Z|\xi)\,d\theta\,dZ,$$

where ξ the proposed design, $p(\theta|Z,\xi)$ the parameter posterior, $p(Z|\xi) = \int p(Z|\theta,\xi)p(\theta)\,d\theta$ is the marginal distribution of the response data Z over the prior distribution of θ. The utility function $U(\theta,\xi,Z)$ is problem specific. When the design goal is the minimisation of parameter uncertainty, several authors (Chaloner and Verdinelli, 1995; Zhu and Stein, 2005) have proposed Shannon information as a useful utility.

Defining the utility as the Shannon information and utilising a Normal approximation to the parameter posterior $p(\theta|Z,S)$, Chaloner and Verdinelli (1995) arrive at the following design criterion:

$$U(\xi) = -\int \ln|\mathcal{F}(\xi,\theta)|\; p(\theta)\; d\theta \tag{5.3}$$

where $p(\theta)$ is the prior on the parameters and $|\mathcal{F}(\xi,\theta)|$ the determinant of the FIM given by Equation (5.2).

Intuitively, as the finite sample FIM approximates the asymptotic distribution of the ML estimator (Section 5.2.2), this criterion integrates the log variance of the ML estimates of the parameters over the prior distribution (Zhu and Stein, 2005).

The integral in (5.3) can be approximated using Monte Carlo techniques:

$$U(\xi) \approx -\frac{1}{N}\sum_{i=1}^{N} \ln|\mathcal{F}(\xi,\theta_i)| \tag{5.4}$$

for N samples from the prior $p(\theta)$. As in Zhu and Stein (2005), in the simulation studies (Section 5.6) a coarse uniform discrete prior $p(\theta)$ is used to speed up the evaluation of the design criterion.

5.5 Optimisation

To complete the specification of the experimental design algorithm the method of optimisation must be defined. The most commonly employed approach is to select a subset of points from a large candidate design set (Zhu and Stein, 2005). A complete enumeration of all possible designs quickly becomes infeasible as the number of candidate points increases. Various search strategies have been proposed in the literature to address this limitation. Some authors have suggested using a stochastic algorithm like simulated annealing with multiple restarts to guarantee robustness (Zhu and Stein, 2005) or random sampling where an information gain is estimated for each candidate point by averaging the design score over all searches in which this point was included (Xia et al., 2006). Computational aspects of the optimisation methods we have utilised are discussed in Section 5.5.1.

We have implemented a simulated annealing (SA) type algorithm (Dréo et al., 2003) which is described in Algorithm 5.1 and includes multiple restarts to ensure robustness. The algorithm parameters were set to the following values:

Parameter	Value
Degree of Parallelism	$d = 8$
Fitness function	$f_f(\mathbf{X}) = -\log\|\mathcal{F}\|$
Initial steps to determine temperature	$N_t = 200$
Maximum iteration count	$M = 5 \times 10^4$

The perturbation function used in SA is described in Algorithm 5.2. In step A of the SA algorithm

described in Algorithm 5.1, the design is perturbed in a continuous fashion and it is not until step B that the optimal design is matched to the candidate set. This discretisation process may significantly alter the continuous design depending on the coarseness of the candidate set. This approach was taken to ensure that the SA optimisation only generates designs that are subsets of the candidate set and may therefore be directly comparable to other optimisation schemes such as the greedy approach discussed below.

Greedy optimisation, described in Algorithm 5.3, is a sequential procedure where at each step the input point from a candidate set which maximises the score gain is included in the selected set. In Xia et al. (2006) the greedy approach is shown to be superior to simple stochastic optimisation schemes through a set of simulation experiments. We confirm this result, providing further experimental results supporting the effectiveness of the greedy approach in Section 5.6.3. Computational aspects for both the Greedy and Simulated annealing methods are discussed in Section 5.5.1.

If the score function is in fact a monotone submodular function, then the solution obtained via the greedy algorithm is guaranteed to be within constant factor of $1-1/e$ of the optimum solution (Krause et al., 2008). A submodular function must satisfy the "diminishing returns" condition where for sets $A \subseteq B \subseteq V$ and $y \in V \setminus B$ it holds that $F(A \cup y) - F(A) \geq F(B \cup y) - F(B)$.

In the machine learning area, the Fisher information has been used for active learning (Hoi et al., 2009) where a submodular function was found to be a good approximation to the FIM in the case of classification assuming small length-scales.

In the case of linear models under conditional suppressor freeness (Das and Kempe, 2008), Bayesian A-optimality is a submodular function (Krause et al., 2008). A variable X_j is a suppressor variable if it "suppresses" the correlation between some other X_i and the predictor variable Z, in the sense that X_i appears not correlated with Z but is much more correlated with Z once X_j has been sampled (Das and Kempe, 2008). As an example, Das and Kempe (2008) state that if variables X_i and Z are independent and $X_j = X_i + Z$ then X_j would be a suppressor variable. In most other cases of optimal design such as Bayesian D-optimal design Krause et al. (2008) have demonstrated that the corresponding objective functions are not submodular. Despite the lack of submodularity, the greedy optimisation method has performed well in our simulation experiments.

As Zimmerman (2006) notes, the Fisher information of stationary isotropic GPs is invariant to translations of the design so optimisation can be sped up by not considering designs that are equivalent. Such optimisation has been not implemented in our experiments.

One challenge with the sequential greedy optimisation method is initialisation. It is necessary to have at least two points to compute the Fisher score in Equation (5.4), with more providing better numerical stability. A potentially useful initialisation is to evaluate the Fisher score for all point pairs. Alternatively a space-filling design, such as the Latin Hyperube, could be used for

initialisation. In our simulation experiments, we initialise the algorithm by selecting the centroid point of the candidate set as the initial design point. This compromise appears to have little effect on the final designs found as shown in Section 5.6.3.

Algorithm 5.1 Simulated Annealing algorithm based on Dréo et al. (2003).

Simulated Annealing algorithm

Input: Candidate points $\mathbf{X}_C$, Target Design size p, degree of parallelism d, fitness function $f_f(\mathbf{X})$, perturbation function $f_p(x)$, initial steps to determine temperature N_t, maximum iteration count M. **Output**: Local optimum design $\mathbf{X}_O$.

I. *Initialisation*. Generate d Latin Hypercube designs to use as starting points. For each initial design use the SA algorithm below to find optimum the design $\mathbf{X}_i^C$.

1. Perform N_t random perturbations (Algorithm 5.2) and evaluate the average change in fitness, also known as energy, denoted by $< \Delta E >$.
2. Calculate the initial temperature $T_0 = \frac{-<\Delta E>}{log(0.5)}$.

A. *Generate Continuous Design* $\mathbf{X}_O^C$. Loop until one of the termination criteria is met.

1. Perform perturbation on current design and calculate energy change ΔE.
2. Metropolis Acceptance Rule: if $\Delta E \leq 0$ the perturbation is accepted. If $\Delta E > 0$ perturbation is accepted with probability $\exp(-\Delta E/T)$ where T is the current temperature.
3. Check termination conditions. If any are met proceed to step B.
   (a) Has the maximum number of iterations M been reached?
   (b) Has thermodynamic equilibrium been reached and is the system deemed frozen? In practical terms, Dréo et al. (2003) suggest if $12p$ perturbations accepted or $100p$ perturbations attempted the system may be deemed to have reached equilibrium. The system is deemed frozen if three temperature changes have been performed without any perturbations accepted.
4. If the system has reached equilibrium and is not frozen, the temperature is lowered according to the annealing schedule. We use a linear schedule $T_{k+1} = 0.9T_k$.

B. *Discretise Continuous Design*

1. Match optimum continuous design $\mathbf{X}_O^C$ to candidate set $\mathbf{X}_C$ by minimising the Euclidean distance of the optimum set to candidate points. Replicate points may be introduced in this process depending on the granularity of the candidate set and the clustering of the optimum design.

5.5.1 Computational Complexity

In the simulation experiments presented in this thesis two optimisation methods have been utilised to generate the Fisher-optimal design, the Greedy and Simulated Annealing algorithms described in Section 5.5. In this section we examine the computational complexity of each algorithm in terms of both the number of function evaluations of the Fisher criterion and the number of primitive arithmetic operations.

Algorithm 5.2 Perturbation function used in the Simulated Annealing algorithm.

Perturbation function

Input: Current design $\mathbf{X}_c$, current temperature T, maximum temperature T_M. **Output**: Perturbed design $\mathbf{X}_O$.

A. *Generate a random number* r in $\mathrm{U}[0,1]$. *If* $r > 0.5$ *use perturbation method* P_1, *else* P_2.
P_1. *Shift Single Point.*

1. Pick point $\mathbf{x}_c^i$ in design $\mathbf{X}_c$ to change at random.
2. Calculate range of shift dependant on temperature ratio T/T_M and shift $\mathbf{x}_c^i$ within the feasible region. At maximum temperature the entire design space is feasible. Specifically given the upper and lower bounds for each dimension $x_i \in [l_i, u_i]$, a random value is generated by

$$\mathbf{x}_c^i = \begin{cases} \mathbf{x}_c^i + (u_i - \mathbf{x}_c^i)\frac{T}{T_M} r_{\ldots D+1} + l_i & ,\mathrm{r}_1 > 0.5 \\ \mathbf{x}_c^i - (\mathbf{x}_c^i - l_i)\frac{T}{T_M} r_{\ldots D+1} + l_i & ,\mathrm{r}_1 \leq 0.5 \end{cases}$$

   where $r = \{r_1, r_{\ldots D+1}\}$ are $D+1$ samples from the uniform distribution $U(0,1)$, where D the dimensionality of $\mathbf{X}_c$.

P_2. *Replace Points.*

1. Calculate the number of points to replace dependant on the temperature ratio T/T_M. At maximum temperature all the points are replaced. Specifically the number of points replaced for a design size M is round$(M \times \frac{T}{T_M})$ where round denotes the integer rounding operation.
2. Replace the selected number of points with randomly generated points that may lie anywhere in the design domain.

Algorithm 5.3 Greedy optimisation for optimal design generation.

Greedy Algorithm

Input: Target design size p, design fitness function $f_f(\mathbf{X})$, Candidate set design $\mathbf{X}_C$ of size $\mathcal{C}$, Initial design $\mathbf{X}_I$ of size l_i. **Output**: Optimal design $\mathbf{X}_O$. **Internal state**: Current proposal design $\mathbf{X}_P$.

A. *Initialise current proposal design to passed in initial design,* $\mathbf{X}_P = \mathbf{X}_I$.
B. *Add to the current proposal design* X_O *the candidate set point which maximises the fitness function* $f_f(\mathbf{X})$ *unless the size of the proposal design has reached the target design size* p.

1. Append each point in the candidate set to the proposal design, i.e. $\mathbf{X}_P^{+i} = [X_P; X_C(i)] \forall i \in \{1, \ldots, \mathcal{C}\}$.
2. Evaluate the criterion function for each proposal, $f_f\left(\mathbf{X}_P^{+i}\right)$.
3. Permanently add the point the maximises the criterion to the current proposal design. Since replication is allowed, the selected point is *not* removed from the candidate set.

The Greedy method described in Algorithm 5.3 requires

$$\mathcal{N}_f = (p - l_i) \times \mathcal{C} \tag{5.5}$$

evaluations of the Fisher criterion function, where p the target design size, l_i the initial design size (see the discussion on Section 5.5 for how the Greedy algorithm is initialised) and $\mathcal{C}$ the size of the candidate design size. The dependence of the algorithmic complexity on the candidate design size is problematic since the candidate size is a discretisation of the design space which grows exponentially with the dimensionality of the space. This is known as the curse of dimensionality and effectively restricts the usage of the presented Greedy algorithm to low-dimensional problems.

Unlike the Greedy method, the Simulated Annealing (SA) optimisation method described in Algorithm 5.1 does not depend on the candidate design size or the target design size. Rather the worse-case complexity depends on the number of initial steps used to determine the temperature N_t and the maximum number of iterations allowed M. Therefore the worse-case number of evaluations of the fitness function is

$$\mathcal{N}_f = N_t + M. \tag{5.6}$$

The SA algorithm therefore does not suffer from the curse of dimensionality. However the settings of the algorithm have to be carefully tuned to the problem at hand as several authors have noted (Dréo et al., 2003). In particular. we have observed that the current maximum iteration limit of 5×10^4 may be too low for our synthetic examples, as it is reached in most of our simulations prior to any other convergence criterion being met. In addition, the linear annealing schedule (Step A.4 in Algorithm 5.1) used may also be too fast as the theoretical convergence guarantees of the algorithm apply only when a logarithmic schedule is used. As a result, in our experiments the SA algorithm on occasions finds a worse solution than the Greedy algorithm. Therefore, although the SA algorithm appears attractive for higher-dimensional design problems, care must be taken to check the parameter settings and convergence of the algorithm.

For the local design simulation experiments presented in Section 5.6.4, a design of size $p = 30$ is constructed using an initial design of one point (the centroid of the candidate set) and a candidate set of 1024 points. By Equation (5.5) therefore 29,696 evaluations of the Fisher criterion function are required. On a standard desktop this takes of the order of two minutes The SA algorithm using $M = 5 \times 10^4$ maximum iterations with $N_t = 200$ initial steps, requires 50,200 evaluations of the Fisher fitness function which on the same hardware takes of the order of 5 minutes of elapsed time. As the Greedy algorithm steps B.1 and B.2 may be run in parallel, the elapsed time may be reduced by using multiple cores.

The computational cost of both algorithms is increased substantially when the Bayesian Fisher

criterion (Equation 5.4) is used since the integration must proceed numerically. In this case, the total number of the local Fisher fitness function (Equation 5.2) evaluations has to be multiplied by the number of Monte Carlo samples used to compute the Bayesian criterion. In the Bayesian design experiments presented in Section 5.6.5, we utilise 16 Monte Carlo samples requiring a total of $50,200 \times 16 = 803,200$ evaluations of the local Fisher function for the SA design and $29,696 \times 16 = 475,136$ evaluations for the Greedy algorithm. We therefore conclude that a quick low cost approximation to the integration will be of great benefit to optimal design optimisation time and consequently to the applicability of the optimal design method we advocate.

The Fisher score criterion itself requires the evaluation of a $l_\theta \times l_\theta$ symmetric matrix where l_θ the number of model parameters. Each of the $\frac{l_\theta(l_\theta-1)}{2}$ elements of the Fisher matrix requires the calculation of Equation (5.1). The inversion of the $p \times p$ training data covariance matrix requires $O(p^3)$ operations for the worst-case scenario of single observations. If replicated design sites are considered, the matrix inversion reduces to the number of unique input training points. The inversion operation can be performed once for the entire Fisher matrix but the matrix multiplication $\Sigma^{-1} \times \frac{\partial \Sigma}{\partial \theta_j}$ has to be performed for each parameter separately requiring $O(l_\theta \times p^2)$ operations. Finally for each pair of parameters the final product requires a further matrix multiplication. Therefore the total computational cost of a Fisher criterion evaluation is:

$$C_{\mathcal{F}} = O(p^3) + O(l_\theta \times p^2) + O\left(\frac{l_\theta(l_\theta - 1)}{2} p^2\right). \tag{5.7}$$

The individual cost of a Fisher criterion evaluation (Equation (5.7)) may then be multiplied by the cost of the corresponding algorithm (Equations (5.5) or (5.6)) given above to obtain the total computational cost of the optimisation method.

Lastly we note that the computational cost of generating the other designs considered such as the Latin Hypercube and Grid designs (Section 5.6.4) is considered negligible since these designs are constructed using simple geometric criteria.

5.6 Simulation Experiments on Maximum Likelihood Estimators

In this section properties of Fisher-information optimised designs are investigated through a range of synthetic examples. The focus is on Maximum Likelihood estimation and Bayesian inference is tackled in the next section. Further, in all experiments except Section 5.6.8, the model used in design generation is the correct model, i.e. the same model is sampled from to generate observations, used in the Fisher criterion to generate the optimal design and perform inference.

In Section 5.6.2 we show the monotonicity of the Fisher information to the empirical parameter covariance of the kernel parameters under different noise regimes. The monotonicity is required

to demonstrate the validity of the Fisher score as a design criterion for parameter estimations.

In Section 5.6.3 a one-dimensional example is used to demonstrate the design optimisation problem as well as the effectiveness of the greedy optimisation approach. Profile likelihood plots are also presented to visualise the effect on the likelihood of a Fisher and a space-filling grid design. The effectiveness of Fisher designs for local design is demonstrated in Section 5.6.4 utilising a range of models of increasing variance complexity.

In Section 5.6.5 a study of Bayesian designs is presented where a coarse discrete prior is used for the kernel parameters. To better understand the differences between designs, the Bayesian optimal and Grid designs are examined using a single GP realisation in Section 5.6.6. The behaviour of Fisher-optimal designs for different design sizes is discussed in Section 5.6.7 and we conclude in Section 5.6.8 by examining the case of structural error, where the assumed model used in design generation is incorrect.

5.6.1 Experimental Methodology

The designs are assessed using two main attributes, prediction and parameter estimation. A GP with known parameters is sampled in order to be able to assess the quality of the ML parameter estimates.

In our simulation experiments the GP subsequently used for inference has an identical mean and covariance function specification to avoid introducing structural error in our results. This assumption is examined in more detail in Section 5.6.8.

In order to measure the accuracy of parameter estimation the relative RMSE (rRMSE) (Zhu and Stein, 2005) is used: $\sqrt{(\hat{\theta}-\theta_0)^2}/|\theta_0|$ where $\hat{\theta}$ is the ML point estimate and $|\theta_0|$ the absolute value of the true parameter. The rescaling by θ_0 ensures the rRMSEs for different parameters are comparable. To ensure robustness in the calculation of the rRMSE, the maximum likelihood optimisation is restarted five times from random initial conditions for all parameters. The solution with the highest likelihood is selected for subsequent validation. The initial value for the log length-scale parameter was set to $\mathcal{N}(-2, 0.01)$, i.e. a Normal distribution centred at -2 corresponding to a length-scale of ≈ 0.1. All other parameters were initialised to $\mathcal{N}(0,1)$.

Another measure of parameter estimation accuracy used is termed the log determinant of the ML estimates or in short the empirical parameter covariance. It is defined as the log determinant of the covariance of the ML estimates of all parameters across all realisations of the experiment under consideration. It is a measure of dispersion of the ML estimates and does not capture the error of the estimations with respect to the true parameters (which the rRMSE does). However the log determinant of the finite sample FIM should approximate the log determinant of the ML estimates (see the asymptotic theory discussion in Section 5.2.2) and the quality of this approximation is a

useful diagnostic for the performance of the Fisher-optimal design. Confidence intervals for the log determinant estimate are computed using the bootstrap algorithm described in Appendix B.1.

The measures used to assess predictive performance are the Mahalanobis error, the Dawid score and the RMSE (Section 2.5). The predictive performance of the model is measured using a random Latin Hypercube test set. Multiple realisations of the experiment are performed and the resulting validation and parameter accuracy metrics are plotted using a box whisker plot where the $\{0.05, 0.25, 0.5, 0.75, 0.95\}$ quantiles are plotted. Values beyond the .05 and 0.95 quantiles are not plotted.

Lastly, in our experiments the design space is set to $\mathbf{X} \in [0,1]^d$ where d the dimensionality of the space. In the simulations $d = 1$ or $d = 2$. To demonstrate the invariance of the properties of the FIM to the choice of kernel, two kernels are utilised in the simulation experiments, the exponential and Matérn with fixed order $\nu = 5/2$.

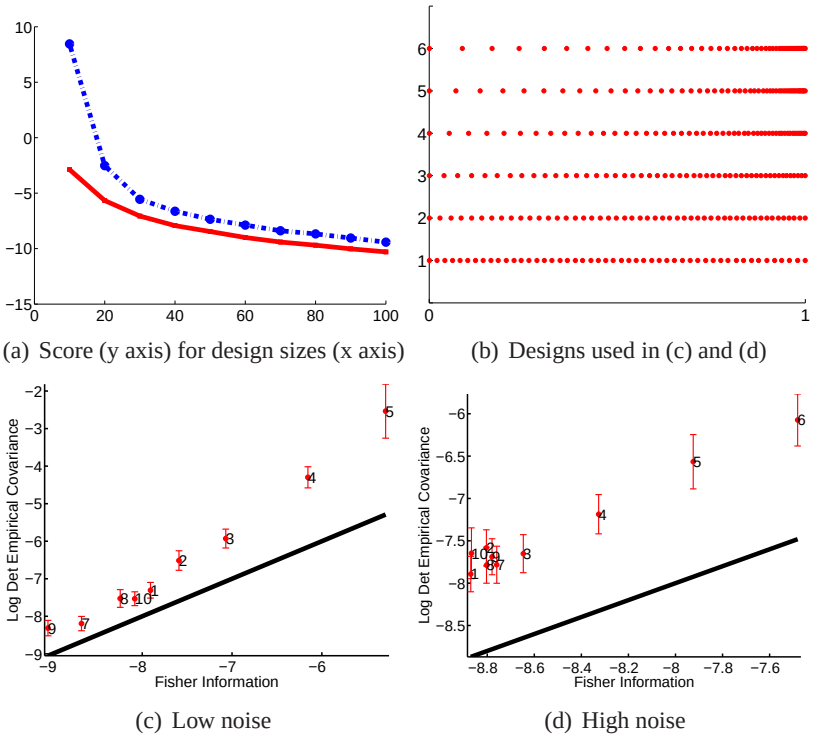
5.6.2 Monotonicity

In this section we show that under different signal-to-noise ratios the Fisher score remains monotonic to the log determinant of the empirical parameter covariance.

The Matérn covariance function is used for the mean, and a linear model for the log variance. The parameters of the generative GP were set to length-scale $\lambda = 0.5$, process variance $\sigma_p = 0.75$ and the variance model linear coefficients to $\beta_1 = 0.01$ and $\beta_2 = -30$ which correspond to a high noise level in the initial part of the design space quickly reducing to low noise. Finally the empirical parameter covariance was calculated using ML parameter estimates from 1000 realisations of the generative GP. The resulting approximation error is shown in Figure 5.1(a) for different design sizes where we observe that the inverse of the FIM provides a lower bound to the empirical parameter covariance and the bound becomes tighter as the number of design points grows.

The next experiment demonstrates the monotonicity of the Fisher information to the empirical parameter covariance. We generate six designs of 50 points with the distance between neighbouring points determined by the quantiles of exponential distributions with different rate parameters (Figure 5.1(b)). In addition three random designs and a Latin Hypercube design were also used.

In the low noise case, a Log-Linear basis variance model was used with the parameters of the GP set to the same values as in the previous experiment. For the high noise case a two-Gaussian basis RBF model was used. The basis functions were positioned at 0.33 and 0.66 in the one-dimensional design space with their variance set to the squared distance between their centres. The parameters were set to length-scale $\lambda = 0.33$, process variance $\sigma_p = 1.8$ and variance model coefficients $\beta_1 = -3.7$, $\beta_2 = -0.8$. Finally, we calculate confidence intervals for our estimates of the log determinant of the empirical parameter covariance using 1000 bootstrap samples (see



(a) Score (y axis) for design sizes (x axis)

(b) Designs used in (c) and (d)

(c) Low noise

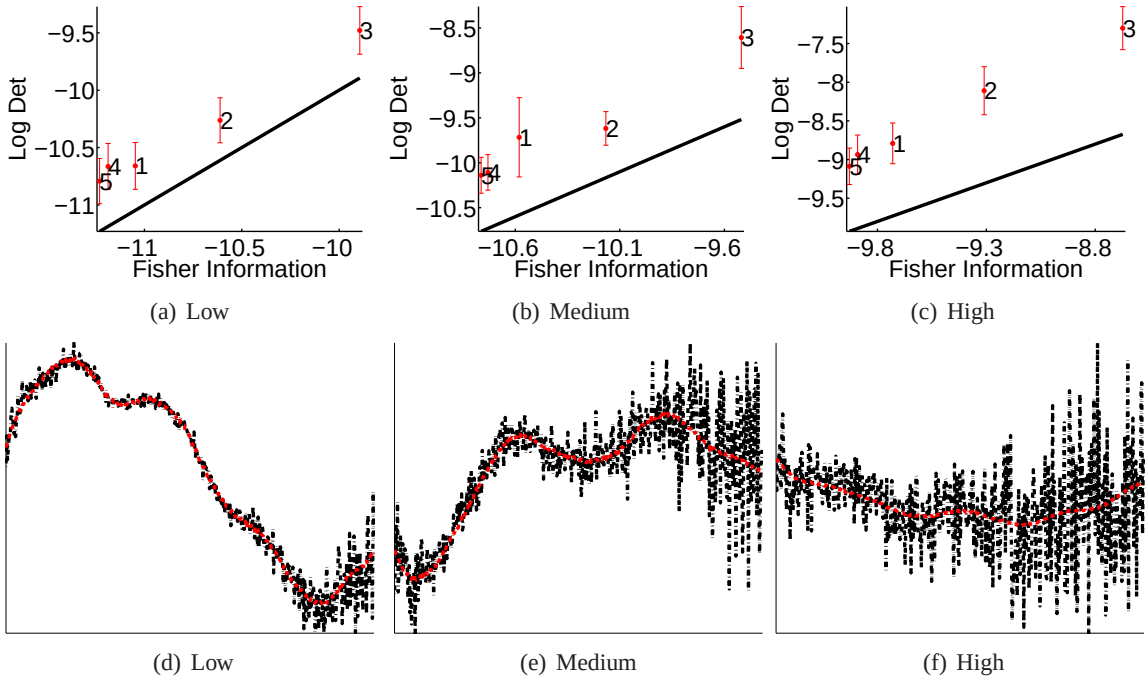
(d) High noise

Figure 5.1: Monotonicity experiment for Fisher Information: (a) The FIM (solid) and empirical parameter covariance (dashed) for designs of size 10 to 100. (b) The non-random designs used in (c) and (d). (c)-(d) Relation of log determinant of the Fisher information (x axis) to the log determinant of the empirical parameter covariance (y axis). The approximation for 50-point designs under different noise levels for a linear basis variance model (c) and an RBF variance model with two Gaussian basis functions centred at the 0.33 and 0.66 of the one-dimensional design space (d). Designs 7-9 are random and 10 is a uniform Latin Hypercube.

Appendix B.1). The results are shown in Figure 5.1(c)-(d) where we observe that for the higher noise level case the approximation error is larger but the monotonicity still holds.

We repeat this experiment on larger designs and varying signal-to-noise ratios. We use designs of 100 points where we sample from a GP with different levels of heteroscedastic noise. Two Gaussian basis functions were used with their centres and widths set as before. Samples from the GP for the different noise scenarios are shown in Figures 5.2(d)-(f). The length-scale and process variance of the Matérn covariance were unchanged. The linear coefficients for the variance model were set to $\beta_1 = -4.7$, $\beta_2 = -2.8$ for the low noise case, $\beta_1 = -3.7$, $\beta_2 = -0.8$ for the the medium noise case and $\beta_1 = -2.7$, $\beta_2 = 1.2$ for the high-noise case. We see in Figures 5.2(a)-(c) that although the approximation of the FIM to the parameter variance gets progressively worse as the noise level increases, the monotonicity holds even for relatively high noise levels.

The monotone relationship between the log determinant of the FIM and the log determinant of the empirical parameter covariance holds in all scenarios tested in this section and affirms the use of the FIM as a design criterion for minimising parameter uncertainty. This conclusion agrees with the findings of Zhu and Stein (2005) which showed this relationship in the homoscedastic case.



(a) Low (b) Medium (c) High

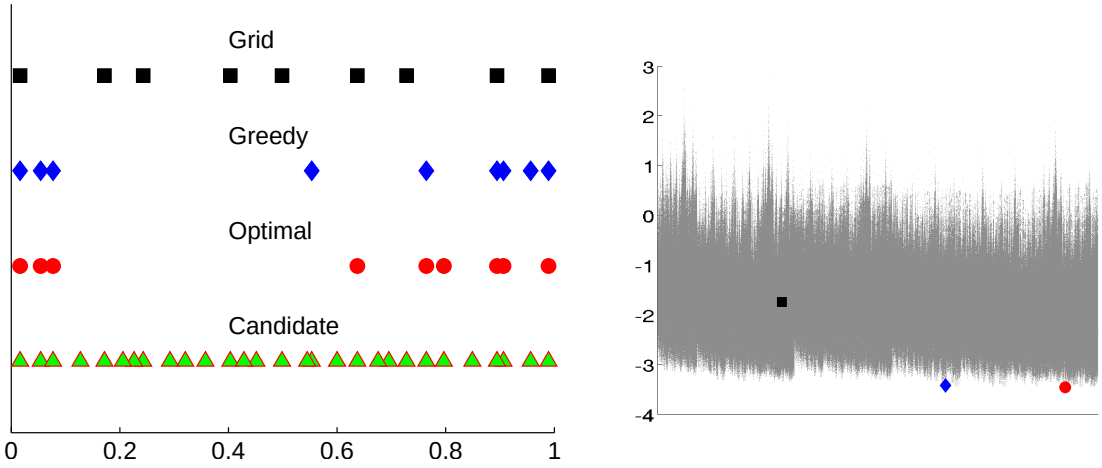
(d) Low (e) Medium (f) High

Figure 5.2: (a)-(c) Effect of noise on the monotonicity of the FIM (x axis) to the log determinant of the empirical parameter covariance (y axis). Designs 1-3 increasing distance designs (Figure 5.1(b)), 4 a Latin design and 5 is random. (d)-(f) Illustrative GP realisations for the various noise levels.

5.6.3 Complete Enumeration

In this simulation study the optimisation problem is examined in more detail. Using a simple one-dimensional example, the properties of the search space are explored through a complete enumeration of all possible solutions.

The experiment considers the selection of 9 locations from a candidate set of 29 points in a locally optimal design. The design is given the point parameter prior $\theta_0 = (\lambda = 0.5, \sigma_p = 0.7, \beta_1 = 0.1, \beta_2 = -10)$ and the FIM score of all $\binom{29}{9}$ combinations is computed. To reduce computational time we have not considered replicate observations in this experiment. The Matérn covariance and a Log-Linear basis function for the variance model is used. The Fisher scores for the solution obtained using greedy optimisation and an approximate grid design selected from the candidate set are also shown in Figure 5.3(b).



(a) Designs (b) FIM solutions

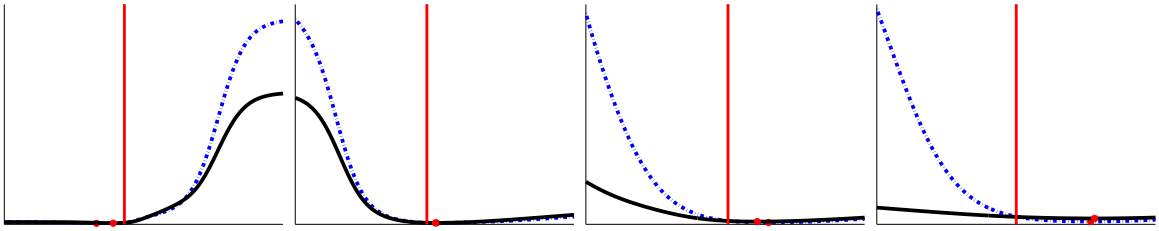
Figure 5.3: Complete enumeration of designs for a locally optimal design. Candidate Set (green triangle), optimal design (red circle), greedy design (blue diamond) and grid design (black square).

In terms of Fisher score, the greedy solution is very close to the optimum while the score for the grid design is significantly worse. Additionally, even for this simple example we notice a very large number of local optima close to the optimum demonstrating the near equivalence of a large number of designs.

The optimal, greedy and grid designs are shown in Figure 5.3(a) alongside the candidate set. The relatively long length-scale of the GP means the noise signal dominates and the optimal designs place the points near the boundaries due to the log-linear form of the variance function.

Since the motivation of using the Fisher information as a design criterion is to minimise parameter uncertainty, we expect the likelihood for the optimal designs be more informative about the optimum θ than the grid design. We demonstrate this effect by plotting the profile likelihood for each parameter (Figure 5.4) using a single GP sample as our training data. For all four param-

eters using only nine training points, the likelihood on the optimal design excludes larger portions of the parameter domain than the grid design.



(a) $\log \lambda$ (b) $\log \sigma_p$ (c) β_1 (d) β_2

Figure 5.4: Profile likelihoods for locally optimal design (dashed blue) and a grid design (solid black). The true parameter value is also shown (vertical red line).

5.6.4 Local Design

In this section Fisher information is applied directly as a design criterion using the true parameter values as used by the sampled GP. This is commonly referred to in the literature as locally optimum design. This approach cannot be used in real-world applications as the parameter values of the underlying process will not be known. However the approach allows us to examine some of the properties of optimal designs without the complexity of prior specifications for the covariance parameters.

The following six designs are utilised in the experiments. We also provide the acronyms that are used to reference these designs in the plots.

1. Greedy (*F*). We obtain the design using greedy optimisation. The algorithm is initialised by placing the first point in the centre of the design space.

2. Grid (*G*). A standard grid design where the distance between neighbouring points is a constant. If the design size is not a perfect square, the remainder points are placed randomly. For a design size of 30 points in two dimensional space for example (Figure 5.5(b)) a 25 point Grid is placed with the remainder 5 points placed randomly.

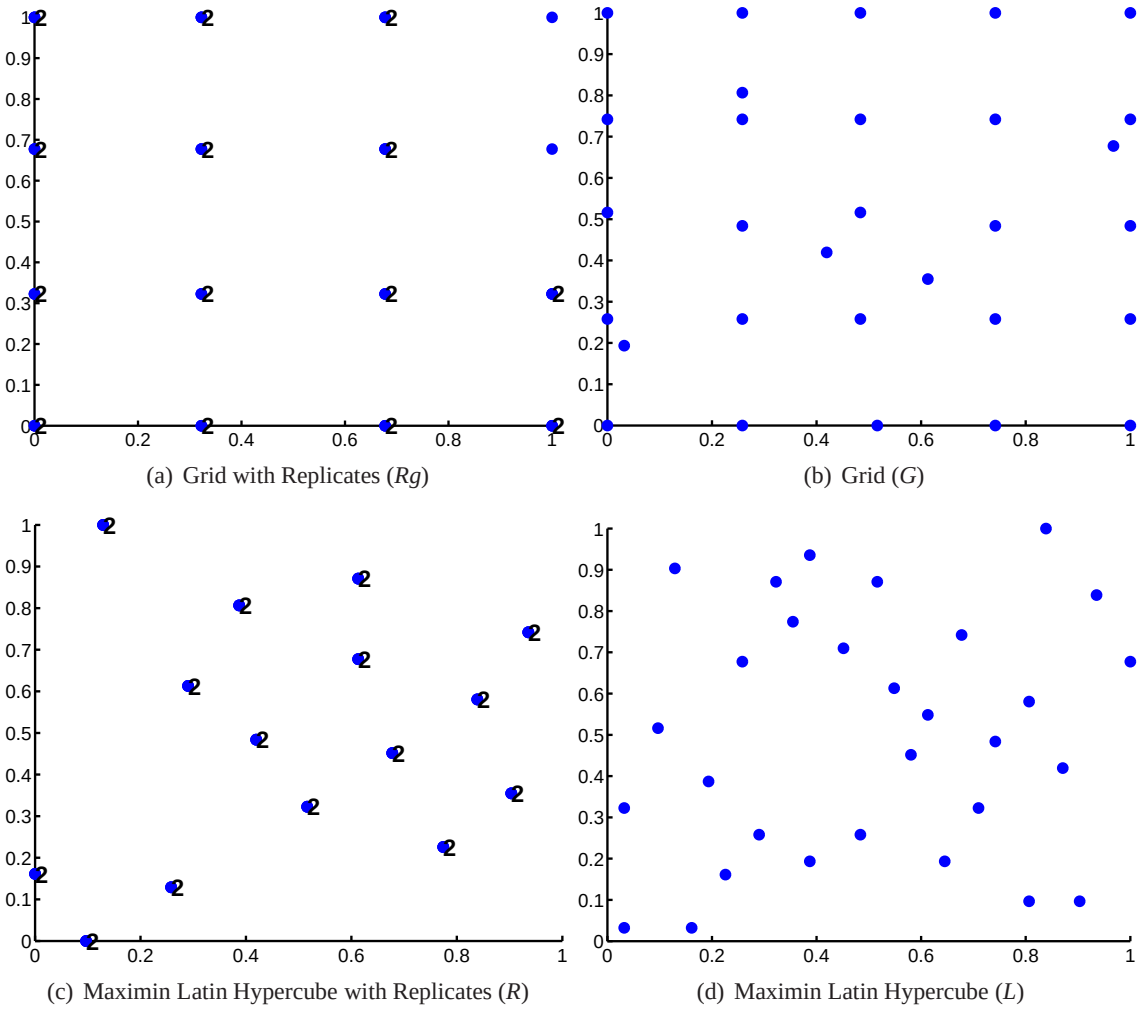
3. Replicated Grid (*Rg*). A standard grid design with two replicate observations at each point.

4. Maximin Latin Hypercube (*L*). Maximise the minimum Euclidean distance between nearest-neighbour points by selecting from 1000 randomly generated uniform Latin Hypercube designs.

5. Replicate Maximin Latin Hypercube (*R*). We use the same configuration as for the Maximin Latin Hypercube but two replicate observations are used at each design point.

6. Simulated Annealing (*S*). We generate a locally optimum design using the Simulated Annealing algorithm described in Section 5.5 to minimise the Fisher score.

Examples of the space filling designs (G,Rg,L,R) are shown in Figure 5.5.



(a) Grid with Replicates (*Rg*)

(b) Grid (*G*)

(c) Maximin Latin Hypercube with Replicates (*R*)

(d) Maximin Latin Hypercube (*L*)

Figure 5.5: Examples of space filling designs used in the simulation experiments. Numbers indicate replicated points.

All designs were generated using a 1024 grid space of candidate points by picking $n = 30$ points allowing for replication. A test set of 1024 points generated using a random uniform Latin Hypercube design is used to compute the Mahalanobis error and the Dawid score. The GP is sampled 500 times, with ML inference and validation performed independently for each realisation. The maximum likelihood optimisation is performed for five independent realisations with different initial conditions to ensure robustness in the solution obtained. The exact initialisation and experimental methodology are described in Section 5.6.1.

Three main cases are explored in our simulation study reflecting increasing levels of complexity in the variance model:

- Nugget: The variance response is constant across the design domain.

- Log-Linear: A fixed-basis linear polynomial model is used for the variance. Fixed basis variance models are described in Section 4.5.2.
- Latent-Kernel: A Latent-Kernel model is used for the variance. Latent-Kernel variance models are described in Section 4.5.3.

5.6.4.1 Nugget Model

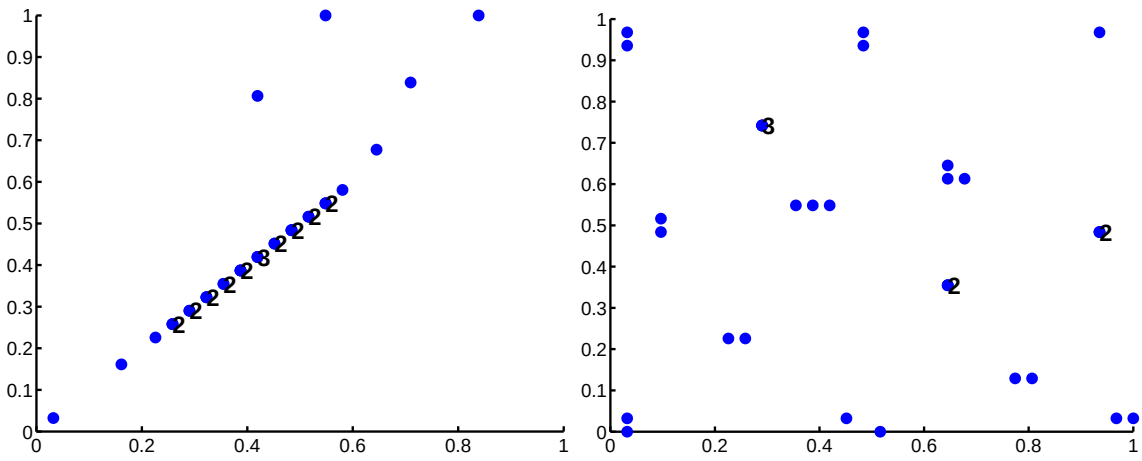
In this simulation experiment a Matérn kernel is used to model input correlations and a homoscedastic constant noise parameter, referred to as the nugget, is used to model the simulator at-a-point variance.

The hyperparameters for the sampled GP were set to $(\lambda, \sigma_p, \beta_1) = (0.1, 1, 0.01)$ corresponding to a short length-scale process with low noise. The resulting design are presented in Figure 5.6. The Greedy and Simulated Annealing (SA) optimisation approaches yield quite different designs, with the SA design covering the space more uniformly. These designs, as well as the replicate Grid and Maximin Latin Hypercube (LH) designs, achieve similar Fisher score (Figure 5.7) confirming the existence of multiple local optima in the search space. The empirical parameter covariance for all designs calculated using the ML point estimates, which the Fisher score approximates, shows a similar effect.

As observed by Zhu and Stein (2005), in the homoscedastic case as the noise level increases, the design becomes more clustered with more sample points per cluster. The experiment in this section corresponds to the smallest level of noise considered by Zhu and Stein (2005). Though the Greedy design appears highly clustered, there is good coverage of the space. In contrast, Zhu and Stein (2005) show for higher noise levels that the optimal designs include few clusters spread evenly in the design space with very small inter-point distances within a cluster. Therefore replicate observations become more prevalent as the noise level increases. The experiment in this section establishes a baseline from which to compare the impact of the heteroscedastic variance models on design.

The performance of the designs in terms of parameter estimation can be further investigated by examining the relative RMSEs and biases for each hyperparameter. These are shown in Figure 5.8 and summarised in Table 5.1. All the replicate designs identify the parameters more robustly, especially the nugget, both in terms of relative RMSE and bias. The replicate Grid, Grid and Latin designs appear less robust in the estimation of the length-scale parameter where significantly longer tails exist in the relative RMSE distribution compared to the other designs.

Lastly, the predictive performance of the designs is examined in terms of Mahalanobis error, Dawid score and root mean squared error (RMSE). The distribution of errors is shown in Figure



(a) Greedy (b) Simulated Annealing

Figure 5.6: Fisher designs obtained for the Nugget variance model using Greedy and Simulated Annealing optimisation methods under Matérn kernel.

Table 5.1: Relative Parameter RMSE for the Nugget model.

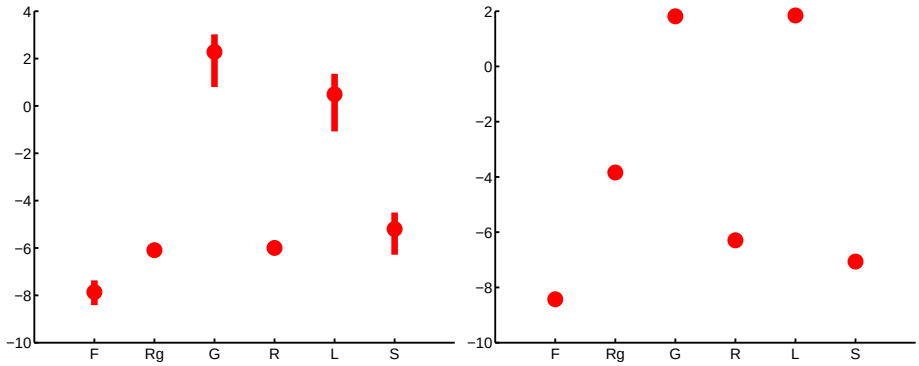
Design	λ	σ_p	β_1
Greedy	0.17 ± 0.14	0.17 ± 0.13	0.08 ± 0.06
Replicate Grid	0.64 ± 0.40	0.14 ± 0.10	0.07 ± 0.05
Grid	15.79 ± 160	0.19 ± 0.19	0.62 ± 0.29
Replicate Maximin LH	0.50 ± 0.34	0.15 ± 0.11	0.07 ± 0.05
Maximin Latin Hypercube	25.95 ± 407	0.19 ± 0.19	0.54 ± 0.27
Simulated Annealing	0.97 ± 14.48	0.15 ± 0.12	0.15 ± 0.13

5.9 and is summarised in Table 5.2. A Latin Hypercube test set of 1024 design points is used to validate each ML estimate realisation. Higher Mahalanobis error and Dawid score is observed for the non-replicate designs. In particular, the distribution of the Mahalanobis error and Dawid score for the Grid and Maximim Latin Hypercube non-replicate designs show significant tails reflecting the lack of robustness in parameter estimation achieved through these designs.

In terms of the RMSE (Figure 5.9(c)), the space filling non-replicate designs achieve somewhat smaller errors compared to the replicate designs. As the former cover the space more fully we expect an overall smaller interpolation error on the mean. This effect is observed later in the heteroscedastic scenarios as well. We conclude therefore that the lower Dawid and Mahalanobis errors for the replicate designs stem from a more accurate prediction of the model covariance which is expected from the lower parameter estimation errors, especially with regards to the nugget parameter.

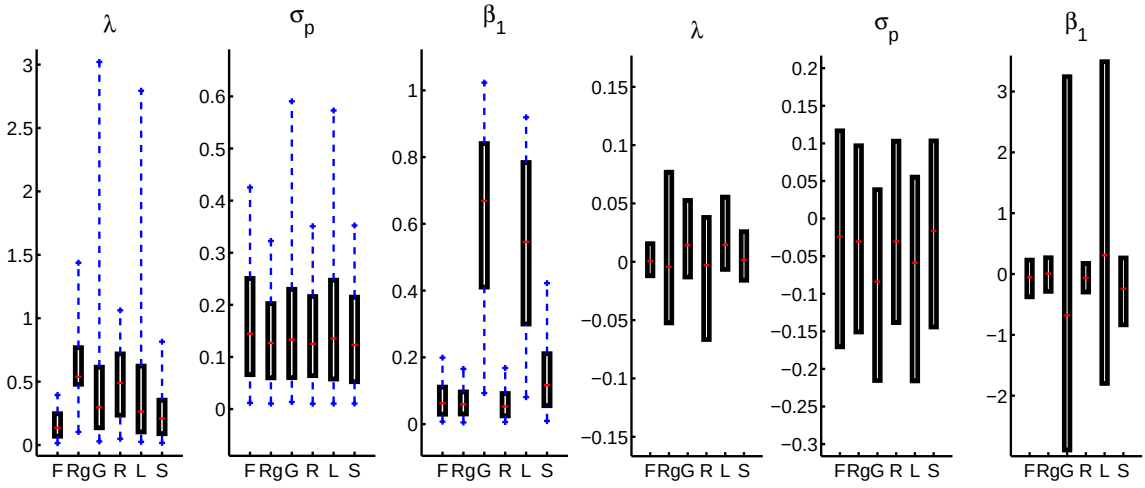
5.6.4.2 Log-Linear Model

We now use a fixed-basis Log-Linear model on the two dimensional input space of the form $\exp(\beta_1 + \beta_2 x_1 + \beta_3 x_2)$. The hyperparameters of the sample GP were set to $(\lambda, \sigma_p, \beta_1, \beta_2, \beta_3) =$



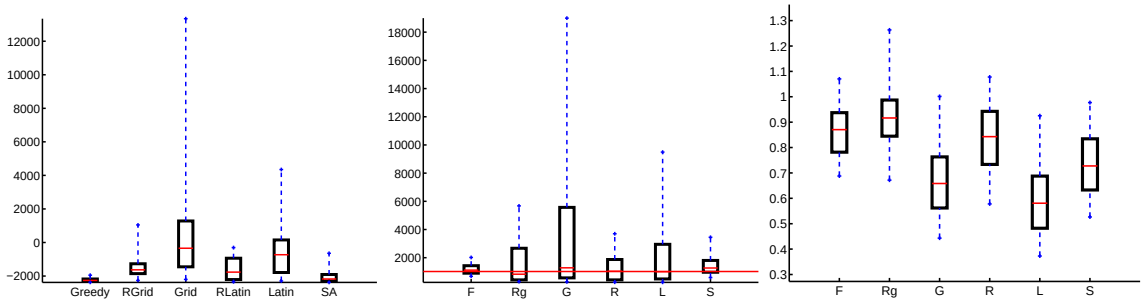
(a) Log Determinant (b) FIM

Figure 5.7: Log Determinant and Fisher Scores for all designs using the Nugget model.



(a) rRMSE (b) Parameter Bias

Figure 5.8: Relative RMSE and parameter bias for the Nugget variance model. Designs compared at the (F)isher design obtained through Greedy optimisation, Replicate Grid (Rg), (G)rid, Maximin Latin Hypercube with Replicates (R), Maximin Latin Hypercube (L) and Fisher design obtained through Simulated Annealing optimisation (S).



(a) Dawid (b) Mahalanobis (c) RMSE

Figure 5.9: Validation performance in terms of Dawid score and Mahalanobis errors using 1024 test points in a Latin Hypercube design and RMSE for the Nugget model.

Table 5.2: Mean and standard deviation of Mahalanobis (1024), Dawid score and RMSE.

Design	Mahalanobis	Dawid	RMSE
Greedy	1208 ± 420	-2220 ± 217	0.86 ± 0.11
Replicate Grid	1796 ± 1908	-1274 ± 1202	0.93 ± 0.18
Grid	4664 ± 7508	1626 ± 6313	0.67 ± 0.16
Replicate Maximin LH	1350 ± 1176	-1537 ± 777	0.84 ± 0.15
Maximin Latin Hypercube	2501 ± 3620	-241 ± 2703	0.60 ± 0.16
Simulated Annealing	1561 ± 1027	-1951 ± 677	0.74 ± 0.14

$(0.2, 1, -4.6, -1.6, -1.6)$. The standard deviation on the input domain using this model is illustrated in Figure 5.10. The noise level at $(0, 0)$ of the design space is exactly the nugget value used in the previous section. The GP prior mean is zero and the sample mean is not shown for brevity.

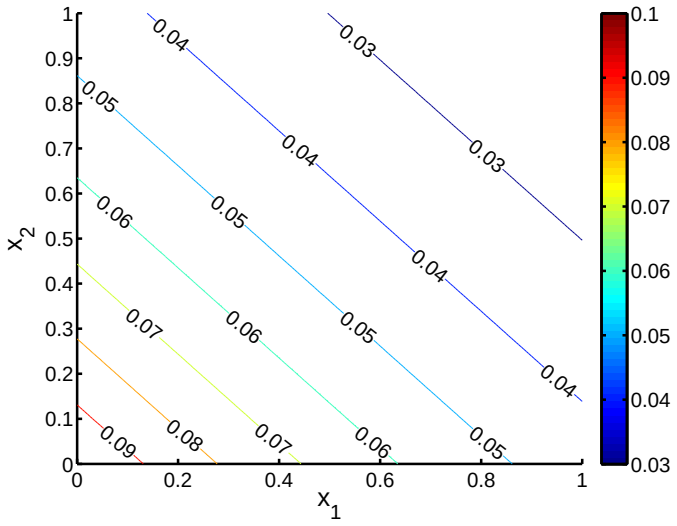
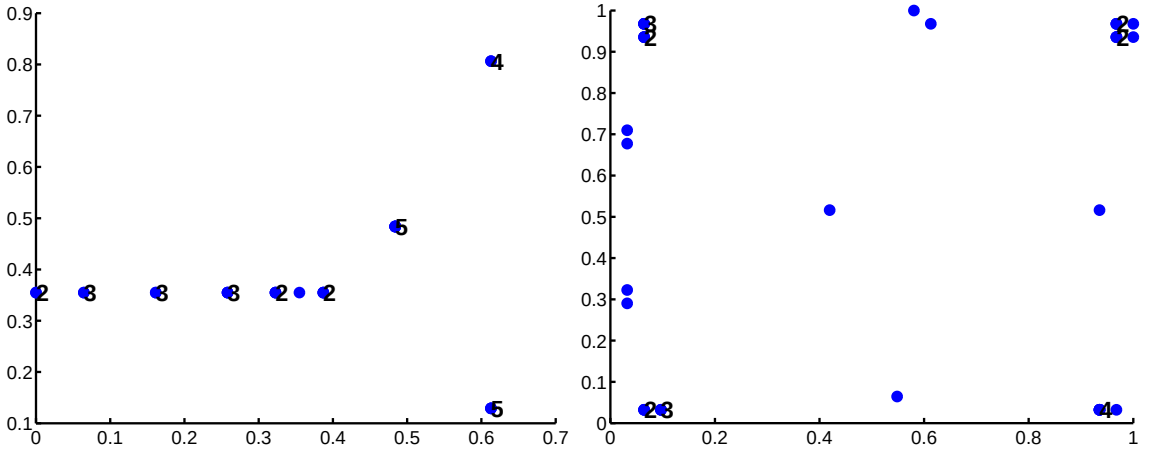


Figure 5.10: Standard Deviation of the Log-Linear model.

The resulting designs using the Greedy and Simulated Annealing optimisation methods are shown in Figure 5.11 where as in the previous section, the designs are quite different in terms of domain coverage but achieve similar Fisher scores (Figure 5.12(b)). The log determinant of the empirical parameter covariance (Figure 5.12(a)) agrees broadly with the Fisher score with regards to separating the poorly performing, in terms of parameter estimation, non-replicate designs from the replicate designs.

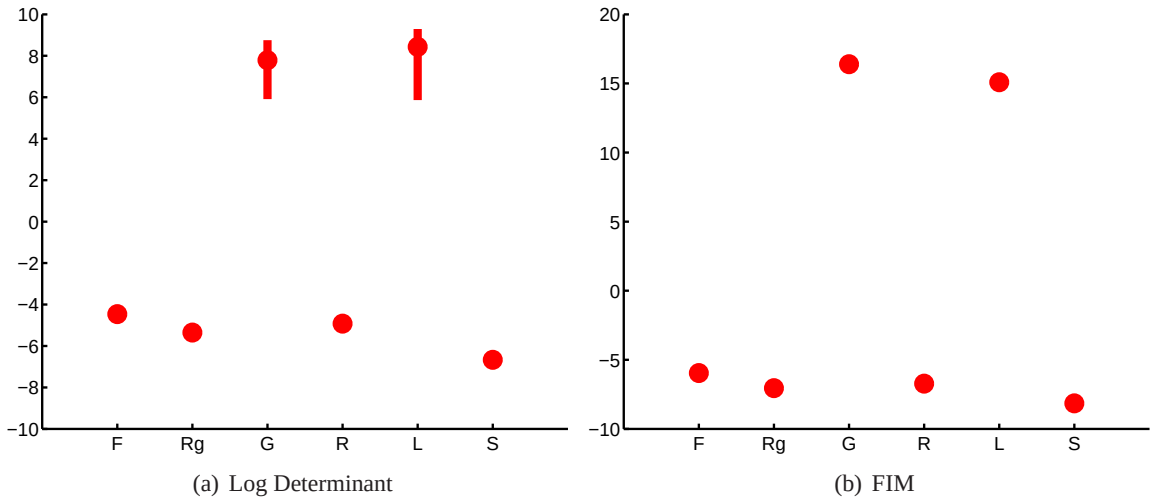
In terms of parameter estimation accuracy (Figure 5.13), all variance parameters β are better identified in the replicate designs in terms of relative RMSE. In terms of the length-scale and process variance parameters, all designs achieve similar errors. In this scenario the replicate designs are therefore superior in identifying the variance model parameters without sacrificing the estimation of the other parameters.

The predictive validation results (Figure 5.14 and Table 5.3) again show a high Mahalanobis



(a) Greedy (b) Simulated Annealing

Figure 5.11: Fisher designs for the Log-Linear model.



(a) Log Determinant (b) FIM

Figure 5.12: Log Determinant and Fisher score for the Log-Linear model.

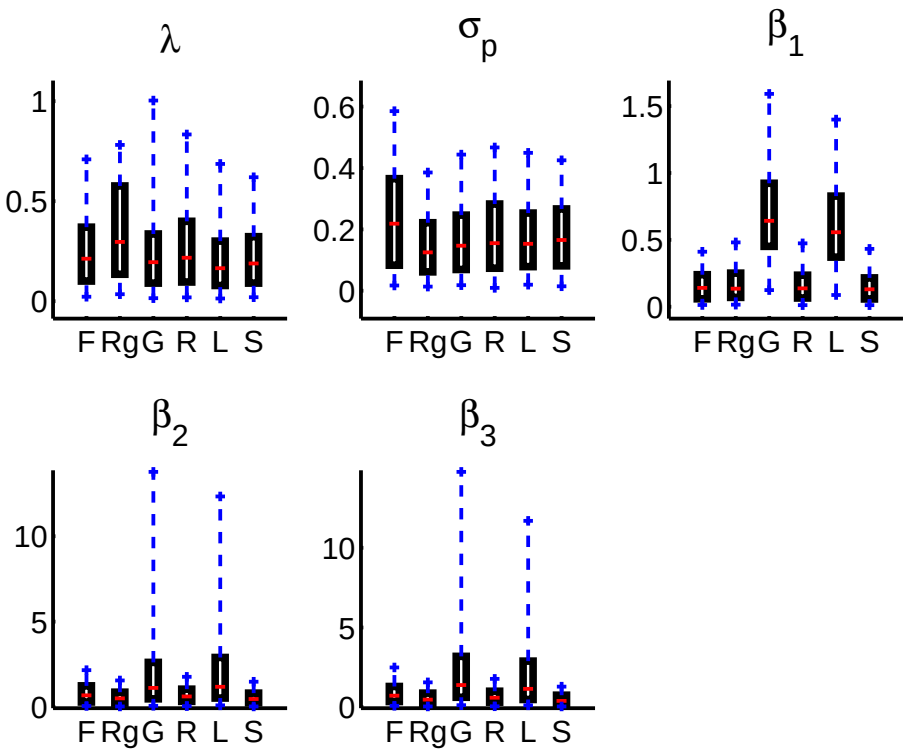


Figure 5.13: Relative RMSE of the ML parameter estimates for the Log-Linear model.

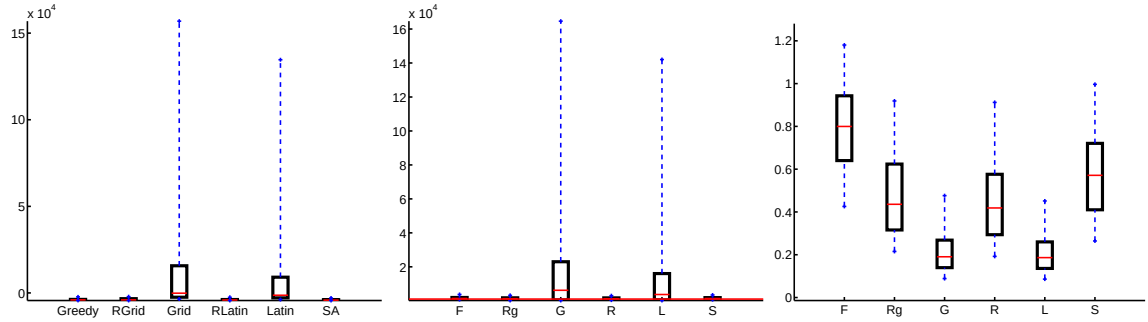
error and Dawid score for the non-replicate designs. The errors are higher than when using a nugget model since the variance response is more complex and the effect of the misidentification of the variance coefficients is thus more pronounced. In terms of mean prediction, we observe a lower RMSE for the non-replicate designs which cover the design space more uniformly, especially when compared to the Greedy and Simulated Annealing designs which are highly clustered, and thus achieve a larger interpolation error. We therefore conclude that as in the Nugget model case, the non-replicate designs have higher Mahalanobis error and Dawid score mainly due to inaccurate variance prediction. In the approach of Krause and Guestrin (2007), discussed in Section 5.2.4, the Fisher design can be considered during the exploration phase to minimise the parameter uncertainty and can be followed by an exploitation phase to minimise the interpolation error.

Comparing the optimal homoscedastic designs (Figure 5.6) to the heteroscedastic designs for the Log-Linear model, the latter place emphasis on the edges of the design space. This is especially evident for the SA design, which achieves the best Fisher score, where only a single point is placed in the interior of the design space. For the homoscedastic optimal designs however, the majority of points are placed in the interior of the design space in regularly spaced clusters. The difference arises due to the nature of the variance model in each case. In the homoscedastic case, the input location where the nugget is sampled is immaterial and the magnitude of the noise level as well as the choice of covariance function and length-scale dictate the placement of the points in the optimal design. In addition to these considerations, for the heteroscedastic Log-Linear model

a design that is optimal for the identification of the coefficients of the linear variance model is required. As is well known in the case of linear regression (Atkinson and Donev, 1992), the optimal design for parameter estimation places points on the corners of the space and this is exactly the effect we observe in the optimal designs for the Log-Linear model. The parameter estimation errors lend further credence to this conclusion as the optimal designs achieve lower errors for the variance model parameters β than the non-replicate space-filling designs while the length-scale and process variance parameters are identified with the same accuracy across all designs. The good performance of the replicate space-filling designs is also explained by this effect since replicated design points are placed on the edges of the design space. As the noise level is quite low across the design space, design points with just two replicated observations are sufficient to capture the variance response. In the case of the non-replicate space-filling designs however, the single observation design points on the edge of the space are not as informative with regards to the variance process.

Table 5.3: Mean and standard deviation of Mahalanobis (1024), Dawid score and RMSE for the Log-Linear model.

Design	Mahalanobis	Dawid	RMSE
Greedy	1749 ± 1898	-3690 ± 1711	0.80 ± 0.24
Replicate Grid	1398 ± 1033	-3652 ± 776	0.49 ± 0.23
Grid	32863 ± 79388	27460 ± 78524	0.23 ± 0.14
Replicate Maximin LH	1420 ± 872	-3848 ± 697	0.46 ± 0.22
Maximin Latin Hypercube	30058 ± 88207	24627 ± 87575	0.22 ± 0.15
Simulated Annealing	1612 ± 949	-3941 ± 584	0.59 ± 0.23



(a) Dawid (b) Mahalanobis (c) RMSE

Figure 5.14: Validation results for the Log-Linear model.

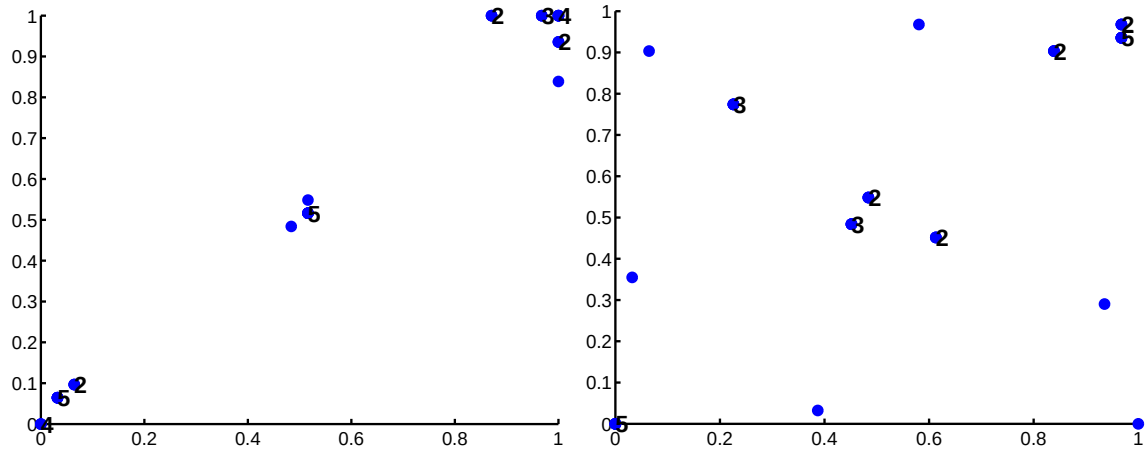
5.6.4.3 Latent-Kernel Model

We conclude the set of simulation experiments for local design by examining the performance of Fisher-based designs under a Latent-Kernel variance model (Section 4.5.3). The variance response is more complex than in the previous experiments as shown in the standard deviation plot in Figure

5.6.4.3. A modified isotropic Gaussian kernel is used for the variance model where the length-scale and process variance parameters are fixed to one. The only free parameters in the Latent Kernel variance model are the linear coefficients $\mathbf{z}$.

The hyperparameters of the sampled GP are set to $(\lambda, \sigma_p, z_1, z_2, z_3) = (0.5, 1, -1.1, -3.8, 4.2)$ and the latent points are set to $X_z = \{(0,0)(1,1)(0.5,0.5)\}$, corresponding to two corners and the mid point of the design space. Finally the exponential kernel is used to model input correlations.

The designs obtained through optimisation of the Fisher score are shown in Figure 5.15. The Simulated Annealing design covers the design space more uniformly while the Greedy design is highly clustered on the corners and mid-point of the design space. Both designs place clusters of points around the latent points X_z of the variance model which we interpret as the most informative locations to learn the parameters of the variance model.

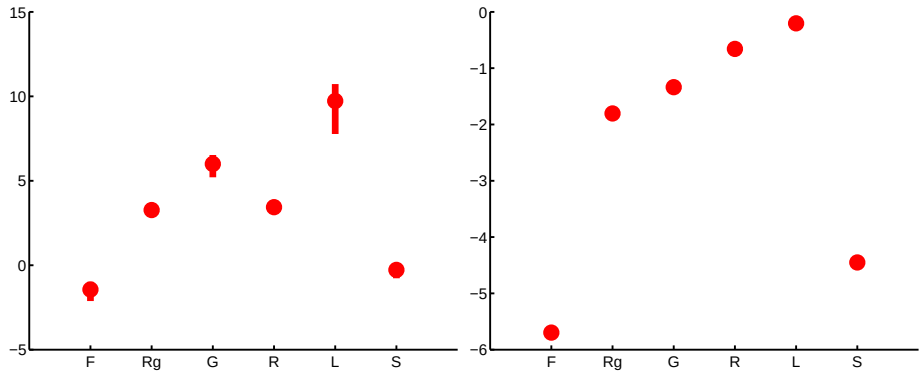


(a) Greedy (b) SA

Figure 5.15: Fisher designs for the Latent-Kernel model.

The Fisher score and corresponding log determinant of the parameter covariance are shown in Figure 5.16. In terms of both Fisher score and the empirical parameter covariance both Fisher-optimised designs are considerably better than the space-filling designs. In this case the extra complexity of the Latent-Kernel variance model separates the model based designs from the replicate space filling designs. Further the Fisher score of the Greedy design is smaller than for the Simulated Annealing design, suggesting more optimisation effort is required in the Simulated Annealing algorithm for this problem. The monotonicity of the FIM to the parameter covariance is violated however in this case with large approximation errors apparent. The log determinant of the parameter covariance for these designs also has the largest errors bars signifying difficulty in estimating parameter uncertainty.

Examining the errors of individual hyperparameters (Figure 5.17 and Table 5.4) we observe the Fisher-based designs achieve smaller relative RMSE particularly for the variance model parameters z_1 and z_2. The z_3 parameter, corresponding to the mid latent point $[0.5, 0.5]$, is identified by all



(a) Log Determinant (b) FIM

Figure 5.16: Log Determinant and FIM for Latent-Kernel model.

designs while the other variance model parameters are only identified by the replicate designs. We believe this is because the effect of the z_3 parameter dominates the variance response while z_1 and z_2 which are related to the corners of the design space have less impact on the variance response across the design space. Additionally the space-filling designs are informative about the mid-point of the design space as opposed to the corners of the space in contrast to the Fisher-optimised design which explicitly emphasize the corners of the space corresponding to z_1 and z_2.

Table 5.4: Relative Parameter RMSE for the Latent-Kernel model.

Design	λ	σ_p	z_1	z_2	z_3
Greedy	1625 ± 16610	0.36 ± 0.25	0.43 ± 0.35	0.13 ± 0.10	0.11 ± 0.09
Replicate Grid	3574 ± 22580	0.49 ± 0.34	0.89 ± 0.75	0.39 ± 0.34	0.09 ± 0.07
Grid	5517 ± 31676	0.53 ± 0.34	1.34 ± 1.86	0.56 ± 0.88	0.08 ± 0.06
Replicate Maximin LH	1580 ± 6128	0.54 ± 0.43	0.96 ± 0.80	0.42 ± 0.37	0.08 ± 0.07
Maximin Latin Hypercube	2057 ± 9172	0.57 ± 0.55	1.67 ± 2.82	0.67 ± 0.94	0.10 ± 0.31
Simulated Annealing	5980 ± 37893	0.42 ± 0.35	0.53 ± 0.44	0.14 ± 0.12	0.08 ± 0.07

We also note the very large errors in identifying the length-scale parameter for all the designs considered. Examination of the profile likelihoods under the Grid and SA designs (Figure 5.18) for one realisation of the experiment allows for a clearer understanding of the issue. The profile likelihoods were constructed by setting the other model parameters to their ML values although as the other parameters are identified with high precision, the profile is not altered if the true values are used instead. In the case of both designs, the issue is not one of an incorrect optimisation as the true length-scale value does not lie on a minimum. In the case of the Grid design where the rRMSE is very high, the multiple restarts avoid the usage of a low likelihood local minimum corresponding to a very small length-scale value. The ML solution however is very far from the true value and the likelihood is flat in the region of very large length-scales effectively signifying that the training design is not informative and cannot exclude a large range of possible large values. We conclude that due to the complexity of the model, the small training size used (30 points) is

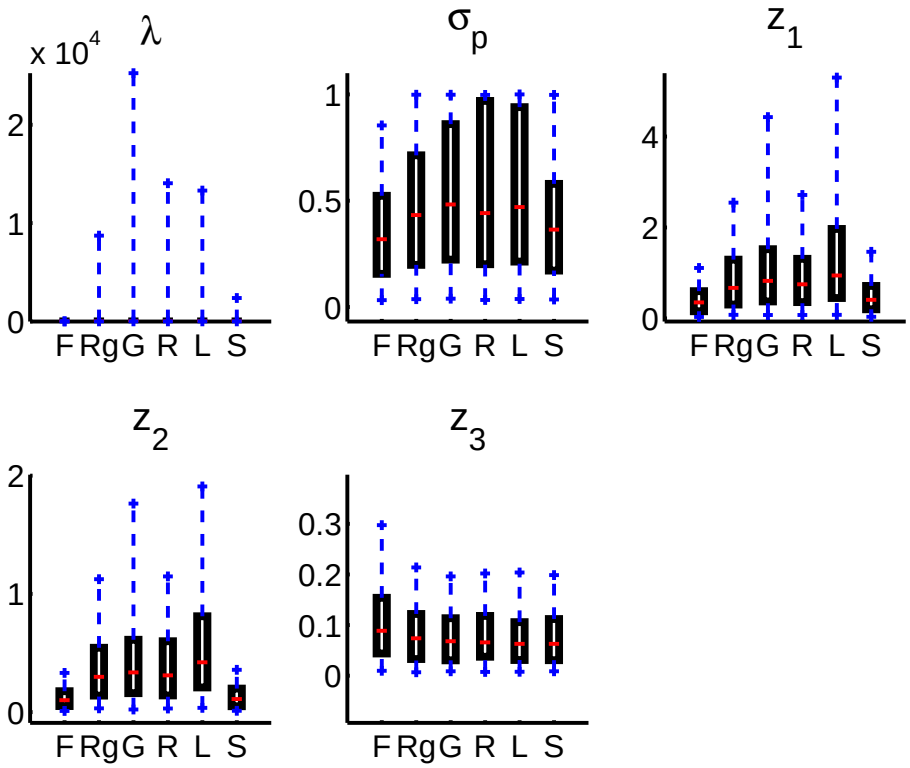
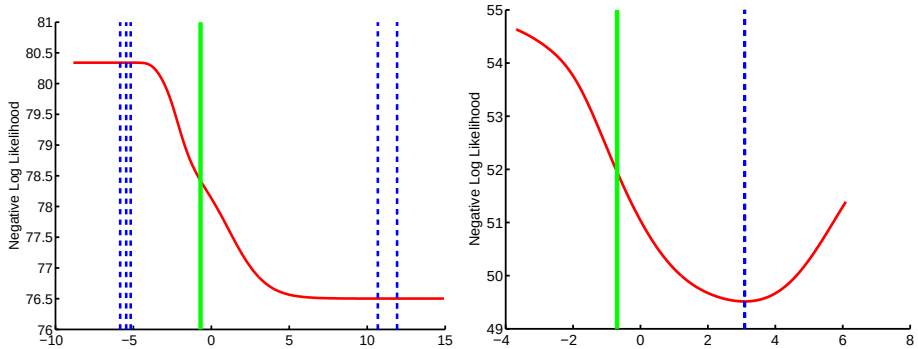


Figure 5.17: Relative RMSE for the Latent-Kernel model.

unable to identify the length-scale of the process.



(a) Grid (b) SA

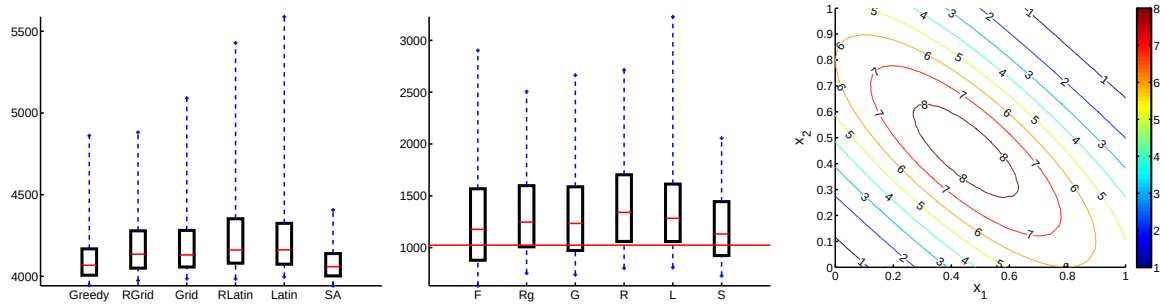
Figure 5.18: Profile likelihood for the length-scale parameter under the Latent-Kernel model. Solid green line is true value, dashed blue is maximum likelihood value under 5 multiple restarts with different initial values. For this realisation the rRMSE is 299,397 for the Grid design and 43 for the SA design. The x-axis denotes the log length-scale value.

Examing the validation errors (Figure 5.19(b), Table 5.5) we note the differences in the metrics are quite small. The RMSE is nearly identical for all designs reflecting the same level of accuracy in mean prediction. The SA design achieves the lowest error in terms of both the Mahalanobis error and Dawid score reflecting a more robust covariance estimation. Although the Greedy design has the lowest parameter estimation error, the highly clustered nature of the design results in the model extrapolating in large areas of the design space and hence incurring higher errors compared to the

more space-filling SA design.

Table 5.5: Local Design Evaluation for the Latent-Kernel model: Mean and standard deviation of Mahalanobis (1024) and RMSE.

Design	Mahalanobis	Dawid	RMSE
Greedy	1361 ± 731	4170 ± 340	1.01 ± 0.01
Replicate Grid	1384 ± 586	4228 ± 352	1.01 ± 0.02
Grid	1549 ± 2010	4405 ± 1819	1.01 ± 0.02
Replicate Maximin LH	1551 ± 1078	4395 ± 992	1.01 ± 0.02
Maximin Latin Hypercube	1551 ± 1165	4397 ± 982	1.01 ± 0.02
Simulated Annealing	1258 ± 556	4119 ± 277	1.01 ± 0.02



(a) Dawid Score (b) Mahalanobis Score (c) Standard Deviation.

Figure 5.19: Validation results and Standard Deviation for the Latent-Kernel model.

5.6.5 Bayesian Design

In the simulation results presented thus far only locally optimum designs have been considered. However such designs cannot be used in practice as they require knowledge of the true parameter values to compute the Fisher score. In this section Bayesian Fisher designs are examined where the resulting design is a compromise across a range of locally optimum designs.

Following Zhu and Stein (2005) a discrete uniform prior is used for ease of computation and interpretation. The designs are computed as in the previous section, the only difference being the criterion function is the Fisher score numerically integrated over the discrete prior (Equation (5.4)).

The evaluation of the designs is performed across all permutations of values of the discrete prior. For each permutation of discrete prior values, the set of designs is evaluated as in the local design case. The simulation experiment in this section can therefore be considered as a set of local simulation experiments where the sample GP parameters are fixed. All evaluation metrics such as the Mahalanobis error and parameter accuracy are plotted across all prior permutations, that is the errors from different prior combinations are plotted jointly in the same figure. The log determinant of the empirical parameter covariance is computed per prior permutation.

We have examined two scenarios:

- *Log-Linear*: A Log-Linear variance model with a Matérn covariance function. The discrete prior was set to $\lambda = \{0.1, 0.5\}$, $\sigma_p = 1$ and $\beta_1 = \{-1.1, -0.5\}$, $\beta_2 = \{-3.8, -0.2\}$, $\beta_3 = \{4.2, 1.2\}$.

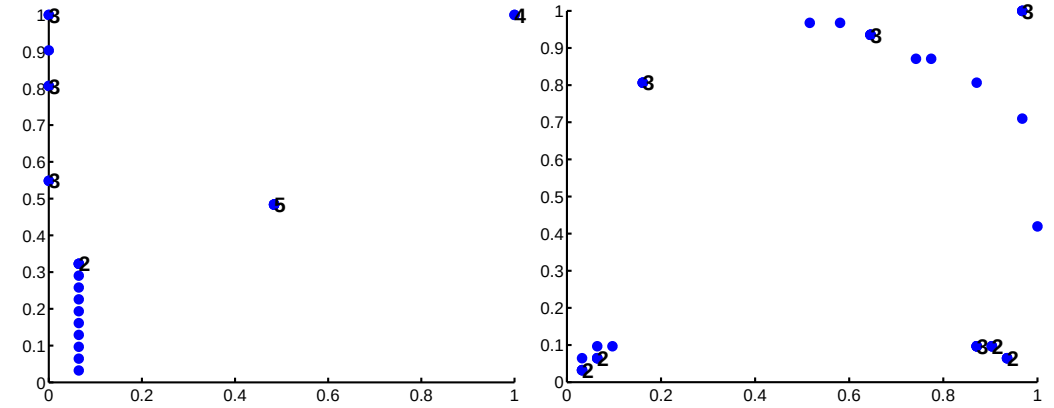
- *Latent-Kernel*: A three-parameter Latent-Kernel model with an exponential covariance function to model input correlations. As in the previous section, the latent points were set to $X_z = \{(0,0),(1,1),(0.5,0.5)\}$, The discrete prior was set to $\lambda = \{0.2, 0.6\}$, $\sigma_p = 1$ and $z_1 = \{0.01, 0.1\}$, $z_2 = \{0.01, 0.2\}$, $z_3 = \{0.01, 0.2\}$.

For both scenarios, the discrete prior has 16 possible permutations. Each prior permutation is treated as a local design and evaluated 30 times, given a total of 480 realisations of the experiment.

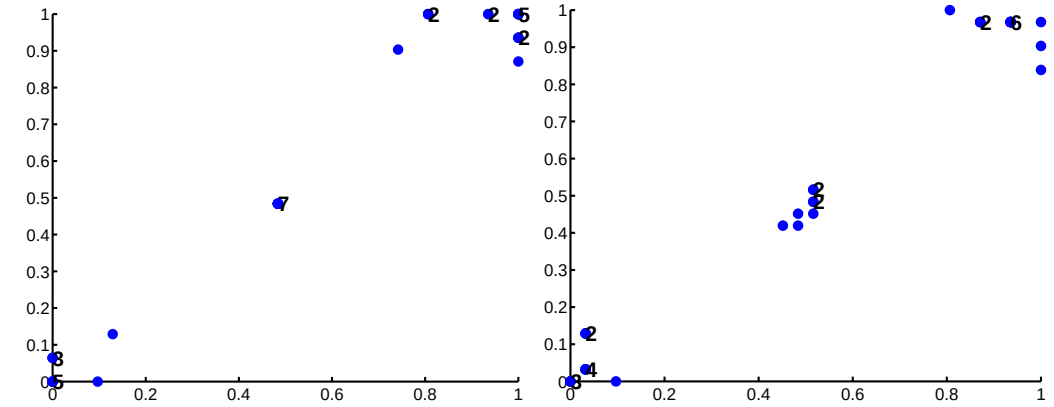
The Fisher-optimal designs are contrasted against the same type of space-filling designs that have been used previously (Figure 5.5). The Fisher based designs obtained through Greedy and Simulated Annealing optimisation are shown in Figure 5.20. The Latent-Kernel optimal designs are clustered around the latent points X_z in both the Greedy and Simulated Annealing cases unlike the local design case discussed in Section 5.6.4.3 where the SA design was more space-filling than the Greedy design. We believe this is due to the consideration of a small length-scale in the scenario examined here whereas for the local design a relatively long length-scale of 0.5 was used. For the Log-Linear model on the other hand, the designs obtained are quite similar to those obtained for the local design case in Section 5.6.4.2 with the Greedy design placing points on a ridge pattern while the SA algorithm results in points being placed on the edges of the design space. Although for both the Log-Linear and Latent-Kernel models the Greedy and SA designs are geometrically quite different, they achieve similar Fisher scores and parameter estimation errors (Figure 5.23) demonstrating the near equivalence of the solutions.

The predictive performance for both models in terms of Mahalanobis error, Dawid score and RMSE is shown in Figure 5.21. For the Log-Linear model the non-replicate designs are not as robust over the wide range of prior values as the replicate designs. In particular we see very large errors in terms of Mahalanobis and Dawid score whilst the RMSE is smallest for these designs. As in the local design experiment (Section 5.6.4.2), the interpolation performance of the space-filling design is superior but in terms of the covariance prediction very large errors are incurred.

For the Latent-Kernel model on the other hand, the RMSE performance is nearly identical across all designs. The covariance performance is also quite similar as reflected by the Mahalanobis error and Dawid score, although we note longer tails in the errors for the non-replicate designs. These results are in agreement with the local design experiment discussed in Section 5.6.4.3.

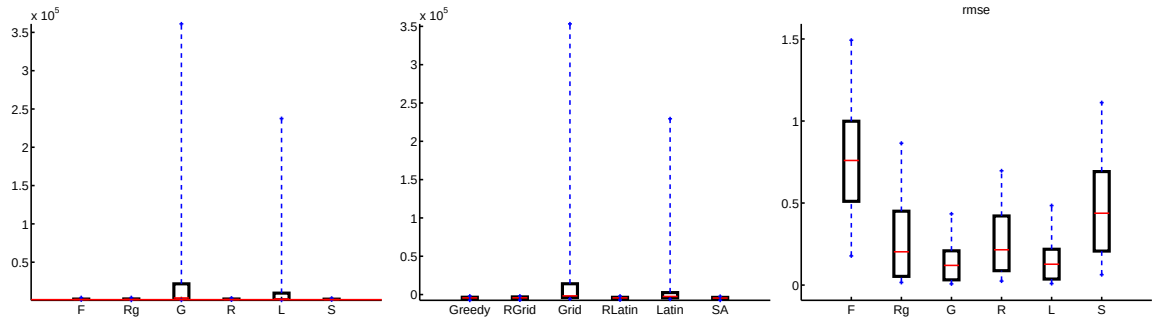


(a) Log-Linear Greedy (b) Log-Linear Simulated Annealing

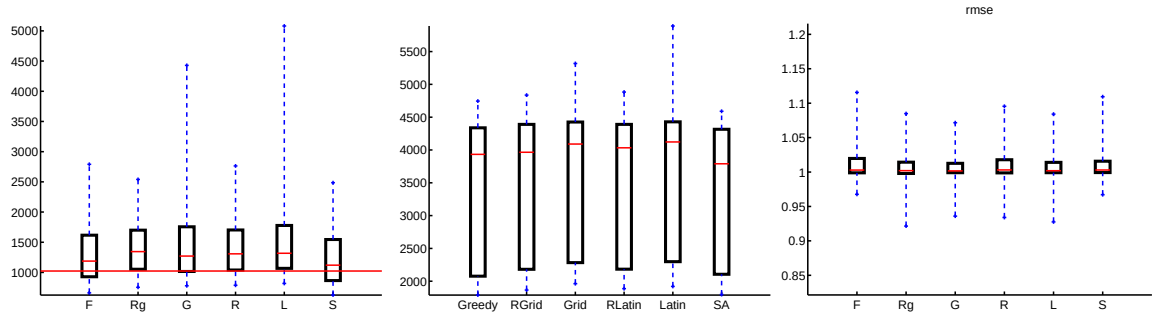


(c) Latent-Kernel Greedy (d) Latent-Kernel Simulated Annealing

Figure 5.20: Fisher-based Bayesian Designs used in the simulation experiments.



(a) Log-Linear: Mahalanobis (b) Log-Linear: Dawid (c) Log-Linear: RMSE



(d) Latent-Kernel: Mahalanobis (e) Latent-Kernel: Dawid (f) Latent-Kernel: RMSE

Figure 5.21: Mahalanobis error, Dawid score and RMSE for the Bayesian Log-Linear and Latent-Kernel models. 1024 test points were used.

For the Latent-Kernel model, the large differences among the different designs become apparent when the parameter accuracy is examined (Figure 5.22). As in the local design case, the variance parameters are better identified in the Fisher-optimised designs with clear differences apparent in the performance of the replicate and Fisher-optimised designs. As in the local design case, the high magnitude of the errors in all designs of the length-scale parameter implies it is not identifiable for this model using such a small training size. For the Log-Linear model, the variance process parameters are better identified in the replicate designs as expected in agreement with the local design experiment.

The log determinant of the empirical parameter covariance and the Fisher score for all designs are shown in Figure 5.23. For the Log-Linear model, the Fisher score is similar for all replicate designs. In the Latent-Kernel model both Fisher designs have the lowest Fisher score which is also reflected in the empirical parameter covariance. As in the local design experiments, we therefore see broad correspondence between the Fisher score and empirical parameter covariance for both models.

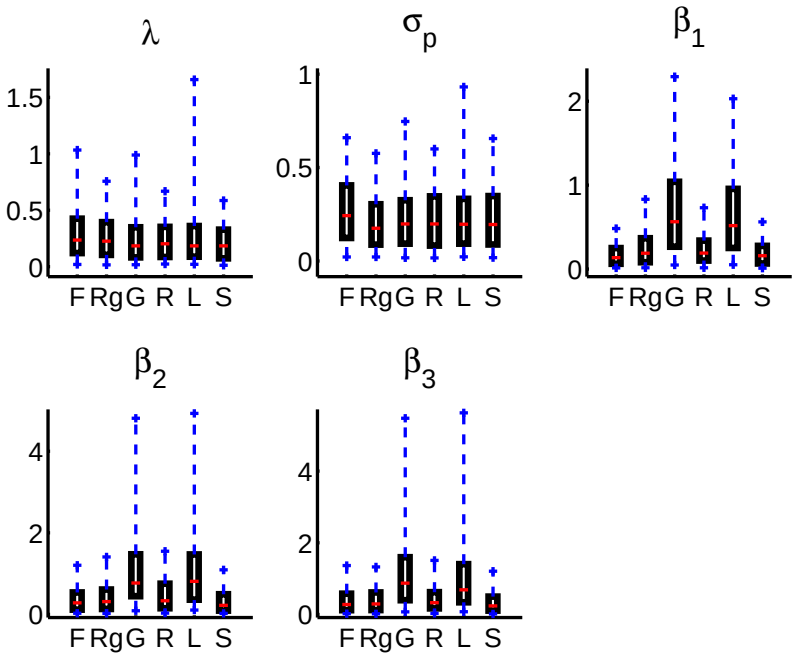
Overall both in terms of validation and parameter accuracy we see significant benefits when using replicate designs. For the more complex Latent-Kernel variance model, the Fisher designs are more differentiated in terms of parameter error performance from the space-filling replicate designs. Due to the higher complexity of the variance response in the Latent-Kernel model, space-filling replicate designs are no longer local optima and the optimisation of the Fisher score is justified. In this section therefore the conclusions drawn from the local design experiments presented in Section 5.6.4 have been generalised to the Bayesian design setting.

5.6.6 Specific Case Example

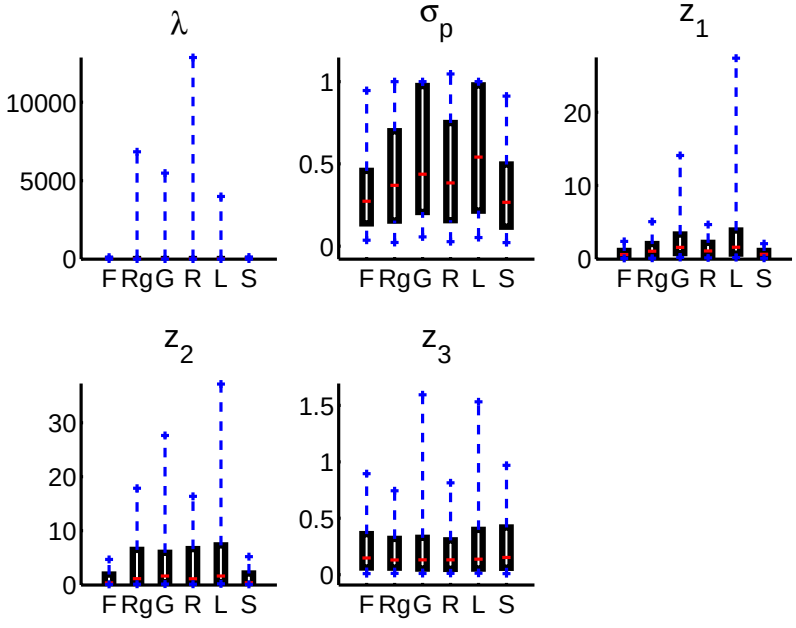
The results presented above summarise the various validation measures across multiple samples of the GP. In this section the Simulated Annealing Bayesian design examined Section 5.6.5 and shown in Figure 5.20(b) is compared to a grid design and the effect of the difference in parameter estimation accuracy on prediction is more closely investigated.

A Matérn kernel is used as before with a Log-Linear fixed-basis variance model. The true parameters of the GP sample are $\lambda = 0.2$, $\sigma_p = 1$ and $\beta = (-4.6, -1.6, -1.6)$.

The Mahalanobis score for the Grid design was 8933 and for the Fisher design 741 with 1024 being the theoretical optimum. The corresponding Dawid score was 4218 and -4364 in agreement with the Mahalanobis results. The corresponding RMSEs on the mean were 0.45 and 0.53, reflecting a more accurate prediction of the mean value for the Grid design. The rRMSE score and bias for the parameters are presented in Table 5.6. The β parameters for the variance process and the length-scale parameter are better identified when the Fisher design is utilised for

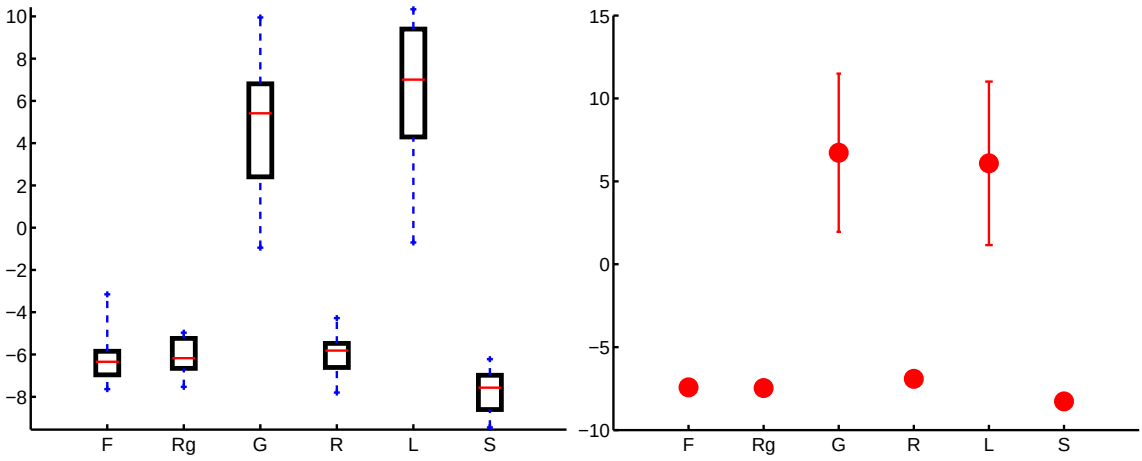


(a) Log-Linear: rRMSE



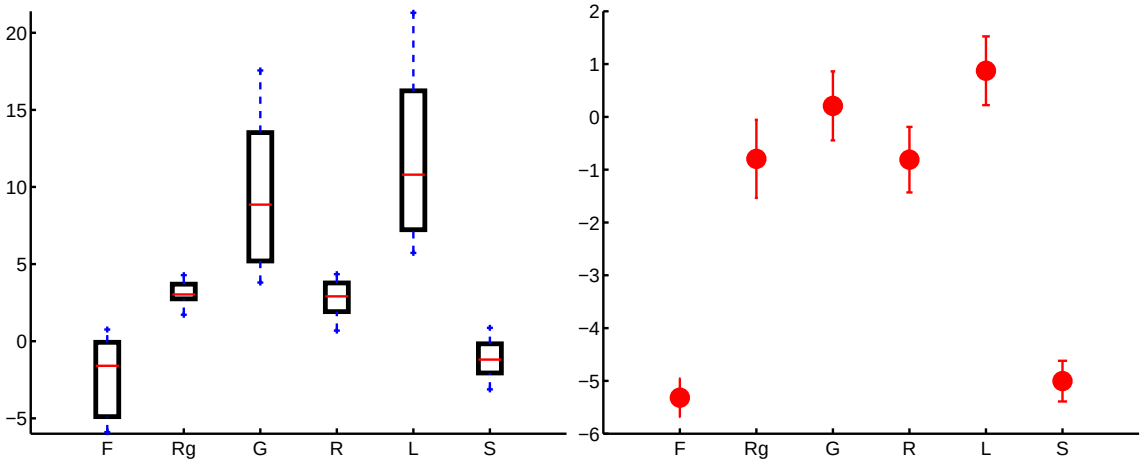
(b) Latent-Kernel: rRMSE

Figure 5.22: Bayesian Design: Parameter accuracy across all discrete prior permutations in terms of relative RMSE and bias for the Log-Linear and Latent-Kernel models.



(a) Log-Linear: Log Determinant

(b) Log-Linear: FIM



(c) Latent-Kernel: Log Determinant

(d) Latent-Kernel: FIM

Figure 5.23: Bayesian Design: Log determinant of the empirical parameter covariance and the Fisher score for the Log-Linear and Latent-Kernel models.

ML estimation as reflected by the improved bias and rRMSE values.

Utilising a 1024 Latin Hypercube test set, the predictive mean and standard deviation for the two designs are shown in Figure 5.24. The mean prediction is best captured by the Grid design. In terms of the standard variance prediction both designs significantly overestimate the true standard deviation of the sampled GP. However the predictive variance is a combination of the variance model as well as uncertainty due to distance from the training points. The latter factor is more critical for the Simulated Annealing design due to the highly clustered nature of the design points.

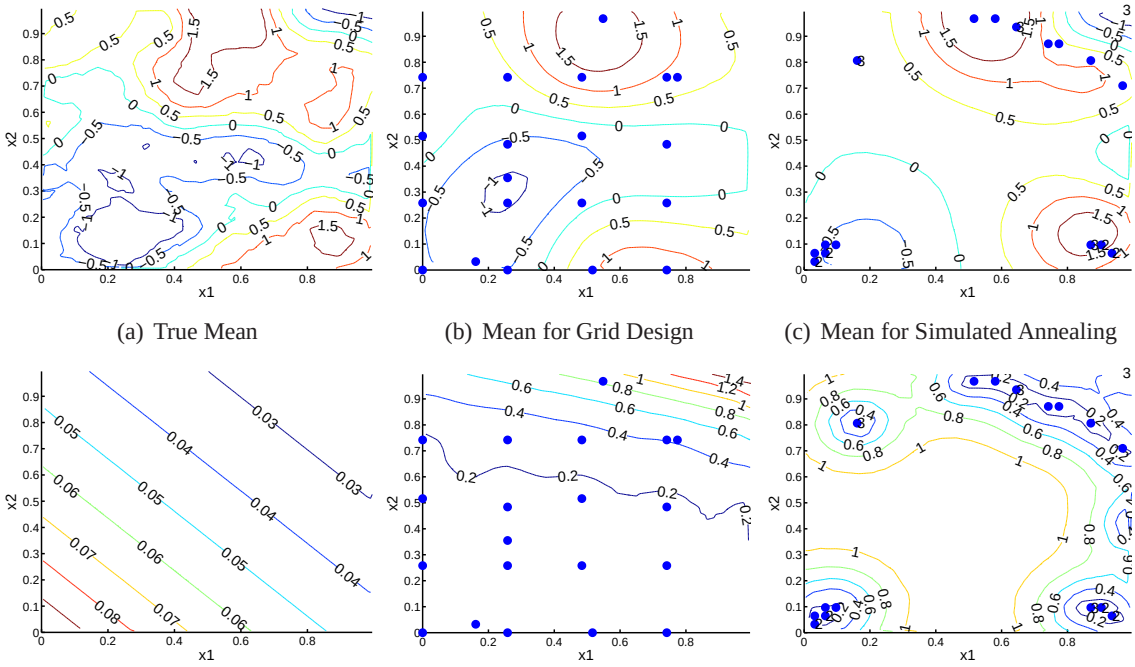
If the training set is fixed to the test set with no parameter re-estimation the two sources of uncertainty can be separated. For this purpose, the parameters estimates obtained for the Grid and Simulated Annealing designs were plugged in a GP with training set the entire test set with no replicate observations. The usage of the test set as training set, essentially cancels the impact on the variance prediction of the uncertainty due to distance of test to training points. The corresponding mean and standard deviation predictions for the two sets of parameter estimates are shown in Figure 5.25. The mean prediction is similar under both sets but the standard deviation is more accurately predicted by the Simulated Annealing parameter set. Examining solely the predictive variance model ($R(X_*)$) in Figure 5.26, we confirm the variance model has been more accurately learnt by the Simulated Annealing design. Further, the variance prediction for the Grid design is dominated by the variance model as the distance of the training set to the test points is considerably less than for the clustered designs such as the Simulated Annealing design.

To better understand the differences in predictive performance we decompose the Mahalanobis error to a vector of individual uncorrelated errors whose theoretical distribution is $N(0,I)$. As proposed by Bastos and O'Hagan (2009) the Mahalanobis error is decomposed using the Pivoted Cholesky Decomposition (PCD) where the order of the individual errors is determined by their conditional variance, i.e. the first point has the highest variance, the second has the highest variance conditioned on the first etc. Diagnostics are further described in Section 2.5.

This diagnostic allows for the interpretation of the errors as unusually large or small errors early in the sequence suggest poor estimation of σ_p or non homogeneity while errors in the latter part point to poor estimation of the correlation structure. The errors in this example are shown in Figure 5.27 where an incorrectly specified correlation structure is suggested for the Grid design which agrees with the parameter errors in Table 5.6.

5.6.7 Increasing Design Size

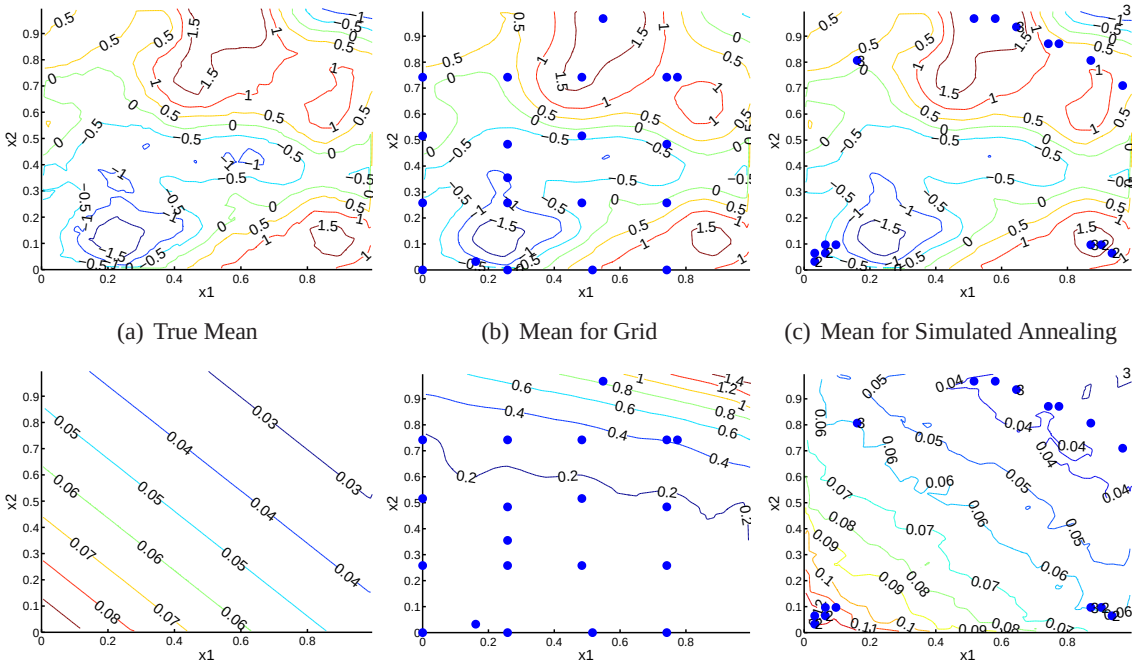
Thus far the simulation experiments have been performed for a single design size. In this section we examine the performance of Fisher-optimised designs as the number of available observations increases. For simplicity, we perform locally optimal design using the same single nugget model



(a) True Mean (b) Mean for Grid Design (c) Mean for Simulated Annealing

(d) True Std (e) Std for Grid (f) Std for Simulated Annealing

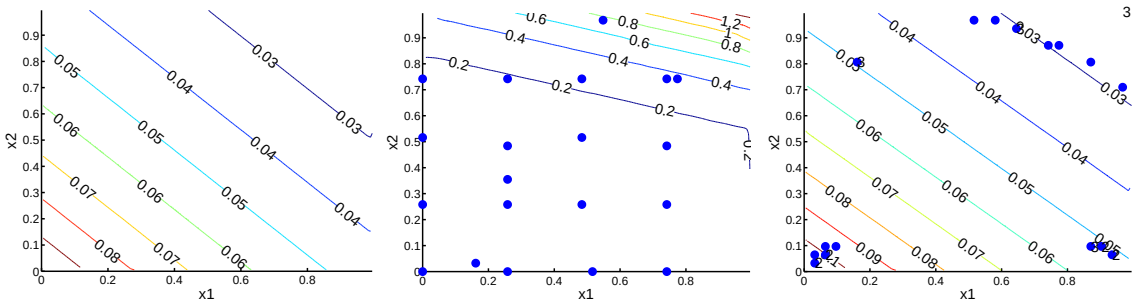
Figure 5.24: Specific Case Example: Predictive mean and standard deviation (std)) using 30 point designs for the Grid and Simulated Annealing designs. Training design points depicted by blue circles.



(a) True Mean (b) Mean for Grid (c) Mean for Simulated Annealing

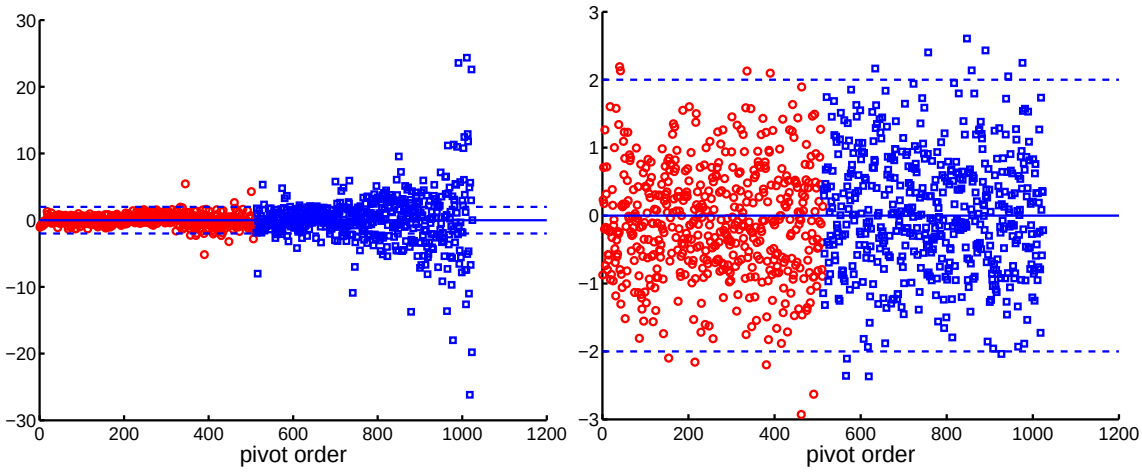
(d) True Std (e) Std for Grid (f) Std for Simulated Annealing

Figure 5.25: Specific Case Example: Predictive mean and standard deviation (std) using 1024 test point design as training set for the Grid and Simulated Annealing designs. Training design points depicted by blue circles.



(a) True Std (b) Std for Grid (c) Std for Simulated Annealing

Figure 5.26: Specific Case Example: Standard deviation of variance model $R(X_*)$ for the Grid and Simulated Annealing designs. Training design points depicted by blue circles.



(a) Grid (b) Simulated Annealing

Figure 5.27: Specific Case Example: Uncorrelated Errors vs Pivot Order for the Grid and Simulated Annealing designs.

Table 5.6: rRMSE and bias of parameter for Grid and Fisher designs.

Statistic	Design	λ	σ_p	β_1	β_2	β_3
rRMSE	Grid	0.71	0.19	1.34	2.62	6.62
	Fisher	0.0241	0.1816	0.0468	0.0446	0.06
Bias	Grid	0.14	-0.19	-6.18	4.22	10.66
	Fisher	-0.01	0.18	0.21	0.07	-0.10

as in Section 5.6.4.1. The design sizes considered are $[30, 100, 200]$.

The Mahalanobis error and Dawid score (Figure 5.28(a)) show the biggest differences in predictive performance for the smaller design size in agreement with the results in Section 5.6.4.1. The Fisher optimised design provides the most robust estimation. Even for the largest design size the Grid design underestimates the variance reflected in the high Mahalanobis error. In terms of parameter error (Figures 5.28(c) - 5.28(d)) the Fisher design has the smallest error with larger errors for the non-replicate designs. The differences in parameter estimation are greatest for the nugget parameter where even for the largest design size considered the non-replicate designs perform poorly.

In terms of Fisher score (Figure 5.28(f)) and the corresponding empirical parameter covariance (Figure 5.28(e)) a similar picture emerges where the Fisher design consistently achieves the smallest error although the differences with the other designs are reduced as the design size increases.

5.6.8 Structural Error

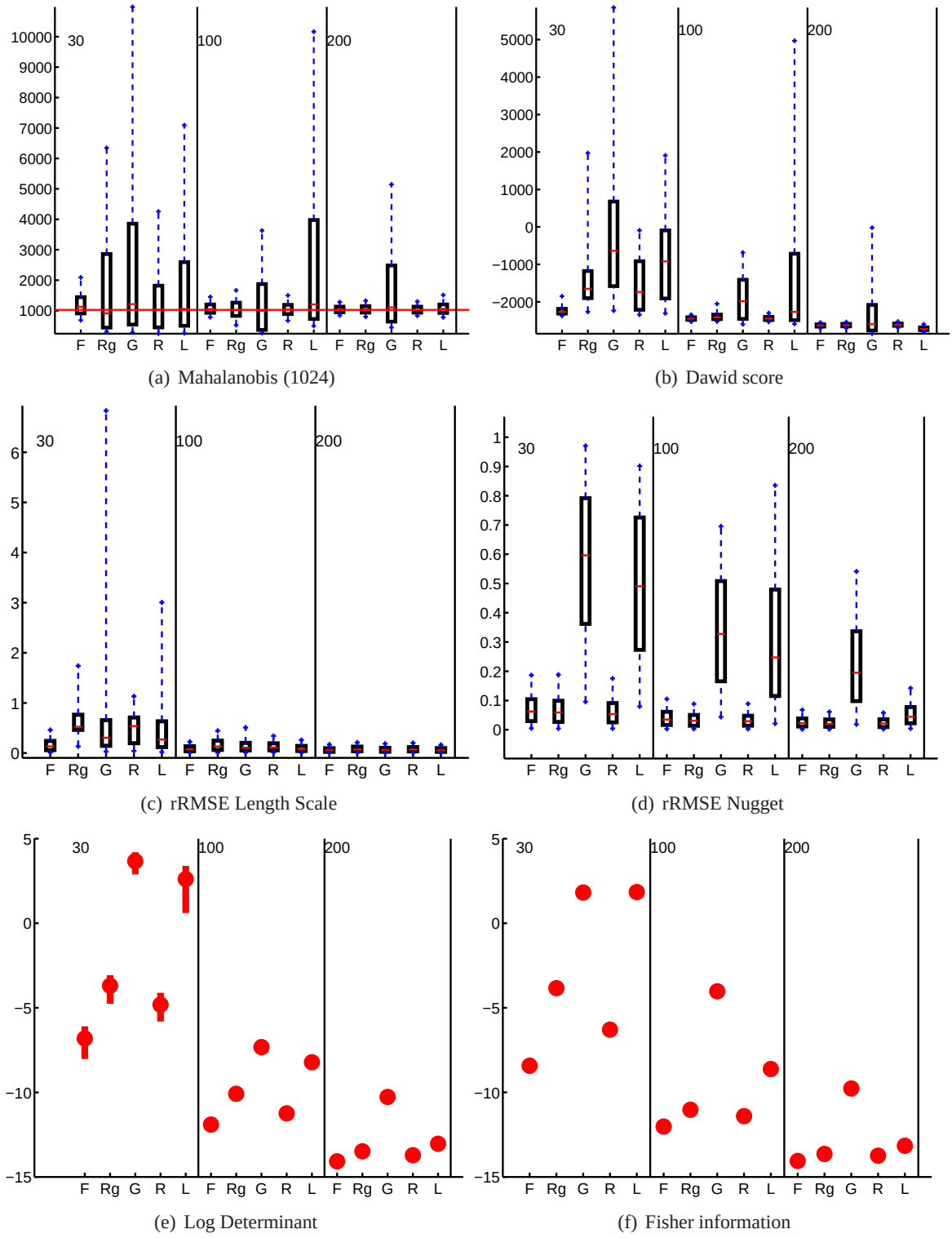
We have so far assumed the absence of structural error, i.e. the model used in the design process is the correct one. We now consider the effects on the performance of Fisher-optimised designs when the true underlying model does not match the assumed model used in inference. This effect is simulated by using GPs with different kernel specifications in the design and inference stages.

The same models and designs are utilised as in Section 5.6.5 but the methodology is modified to introduce structural error:

- *Log-Linear to Latent*. The Bayesian designs generated using the "assumed" Log-Linear model are evaluated using the Latent-Kernel model as the true process.
- *Latent to Log-Linear*. The "assumed" Latent-Kernel model designs are evaluated using the Log-Linear model as the true process.

The designs utilised are shown in Figure 5.20.

In the first experiment, a design generated assuming a Log-Linear variance model is evaluated on the more complex Latent-Kernel model variance (Figure 5.29). All designs achieve similar predictive accuracy as reflected by the Mahalanobis error. The Dawid score is omitted as it provides



(a) Mahalanobis (1024)

(b) Dawid score

(c) rRMSE Length Scale

(d) rRMSE Nugget

(e) Log Determinant

(f) Fisher information

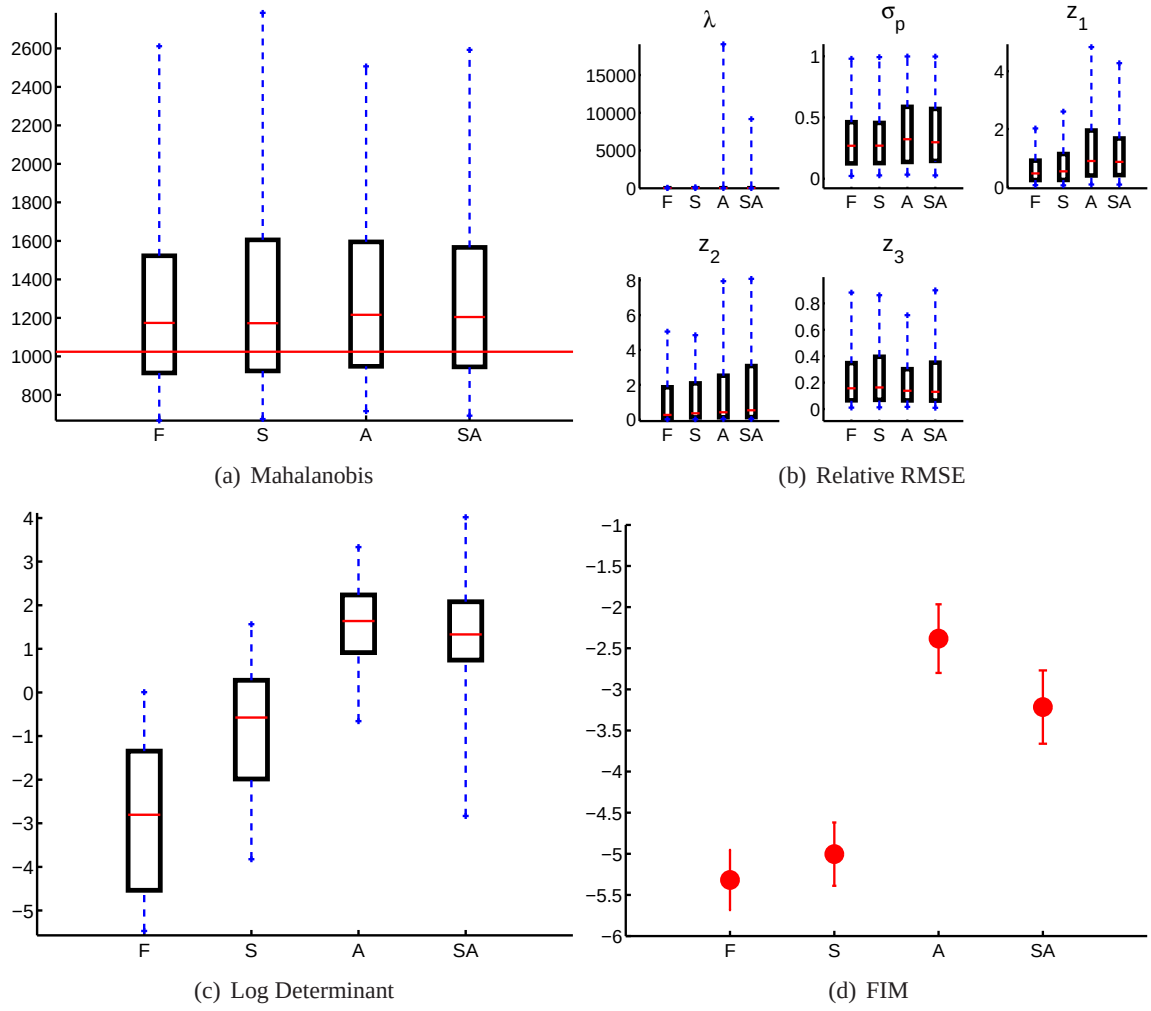
Figure 5.28: Effect of increasing design size on Fisher information (Section 5.6.7). F=Greedy design, Rg=Replicate Grid, G=Grid, R=Replicate Maximin Latin Hypercube, L=Maximin Latin Hypercube.

an identical picture.

Further, in terms of parameter estimation accuracy (Figure 5.29(b)) there are small differences, mostly in the tails of the distributions between the designs. The length-scale as was noted in Section 5.6.4.3 is not identifiable under this model for such a small training size and the differences are not meaningful. For the variance model parameters, we note utilising the correct model leads to more accurate estimation, especially with regards to the parameters corresponding to the corners of the design space, z_1 and z_2, where the Log-Linear designs do not place as strong an emphasis as the Latent-Kernel optimal designs. In terms of Fisher score and the corresponding empirical parameter covariance, we confirm that model misspecification has negatively impacted the parameter accuracy The Fisher score (Figure 5.29(d)), computed using the correct model for all designs, corresponds well to the empirical parameter covariance (Figure 5.29(c)). We note the impact of the model misspecification, however, has been minor as evidenced by the small separation in the latter and the lack of detriment on the predictive performance.

The reverse experiment of assuming a more complex variance response when the true process has a simpler log-linear form is summarised in Figure 5.30. In this case we see large differences in performance both in terms of the Mahalanobis error and the relative RMSE of the variance parameters for the Latent-Kernel designs. As in the previous experiment, the Dawid score is not included as it provides an identical ranking. In terms of parameter estimation (Figure 5.30(b)), we observe very large errors in the identification of the higher order variance coefficients, β_2 and β_3 when utilising the Latent-Kernel designs while the length-scale, process variance and first order variance coefficient, β_1, are estimated with similar accuracy across all designs. Examining the designs shown in Figure 5.20, we note the Latent-Kernel Designs place most points in a diagonal across the design space while the Log-Linear design have points in at least three corners of the space allowing for the separation of the effects of the two input factors. As was noted in Section 5.6.4.2, the linear form of the Log-Linear variance model requires placement of points on the edges of the design space to allow for the accurate estimation of the model coefficients β.

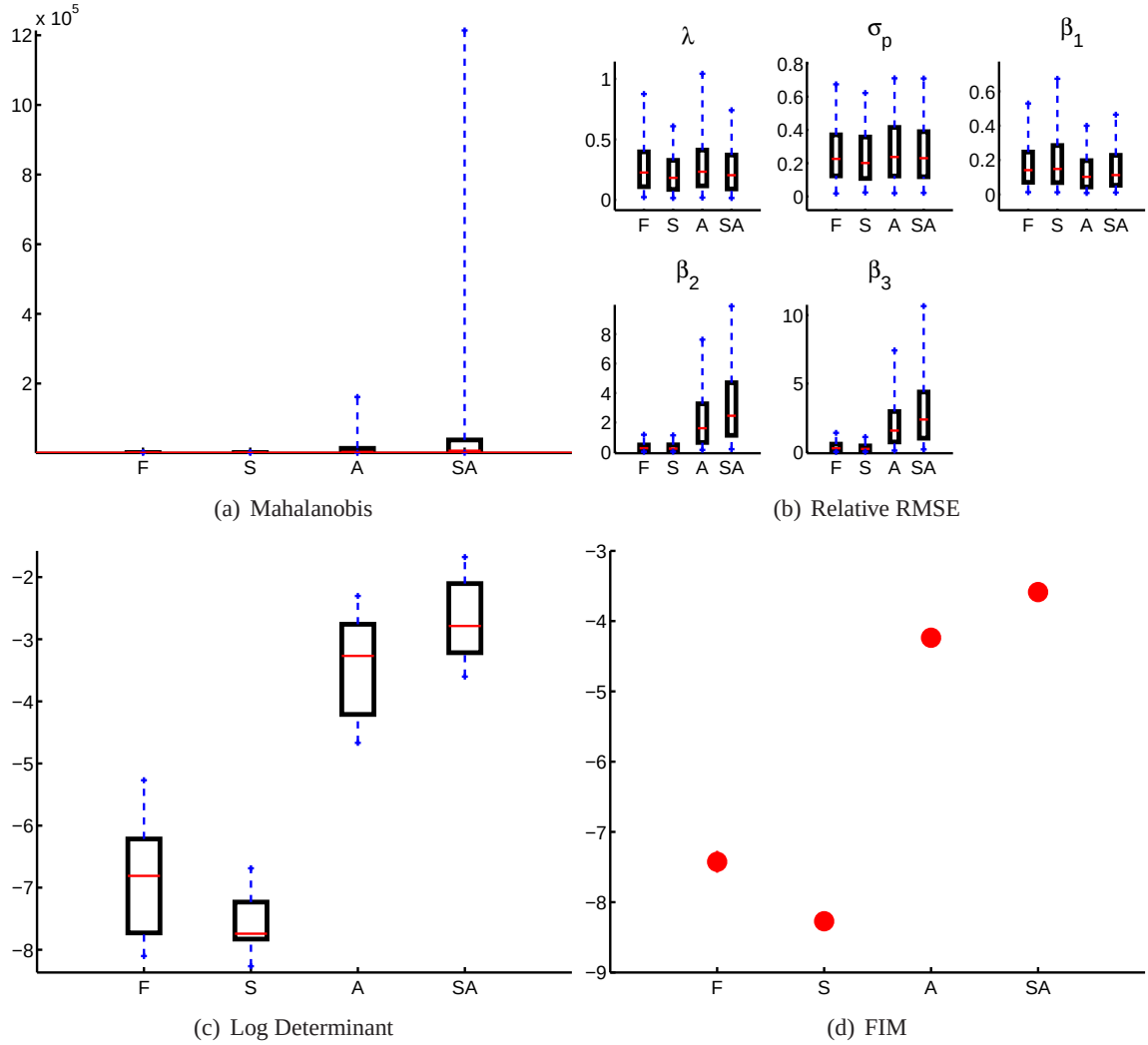
The Fisher score (Figure 5.30(d)) and empirical covariance (Figure 5.30(c)) reflect the higher errors for the Latent-Kernel designs. Specifically with regards to the latter, we note a large separation in the performance of the optimal and model misspecified designs. We conclude therefore the Latent-Kernel designs assume a more complex variance model and hence can capture a more limited set of models than the designs generated under the simpler Log-Linear variance model. The latter designs thus appear more robust to structural error, i.e. the misspecification of the model.



(a) Mahalanobis (b) Relative RMSE

(c) Log Determinant (d) FIM

Figure 5.29: Structural Error: Evaluating Log-Linear designs (Figure 5.20) on the Latent-Kernel model. F=Fisher design optimised using the Greedy optimisation under the Latent-Kernel model, S=Simulated Annealing Design for the Latent-Kernel model, A=Fisher Greedy design optimised under the Log-Linear model, SA=Simulated Annealing design under the Log-Linear model.



(a) Mahalanobis (b) Relative RMSE

(c) Log Determinant (d) FIM

Figure 5.30: Structural Error: Evaluating Latent-Kernel designs (Figure 5.20) on the Log-Linear model. F=Fisher design optimised using the Greedy optimisation under the Log-Linear model, S=Simulated Annealing Design for the Log-Linear model, A=Fisher Greedy design optimised under the Latent-Kernel model, SA=Simulated Annealing design under the Latent-Kernel model.

5.7 Bayesian Inference

In this section, the effect of the Fisher, uniform replicate and non-replicate designs on the GP parameter posterior using Bayesian inference is examined. The Hybrid Monte Carlo (HMC) (Nabney, 2001; Bishop, 2007) algorithm is utilised to perform sampling over a vague prior on the GP parameters and the posterior under the different designs is discussed.

In Section 5.7.1 a brief overview of the sampling approach to emulation is given. The issue of convergence is discussed in Section 5.7.2 followed by the simulation results in Section 5.7.3.

5.7.1 Methodology

A sampling approach is used to incorporate parameter uncertainty into the predictive GP variance. The HMC algorithm combines the Metropolis-Hastings algorithm with dynamical simulation methods utilising gradient information to bias the directions of exploration (Nabney, 2001). The resulting transitions have the ability of making large steps while keeping the rejection rate small (Bishop, 2007). Specifically HMC tries to avoid random walk behaviour by introducing an auxiliary momentum vector and implementing Hamiltonian dynamics where the potential function is the target density. The momentum samples are discarded after sampling. The end result of Hybrid MCMC is that proposals move across the sample space in larger steps and are therefore less correlated and converge to the target distribution more rapidly. We refer the reader to Nabney (2001) or Bishop (2007) for a detailed description and discussion of the HMC algorithm.

Given GP hyperparameter samples from the HMC procedure, the predictive mean and variance are calculated using the corresponding mean and variance of the GP samples (see MUCM Toolkit (World Wide Web electronic publication, Release 6, 2010) `ProcPredictGP` page). Using N samples $\{\theta_1, \theta_2, \ldots, \theta_N\}$ from the GP parameter posterior for a training design set ξ, each sample θ_i corresponds to a conditional GP prediction at a new point x_* with mean $\mu^i(x_*|\xi,\theta_i)$ and covariance $V^i(x_*|\xi,\theta_i)$. The combined predictive mean $\mu^c(x_*|\xi)$ and covariance $V^c(x_*|\xi)$ are:

- $\mu^c(x_*|\xi) = \frac{1}{N}\sum_i^N \mu^i(x_*|\xi,\theta_i)$, i.e. the combined mean is the average of the conditional predictive means.
- The calculation of the predictive covariance is more complex:
  1. Calculate the average covariance $\bar{V} = \frac{1}{N}\sum_i^N V^i(x_*|\xi,\theta_i)$.
  2. Calculate the covariance of the conditional means

$$W = \frac{1}{N}\sum_i^N \left(\mu^i(x_*|\xi,\theta_i) - \mu^c(x_*|\xi)\right)\left(\mu^i(x_*|\xi,\theta_i) - \mu^c(x_*|\xi)\right)^T.$$

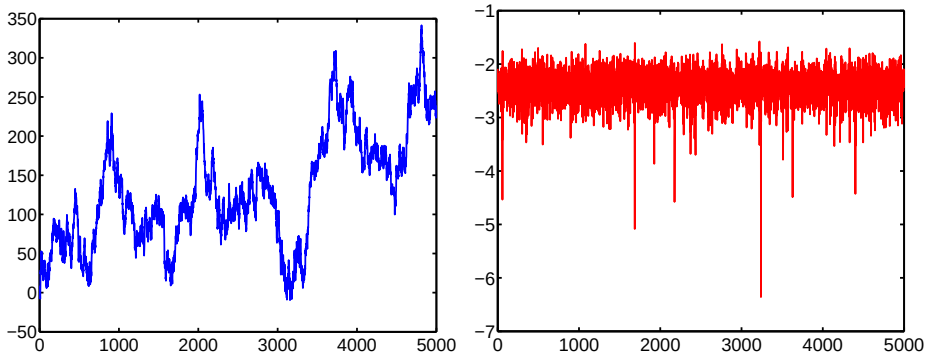
3. The predictive covariance is $V^c(x_*|\xi) = \bar{V} + W$.

5.7.2 Convergence Diagnostics

The convergence of the HMC chain was assessed by starting 8 parallel chains from perturbed initial conditions. The ML estimate of the parameters was perturbed by adding independent Gaussian noise, $N(0,1)$, which was used to initialised the HMC chain. A total of 5000 samples was used with a trajectory size of 200 and step size of 0.025.

We also use the EPSR measure to check if the chains have converged (Gelman and Rubin, 1992). This compares the within-chain variability to between-chain variability and as a guide should be less than 1.1 which was the case in our experiments. Following the recommendation in Nabney (2001), the first half n/2=2500 of samples is ignored.

In the pilot runs of the HMC simulation, the chains did not converge when no prior was used. When the vague prior described in Section 5.7.3 was used however, all chains converged. This phenomenon was observed under both Matérn and Exponential kernels and was especially pronounced for the length-scale parameter. An example of converging and non-converging chains for the length-scale parameter is shown in Figure 5.31. Therefore in the subsequent experiments, the prior was utilised as it has been shown to stabilise the HMC algorithm and ensure convergence within the allotted 5000 time steps.



(a) No Prior (b) Prior

Figure 5.31: Effect of Prior on convergence of chain for length-scale parameter. When no prior is used the chain is not converging to a stable distribution whereas under the vague prior described in Section 5.7.3 it does.

5.7.3 Simulation Results

A simulation study of the Nugget and Log-Linear variance models described in Section 5.6.4 is presented. The locally optimum designs are utilised and for the purposes of Bayesian inference a vague prior is placed on all GP parameters.

Independent normal priors are used for all GP parameters. For the log length-scale the prior is $N(-1,5)$, the log process variance $N(-1,8)$ and for the nugget and higher order variance coefficients $N(0,6)$. This translates in the following 95 percent intervals: $\log\lambda = (-5.4, 3.4)$, $\log\sigma_p = (-6.6, 4.6)$ and $\beta_i = (-4.8, 4.8)$.

The true parameter values for the Nugget model are $\log\lambda = -1.6$, $\log\sigma_p = 0$, $\beta_1 = -4.6$ and for the Log-Linear model $\log\lambda = -2.3$, $\log\sigma_p = 0$, $\beta_1 = -4.6$, $\beta_2 = \beta_3 = -1.6$. Thus, the priors contain the true values within the 95 intervals but are not centred at those values and in some cases such as for the nugget, the true parameter value is at the tail of the prior distribution. The prior was determined to allow a wide range of length-scales and noise levels. Specifically, the length-scale prior covers any credible value in the $[0,1]$ design region used.

For the prediction and histograms shown in the simulation results, the HMC chain is subsampled to every 25^{th} sample to ensure the remaining samples are approximately uncorrelated. The subsample size was derived by examining the autocorrelation. The HMC implementation in the Netlab (Nabney, 2001) software is used.

5.7.3.1 Nugget Model

An HMC simulation experiment for the Nugget model described in Section 5.6.4 is presented. Three realisations of the experiment are utilised to examine the impact of the Fisher-optimal and other designs on the parameter posterior. The Nugget model consists of three parameters, a Matérn kernel and a single nugget. The locally optimal designs shown in Figure 5.6 are utilised.

The validation errors, presented in Table 5.7, are calculated using a 1024 Latin Hypercube test set. The high uncertainty in most of the designs leads to very low Mahalanobis score reflecting the underconfidence of the predictors. The smallest error is observed for the Greedy design and the largest for the replicate Grid, Grid and replicate Latin designs. Of note is the high Mahalanobis error for the SA design which as explained below is caused by the high posterior variance for the nugget parameter. In terms of the Dawid score, the Greedy and Simulated Annealing Fisher-optimised designs are again ranked better than the competing space-filling designs confirming the Mahalanobis score ranking.

The parameter accuracy of the ML estimates is shown in Table 5.8 in terms of the relative RMSE score. The posterior mode accuracy results are given in Table 5.9 and in general we see broad agreement with the ML errors signifying that the parameter posterior mode agrees with the ML estimate. The errors for the Replicated Latin are an exception with the errors for the length-scale and nugget parameters being significantly smaller for the posterior mode. For the non-replicate Maximin Latin design, a significant drop of error for the length-scale parameter is also observed. Of note is the high error of the SA design for the nugget parameter β_1 which is

Table 5.7: HMC Validation results for the Nugget model. Mean value and standard deviations across three realisations of the experiment.

Design	Mahalanobis (1024)	RMSE	Dawid Score
Greedy	745 ±368	0.86 ±0.03	-2124 ±243
RGrid	249 ±64	0.92 ±0.12	-887 ±242
Grid	370 ±197	0.92 ±0.04	225 ±206
RL	164 ±64	0.86 ±0.06	-530 ±428
Latin	414 ±139	0.87 ±0.12	-77 ±418
SA	543 ±217	0.87 ±0.15	-1906 ±148

higher than for the other replicate designs. Overall, the Fisher-based designs have on average the lowest parameter errors.

The posterior variances are summarised in Table 5.10 and the full posterior distributions are shown in Figures 5.32, 5.33 and 5.34 for a single realisation of the experiment. In general, a larger variance in parameter posterior leads to larger variance in prediction. In the case of the non-replicate designs large posterior variances are observed for all parameters and are reflected in very high Mahalanobis errors demonstrating underconfidence of the HMC prediction.

For the length-scale parameter, the Fisher-optimised designs have the lowest variance in the posterior. and appear to be more effective in restricting the range of plausible values for the length-scale parameter than all other designs.

For the process variance parameter the replicate designs achieve similar variance in the posterior while the non-replicate Grid and Latin designs have the highest posterior variance.

Finally the posterior variance for the nugget parameter is low for the Greedy, replicate Grid and replicated Latin designs while for the SA, Grid and Latin designs it is higher. Examining the relative RMSE results of the posterior mode in Table 5.9 As we have noted the SA algorithm has a high average relative RMSE for the nugget parameter compared to the other replicate designs (Table 5.9) and the corresponding posterior variance is also higher which explains the large predictive variance and hence low Mahalanobis score of the SA design. This is also supported by the Fisher scores (Figure 5.7) where the SA design achieved a worse Fisher score than the Greedy design so we would expect the Greedy design to have lower errors in this experiment.

Overall, the Fisher-optimal design, the Greedy design, is shown to lead to robust estimation of all GP parameters and as expected by the Fisher scores (Figure 5.7) the replicate designs outperform and non-replicate Grid and Maximin Latin Hypercube designs. However we caution that due to the small number of realisations of the experiment, the conclusions drawn are preliminary.

Table 5.8: Relative RMSE of the ML estimate for the Nugget model. Mean and standard deviation for three realisations of the experiment shown.

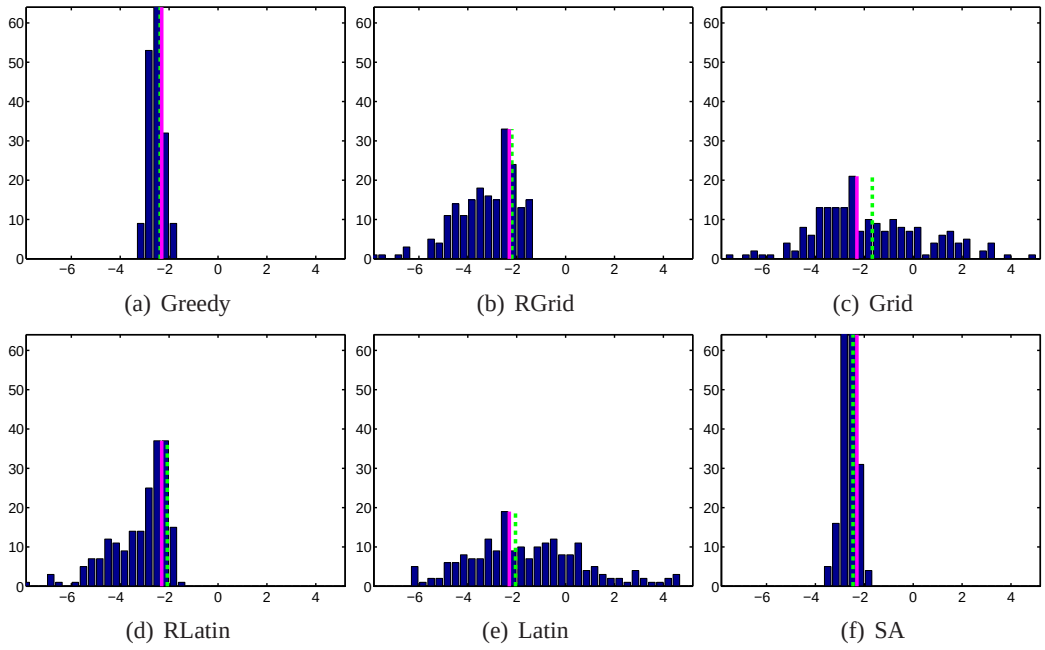
Parameter	Greedy	RGrid	Grid	RL	Latin	SA
Length Scale	0.28 ±0.21	0.48 ±0.64	4.56 ±4.85	6.68 ±11.19	4.08 ±6.22	0.30 ±0.34
Process Variance	0.29 ±0.35	0.09 ±0.05	0.70 ±0.25	0.16 ±0.17	0.49 ±0.30	0.05 ±0.06
β_1	0.05 ±0.02	0.08 ±0.05	0.96 ±0.08	0.37 ±0.59	0.78 ±0.27	0.20 ±0.14

Table 5.9: Relative RMSE of the posterior mode for the Nugget model. Mean and standard deviation for three realisations of the experiment shown.

Parameter	Greedy	RGrid	Grid	RL	Latin	SA
Length Scale	0.29 ±0.21	0.56 ±0.16	3.66 ±5.29	0.44 ±0.14	1.16 ±0.48	0.26 ±0.20
Process Variance	0.30 ±0.39	0.10 ±0.04	0.66 ±0.41	0.15 ±0.20	0.52 ±0.35	0.07 ±0.04
β_1	0.05 ±0.03	0.10 ±0.05	0.95 ±0.08	0.04 ±0.02	0.83 ±0.15	0.21 ±0.13

Table 5.10: Parameter posterior variance for the Nugget model. Mean and standard deviation for three realisations of the experiment shown.

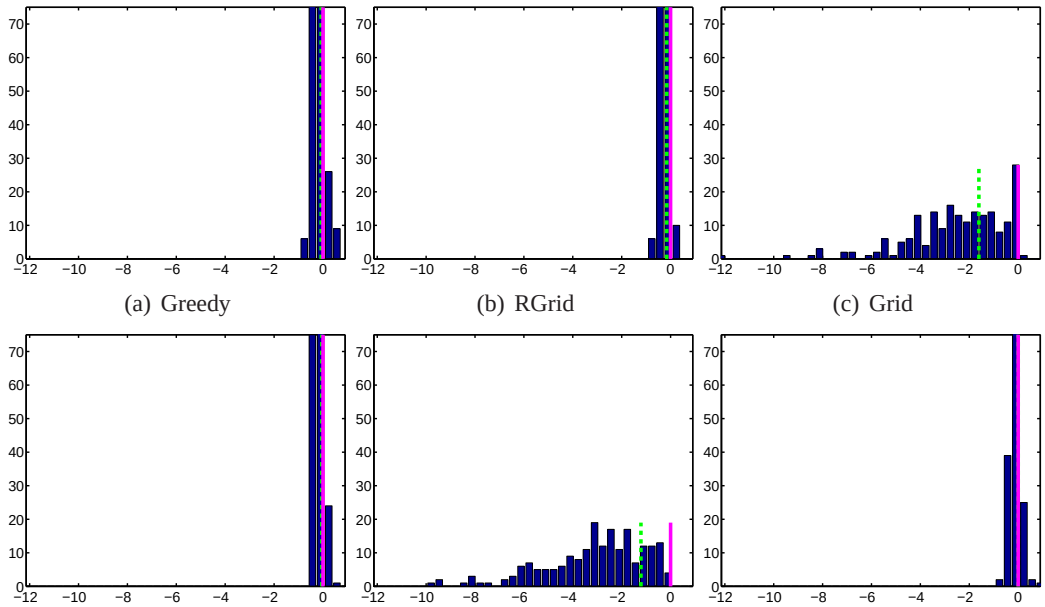
Parameter	Greedy	RGrid	Grid	RL	Latin	SA
Length Scale	0.09 ±0.06	1.44 ±0.20	4.98 ±1.07	1.50 ±0.26	3.56 ±1.89	0.13 ±0.03
Process Variance	0.07 ±0.01	0.04 ±0.01	2.69 ±1.65	0.04 ±0.00	2.99 ±1.07	0.04 ±0.00
β_1	0.20 ±0.02	0.18 ±0.01	1.08 ±0.62	0.14 ±0.01	1.47 ±0.94	0.56 ±0.17



(a) Greedy (b) RGrid (c) Grid

(d) RLatin (e) Latin (f) SA

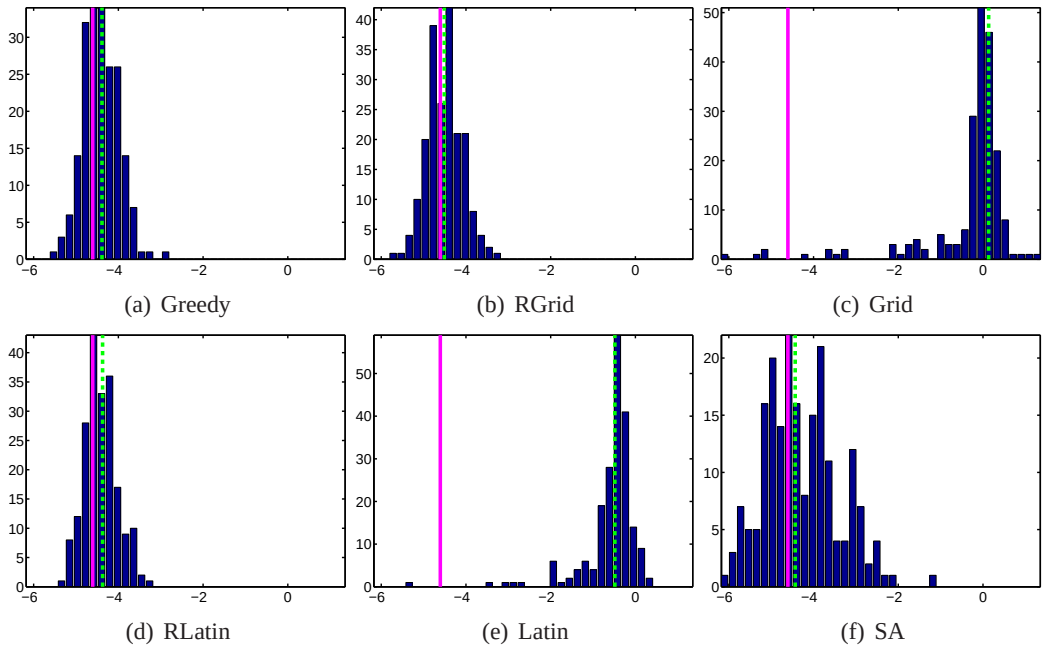
Figure 5.32: Posterior variance for the log Length scale parameter of the Nugget model. Solid magenta line is true value and green dashed line is the ML estimate.



(a) Greedy (b) RGrid (c) Grid

(d) RLatin (e) Latin (f) SA

Figure 5.33: Posterior variance for the log Process Variance parameter of the Nugget model. Solid magenta line is true value and green dashed line is the ML estimate.



(a) Greedy (b) RGrid (c) Grid

(d) RLatin (e) Latin (f) SA

Figure 5.34: Posterior variance for the Nugget parameter of the Nugget model. Solid magenta line is true value and green dashed line is the ML estimate.

5.7.3.2 Log-Linear Model

A single instance of HMC prediction for the Log-Linear model is examined in this section. The Matérn kernel is also used but the variance model now has 3 parameters. The locally optimum designs are shown in Figure 5.11 and the model is described in Section 5.6.4.

In Table 5.11 we see the validation results on a 1024 point test set in terms of the Mahalanobis error, Dawid score and RMSE. The SA design performs well with the Greedy design performing less well but significantly better than the other designs which all make underconfident predictions as reflected by the very low Mahalanobis score. The Dawid score results in an identical design ranking with the Mahalanobis error.

Table 5.11: HMC Validation results for the Log-Linear model.

Design	Mahalanobis (1024)	RMSE	Dawid score
Greedy	515	0.72	-4073
RGrid	35	0.59	-744
Grid	93	0.39	-1657
RL	48	0.61	-1387
Latin	128	0.37	-1947
SA	1036	0.48	-4294

The relative RMSE of the ML estimate for all parameters is presented in Table 5.12 and the corresponding errors for the posterior mode in Table 5.13. Posterior parameter variance is shown in Figure 5.14. As before, the posterior for the length-scale parameter is lowest for the Fisher-optimised designs though in this example the non-replicate Latin design also has small variance. It is worth noting that the ML relative RMSE for the length-scale is lowest for the Replicated Grid design for which the corresponding posterior mode error variance is high. The process variance parameter is identified with similar accuracy for all designs except the SA design, in terms of both relative RMSE and posterior variance. The SA design in this example has a higher relative RMSE.

For the variance model parameters, the non-replicate designs have consistently high posterior variance even though the corresponding relative RMSE is sometimes quite low. For instance, the Latin design has the smallest relative RMSE for β_3 but the second highest posterior variance for that parameter. The SA design which has the lowest Fisher score of all considered designs (Figure 5.12(b)) consistently achieves the lowest posterior variance even though it does not have the lowest rRMSE for every variance parameter.

Table 5.12: Relative RMSE of the ML estimate for the Log-Linear model.

Parameter	Greedy	RGrid	Grid	RL	Latin	SA
Length Scale	0.05	0.02	4.74	0.25	0.19	0.39
Process Variance	0.04	0.10	0.14	0.05	0.13	0.44
β_1	0.01	0.18	0.89	0.11	0.62	0.00
β_2	0.45	0.18	0.50	0.27	1.29	0.22
β_3	0.12	1.12	1.28	0.53	0.01	0.10

Table 5.13: Relative RMSE of the posterior mode estimate for the Log-Linear model.

Parameter	Greedy	RGrid	Grid	RL	Latin	SA
Length Scale	0.06	0.57	0.10	0.43	0.16	0.37
Process Variance	0.02	0.16	0.02	0.01	0.03	0.44
β_1	0.04	0.15	0.49	0.08	0.52	0.02
β_2	0.38	0.20	0.79	0.35	0.79	0.04
β_3	0.16	1.17	0.44	0.32	0.01	0.04

Table 5.14: Parameter posterior variances for the Log-Linear model.

Parameter	Greedy	RGrid	Grid	RL	Latin	SA
Length Scale	0.11	2.18	0.43	1.36	0.13	0.10
Process Variance	0.13	0.04	0.13	0.05	0.09	0.10
β_1	0.62	0.94	3.23	0.81	2.12	0.50
β_2	1.77	0.93	4.01	2.49	5.14	0.73
β_3	1.64	1.51	4.70	1.60	3.50	0.81

5.8 Conclusions

This chapter has presented a new approach to model based optimal design for heteroscedastic GP emulators and examined empirically the performance of the produced designs through an extensive set of simulation studies.

In Section 5.2 an overview of the optimal design literature was given. When the design criterion function is concave, such as for linear models, the General Equivalence Theorem applies which allows to check whether the proposed design is the minimally supported optimal design. Further, the additivity of the information matrices for linear models allows us to calculate a bound for the design size. However in the correlated error setup considered in this thesis, neither result applies as the criterion is not concave. The motivation for using Fisher information in this context stems from the asymptotic analysis of Mardia and Marshall (1984) who showed under increasing domain asymptotics that the ML estimator $\hat{\theta}$ converges in probability to the true parameter θ, $\hat{\theta} \rightarrow N(\theta, I^{-1}(\theta))$ where $I(\theta)$ is the Fisher information matrix. Therefore the minimisation of the finite sample FIM is justified when the goal is to minimise parameter uncertainty. Some authors (Zhang and Zimmerman, 2005) argue that infill asymptotics, where inter-point distances go to zero, are more appropriate for interpolation and computer experiments. Under this asymptotic framework however, the convergence of the finite sample FIM has not been proven in general and results exist only for specific cases.

In the current design literature for computer experiments, only deterministic simulators are considered and as a result replicated observations are not handled (Müller and Stehlík, 2009). The extension of the Fisher criterion for replicated observations under our likelihood model was presented in Section 5.3. The Bayesian formulation of the design problem was discussed in Section 5.4 where the parameter uncertainty is numerically integrated out using Monte Carlo. The design methodology is completed by specifying the optimisation method used. We have considered the Greedy and Simulated Annealing algorithms. The former is simple to implement and requires little computational effort although as was discussed in Section 5.5.1 it cannot be utilised in high-dimensional spaces due to the curse of dimensionality. However as the criterion used is not a submodular function, no theoretical guarantee exists on its performance. The Simulated Annealing algorithm does not suffer from the curse of dimensionality and is a well known global optimisation method which has been shown to avoid local minima but requires significantly higher computational effort as well as tuning of a set of parameters.

The first set of simulation experiments in Section 5.6 focused on Maximum Likelihood (ML) estimators. The monotonicity of the FIM to the log determinant of the parameter covariance was demonstrated for the fixed basis variance model under different noise levels in Section 5.6.2. The

approximation error was found to increase as the noise level was increased but the monotonicity relationship was not violated.

In Section 5.6.3 a complete enumeration of all nine-point non-replicate designs from a twenty nine-point candidate set was used to demonstrate the existence of multiple local minima in the optimisation search space and the effectiveness of the greedy algorithm to locate a near-optimal solution. Further, the profile likelihoods of all model parameters for the optimal and a grid design were compared. The FIM design was found to exclude a larger range of parameter values from consideration and hence identify the ML estimate with higher certainty.

The performance of FIM designs was examined in more depth in Section 5.6.4. The utilisation of local designs, where the true parameters values are used in design generation, allowed for the study of the Fisher approximation without errors due to the numerical integration of the Bayesian criterion (Equation (5.4)). The experiments were performed across multiple realisations and the performance examined in terms of both predictive and parameter estimation accuracy. The three models used ranged in order of complexity. For the Nugget model (Section 5.6.4.1), where a constant variance model is used, the predictive performance of all replicate designs was found to be superior to that of the non-replicate Grid and Maximin Latin Hypercube designs. The finite-sample FIM design ordering corresponded to the empirical parameter covariance with the lowest Fisher score design also having the lowest parameter error both in terms of the empirical parameter covariance and the relative RMSE of individual parameters.

These results extend to the Log-Linear model (Section 5.6.4.2) where a linear variance model was used. The approximation of the FIM to the parameter covariance was worse than for the Nugget model with the replicate designs achieving lower Fisher score and higher parameter estimation accuracy as reflected by the log determinant of the parameter covariance than the non-replicate Grid and Maximin Latin Hypercube designs. The relative RMSE showed the Fisher-optimised designs obtained through Simulated Annealing and Greedy optimisation identified the length-scale parameter more reliably than the other designs. The variance-model parameters were identified by all replicate designs with the non-replicate designs showing significantly higher errors.

The largest approximation error is observed for the Latent-Kernel model (Section 5.6.4.3) where the ordering of the space-filling designs as predicted by the Fisher score does not match the empirical parameter covariance. However the Fisher-optimised designs achieve the lowest score in terms of both measures as well as the individual parameter relative RMSEs. Thus, although the approximation error is larger, the Fisher designs are more differentiated from the space-filling designs in terms of parameter accuracy than for the simpler Nugget and Log-Linear variance models considered.

The local design experimental results are summarised in Table 5.15 where both the computational complexity necessary to generate each design and the average error in the estimation of the variance model parameters β are shown. As discussed in Section 5.5.1, the computational cost of generating the space-filling designs is negligible and does not depend on the model complexity as is reflected by the constant cost of generating the Grid and Maximin Latin Hypercube designs. The Simulated Annealing design is the most expensive to generate and is roughly 10 times as expensive to calculate as the Greedy design for the configurations used in the experiments. In terms of parameter estimation error for the variance model parameters, the computational complexity of model-based design based on the Fisher information is only justified for the Latent-Kernel model, where the variance model is the most complex. For the simpler Nugget and Log-Linear variance models, the geometric space-filling designs with uniformly spread replicated observations have performed as well as the Fisher optimal designs.

Table 5.15: Summary of design performance for all local design experiments. The elapsed time T to generate each design is provided in seconds. The optimisation for both the Greedy and Simulated Annealing designs was run in parallel as described in Section 5.5 and the elapsed time is reported for the entire optimisation process. Also shown is the average rRMSE for all variance model parameters β where the lowest errors are marked in bold.

Design	Nugget		Log-Linear		Latent-Kernel	
	T	β	T	β	T	β
Greedy	28	0.08	65	0.65	68	**0.22**
Replicate Grid	< 1	**0.07**	< 1	**0.47**	< 1	0.46
Grid	< 1	0.62	< 1	2.30	< 1	0.66
Replicate Maximin Latin Hypercube	3	**0.07**	3	0.55	3	0.49
Maximin Latin Hypercube	3	0.54	3	2.10	3	0.82
Simulated Annealing	301	0.15	743	**0.43**	903	**0.25**

In general, these results suggest that the approximation error of the FIM to the empirical covariance increases with the complexity of the variance model. However the benefits of optimising the Fisher score become more apparent under such complex models. The usage of Fisher-optimal designs can be justified up to the point where the approximation error is too large and the monotonicity of the designs considered is violated. However we are aware of no theoretical results to help estimate the magnitude of the approximation error and heuristic approaches have to be used.

In terms of prediction error of the simulator mean as reflected by the RMSE, the non-replicate space-filling designs on average achieve lower error. Even though such designs have higher errors in parameter estimation, they cover the space more uniformly and hence are more likely to predict accurately the mean value. However, as reflected by the Mahalanobis error and Dawid score, the replicate designs capture the variance response more accurately as the length-scale and variance parameters are estimated more precisely. The length-scale parameter is most reliably identified

by the lowest Fisher score design while for the non-replicate designs, consistently large errors are observed in length-scale estimation. The variance model parameters are identified reliably by all replicate designs in the case of the simpler Nugget and Log-Linear models or by the Fisher-optimal designs in the case of the more complex Latent-Kernel model. The similarity in performance for the replicate and Fisher-optimal designs in the case of the Nugget and Log-Linear models, is reflected by the Fisher score .

We note that in the entirety of simulation experiments presented in this chapter, the Dawid score agrees with the Mahalanobis error in the ranking of the competing designs. Therefore in the subsequent chapter, for brevity of presentation we focus on the Mahalanobis error which can be interpreted more readily due to its known sampling properties (Section 2.5). Different rankings of emulators are possible through the two metrics but such have only been observed in our experience when the difference of the Mahalanobis error between two models is small, i.e. less than two standard deviations of the Mahalanobis error sampling distribution. In particular as Bastos (2010) discusses, the validity of an emulator may be judged by checking that the Mahalanobis error is within two standard deviations of the expected value. In the results presented in this chapter, large differences in the Mahalanobis error have signified a comparison between a valid and an invalid emulator. In our experience under such a circumstance, the Dawid score will provide the same ranking as the Mahalanobis error.

Regarding the optimisation methods used, the solution found by the Simulated Annealing algorithm for the Nugget (Section 5.6.4.1) and Latent kernel (Section 5.6.4.3) local design experiments, has a larger Fisher score than the Greedy solution suggesting further effort in terms of computational time is required for the algorithm to find a solution closer to optimal than the Greedy solution. As was seen in Section 5.7.3.1, the lower Fisher score for the Greedy design in the case of the Nugget model was reflected in lower parameter posterior variance and more accurate prediction. To improve the performance of the Simulated Annealing algorithm, the annealing schedule could be changed from linear to a more conservative (e.g. log) schedule and the maximum number of iterations increased.

In Section 5.6.5 a set of Bayesian design simulation results was presented. For design generation, as in Zhu and Stein (2005) a discrete prior was used. The discrete values were selected to represent a wide range of simulator behaviours with short and long correlation length-scales and varying levels of noise. The Log-Linear and Latent-Kernel models from Section 5.6.4 were used and the evaluation of the Bayesian designs included all permutations of prior values to ensure performance was measured across the entire parameter domain defined by the prior. The conclusions from the local design simulation experiments continue to hold under the Bayesian framework. All variance models parameters were better identified by the replicate designs in the case of the Log-

Linear model whereas solely the Greedy and Simulated Annealing designs captured accurately the variance model parameters in the case of the Latent-Kernel model. The non-replicate Grid and Maximin Latin Hypercube design perform poorly under both models with very high Mahalanobis errors and Dawid scores evident.

The performance of the Fisher-optimal Simulated Annealing Bayesian design was examined more closely in Section 5.6.6 where it was compared to a Grid design for a specific realisation of the GP using the Log-Linear model. The two sources of uncertainty in prediction, code uncertainty, (Kennedy and O'Hagan, 2001) which stems from distance of test to train points, and intrinsic model variance, which is present only in stochastic computer models, were separated by using the test set as the training set. In this setup, the effect of the more accurate parameter estimation of the Simulated Annealing design is evident as the predictive variance closely matches the true variance while for the Grid design the variance estimation is highly inaccurate. Under the smaller thirty-point training design, the intrinsic model variance dominates the predictive variance for the space-filling Grid design while for the highly clustered Simulated Annealing design code uncertainty dominates as the true variance is significantly smaller. The decomposition of the Mahalanobis score using the Pivoted Cholesky Decomposition confirms the higher error for the Grid designs stems from an inaccurately identified correlation structure. This validation method suggested by Bastos and O'Hagan (2009) offers a practical method of validating emulators and can point to the source of estimation error.

In Section 5.6.7, the performance of Fisher designs was examined under increasing design size for the Nugget model. The parameter and prediction errors decrease for all designs but even for the largest design size considered (200) the Grid design has higher errors than the other designs. The parameter estimation error as reflected by the log determinant of the parameter covariance is smaller for the Fisher design even under the larger designs sizes.

Structural error, where the model used in design generation is not the correct one, is discussed in Section 5.6.8. When the Log-Linear design was used with the Latent-Kernel model little loss of predictive or parameter estimation accuracy was observed. However the reverse experiment where the Latent-Kernel optimal design was used with the Log-Linear model resulted in large errors in terms of both sets of measures. As the principle of parsimony suggests, optimal designs generated using simpler models are more robust to model misspecification.

In Section 5.7 Fisher-optimal designs were examined under Bayesian inference. The parameter posterior for three realisations of the Nugget and one realisation of the Log-Linear models was examined. Incorporating prior uncertainty into the prediction inflates the predictive variance for all designs considered as reflected by the low Mahalanobis scores. However Fisher-optimal designs achieve the lowest Mahalanobis errors and Dawid scores as they have the lowest variance in the

parameter posterior in addition to lower ML estimation error. Even for parameters where the ML error is higher than for other designs, Fisher-optimal designs have lower parameter posterior variance reflecting the informativeness of the design to constrain the range of plausible parameter values. This effect was also observed in a more limited context in Section 5.6.3 where the profile likelihoods were examined for a one-dimensional nine-point design.

Overall explicit optimisation of the Fisher criterion has been shown to facilitate reliable inference of model parameters under a range of models of differing complexities. This conclusion is corroborated by predictive and parameter accuracy results for both ML and Bayesian estimation. The effect is more pronounced under the latter where the parameter uncertainty is included in the model prediction. Our work agrees and extends the results of Zhu and Stein (2005) where only ML estimation of single nugget models was considered and predictive performance was not examined.

5.8.1 Future Work

The Fisher design methodology we have presented could be extended in a variety of ways.

Adaptive experimental design where the new simulator observations are requested and incorporated in the design approach was briefly discussed in Section 5.2.4. The Fisher-based approach can directly be extended in this direction. Rather than examining the GP prior process, the GP posterior may be used in the design criterion:

$$\begin{aligned} \mu_* &= K_{\Sigma*}^T C_\Sigma^{-1} t \\ \Sigma_* &= K_{\Sigma**} + R_{**} - K_{\Sigma*}^T C_\Sigma^{-1} K_{\Sigma*}, \end{aligned}$$

where t the observed values. The correlation parameters θ appear both in the mean and covariance of the GP posterior. Given $\mathbf{X}$ distributed as $N(\mu(\theta), \Sigma(\theta))$, the i, j element of the FIM is:

$$\mathcal{F}_{ij} = \frac{\partial \mu^T}{\partial \theta_i} \Sigma^{-1} \frac{\partial \mu}{\partial \theta_j} + \frac{1}{2} \mathrm{tr} \left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_i} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_j} \right).$$

This result can be readily extended to the replicated observations using the approach presented in Section 5.3. Further research is required however to show the usefulness of this approach.

In addition, Fisher-based designs can be incorporated into design approaches that minimise other criteria. In particular, we envisage utilisation of Fisher designs for design approaches that linearise the correlated process using expansions that depend on parameter accuracy. In Fedorov and Müller (2004) the GP covariance is approximated by a truncated eigenvector expansion. Youssef (2010) proposes the usage of Haar wavelets for the expansion. The approximation error of the ex-

pansion critically depends on the parameter accuracy. Youssef (2010) proposes a Latin Hypercube for the initial design but a much more natural choice would be a Fisher design where the parameter estimation variance is explicitly minimised.

Following the discussion in Section 5.2, Zhang and Zimmerman (2005) suggest that the infill asymptotic framework is preferable when interpolation is the aim. However when considering infill asymptotics, the issue of parameter consistency must be addressed. Under increasing domain asymptotics all kernel parameters are consistently estimable. Under infill asymptotics this is no longer the case and consistency must be established. Zhang and Zimmerman (2005) demonstrate through simulation that for parameters that are not consistently estimable, infill asymptotic results seem to approximate finite sample properties more closely than increasing domain. We propose that for specific models, consistency of parameters is established and when this is not true use approximations based on infill asymptotic results rather than Fisher information or other increasing domain results. However such an approach would be model specific.

In this work the focus has been exclusively on design for identifying the covariance parameters. A mean function in the GP prior is used in practice in the emulation context as prior information can be easily incorporated and the residual process is more likely to be stationary. It is well known in the literature (e.g. Müller and Stehlík (2010)) that design for trend parameters is usually antithetical to that of covariance parameters. Combining design for trend and covariance parameter estimation in the heteroscedastic emulation context is an area for future research.

One limitation of the approach proposed in this work is the discrete nature of the optimisation. The utilisation of a candidate set, which is typically obtained by a discretisation of the design region, for optimisation scales poorly with the input dimensionality. This effect is known as the curse of dimensionality (Bishop, 2007) and limits our method to low-dimensional spaces. A possible extension would be to use a continuous global optimisation method such as genetic algorithms to remove this restriction.

The computational requirement of design generation may be further reduced by approximating the Bayesian criterion integral in Equation (5.4) using more sophisticated approaches than Monte Carlo. One possibility would be the emulation of the integral using a GP. The introduction of another approximation error in the design process however would need further consideration.

6 Applications

CONTENTS

6.1 Introduction

In this chapter, the screening, emulation and optimal design frameworks discussed in the previous chapters are applied to real-world stochastic models. In Section 6.2, the sequential screening described in Chapter 3 is applied to a stochastic rabies model supplied to us by the Food and Environment Reseach Agency (FERA). The most important factors as identified by the screening procedure are compared to published results utilising computationally intensive methods and verified by subsequent emulation.

The optimal design methods discussed in Chapter 5 are applied to two stochastic System Biology models in Section 6.3. The resulting designs are validated through a range of simulation experiments.

6.2 Screening: Rabies Model

In this section we discuss the application of the Morris sequential screening method described on Section 3.2 on a stochastic model provided by the Food and Environment Reseach Agency (FERA) (Singer et al., 2008, 2009).

An overview of the stochastic simulator is given in Section 6.2.1, followed by a description of the screening methodology (Section 6.2.2) and a discussion of the results (Section 6.2.3). The effect of screening on emulation is discussed in Section 6.2.4 and the a summary of the screening results is provided in Section 6.2.5.

6.2.1 Model Description

Although wildlife rabies was eradicated from large parts of Europe, there is a remaining risk of disease re-introduction. The situation is aggravated by an invasive species, the raccoon dog (*Nyctereutes procyonoides*) that can act as a second rabies vector in addition to the red fox (*Vulpes vulpes*). The purpose of the rabies model is to analyse the risk of rabies spread in this new type of vector community (Singer et al., 2008). The individual-based, non-spatial, time-discrete model incorporates population and disease dynamical processes such as host reproduction and mortality rates as well as disease transmission. These processes are modelled stochastically to reflect natural variability (e.g. demographic stochasticity). Thus model analysis (e.g. sensitivity analysis) has to contend with stochastic, indeed heteroscedastic, model output (Boukouvalas et al., 2009).

The model includes two vector species: raccoon dogs and foxes. The model is non-spatial and disease propagation is calculated solely with respect to population dynamics. As depicted in Figure 6.1 the model consists of an input generation phase, the actual calculation of the model

which is implemented in Java and two types of output, time series and summary statistics.

There are 132 individual inputs to the Java code but typically most are varied in a dependent fashion, being separated in 16 different groups. Each individual input has a deterministic relationship with its respective grouping variable. Thus the model can be run in two configurations; in the stand-alone setup 132 inputs can be independently set for the model run, while in the hierarchical setup 16 grouping variables are set from which the individual 132 inputs are generated.

In the experiments that follow only the hierarchical mode is used as this is the setup currently employed by FERA. The grouping variables are shown in Table 6.1. For each input variable, FERA has specified upper and lower bound values which are also shown. For our experimental results to be comparable to Singer et al. (2008) the number of steps input variable was kept fixed to 400 steps, the cross infection input at 0.002 and the area size at 5400 km^2. We therefore allow 13 parameters to vary freely in the simulator.

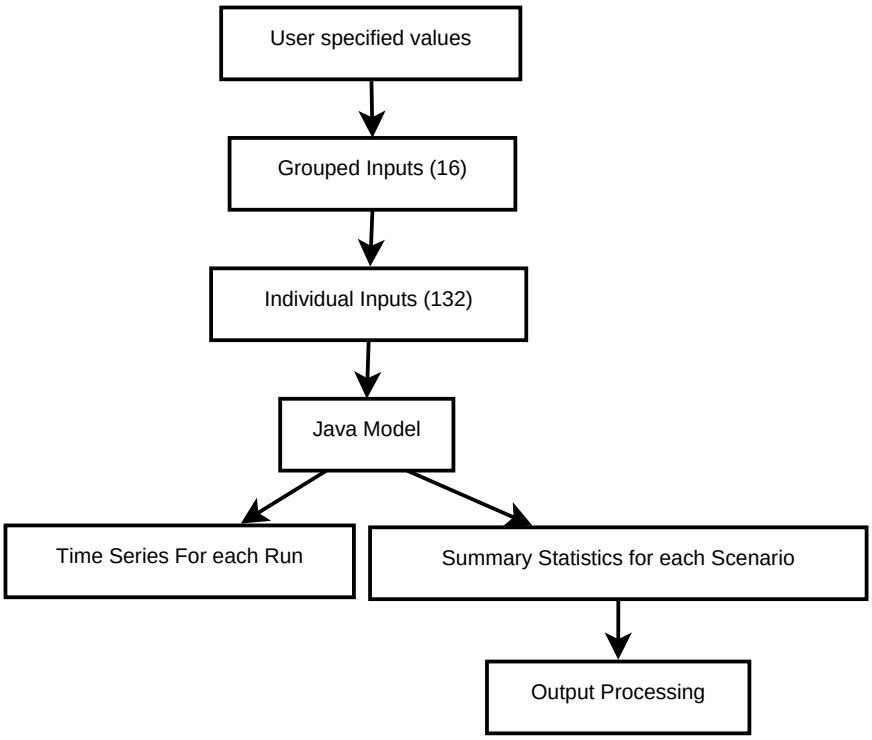


Figure 6.1: Overview of the rabies model.

After the inputs have been generated, the model is run. It is currently implemented in the Java programming language and is relatively computationally cheap to run; each run taking approximately one minute on a recent desktop machine. The exact computational time depends on the 'number of steps' input variable (NumSteps) which determines the maximum number of time steps in a single simulation. The simulation will terminate when the rabies disease becomes extinct in both species populations or the number of steps reaches the maximum specified by NumSteps. Thus the computational runtime depends on the input configuration at which the simulator is run.

Table 6.1: Grouping parameters for the rabies model and their associated Lower and Upper Bounds (LB & UB).

id	*Grouping Parameter*	*Description*	LB	UB
1	NumRuns	Number of repetitions of the experiment at a specific parameter setting.	200	300
2	FoxStableDensityWin	Fox population winter density (individuals/km^2)	0.1	0.5
3	RacStableDensityWin	Raccoon Dog population winter density (individuals/km^2)	0.1	1
4	RacInfProb1	Shape parameter for the probability distribution of raccoon dog infection	0.39	0.47
5	dummy	Dummy variable with no influence	0.9	1.1
6	fox.death	Fox population mortality	0.9	1.1
7	rac.death	Raccoon Dog population mortality	0.9	1.1
8	win.hunting.prop	Winter hunting proportion	0.9	1.1
9	fox.birth	Fox population birth rate	0.9	1.1
10	rac.birth	Raccoon Dog population birth rate	0.9	1.1
11	fox.inf	Fox population infection rate	0.9	1.1
12	fox.rabid	Fox population rabies individual density	0.95	1.05
13	rac.rabid	Raccoon Dog population rabies individual density	0.95	1.05
14	cross.inf	Cross infection rate	0.002	0.002
15	NumSteps	Length of simulation run	400	400
16	AreaSize	Area size (km^2)	5400	5400

For input regions where the disease becomes extinct quickly, the simulator is quick to evaluate while in other input regions, the simulator can take a maximum runtime determined by the NumSteps input variable. In the future, design for screening and emulation could take into account this input-dependent simulator cost.

The model itself is stochastic in nature and thus in our analysis each simulation is repeated multiple times to enable estimation of the stochastic process. The output of each simulation is a time series data file, the i^{th} row corresponding to the state of the system at the i^{th} time step.

At the end of each simulation, summary statistics are calculated for the time series data and stored as a single row in a scenario file. The latter contains one row for each repetition of a simulation for a given set of inputs. The time series data has not been investigated at this point for the purposes of screening and remains an open research area.

The summary outputs are further processed and the output that measures the probability that the rabies disease becomes extinct in both species after 5 years is used for subsequent analysis (Both.Inf.percent.ext.5years). This output is important in deciding on the response to a potential rabies outbreak since it indicates the risk of long term rabies disease persistence (Singer et al., 2008, 2009).

The probability is calculated by first measuring the time that it took the disease to become extinct in both species. Provided the disease started in the raccoon dog population and went extinct in both populations during the run the formula used is: max(RacInfExtTimeLast, FoxInfExtTimeLast)-RacInfFirstInt. The output RacInfFirstInt records the time step when the rabies infection in the

raccoon dog population commenced. Note that this formula can take the value Not Available (NA) if the disease did not become extinct in the maximum number of steps allowed. The probability is then calculated by measuring the frequency of instances for a single scenario that are less than 20 time steps since disease introduction (1 time step equalling 3 months bringing the total to 5 years). Thus the output Both.Inf.percent.ext.5years is available only once for each scenario and cannot hold the NA value by its definition (if for all the runs in the scenario the disease did not become extinct the probability will be 0). Furthermore, to calculate the probability of disease extinction to a high accuracy requires multiple repetitive runs of the simulator for a fixed parameter set. The number of repetitions is determined by the NumRuns model parameter and the dataset generation for a single output can now be computationally demanding.

The probability output for the two dominant inputs and averaged over all others is shown in Figure 6.2.

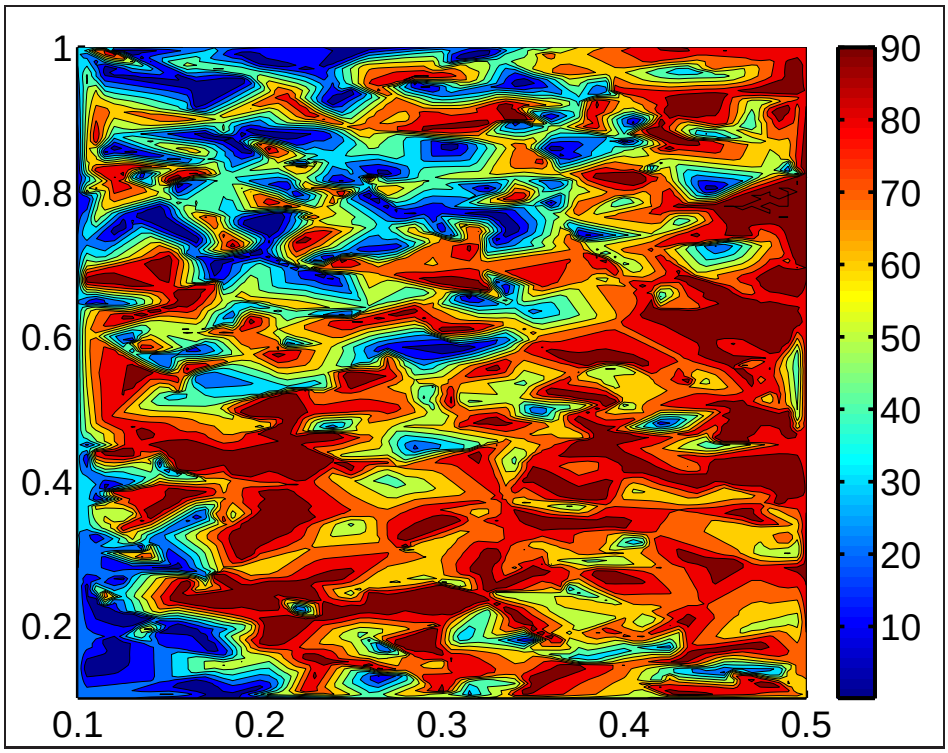


Figure 6.2: Probabilistic Output of Rabies simulator using inputs Fox winter density (X axis) and Raccoon Dog winter density (Y axis) and averaging over all others.

6.2.2 Screening Methodology

Singer et al. (2008) performed sensitivity analysis on the rabies model using a variety of standard sensitivity analysis methods. In particular, results have been obtained using the Morris (Section 3.1.3) and Sobol' (Section 3.1.2) methods.

In order for our experimental results to be directly comparable we have used the same setup

as in Singer et al. (2008). The setup used is based on a Morris design with number of trajectories $R = 20$ resulting in $(k+1) \times R = 14 \times 20 = 280$ simulator evaluations, where k is the number of input variables. Two experimental designs are employed in the experiments, a Morris standard design without clustering and a Latin Hypercube design. The latter design is used to validate the emulators constructed on the screened input factors. For the Morris screening method we use the mean μ_* of the absolute values of the Elementary Effects to rank the input factors. Following Singer et al. (2008) we set the number of levels to $p = 6$.

The sequential Morris method (Section 3.2) requires the specification of a variance γ from which the threshold on the Elementary Effect deviation σ_0 is derived. We set $\gamma = 3.5$ which reflects a prior belief that individual factor effects on the output are considered near-linear if the effect on the output is within three standard deviations of purely linear, i.e. $\pm 3\sqrt{\gamma} = 5.6$. Since the output is hard bounded in the range $[0, 100]$ a factor has near-linear effect if the output varies no more than 5.6% from linear. This variability encapsulates both the internal variability of the stochastic model and our prior definition of a near-linear effect.

6.2.3 Screening Results

Singer et al. (2008) performed sensitivity analysis on this model using the standard Morris method with the same setup as here as well as the Sobol' method, described in Section 3.1.2. They noted the most important parameters are species winter densities and mortalities. They also noted the least influential factors are the dummy variable that has no explicit influence on the model output and RacInfProb1, a shape parameter for the probability distribution of raccoon dog infection. It is also noted that the Sobol method is prohibitively expensive and offers low accuracy with a sample size of 300. They suggested increasing the sample size and reducing the dimensionality of the problem by fixing some of the factors to their nominal values. For expensive simulators this motivates the usage of the Morris method.

The standard Morris method variable ranking with $R = 20$ trajectories is presented in Figure 6.3(a) and agrees with the results of Singer et al. (2008) where the four dominant factors were found to be the winter densities for both species and the associated mortality rates. Singer et al. (2008) conclude that both Sobol and Morris methods show that the main effects are not sufficient to characterise the parameter sensitivity in this model, i.e. the dominant factors have strong non-linear and interaction effects which is reflected in the high σ value in the Morris method.

The sequential Morris method is initialised with $R = 2$ trajectories on all 13 factors requiring $(k+1)R = 28$ simulator runs. We note that since the same jump, Δ, is used for all factors, the computed threshold is the same for all factors. The Morris plot with the associated threshold value is shown in Figure 6.3(b). Two factors are significantly over the threshold, the Raccoon Dog

winter density (3) and the Raccoon population rabies individual density (13), and are eliminated from further consideration since they have strong non-linear effects on the simulator output.

Another trajectory design for the remaining 11 factors is evaluated and requires 12 further simulator evaluations (Figure 6.3(c)). The NumRuns (1), Fox winter density (2) and Raccoon Dog population birth rate (10) parameters are found to have non-linear effects and are removed from further consideration. As evidenced by the Morris plot, the σ value for parameter 10 changed significantly from the previous step where the effect was considerably below the threshold and very close to linear.

For the third step, the eight-factor trajectory requires 9 more simulator evaluations (Figure 6.3(d)). Four further parameters are eliminated, the fox (6) and raccoon dog (7) mortality rates, the fox birth rate (9) and the fox population rabies individual density (12) where large changes in the moments of the elementary effects are again observed due to the increased accuracy from the increased Morris design size.

No more factors are eliminated until step 7 requiring a further $(4+1)\times 4 = 20$ simulator evaluations (Figure 6.3 (e)-(h)). At step 7 the winter hunting proportion (8) and fox population infection rate (11) parameters are removed from further consideration.

The remaining two factors, the shape parameter for the probability distribution (4) and the dummy variable (5) are found to be below the σ threshold for all subsequent twelve steps requiring $(2+1)12 = 36$ simulator evaluations.

The total number of simulator evaluations for the sequential procedure is 105 compared to the 280 evaluations required by the standard batch Morris method with $R = 20$ trajectories.

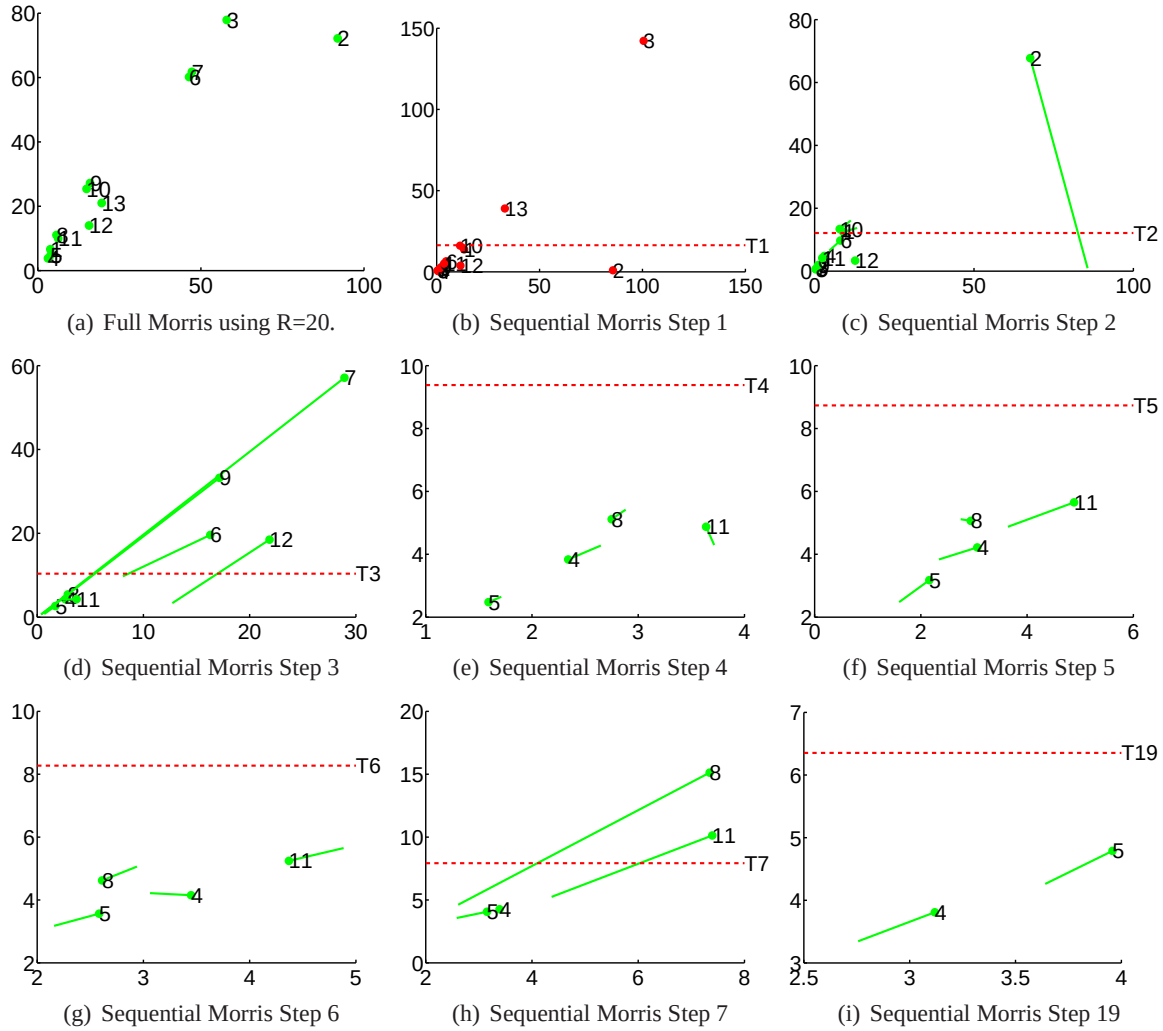
We have also performed the threshold calculation on the full Morris set with $R = 20$ trajectories and the same factors as with the sequential version are identified as near-linear.

In summary, the sequential Morris method for the Rabies model has been successfully used to identify factors with no or near-linear effects on the simulator response at a significant savings to the standard Morris method.

6.2.4 Standard Gaussian Process Emulation

In this section, the screening results are further examined by performing emulation on the rabies simulator using different sets of input factors. In particular the predictive performance of the following configurations is compared:

- *All*. The entire set of thirteen input factors is used.
- *Low Order*. The shape parameter for the probability distribution (4) and the dummy variable (5) factors identified by the sequential procedure as having near-linear effects are discarded.



(a) Full Morris using R=20. (b) Sequential Morris Step 1 (c) Sequential Morris Step 2

(d) Sequential Morris Step 3 (e) Sequential Morris Step 4 (f) Sequential Morris Step 5

(g) Sequential Morris Step 6 (h) Sequential Morris Step 7 (i) Sequential Morris Step 19

Figure 6.3: Morris Screening on Rabies simulator. X axis is μ_* and Y axis σ of Elementary Effects. Green line denotes path from previous value of (μ_*, σ) for each factor. Horizontal dashed red line denotes the σ_0 threshold value for the given step.

- *Neg Low Order*. Only the shape parameter for the probability distribution (4) and the dummy variable (5) factors are used.

- *High Order*. Only the four dominant factors are used by the emulator. As identified in the previous section these are the raccoon dog and fox population winter density factors (2,3) and their associated mortality rates (6,7).

- *Neg High Order*. The remaining eight factors are used by the emulator.

The emulator used is a zero-mean Gaussian Process with a fixed differentiability $\nu = 5/2$ Matérn kernel and a single nugget. A nugget is used as the simulator output is stochastic.

The emulators were trained on three instances of Morris designs of 280 model observations ($R = 20$), each instance being validated using a separate Latin Hypercube test set of 280 points. The Morris designs are used for training since model observations obtained during screening should be usable in subsequent stages of emulation.

The validation results for all five input configurations are shown in Table 6.2 in terms of the Mahalanobis and RMSE scores. In order to compute the probability output at each design point, multiple repetitions of the simulator output are required. Therefore due to the computational expense only six designs have been used, three training and three test sets, to generate three realisations of the experiment. The validation measures have been averaged over the realisations of the experiment.

When discarding the two parameters identified in the sequential procedure (Low Order) the emulation predictive performance actually improves. Eliminating unimportant factors benefits the emulation as the inference is performed on a lower-dimensional space which can be critical especially when a relatively small training set is available.

When considering only the two unimportant factors (Neg Low Order) the predictive performance both in terms of RMSE and Mahalanobis score deteriorates sharply signifying the inability of the emulator to predict the simulator output using these inputs.

For the High Order scenario, in terms of RMSE there is little loss of accuracy from not including the eight least influential simulator factors in the analysis. The Mahalanobis score shows a loss of predictive power which is however much greater when the four dominant factors are discarded (Neg High Order).

We note that for factors with linear effect on the simulator output, as reflected by a high mean μ_* and low deviation σ values in the Morris procedure (Section 3.1.3) a preprocessing step whereby the linear effect is removed from the output could be utilised to further minimise the impact of discarding such factors from subsequent emulation. This is discussed in Section 3.3.

Table 6.2: Validation of Emulators on Rabies model using different sets of input factors as described in Section 6.2.4. Mean and 2 standard deviations are shown based on three realisation of the experiments. Emulators trained on Morris design and evaluated on a Latin Hypercube design of the same size (280 points).

Input Set	Mahalanobis (280)	RMSE
All	381.9 ± 145.12	28.65 ± 4.91
Low Order	274.2 ± 59.3	27.9 ± 5.62
Neg Low Order	1236.9 ± 1495.5	32.5 ± 1.34
High Order	559.8 ± 136.3	29.42 ± 5.15
Neg High Order	2349 ± 1183.26	33.75 ± 3.14

In Boukouvalas et al. (2009) and Boukouvalas and Cornford (2009) the number of steps to disease extinction output of the rabies simulator has been emulated using the Coupled Model described in Chapter 4 as, unlike the probability of disease extinction output, the variance is heteroscedastic. Therein, it was found empirically that the Coupled Model utilising Latin Hypercube training designs with replicate observations obtained better validation results than without replication. The heuristic nature of the design generation motivated us to examine model-based optimal design (Chapter 5), an application of which is discussed in the next section.

6.2.5 Conclusions

In Section 6.2 the application of the sequential screening method described in Section 3.2 to the rabies model has demonstrated the effectiveness of the method on a real-world high-dimensional stochastic simulator. The number of required model evaluations was considerably less than would be required from a straightforward application of the standard Morris method with the same number of trajectories.

The only requirement of the method is the definition of near-linear effects via the specification of the γ variance parameter. In the case of stochastic simulators, this parameter includes the internal simulator variability whereas for deterministic simulators near-linear definitions only include errors due to machine precision and degree of departure from truly linear effects on the output.

Lastly, the effect of screening on emulator performance was examined in Section 6.2.4 where discarding the two factors identified as near-linear actually improves predictive performance and using the four dominant input factors incurred a small loss of predictive accuracy compared to a full model with a much higher degree of complexity. Such lower-dimensional representations simplify subsequent stages of the analysis such as optimal design generation.

6.3 Optimal Design: Systems Biology Simulators

In this section we present a case study of systems biology models with the aim of demonstrating the emulation and design aspects discussed in Chapters 4 and 5 respectively.

In Section 6.3.1 an introduction to stochastic modelling for systems biology is provided, followed by a review of existing work on emulation of such systems in Section 6.3.2. In Sections 6.3.3 and 6.3.4 the experimental results on emulation and design are presented for two systems biology models, the Dimerisation Kinetics and Prokaryotic Auto-regulatory Network models. A discussion of the results and directions for future research are presented in Section 6.3.5.

6.3.1 Introduction to Systems Biology Modelling

In this section we provide a short introduction to the stochastic simulation of systems biology models. The discussion is heavily based on the book by Wilkinson (2006) to which the reader is referred for further details.

The traditional method of modelling the kinetics of biological processes is via the solution of a deterministic system of differential equations. However at the intra-cellular level the kinetics are inherently stochastic and cellular functions cannot be properly understood without explicitly modelling that stochasticity.

The deterministic approach to kinetics fails to capture the discrete and stochastic nature of the molecular reactions involved, especially when at least some of the reactant molecules appear at low concentrations where stochastic variability can dramatically change system behaviour away from the deterministic solution.

The approach we follow is to perform exact stochastic simulation of a system of molecular reactions. A reaction R_i may be represented as:

$$X_1 + X_2 + \cdots + X_u \xrightarrow{k_i} Y_1 + Y_2 + \cdots + Y_n,$$

where X_* are the molecules reacting, known as reactant species, k_i is the rate constant which is linked to the likelihood of the reaction occurring and Y_* are the resulting molecules.

Provided the volume and temperature are fixed and the distribution of molecules is uniform, the probability of the reaction occurring, known as the hazard, is provably constant. The rate constant k_i associated with a reaction R_i is necessary but not sufficient to describe the hazard of a reaction. The hazard also depends on the quantities of the reactant molecules and can be written as $h_i(x,k_i)$ where $x = (x_1, x_2, \ldots, x_u)$ the current state of the system, i.e. the number of molecules for each reactant species.

Formally, conditional on the state being x at time t, the probability that a reaction R_i will occur in the interval $(t, t+dt]$ is given by $h_i(x,k_i)dt$.

The exact form of h_i depends on the *order* of the reaction.

A zeroth-order reaction is of the form:

$$\emptyset \xrightarrow{k_i} X.$$

The hazard function depends solely on the rate parameter $h_i(x,k_i) = k_i$. In some cases, it is convenient to model an external influx of molecules via a zeroth-order reaction.

A first-order reaction is of the form:

$$X_j \xrightarrow{k_i} ?$$

where ? is any outcome. Given x_j number of X_j molecules, the probability of each molecule reacting is k_i. Therefore the probability of any one reacting is the combined hazard $h_i(x,k_i) = k_i x_j$.

A second-order reaction is of the form:

$$X_j + X_k \xrightarrow{k_i} ?$$

There exist $x_j \times x_k$ different pairs of molecules that may react. Therefore the combined hazard is $h_i(x,k_i) = k_i x_j x_k$. A special type of second-order reaction is:

$$2X_j \xrightarrow{k_i} ?$$

In this case only $x_j(x_j-1)/2$ pairs of molecules may react and the combined hazard becomes $h_i(x,k_i) = k_i x_j(x_j-1)/2$.

Most higher-order reactions can be modelled as a set of second-order reactions which quite often is chemically more realistic and may result in different dynamics compared with modelling the higher-order reaction directly. A set of second-order reactions is generally believed to be more biologically plausible as reactions where three or more species react simultaneously are rare (Wilkinson, 2006).

For simple systems where all reactions in a system are zero and first-order mass action kinetics, the deterministic solution will correctly describe the expected value of the stochastic kinetic model (Section 6.7 of Wilkinson (2006)). However no estimate of the variability will be available and this link fails for systems with higher-order reactions.

Since the hazards only depend on the current state of the reaction systems, the dynamics can be modelled as a continuous-time Markov process with a discrete state space. Detailed mathemat-

ical analysis of such systems is usually not tractable but stochastic simulation of the dynamics is straightforward. The Gillespie algorithm is one option to perform simulation from such systems and is described in Algorithm 6.1.

Algorithm 6.1 Description of the Gillespie algorithm for exact simulation of stochastic systems.

Gillespie Algorithm

Input: U reactants with initial concentrations $\mathbf{X}_0 = \{x_1, \ldots, x_u\}$, V reactions $\mathbf{R} = \{R_1, \ldots, R_V\}$ with rate constants $\mathbf{k} = \{k_1, k_2, \ldots, k_V\}$, number of timesteps T.
Output: Time series of state vector $\mathbf{X}_T$.

A. *Iterate until the number of timesteps exceeds threshold T.*

1. For each reaction R_i calculate its hazard $h_i(x, k_i)$.
2. Calculate the combined hazard of any system reaction occurring $h_0(\mathbf{X}, \mathbf{k}) = \sum_{i=1}^{V} h_i(\mathbf{X}, k_i)$.
3. Simulate the time to next event t' by sampling from an exponential distribution with $\lambda = h_0(\mathbf{X}, k)$.
4. Move the current time to $t = t + t'$.
5. Probabilistically select which reaction will occur by sampling from a discrete distribution with probabilities $h_i(\mathbf{X}, k_i)/h_0(\mathbf{X}, \mathbf{k})$ for $i = 1, 2, \ldots, V$.
6. Update the current state $\mathbf{X}$ according to the selected reaction and append it to the time series $\mathbf{X}_T$.

6.3.2 Existing Work

Emulation of systems biology stochastic systems is a relatively new research area and we are only aware of the work of Henderson et al. (2009a) that directly tackles the emulation of these biologically inspired models.

Henderson et al. (2009a) present a method to perform emulation and calibration of a five reaction system that describes mitochondrial DNA deletions in Substantia Nigra neurons. Due to conservations laws in the system only two rate parameters are needed. The authors state that although exact Bayesian inference via Markov Chain Monte Carlo can still proceed in theory, using simulations from the biological model, it becomes infeasible in practice due to the computationally demanding simulator. They propose to replace the simulator by an inexpensive statistical surrogate, an emulator. Due to the heteroscedasticity of the simulator variance, the proposed emulator is an independent set of two GPs that model the mean and log standard deviation of the response respectively. Both GP priors are set with a constant mean and a squared exponential kernel with a nugget parameter.

The authors tackle the design by heuristically combining three designs (Henderson et al., 2009b). The design space is four-dimensional, the two rate parameters, a threshold value for cell death and the age of the individual. A 250-point design is constructed from the following combination of designs:

- A 2^4 factorial design on the extreme points of the input space. An overview of factorial designs is given in Section 2.3.

- A Cartesian product of an 8-point Latin Hypercube on the first three parameters with a 13 unique value design for the age parameter where observational data are available

- A 130-point design consisting of a random sample from the prior distribution of the rate and threshold parameters and a corresponding sample from a uniform distribution of integer values for the age parameter.

The authors state this composite design aims to give good coverage over the support of the prior distribution, i.e. provide an emulator that is a good approximation to the simulator across the whole parameter space. Furthermore, it is hoped that the large number of inter-point distances available in this design will be of benefit when estimating GP parameters.

The simulator is run for each of the 250 input configurations 1000 times to obtain replicated observations. In some cases the simulation experiment concludes prematurely due to cell death and these runs are discarded. The authors do not utilise the bias correction due to the log transformation of the sample variance described in Section 4.4.1.

In order to speed up computation two GP emulators are trained on the mean and log standard deviation of the replicated observations where to achieve robustness in the estimation of the sample moments, the authors further restrict the training design to points where at least four replicates observations resulted in successful runs. The final design is 171 points.

Finally uniform priors are assigned to all kernel parameters and a two step MCMC scheme is used to perform emulation and calibration. The authors mention two further simplifications they have found had little impact on the performance of their scheme. The MAP estimate is used for the GP hyperparameters and only the mean predictions of the mean and variance GPs are utilised in the MCMC inference.

In the case study we present, the focus is exclusively on the design and emulation aspect of the analysis and calibration is not considered. The emulation models we use also utilise the moments of replicated observations but the mean and variance models are coupled for better predictive performance. Since the emphasis is on design, the variance model we use is simpler than a full GP although in Section 6.3.4.3 we present emulation results based on the latent GP model which is similar to a GP model for the log variance.

The design question is approached using the Fisher information criterion to obtain an optimal design for parameter estimation rather than heuristically constructing a design with a large set of inter-point distances.

6.3.3 Dimerisation Kinetics

The first model simulates the dimerisation kinetics reversible reaction (Wilkinson, 2006). It consists of two reactions:

$$2P \xrightarrow{k_1} P_2$$
$$P_2 \xrightarrow{k_2} 2P$$

The two reactants are the proteins P and P_2. The rate parameters k_1 and k_2 control the probability of the reaction firing.

Given the initial conditions, i.e. the number of P and P_2 molecules, and values for the rate constants k_1 and k_2 the ordinary differential equation describing this model can be solved analytically to describe the full dynamical behaviour of the system. The focus in this section is on emulation of the stochastic simulator under uncertain rate values.

The Gillespie algorithm described in Section 6.3.1 is used to simulate from this model. The input domain space is two-dimensional with the domain for $k_1 \in [0.0005, 0.03]$ and $k_2 \in [0.0005, 0.5]$. The initial number of molecules were set to $P = 301$ and $P_2 = 0$. The model was run to time $T = 10$ with step $dt = 0.1$.

The mean and standard deviation of the simulator across the input domain is shown in Figure 6.4.

The distribution of the output at a point for this model is approximately normal as discussed in Wilkinson (2006) so we expect the mean and variance to describe the output reasonably well. Simulation results confirming the normality approximation are shown in Figure 6.5.

The near-normal output and the input dependency of the model variance motivates the usage of the heteroscedastic emulation methods.

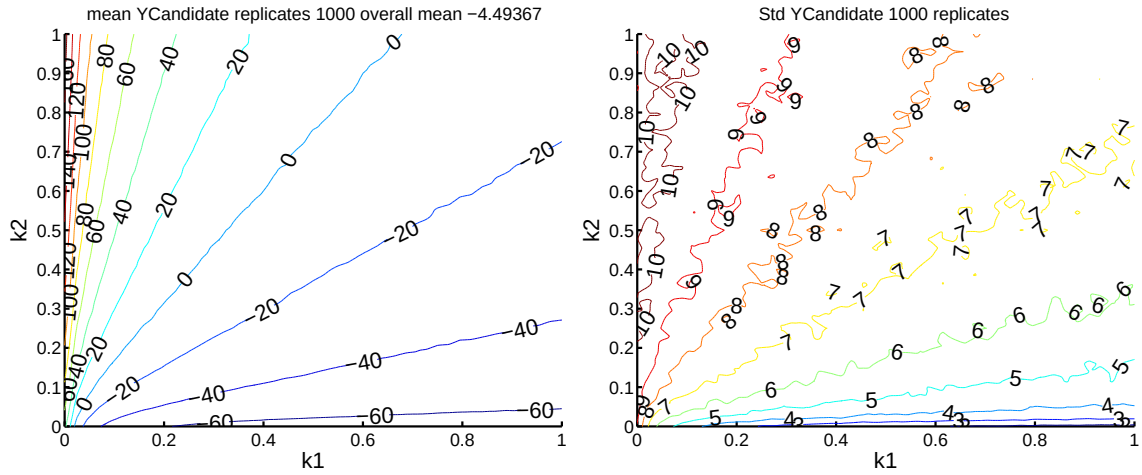
More details on this model can be found in Wilkinson (2006).

6.3.3.1 Design and Emulation Results

As the variance plot of the simulator (Figure 6.4(b)) shows, a linear variance model would appear to be appropriate.

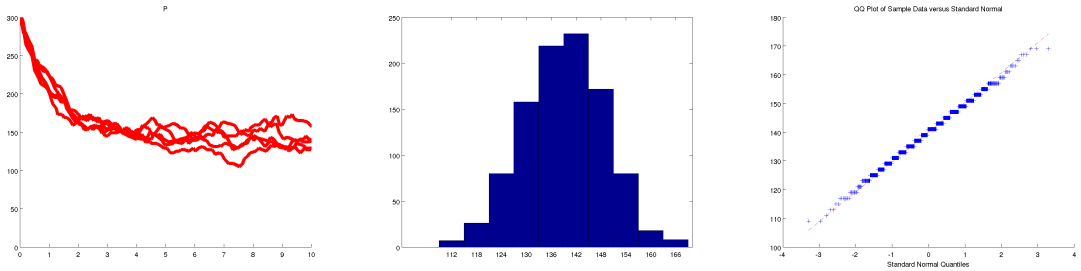
We discretise the input space into a grid of 2025 candidate points from which we wish to select a 30-point design allowing for replication. A Latin Hypercube design of 2025 design points with 1000 replicates at each design point is used to validate the emulator.

The model used both in design and emulation is a zero-mean GP prior with a Matérn kernel with fixed differentiability $\nu = 5/2$ and a log-linear function for the variance. A non-stationary



(a) Mean (b) Standard Deviation

Figure 6.4: Mean and standard deviation of dimerisation model at time step 10, using 1000 realisations on a grid of 2025 input points. The inputs are the rate parameters and the initial conditions are kept fixed.



(a) 5 realisations of dynamics. X axis signifies time steps. (b) Histogram of values at $T = 10$. (c) QQ-plot of values at $T = 10$.

Figure 6.5: Dynamics from the Dimerisation Kinetics model showing 10 realisations. Initial conditions $P = 301$, $P_2 = 0$ and $k_1 = 1.66e-3$, $k_2 = 0.2$. Output plotted is number of P molecules. Also plotted histogram and QQ-plot of 1000 realisations of model output at time step $T = 10$ which can be seen to be approximately normal. The QQ-plot shows the Standard Normal Quantiles (X-axis) vs the Quantiles of the input sample (Y-axis).

GP prior with non-constant mean function would be more appropriate as the exploratory plots of the simulator output in Figure 6.4 suggest. However elicitation and design for such a model is more complex and could be pursued as a future research direction. A discrete prior is placed on all the model parameters; for the length-scale $\lambda \in (0.2, 0.6, 1.5)$, the process variance $\sigma_p = 1$ and the variance coefficients $\beta_1 \in (-4.6, -2.3)$, $\beta_2, \beta_3 \in (-1.6, 1.6)$. The prior is constructed to allow for short and long length-scale values, a range of noise levels and slopes for the variance function. Following Zhu and Stein (2005) a point prior is set for the process variance since only the signal-to-noise ratio is of important for model-based design.

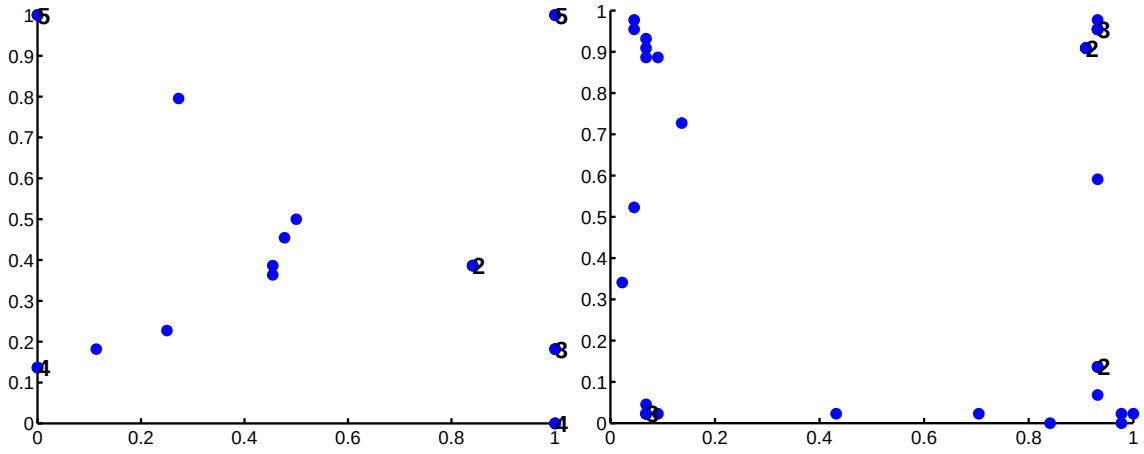
The validation results are obtained using Maximum Likelihood (ML) optimisation without reference to the discrete prior. This was done to separate the impact of the design on the ML parameter estimates from the impact of the prior on the generated Fisher designs.

Lastly the model is fit on the entire candidate set with four replicates at each design point to obtain a reliable estimate of the model hyperparameters which are treated as the true values for the purposes of calculating the RMSE of the ML parameter estimates of the different candidate designs. The estimated parameter values from utilising the entire candidate set are $\lambda = 3.8$, $\sigma_p = 3 \times 10^8$, $\beta_1 = 3.9$, $\beta_2 = -0.99$ and $\beta_3 = 0.99$.

The Fisher-generated design are shown in Figure 6.6, where both Greedy and Simulated Annealing optimisation methods result in the placement of a significant proportion of the design points on the corners of the design space. As the variance model used in the emulator is linear, this is consistent with traditional optimal design where the variance of the coefficients of a linear model is minimised.

The predictive validation results are shown in Table 6.3. In terms of Mahalanobis score the non-replicate designs do badly with significant high values in the tail of the distribution which skew the mean values. The RMSE score is worse for the Greedy design whose Mahalanobis score is closest to optimal which leads to the conclusion that the variance prediction is the source of the predictive improvement. The replicated Latin design achieves the smallest RMSE. We note here that as was described in Section 5.8, the Dawid score is not included as it provides an identical emulator ranking under such large differences in the Mahalanobis error.

The predictive performance results can be better understood by examining the parameter accuracy of the designs. In terms of relative RMSE on the parameters (Figure 6.7), the biggest differences are observed in terms of the estimation of the noise variance terms β_1 and β_2. Of note is also the error for the length-scale parameter when the inference is done under the Greedy design. The Fisher information and the empirical log determinant of ML estimates are calculated using the logarithm of the length-scale parameter while the relative RMSE is computed on the natural space. For this reason larger differences in RMSE may not correspond to large differences in the



(a) Greedy (b) Simulated Annealing

Figure 6.6: Fisher Designs Produced for the Protein Dimerisation model.

empirical log determinant.

The Fisher information score for each design is provided in Figure 6.8 and we see a clear correspondence to the empirical log determinant. The fact that all replicate designs achieve similar Fisher scores may stem from either optimisation getting trapped in local minima for the Greedy and Simulated Annealing schemes or from the fact that uniform replicate designs are close to optimum for this choice of prior. Due to the good performance of the Greedy/SA algorithms in our previous experiments we believe the latter is more likely in this scenario.

In summary, in terms of both parameter accuracy as reflected in the empirical log determinant and prediction accuracy the replicate designs outperform the non-replicate designs.

Table 6.3: Mean and standard deviation of the Mahalanobis score (2025) and RMSE for the Protein Dimerisation model. 1000 realisations of the experiment were used.

Design	Mahalanobis	RMSE
Greedy	2866.28 ± 1476.09	23.14 ± 3.24
Replicate Grid	3985.77 ± 2346.76	20.28 ± 2.04
Grid	$38 \times 10^6 \pm 355 \times 10^6$	15.65 ± 2.71
Replicate LH	3399.83 ± 1873.07	13.91 ± 2.38
LH	$7 \times 10^6 \pm 82 \times 10^6$	14.13 ± 2.09
Simulated Annealing	3704.42 ± 2261.49	16.65 ± 3.05

6.3.4 Prokaryotic Auto-regulatory Network

The simulator used in this section describes a simple gene expression auto-regulation mechanism often present in prokaryotic gene networks. It is composed of five reactant species, the gene g, protein P and its dimer P_2, and the mRNA molecule. The eight reactions complete the specification of the model:

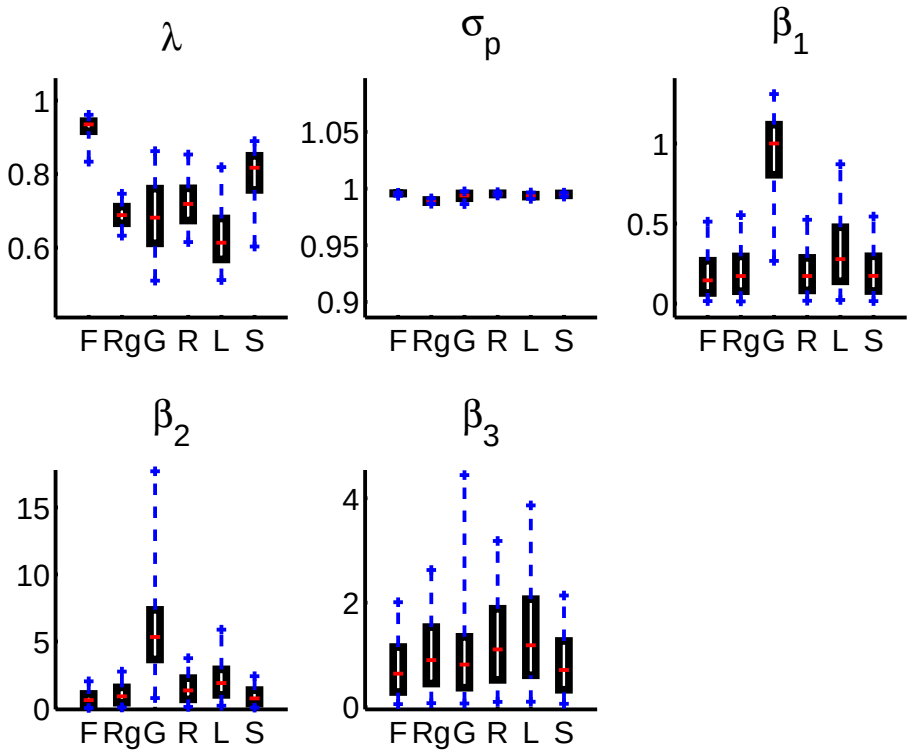
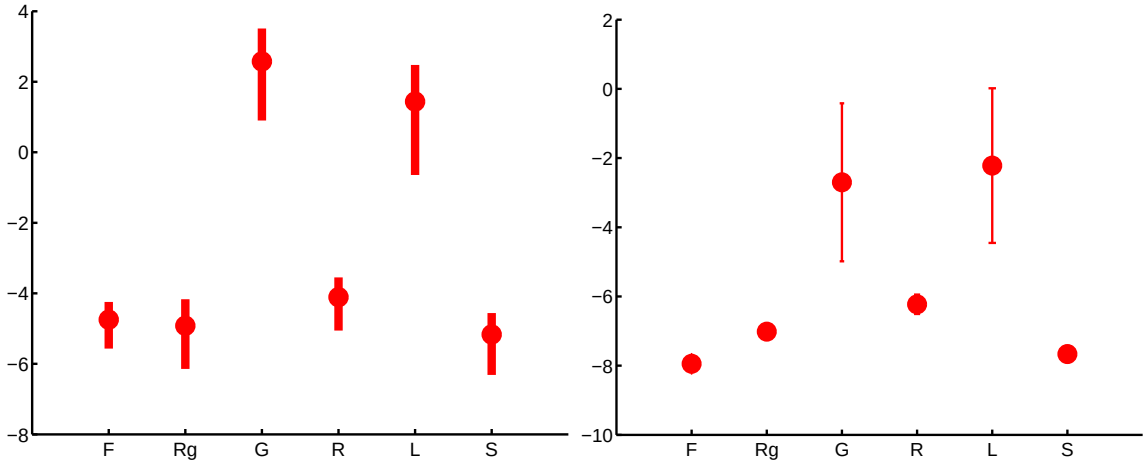


Figure 6.7: Parameter Estimation Relative RMSE for the Protein Dimerisation model. F=Greedy design, Rg=Replicate Grid, G=Grid, R=Replicate Maximin Latin Hypercube, L=Maximin Latin Hypercube.



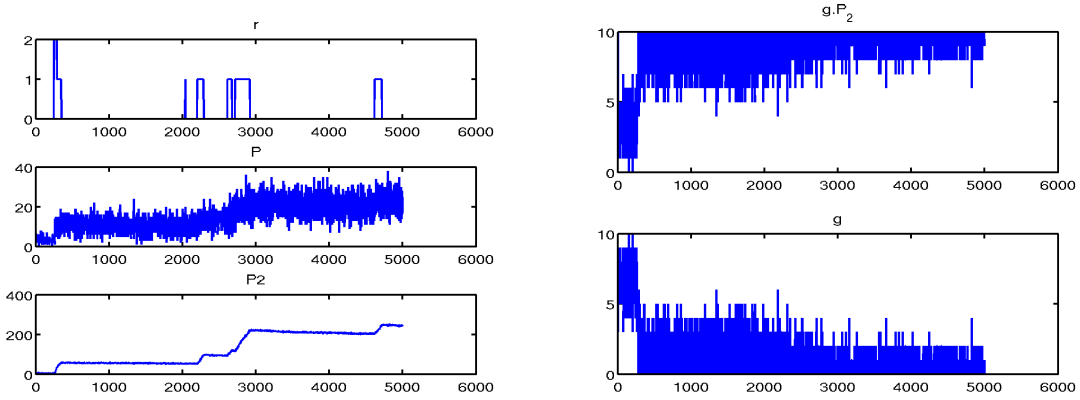
(a) Log Determinant (b) FIM

Figure 6.8: Parameter Fisher score and Empirical Log Determinant for the Protein Dimerisation model.

$$
\begin{aligned}
g + P_2 &\xrightarrow{k_1} g.P_2 && \text{Repression} \\
g.P_2 &\xrightarrow{k_2} g + P_2 && \text{Reverse Repression} \\
g &\xrightarrow{k_3} g + r && \text{Transcription} \\
r &\xrightarrow{k_4} r + P && \text{Translation} \\
2P &\xrightarrow{k_5} P_2 && \text{Dimerisation} \\
P_2 &\xrightarrow{k_6} 2P && \text{Dissociation} \\
r &\xrightarrow{k_7} \emptyset && \text{mRNA degradation} \\
P &\xrightarrow{k_8} \emptyset && \text{Protein degradation}
\end{aligned}
$$

Dimers of the protein P (P_2) coded for by the gene g repress their own transcription by binding to a repressive regulatory region upstream of g. This model is minimal in terms of biological detail included but contains many of the interesting features of an auto-regulatory feedback network (Wilkinson, 2006).

Figure 6.9 shows the dynamics for all five species for 5000 time steps. The mRNA transcript events producing the reactant g are comparatively rare and random in their occurrence (top Figure 6.9(a)). The number of protein dimers P_2 jumps abruptly at random times and coincides with the mRNA transcription events. So even though there exist a large number of protein dimer molecules, their behaviour is strongly stochastic due to the fact they are affected by the number of mRNA transcripts which are few in number (Wilkinson (2006) page 173). Due to this inherent randomness, a continuous deterministic model will not adequately capture its behaviour and stochastic simulation is justified in the analysis of this model.

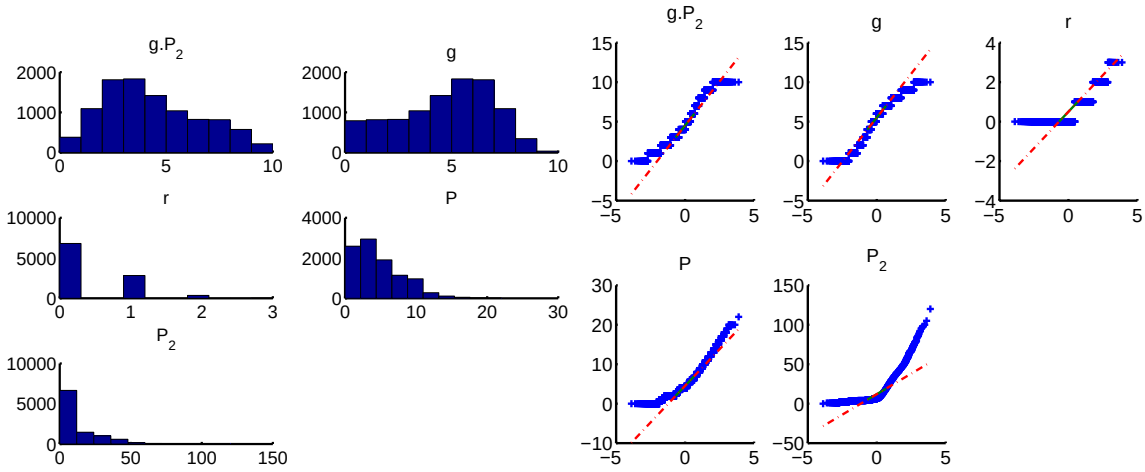


(a) r,P,P_2 (b) $g.P_2$,g

Figure 6.9: Dynamics for all five species up to time T=500, dt=0.1 of the Prokaryotic Auto-regulatory Network. Initialised with $g.P_2 = 10$, $g = r = P = P_2 = 0$ number of molecules.

Figure 6.10 shows histograms of the empirical distributions of all species. Due to the approximate normality of species $g.P2$, it is utilised for the design and emulation experiments that follow

in next section.



(a) 10 step histogram (b) QQ-plot

Figure 6.10: Histograms of all five species taken at time T=10, dt=0.1 using 10,000 realisations for the Prokaryotic Auto-regulatory Network. The QQ-plot shows the Standard Normal Quantiles (X-axis) vs the Quantiles of the input sample (Y-axis).

In Section 6.3.4.1 different design strategies are evaluated using multiple realisations of the experiment. In Section 6.3.4.2 a specific validation instance is examined for the Greedy and Grid designs to better understand the sources of design error and in Section 6.3.4.3 the impact on design performance of a more complex variance model is examined.

6.3.4.1 Design and Emulation Results

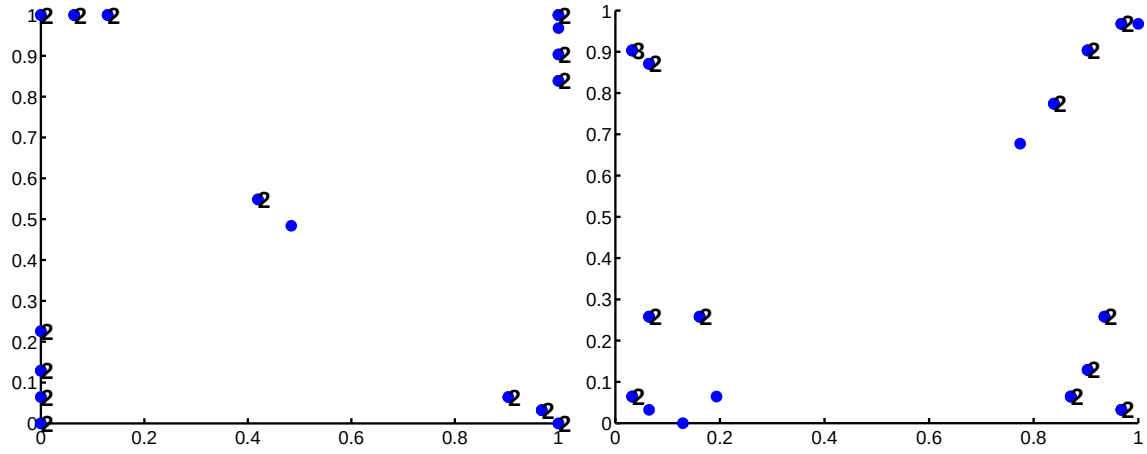
In this section the performance of six different designs are compared both in terms of predictive performance and parameter estimation error.

A single nugget variance model is used as a more complex model would require the specification of prior beliefs for parameters whose effect on the response is complex and does not facilitate elicitation. Furthermore, as was demonstrated in Chapter 5 an incorrect prior specification on a complex model can lead to very inefficient designs.

Thirty points are selected from a candidate set of 1024 points. We use 2025 test points and perform 500 realisations of the experiment. An exponential kernel with a single nugget variance model is used. The prior used is: $\lambda^2 = (0.1, 5, 10)$, $\sigma_p = (1, 3, 5)$ and $\tau = (0.1, 0.5, 4)$ which allows a wide range of noise levels and correlation length-scales.

As before the Greedy and Simulated Annealing designs obtained through the minimisation of the Fisher score are compared to Grid and Maximin Latin Hypercube designs with and without replicate observations. The Fisher designs are shown in Figure 6.11.

As in the Protein Dimerisation model study (Section 6.3.3), parameter accuracy is estimated by treating as true the hyperparameters values inferred when leveraging the entire candidate set



(a) Greedy (b) SA

Figure 6.11: Fisher Designs obtained for the Prokaryotic Auto-regulatory Network.

with four replicates at each site as the training set.

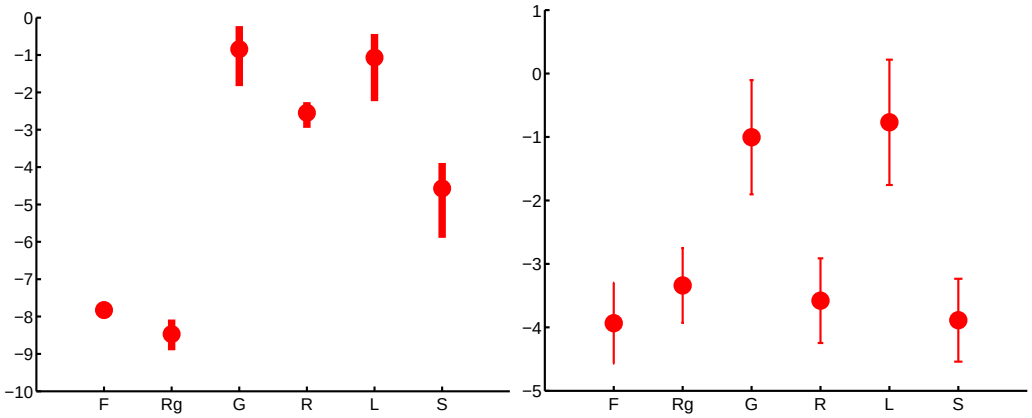
In terms of predictive validation (Table 6.4) the Mahalanobis score is closer to optimal for the replicate designs and the RMSE is similar for all designs.

Table 6.4: Mean and standard deviation of Mahalanobis score (2025) and RMSE for the Prokaryotic Auto-regulatory Network. 500 realisations of the experiment were performed.

Design	Mahalanobis	RMSE
Greedy	2490.84 ± 808.29	2.35 ± 0.11
RGrid	2521.42 ± 1043.49	2.21 ± 0.09
Grid	5520.03 ± 6763.81	2.30 ± 0.14
RLatin	2098.68 ± 546.42	2.39 ± 0.14
Latin	4361.50 ± 4603.53	2.34 ± 0.12
SA	2284.77 ± 730.00	2.28 ± 0.10

The Fisher score approximates the log determinant (Figure 6.12) with significant error but the overall ordering of the non-replicate to replicate designs is maintained. In terms of the empirical log determinant the Greedy and Replicated Grid designs have the smallest dispersion. Both of these designs place replicated design points on the boundary of the variance response and therefore perform better than predicted by the Fisher score in terms of parameter estimation. The larger approximation error of the Fisher score to the empirical log determinant suggests the prior used is not completely appropriate for the simulator data. A more informative prior closer to the simulator function would improve the Fisher approximation.

In terms of relative RMSE (Figure 6.13) very high errors are observed for the nugget parameter in the case of the non-replicate designs. Differences in estimation of the process variance term are also evident with the replicate Latin Hypercube design having the highest error. However due to the log transformation of the length-scale and process variance terms mentioned previously



(a) Log Determinant (b) FIM

Figure 6.12: Log Determinant and Fisher Information for the Prokaryotic Auto-regulatory Network. For the log determinant, .5 and .95 quantiles calculated using bootstrap. For Fisher calculation 2 standard deviations error bars are estimated from the Monte Carlo sample.

(Section 6.3.3.1) in the optimisation, the observed errors only have a small contribution to the empirical log determinant where the error on the nugget term dominates.

6.3.4.2 Individual Example

In this section a specific realisation for the Greedy and Grid designs is more closely examined to better understand the differences in behaviour of the replicate and non-replicate designs. The realisation selected has relative RMSE close to the average errors presented in the Section 6.3.4.1 and can therefore be considered representative.

The predictive validation and parameter relative RMSE for this realisation are shown in Table 6.5. The Greedy design has significantly lower Mahalanobis error and lower RMSE. In terms of relative RMSE there exists a striking difference in the error for the nugget parameter consistent with the summary results across multiple realisations of the experiment shown in Figure 6.13 and discussed in Section 6.3.4.1.

Visually the differences in mean (Figure 6.14) and standard deviation (Figure 6.15) prediction are apparent between the two designs. For the mean response the fit achieved using the Greedy design is functionally closer to the simulator output than the inference based on the Grid design. The standard deviation appears too large for the Greedy design but that is deceptive. As was noted in Section 5.6.6 for highly clustered designs the code uncertainty arising from the distance of test to training points dominates the predictive variance. The Greedy design is highly clustered (Figure 6.11) and as revealed by the Mahalanobis score and parameter estimation errors, the variance response is captured well. For the Grid design the standard deviation is unrealistically small close to training points reflecting the problem in estimating the nugget and this is mirrored by the high Mahalanobis error. The Greedy design places replicated points on the corners of the space and is

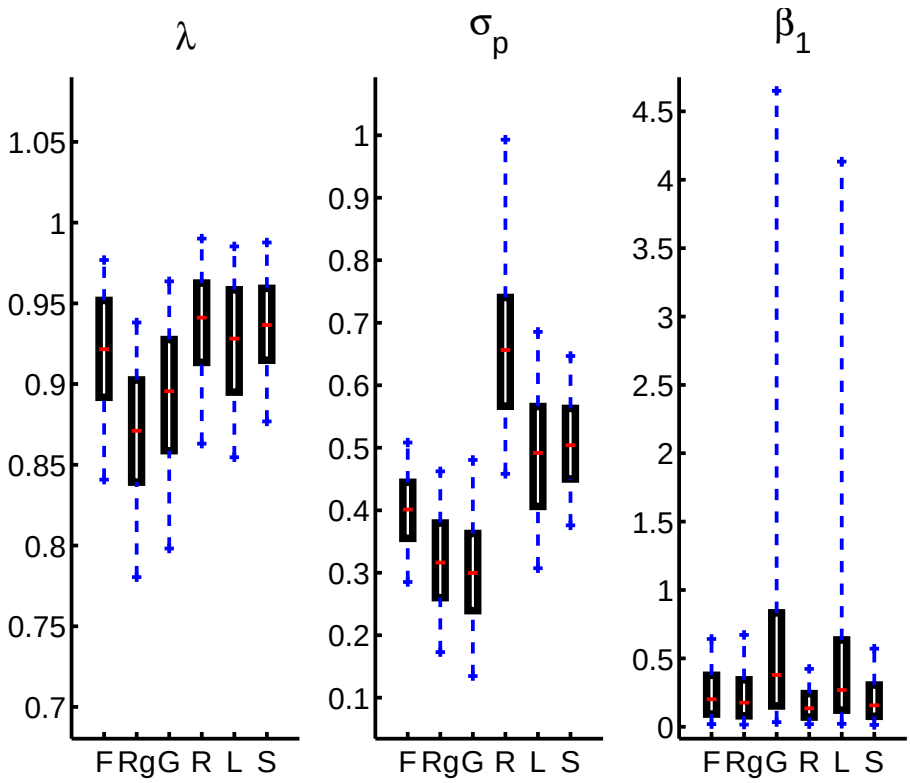


Figure 6.13: Parameter accuracy (RMSE) for the Prokaryotic Auto-regulatory Network. F=Greedy design, Rg=Replicate Grid, G=Grid, R=Replicate Maximin Latin Hypercube, L=Maximin Latin Hypercube.

able to approximately capture the functional form of the variance response.

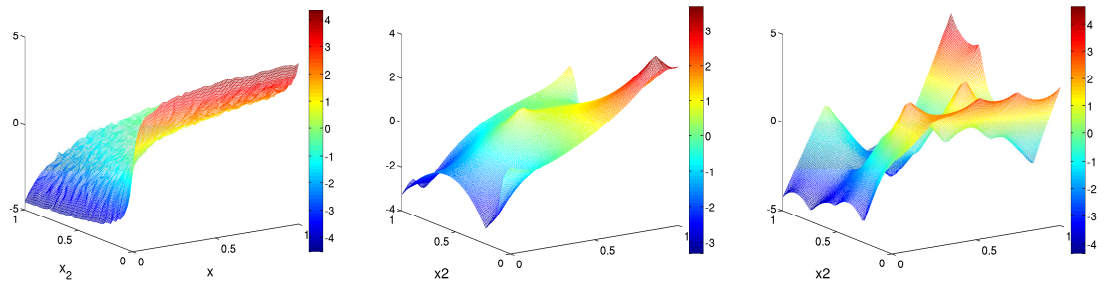
Table 6.5: Prokaryotic Auto-regulatory Network: Validation Measures and relative RMSE for length-scale (λ), process variance (σ_p) and nugget parameters (τ) for two realisations from simulator. 30-point design.

Design	Mahalanobis (2025)	RMSE	Relative RMSE $(\lambda, \sigma_p, \tau)$
Greedy	1904.26	2.23	(0.96 , 0.44 , 0.05)
Grid	5178.24	2.49	(0.98 , 0.37 , 0.99)

6.3.4.3 Fitting Complex Variance Model

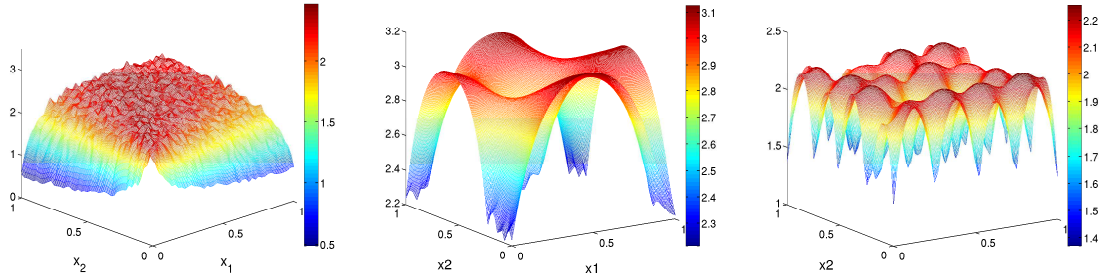
The constant variance model used to generate the model is clearly incorrect as the simulator variance exhibits structure, especially at the boundary of the design domain.

In this section, a more complex model is assumed for the variance and the resulting fit is examined for both a replicate and a non-replicate Grid design. From Section 6.3.4.1, we expect the former design to identify the model parameters more robustly and lead to lower prediction errors when compared to the Grid design. This conclusion is confirmed under a more complex variance model where the simulator variance can be captured.



(a) Simulator (b) Greedy (c) Grid

Figure 6.14: Prokaryotic Auto-regulatory Network: Comparison of Mean Prediction for two realisations from simulator.



(a) Simulator (b) Greedy (c) Grid

Figure 6.15: Prokaryotic Auto-regulatory Network: Comparison of Standard Deviation Prediction for two realisations from simulator.

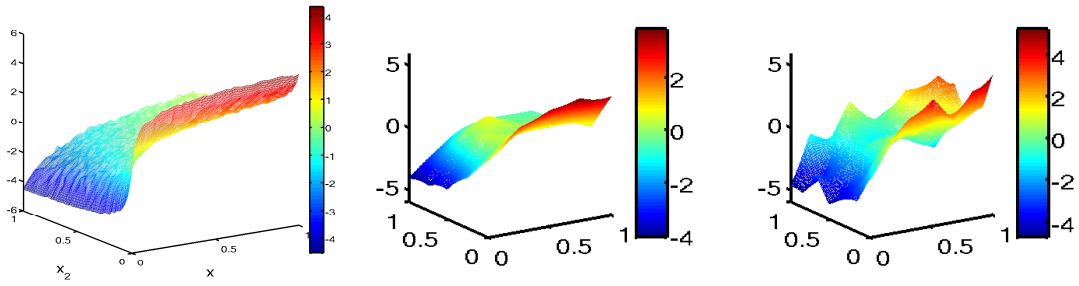
A latent kernel GP model is used for the variance with six latent points arranged in a Maximin Latin Hypercube design. Due to the higher complexity of the inferred model the training design size is increased from 30 to 50 points.

As in the previous section, the test set is a 2025 grid point design. The predictive validation results are shown in Table 6.6. The replicated Grid design achieves lower Mahalanobis error and RMSE than the non-replicate Grid design.

The emulator fit in terms of mean and standard deviation is shown in Figures 6.16 and 6.17 respectively. Note all the plots are on the same scale to facilitate comparison. In terms of the mean prediction the replicate design is smoother and approximates the simulator mean more accurately than the non-replicate design mean prediction. The standard deviation plots reveal that the simulator variance response is best captured under the replicated Grid design while for the non-replicate Grid design the functional form of the variance is not captured. The replicate Grid design captures the variance structure in this case because it places replicated points on the corners of the space where the simulator variance varies significantly from the previously assumed constant nugget.

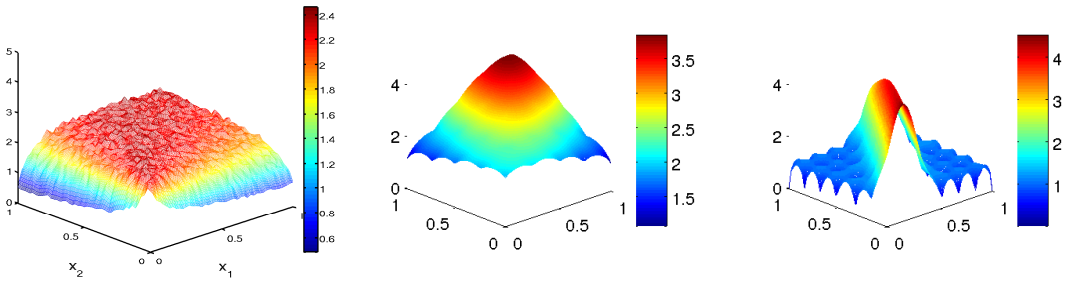
Table 6.6: Validation Measures for 50 training point design on the Latent Kernel Variance model.

Design	Mahalanobis (2025)	RMSE
Replicated Grid	1727.48	2.10
Grid	16220.3	2.18



(a) Simulator (b) Replicated Grid. (c) Grid.

Figure 6.16: Prokaryotic Auto-regulatory Network: Prediction of mean simulator value using a latent GP variance model.



(a) Simulator (b) Replicated Grid. (c) Grid.

Figure 6.17: Prokaryotic Auto-regulatory Network: Prediction of standard deviation simulator value using a latent GP variance model.

6.3.5 Conclusions

The applicability of Fisher and uniform replicate designs to two complex systems biology models was examined in Section 6.3. Section 6.3.1 motivated the usage of stochastic simulation methods for systems biology models and provided a brief overview of how exact simulation from such models may be achieved using the Gillespie algorithm. The existing literature on emulation of stochastic models for systems biology was reviewed in Section 6.3.2 where the main differences to our work were highlighted.

The experimental results for the Dimerisation Kinetics model were presented in Section 6.3.3. A linear variance model with a vague prior was used for design. All replicate designs considered achieved similar Fisher score and the approximation error to the empirical log determinant of the maximum likelihood estimates was small. In addition the predictive performance of all replicate designs was superior to the non-replicate designs considered, the Grid and Maximin Latin Hypercube designs. The similar Fisher and log determinant scores imply that for this simulator there exist multiple near-equivalent designs that perform well both in terms of minimising parameter error and predictive performance. The presence of replicate observations however seems necessary at least when considering smaller design sizes in order to capture the variance response reasonably well.

Another issue that was raised was the impact on the Fisher score considering the logarithm of the length-scale and process variance parameters versus the natural space of the variance coefficients. This has the effect of emphasising the minimisation of the variance coefficient parameter errors in the Fisher score as was clearly seen in Figure 6.7 where the larger error for the length-scale parameter in the Greedy design was not reflected in either the Fisher information or the empirical log determinant of the ML parameter covariance.

The other model examined was the Prokaryotic Auto-regulatory Network in Section 6.3.4 where a simple nugget variance model was used rather than a complex variance response which would fit the simulator variance more closely on the edges of the design space due to the corresponding difficulty in specifying a prior for such a complex model. We therefore selected to use the simpler nugget model rather than imposing an uninformative prior on a complex variance model. As was noted in Section 5.6.8 optimal design under complex models is less robust to model misspecification. We envisage the usage of complex variance models for optimal designs only in cases where informative priors can be elicited for the model parameters.

The approximation error of the Fisher information to the empirical log determinant was significantly larger than for the previous model (Figure 6.12). In particular, the dispersion as revealed by the empirical log determinant was much smaller than predicted by the Fisher information for

the Greedy and Replicated Grid designs. Both of these designs place replicated design points on the boundary of the variance response and therefore perform better than anticipated in terms of parameter estimation.

In Section 6.3.4.2 a representative instance of inference using the nugget model with the Greedy and Grid designs was examined in detail. The crucial difference between the designs was in the identification of the nugget parameter which caused the predictive variance of the Grid model to be too small close to the training points. The Greedy design which placed replicated points on the edge of the design space provided a closer fit to the simulator variance.

In Section 6.3.4.3 we abandon the nugget model and compare the inference of a latent kernel variance model on Replicated Grid and non-replicate Grid designs. The replicated design achieved better predictive performance than the non-replicate Grid design and was able to capture the functional form of the simulator variance response. This experiment provides some indication that, as the principle of parsimony suggests and as discussed in Chapter 5, optimal designs under simpler models are likely to provide support for inference of more complex models provided the original simpler model is a reasonable approximation to the true function.

The application of optimal design on the systems biology simulators has highlighted the importance of prior elicitation for model-based design. Using fairly uninformative priors limits the complexity of models that can be used as the approximation error of an incorrect complex model can be very high (see Section 5.6.8). We have chosen to use simpler models for design which are more robust to prior misspecification. However one can reasonably expect with expert elicitation reducing prior uncertainty, model-based design for stochastic simulators to be more efficient in terms of number of points required and further reduce parameter uncertainty and hence errors in predictive uncertainty.

A complementary approach to prior elicitation would be adaptive sequential design where the simulator observations are acquired sequentially and the design can be adapted as more observations are collected. Some initial thoughts on how this could be achieved in the case of Fisher design were outlined in Section 5.8. We envisage this approach to be most useful in instances where little prior information can be obtained and the simulator response is complex.

7 Conclusions and future directions.

CONTENTS

7.1 Thesis Summary

In this section the thesis is summarised and in Section 7.2 we propose directions for future research. The focus of this thesis has been on extending the emulation methodology of deterministic simulators to high-dimensional input spaces and stochastic simulators. Emulation for deterministic simulators is a well developed field where Gaussian Processes have been successfully applied as probabilistic surrogate models of the simulator. The emulation methodology was briefly described in Chapter 2.

One of the first stages in the emulation methodology is the employment of screening techniques to reduce the input dimensionality of the simulator by identifying inactive variables. In Chapter 3 screening methods for scalar output simulators were reviewed and a new sequential method based on the method of Morris (1991) proposed. The method of Morris also known as the Elementary Effects method, has found wide-spread use in the area of computer experiments due to its simplicity and effectiveness. A reliable ranking order of factor effects can be produced with a fraction of simulator runs typically required by traditional sensitivity analysis techniques such as the method of Sobol'. In some cases the number of simulator runs required by the Morris method can still be prohibitively large. The proposed sequential Morris method can be utilised when the goal of the screening process is to separate factors with non-linear effects from factors with linear, near-linear or no effects. Near-linear effects are defined as linear within some elicited variance γ (see Section 3.2.1.2 for an example). Linear and near-linear effects can be removed from the simulator output at a preprocessing stage prior to emulation and only factors with non-linear and interaction effects need to be considered in the subsequent stages of the emulation methodology. This results in performing optimal design and emulation in lower-dimensional spaces which can simplify inference and validation.

The sequential Morris method relies on the specification of a threshold value for the elementary effect variance of each factor. This quantity can be difficult to elicit directly and a new method is proposed to elicit the variance γ instead. We prove that the elementary effect variance is distributed as a scaled chi-square distribution with $R-1$ degrees of freedom, where R the number of trajectories. The threshold is then defined as the 99^{th} percentile of this distribution. A simulation experiment has demonstrated the utility of the threshold even when the factor effect is near-linear but the additional noise is not i.i.d Gaussian as assumed by the derivation.

The Morris method is applicable to deterministic as well as stochastic simulators with a high signal-to-noise ratio as the method does not account for internal simulator variability. In Chapter 6 the Morris and sequential Morris methods were applied to a stochastic simulator that models the propagation of the rabies disease in a two-species population. The screening methods were applied

on the "probability of disease extinction within five years" output. This output is of great interest to the users of this simulator and is appropriate for the Morris class of screening methods as it is approximately deterministic with a small amount of noise due to the finite number of simulator repetitions used to calculate the probability output. The results of the Morris and sequential Morris screening procedures identified the same two inputs as having near-linear effects. The sequential Morris method however required considerably fewer simulator evaluations than the batch Morris method.

The next stage in the emulation framework is to construct the statistical surrogate model using the GP formalism. In Chapter 4 two new types of heteroscedastic GP were introduced. The Coupled Model allows for the flexible, non-parametric modelling of both the mean and variance response of a simulator through a coupled system of two GPs. The method is based on the model of Kersting et al. (2007), extended to efficiently handle replicate observations and to correct the bias introduced by the log transformation of the sample variance. A new interpretation of the method of Kersting et al. (2007) was also discussed and a correction given for a systematic bias due to the non-linear transformation of the most likely variance prediction.

The Coupled Model however is too complex for the purposes of optimal design. The Joint Likelihood model was proposed as a simpler alternative where a deterministic function is used to model the variance response. Elicitation of prior beliefs for the model parameters is simplified under this model as their effect on the variance output is easier to understand.

The issue of optimal experimental design was discussed in Chapter 5. Geometric model-free designs such as the Maximin Latin Hypercube are used for their simplicity and good coverage of the input space (see Chapter 2). Such model-free designs which can be used for a variety of simulators, are quick to generate and permit the checking of modelling assumptions across the entire input domain. Model-based experimental designs are computationally more demanding to generate but allow for the incorporation of prior beliefs and optimisation of desired criteria such as the minimisation of parameter uncertainty.

A model-based Bayesian approach is suggested in Chapter 5. The methodology is based on the approach taken by Zhu and Stein (2005) and extended to the heteroscedastic GP framework with explicit consideration of replicated design points. The Fisher Information for the Joint Likelihood model with replicate observations is analytically derived. The Fisher Information for non-linear models depends on the unknown parameters and therefore a Bayesian approach is needed in practice where the Fisher Information is integrated over a parameter prior. In this work as in Zhu and Stein (2005) a coarse discrete prior is used and the Bayesian integral is approximated using Monte Carlo integration.

A series of simulation experiments was performed to investigate the performance of Fisher-

optimal designs under both Maximum Likelihood (ML) and fully Bayesian inference. The monotonicity of the Fisher score to the logarithm of the determinant of the parameter covariance of ML estimates for a range of noise levels was first established. In Zhu and Stein (2005) a similar result was established via simulation but this was extended and confirmed for the case of heteroscedastic variance models.

Three sets of locally optimal design simulation experiments for the Nugget, Log Linear and Latent Kernel variance models were presented in Chapter 5. In locally optimal designs, the Fisher score is calculated at the true parameter values and errors due to prior misspecification or Monte Carlo approximation in the Bayesian integral are absent. Therefore this set of simulation experiments focuses solely on the effectiveness of the Fisher score to minimise parameter uncertainty and the ability of the Greedy and SA optimisation methods to find a solution close to optimal.

In all experiments, the designs with lowest Fisher score resulted in GPs with the smallest parameter estimation and prediction errors. In terms of the latter, lower Mahalanobis errors and Dawid scores were observed which reflect a more accurate prediction of the variance response. Both the Fisher-optimal and uniform replicate space-filling designs achieved similar Fisher scores and prediction errors for the Nugget and Log Linear models. For the more complex Latent Kernel model however, the Fisher scores of the optimal designs were considerably lower than all other designs considered, including the uniform replicate designs. The lower Fisher score was reflected by lower parameter estimation and prediction errors for the optimal designs, although the approximation of the Fisher score to the empirical log determinant of the ML parameter covariance was worse than for the simpler models. Thus it can be said that the Fisher score is overall a good predictor of ML parameter estimation error and predictive accuracy although the approximation gets worse for the more complex models. These conclusions extend to the Bayesian optimal design context where a parameter prior is specified rather than using a plug-in estimate.

In addition, the case of structural error was examined where the model used during design is incorrect. A simulator experiment was conducted where the simpler Log Linear model is used to generate an optimal design on which the parameters of the Latent Kernel model are inferred. Although the design generated is clearly sub-optimal, the parameter accuracy and prediction errors under the Log Linear optimal design compared to utilising the optimal Latent Kernel design, were only marginally increased. The converse simulation experiment was also performed where the Latent Kernel design was utilised for inference of the Log Linear model. In this case larger errors were observed. We conclude that optimal designs generated using simpler models are more robust to model misspecification compared to more complex models.

The simulation experiments were concluded by examining the effect of Fisher-optimal designs under fully Bayesian inference where all GP hyperparameters are integrated out of the predictive

distribution. The hybrid Monte Carlo algorithm was used for the integration with a set of vague independent priors for all GP parameters. It was found that Fisher-optimal designs minimise the parameter posterior variance and result in more robust prediction. The Fisher-optimal designs are more informative about the parameter posterior and under fully Bayesian inference this has a stronger effect on the predictive variance. Therefore the benefits of utilising Fisher-optimal designs under Bayesian inference are magnified.

The optimal design approach was applied in Chapter 6 to two system biology models. A Joint likelihood model with a single nugget was used to emulated the Prokaryotic Auto-regulatory Network simulator. The Protein Dimerisation simulator was emulated using a Joint likelihood model with a Log Linear variance model. Previous work on emulating this type of simulator utilised independent emulators to model the mean and variance responses. For both simulators, the uniformly replicate space-filling designs achieved similar Fisher scores to the optimal designs and performed quite similarly in terms of predictive errors. Utilising non-replicate space-filling designs resulted in significantly higher errors.

In summary, Fisher-optimal designs can be used to more robustly identify the model parameters and this results in more accurate prediction of the variance response especially when considering fully Bayesian inference. Minimising estimation error of parameters can also be of use when an interpretation is attached to the GP hyperparameters such as in the case of ARD (Section 3.1.1) where the kernel length scales are used for screening. Even for simpler variance models, considering replicate observations in the designs has been shown to be of benefit in terms of both parameter estimation and prediction.

7.2 Future Work

In the future, the Morris method may be extended by investigating more economical designs where more elementary effects are calculated using the same number of simulator runs. A more complex one-at-a-time design is required and correlation is introduced in the calculation of the moments of the elementary effect distribution. However how to construct maximum economy designs remains an open question. Extending the Morris method to stochastic simulators would widen the applicability of the approach. Another direction of future research would be to develop screening methods for multiple simulator outputs where the between-output correlation is utilised to discover a common set of relevant factors.

The heteroscedastic models developed in Chapter 4 could also be extended in a variety of ways. The assumption of Gaussian errors could be relaxed by including other noise models and performing approximate inference using algorithms such as Expectation Propagation (Minka, 2001). This

could be extended further by modelling the output distribution non-parametrically using methods such as indicator Kriging (Oh and Lindquist, 1999) where a multiple output emulator is utilised to predict the quantiles of the output distribution.

The optimal design framework presented could also be extended in a variety of ways. Future research directions include a modification of the Fisher criterion for adaptive design where model observations made at previous emulation stages are included in the design criterion (see Section 5.8.1). This would allow for larger designs sizes to be considered and reduce the impact of the prior on the design process as more observations are included. However the validity of the proposed criterion needs to be investigated. Further we envisage the inclusion of Fisher-optimal designs, also known as D-optimal, in hybrid criteria that include multiple design goals such as minimising predictive variance and maximising information around mean function parameters. Such criteria often lead to very different designs so a hybrid approach where a multiple objective optimisation is performed could potentially yield designs useful for a multitude of purposes and of great practical use. Also optimal designs that focus on specific input regions or output threshold such as in Picheny et al. (2010) can be investigated in the context of stochastic emulation.

The impact of optimal designs on parameter posteriors when using fully Bayesian inference was examined in Section 5.7. However we stress that the simulation results presented are based on only a few realisations of the experiment. A more extensive study considering more realisations and possibly a wider range of priors and models can be pursued as a possible future research to test these conclusions more generally.

In the framework developed, the Greedy and Simulated Annealing optimisation methods are used to obtain the optimal designs. The search space for both methods is a discretised version of the design space. This approach allows for arbitrary constraints to be easily imposed on the design region but suffers from the curse of dimensionality problem since the number of candidate points grows exponentially as the number of input dimensions increases. A continuous optimisation strategy may alleviate this problem and would allow the optimal design approach proposed to scale to higher-dimensional input spaces.

Bibliography

M. Abt and W. J. Welch. Fisher information and maximum likelihood estimation of covariance parameters in Gaussian stochastic processes. *Canadian Journal of Statistics*, 26:127–137, 1998.

G. E. B. Archer, A. Saltelli, and I. M. Sobol. Sensitivity measures, anova-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58(1):99–120, 1997.

A. C. Atkinson and A. N. Donev, editors. *Optimum Experimental Designs*. Oxford University Press, 1992.

L. S. Bastos. *Validating Gaussian Process Models in Computer Experiments*. PhD thesis, University of Sheffield, 2010.

L. S. Bastos and A. O'Hagan. Diagnostics for Gaussian process emulators. *Technometrics*, 2009.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007. ISBN 0387310738. URL `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0387310738`.

E. V. Bonilla, K. Ming, A. Chai, and C. K. I. Williams. Multi-task gaussian process prediction. *Advances in Neural Information Processing Systems*, 20, 2008.

A. Boukouvalas and D. Cornford. Screening strategies for high dimensional spaces. Research note, Aston University, 2007. URL `https://wiki.aston.ac.uk/foswiki/pub/AlexisBoukouvalas/WebHome/screeningStrategiesReview.pdf`.

A. Boukouvalas and D. Cornford. Dimension reduction for multivariate emulation. Research note, Aston University, 2008. URL `https://wiki.aston.ac.uk/foswiki/pub/AlexisBoukouvalas/WebHome/PlaygroundDimRed.pdf`.

A. Boukouvalas and D. Cornford. Learning heteroscedastic gaussian processes for complex datasets. Technical Report NCRG/2009/001, Non-Linear Complexity Group, Aston University, 2009. URL `https://wiki.aston.ac.uk/foswiki/pub/AlexisBoukouvalas/WebHome/hetGPTechReport.pdf`.

A. Boukouvalas, D. Cornford, and A. Singer. Managing uncertainty in complex stochastic models: Design and emulation of a rabies model. In *St. Petersburg Workshop on Simulation*, 2009. Available upon request.

A. Boukouvalas, J.P. Gosling, and H. Maruri-Aguilar. An efficient screening method for computer experiments. Research note, Aston University, 2010. URL `https://wiki.aston.ac.uk/foswiki/pub/AlexisBoukouvalas/WebHome/screenReport.pdf`.

D. Braess and H. Dette. On the number of support points of maximin and bayesian optimal designs. *Annals of Statistics*, 35(2):772–792, 2007.

D. Bursztyn and D. Steinberg. Comparison of designs for computer experiments. *Journal of Statistical Planning and Inference*, 136(3):1103–1119, March 2006. ISSN 03783758. doi: 10.1016/j.jspi.2004.08.007. URL http://dx.doi.org/10.1016/j.jspi.2004.08.007.

F. Campolongo, J. Cariboni, A. Saltelli, and W. Schoutens. Enhancing the Morris Method. In *Sensitivity Analysis of Model Output*, pages 369–79, 2004.

F. Campolongo, J. Cariboni, and A. Saltelli. An effective screening design for sensitivity analysis of large models. *Environ. Model. Softw.*, 22(10):1509–18, 2007. ISSN 1364-8152. doi: http://dx.doi.org/10.1016/j.envsoft.2006.10.004.

K. Chaloner and K. Larntz. Optimal bayesian design applied to logistic regression experiments. *Journal of Statistical Planning and Inference*, 21(2):191 – 208, 1989. ISSN 0378-3758. doi: DOI:10.1016/0378-3758(89)90004-9. URL http://www.sciencedirect.com/science/article/B6V0M-45FJY0Y-1R/2/1094b254fe4dbcd6ad1bb1d7250a0df6.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10: 273–304, 1995.

S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640 – 651, 2007. ISSN 0378-3758. doi: DOI:10.1016/j.jspi.2009.08.006. URL http://www.sciencedirect.com/science/article/B6V0M-4X1SBB7-2/2/ff326cc0fa03e9a33e062e4d43c8788c.

D. R. Cox and P. J. Solomon. *Components of Variance*. Chapman and Hall CRC, 2003.

L. Csato. *Gaussian Processes - Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.

A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 45–54, 2008.

J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard. *Meta heuristics for Hard Optimization, Methods and Case Studies*. Springer, 2003.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. Technical report, Department of Statistics, Stanford University, 2002. URL citeseer.ist.psu.edu/efron02least.html.

V. Fedorov and W. Müller. Optimum design for correlated processes via eigenfunction expansions. Technical Report 6, Department of Statistics and Mathematics, University of Vienna, 2004.

V. V. Fedorov, editor. *Theory of optimal experiments*. Academic press, 1972.

P. H. Garthwaite and J. M. Dickey. Quantifying expert opinion in linear regression problems. *J. Roy. Statist. Soc. Ser. B*, 50(3):462–474, 1988.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992. ISSN 08834237. doi: 10.2307/2246093. URL http://dx.doi.org/10.2307/2246093.

P. W. Goldberg, C. K. I. Williams, and C. M. Bishop. Regression with input-dependent noise: A Gaussian process treatment. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

M. Goulard and M. Voltz. Linear coregionalization model: Tools for estimation and choice of cross-variogram matrix. *Mathematical Geology*, 24:269–286, 1992. ISSN 0882-8121. URL http://dx.doi.org/10.1007/BF00893750. 10.1007/BF00893750.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

J. C. Hargreaves, J. D. Annan, N. R. Edwards, and R. Marsh. An efficient climate forecasting method using an intermediate complexity Earth System Model and the ensemble Kalman filter. *Climate Dynamics*, 23:745–60, 2004.

A. J. Hayter. *Probability and Statistics for Engineers and Scientists*. Duxbury, 2002.

D. A. Henderson, R. J. Boys, K. J. Krishnan, C. Lawless, and D. J. Wilkinson. Bayesian emulation and calibration of a stochastic computer model of mitochondrial dna deletions in substantia nigra neurons. *Journal of the American Statistical Association*, 104(485):76–87, 2009a. URL http://econpapers.repec.org/RePEc:bes:jnlasa:v:104:i:485:y:2009:p:76-87.

D. A. Henderson, R. J. Boys, K. J. Krishnan, C. Lawless, and D. J. Wilkinson. Supplemental material on: Bayesian emulation and calibration of a stochastic computer model of mitochondrial dna deletions in substantia nigra neurons. Research note, Newcastle Upon Tyne University, 2009b.

S. C. H. Hoi, R. Jin, and M. R. Lyu. Batch mode active learning with applications to text categorization and image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 99(1), 2009. ISSN 1041-4347. doi: 10.1109/TKDE.2009.60. URL http://dx.doi.org/10.1109/TKDE.2009.60.

J. B. Kadane, J. M. Dickey, R. L. Winkler, W. S. Smith, and S. C. Peters. Interactive elicitation of opinion for a normal linear model. *J. Amer. Statist. Assoc.*, 75(372):845–854, 1980.

M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society*, B63:425–464, 2001.

M. C. Kennedy, C. W. Anderson, A. O'Hagan, M. R. Lomas, F. I. Woodward, J. P. Gosling, and A. Heinemeyer. Quantifying uncertainty in the biospheric carbon flux for England and Wales. *J. R. Statist. Soc. Ser. A*, 171:109–135, 2008.

K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic Gaussian process regression. In Zoubin Ghahramani, editor, *Proc. 24th International Conf. on Machine Learning*, pages 393–400. Omnipress, 2007.

J. P. C. Kleijnen and W. C. M. van Beers. Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research*, 165(3):826–834, 2005. URL http://econpapers.repec.org/RePEc:eee:ejores:v:165:y:2005:i:3:p:826-834.

A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 449–456, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273553. URL http://dx.doi.org/10.1145/1273496.1273553.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008. ISSN 1533-7928. URL http://portal.acm.org/citation.cfm?id=1390689.

L. Kuipers and H. Niederreiter. *Uniform distribution of sequences*. Dover Publications, 2005. ISBN 0-486-45019-8.

P. Latouche. Distributed machine learning. Master's thesis, Aston University, 2007.

D. Lowe and M. E. Tipping. NeuroScale: Novel topographic feature extraction using RBF networks. *Advances in Neural Information Processing Systems*, 1997.

D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4:590–604, 1992.

D. J. C. Mackay. Introduction to gaussian processes. *Neural Networks and Machine Learning*, 1998.

K. V. Mardia and R. J. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71:135–146, 1984.

G. P. McCabe. Principal variables. *Technometrics*, 26(2):137–144, 1984.

J.M. McGree, J.A. Eccleston, and S.B Duffull. Compound optimal design criteria for non-linear models. *Journal of Biopharmaceutical Statistics*, 18:646–661, 2008.

T. Minka. Expectation propagation for approximate bayesian inference. In *In Proceedings UAI*, pages 362–369, 2001.

M. D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics,*, 33(2):161–174, May 1991.

M. D. Morris and T. J. Mitchell. Exploratory designs for computer experiments. *J. Statist. Planning and Inference*, 43:381–402, 1995.

MUCM Toolkit (World Wide Web electronic publication, Release 6, 2010). The mucm toolkit. World Wide Web electronic publication, 2010. URL `http://mucm.aston.ac.uk/MUCM/MUCMToolkit/`.

W. Müller. A comparison of spatial design methods for correlated observations. *Environmetrics*, 16(5):163–177, 2005. URL `http://dx.doi.org/10.1002/asmb.740`.

W. Müller and M. Stehlík. Issues in the optimal design of computer simulation experiments. *Applied Stochastic Models in Business and Industry*, 25(2):163–177, 2009. ISSN 1526-4025. doi: 10.1002/asmb.740. URL `http://dx.doi.org/10.1002/asmb.740`.

W. G. Müller and M. Stehlík. Compound optimal spatial designs. *Environmetrics*, 21(3-4):354–364, 2010. ISSN 1099-095X. doi: 10.1002/env.1009. URL `http://dx.doi.org/10.1002/env.1009`.

I. T. Nabney. *Netlab, Algorithms for Pattern Recognition*. Springer, 2001.

R. M. Neal. *Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification*. PhD thesis, University of Toronto, 1997.

A. Neumaier and E. Groeneveld. Restricted maximum likelihood estimation of covariances in sparse linear models. *Genetics, Selection, Evolution*, 30:3–26, 1998.

H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

W. Oh and B. Lindquist. Image thresholding by indicator kriging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(7):590 –602, jul. 1999. ISSN 0162-8828. doi: 10.1109/34.777370.

A. O'Hagan and J. F. C. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978. ISSN 00359246. doi: 10.2307/2984861. URL http://dx.doi.org/10.2307/2984861.

A. O'Hagan, M. C. Kennedy, and J. E. Oakley. Uncertainty analysis and other inference tools for complex computer codes. *Bayesian Statistics*, 6:503–524, 1998.

A. Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, 4th edition, 2002. ISBN 0071226613.

A. Pázman. Correlated optimum design with parameterized covariance function: Justification of the fisher information matrix and of the method of virtual noise. Technical Report 5, Department of Statistics and Mathematics, Wirtschaftsuniversitat Wien, June 2004. URL http://statistic.wu-wien.ac.at/.

A. Pázman. Criteria for optimal design of small-sample experiments with correlated observations. *Kybernetika*, 43(4):453–462, 2007.

V. Picheny, D. Ginsbourger, O. Roustant, R.T. Haftka, and N-H. Kim. Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7), 2010.

Christian Plagemann, Kristian Kersting, and Wolfram Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5212 of *Lecture Notes in Computer Science*, pages 204–219. Springer Berlin / Heidelberg, 2008.

G. Pujol. Simplex-based screening designs for estimating metamodels. *Reliability Engineering & System Safety*, 94:1156–60, 2009.

Y. Qi, T. P. Minka, R. W. Picard, and Z. Ghahramani. Predictive automatic relevance determination by expectation propagation. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 85, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5. doi: http://doi.acm.org/10.1145/1015330.1015418.

J. Quinonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, 2005. ISSN 1532-4435. URL http://portal.acm.org/citation.cfm?id=1194909.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–435, 1989.

A. Saltelli, K. Chan, and E. M. Scott, editors. *Sensitivity Analysis*. Wiley, 2000.

A. Saltelli, S. Tarantola, and F. Campolongo, editors. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley, 2006.

M. W. Seeger. Bayesian inference and optimal design for the sparse linear model. *J. Mach. Learn. Res.*, 9:759–813, 2008. ISSN 1532-4435.

A. Singer, F. Kauhala, K. Holmala, and G.C. Smith. Rabies risk in raccoon dogs and foxes. *Biologicals, In press.*, 2008.

A. Singer, F. Kauhala, K. Holmala, and G.C. Smith. Rabies in northeastern europe - the threat from invasive raccoon dogs. *Journal of Wildlife diseases.*, 45(4):1121–1137, 2009.

A. Smola, S. V. N. Vishwanathan, and E. Eskin. Laplace propagation. *Advances in Neural Information Processing Systems*, 2004.

E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. *Neural Information Processing Systems 18*, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.2209.

E. Snelson and Z. Ghahramani. Variable noise and dimensionality reduction for sparse Gaussian processes. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Arlington, Virginia, 2006. AUAI Press.

Edward Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2007.

I. M Sobol. Sensitivity analysis for non-linear mathematical model. *Math. Modelling Comput. Exp.*, 1:407414, 1993.

M. L. Stein, editor. *Interpolation of Spatial Data: Some Theory for Kriging.* New York: Springer-Verlag, 1999a.

M. L. Stein. *Statistical Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999b.

L. Tack, P. Goos, and M. Vandebroek. Efficient bayesian designs under heteroscedasticity. *Journal of Statistical Planning and Inference*, 104(2):469 – 483, 2002. ISSN 0378-3758. doi: DOI:10.1016/S0378-3758(01)00256-7. URL http://www.sciencedirect.com/science/article/B6V0M-45NWV5F-H/2/81fcdfa2c11023d14e91ab0870c50459.

V. Tresp. A bayesian committee machine. *NEURAL COMPUTATION*, 12:2000, 2000.

C. Walder, Kwang I. Kim, and B. Schölkopf. Sparse multiscale gaussian process regression. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1112–1119, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390296.

W. J. Welch, J. B. Robert, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris. Screening, predicting, and computer experiments. *Technometrics,*, 34(1):15–25, 1992.

D. J. Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC, 1 edition, 2006.

G. Xia, M. L. Miranda, and A. E. Gelfand. Approximately optimal spatial design approaches for environmental health data. *Environmetrics*, 17(4):363–385, 2006. ISSN 0378-3758. doi: 10.1002/env.775.

N. Youssef. *An orthonormal function approach to optimal design for computer experiments*. PhD thesis, London School of Economics, UK, 2010.

K. Yu and J. Bi. Active learning via transductive experimental design. In *In Machine Learning, Proceedings of the Twenty-Third International Conference (ICML*, pages 1081–1088. ACM Press, 2006.

M. Yuan and G. Wahba. Doubly penalized likelihood estimator in heteroscedastic regression. *Statistics and Probability Letters*, 69:11–20, 2004.

H. Zhang and D. L. Zimmerman. Towards reconciling two asymptotic frameworks in spatial statistics. *Biometrika*, 92(4):921–936, December 2005. ISSN 0006-3444. doi: 10.1093/biomet/92.4.921. URL http://dx.doi.org/10.1093/biomet/92.4.921.

Z. Zhu and M. L. Stein. Spatial sampling design for parameter estimation of the covariance function. *Journal of Statistical Planning and Inference*, 134(2):583 – 603, 2005. ISSN 0378-3758. doi: DOI:10.1016/j.jspi.2004.04.017. URL http://www.sciencedirect.com/science/article/B6V0M-4CYTYTS-N/2/2aa343e9d658f63d3d1e4333f747579c.

Z. Zhu and M. L. Stein. Spatial sampling design for prediction with estimated parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 11(1):24–44, March 2006. ISSN 1085-7117. doi: 10.1198/108571106X99751. URL http://dx.doi.org/10.1198/108571106X99751.

D. L. Zimmerman. Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics*, 17(6):635–652, 2006. ISSN 1099-095X. doi: 10.1002/env.769. URL http://dx.doi.org/10.1002/env.769.

A Heteroscedastic Gaussian Process Derivations

In the chapter, the derivations related to the Heteroscedastic GP framework are given.

A.1 Obtaining the Kersting approach through explicit maximisation

In this appendix we discuss how the Kersting method presented in Section 4.3 may be obtained through an explicit maximisation of the posterior density of the noise levels. Specifically we examine how Equation (4.9), used to estimate the empirical noise levels, is arrived at through a maximisation of Equation (4.5).

To obtain maximum likelihood (ML) estimates of the most likely noise levels, the explicit maximisation problem we wish to solve during the training stage is:

$$\arg\max_{z_i, \mathbf{z}_{-i}} p(z_i, \mathbf{z}_{-i} | \mathbf{t}).$$

In step 1 of the Kersting algorithm, we obtain estimates for the noise-free process hyperparameters, θ_y, and an input-dependent nugget, σ_H^2, which is useful to obtain initial estimates on the noise process.

We reformulate the problem:

$$\arg\max_{z_i, \mathbf{z}_{-i}} p(z_i | \mathbf{z}_{-i}, \mathbf{t}) p(\mathbf{z}_{-i} | \mathbf{t}).$$

We now make a crucial simplification to maximise the noise levels one point at a time. This is clearly suboptimal but allows for a simpler derivation and emphasizes the iterative nature of the algorithm since the values for the non-optimised noise levels $\mathbf{z}_{-i}$ may be used from the previous iterative step. We note here that although it is possible to immediately use the new estimate for z_i at the optimisation for the next point similar to a Gibbs sampler, for simplicity and to minimise computational complexity we have elected to utilise the new optimised estimates for the noise levels only at the subsequent iteration of the Kersting algorithm. The optimisation problem is thus

simplified to:

$$\arg\max_{z_i} p(z_i|\mathbf{z}_{-i},\mathbf{t})p(\mathbf{z}_{-i}|\mathbf{t}).$$

Only the first term is relevant as the latter does not include z_i. Using Bayes' theorem we can reformulate the distribution of interest to:

$$p(z_i|\mathbf{z}_{-i},\mathbf{t}) \propto p(\mathbf{t}|z_i,\mathbf{z}_{-i})p(z_i|\mathbf{z}_{-i}) = \mathcal{N}(\mathbf{t}|0,K_y+R)\ln\mathcal{N}(z_i|\mu_z,\lambda_z),$$

where $\ln\mathcal{N}$ denotes the Log Normal distribution with mean μ_z and variance λ_z. Note these are related to the mean and variance of the variance GP but are not identical - see Section 4.3.3. As before the R matrix is $\text{diag}(z_1,\ldots,z_N)$.

The likelihood term can be further decomposed:

$$p(\mathbf{t}|z_i,\mathbf{z}_{-i}) = p(t_i|\mathbf{t}_{-i},z_i,\mathbf{z}_{-i})p(\mathbf{t}_{-i}|z_i,\mathbf{z}_{-i}) = p(t_i|\mathbf{t}_{-i},z_i,\mathbf{z}_{-i})p(\mathbf{t}_{-i}|\mathbf{z}_{-i}).$$

The last step follows from the model dependency structure (see Goldberg et al. (1998)), i.e. given $\mathbf{z}_{-i}$, the distribution of the noisy observations $\mathbf{t}_{-i}$ can be uniquely determined without reference to z_i. The latter term is thus irrelevant to the optimisation task since it does not depend on z_i.

To summarise the optimisation task is:

$$\arg\max_{z_i} p(z_i,\mathbf{z}_{-i}|\mathbf{t}) \equiv \arg\max_{z_i} p(t_i|\mathbf{t}_{-i},z_i,\mathbf{z}_{-i})p(z_i|\mathbf{z}_{-i}). \tag{A.1}$$

As mentioned previously the term $p(z_i|\mathbf{z}_{-i})$ is a Log Normal whose moments are determined by the variance GP inferred at the previous iteration step. The homoscedastic nugget term σ_H^2 can be used to initialise this distribution for the first step.

The univariate likelihood term is a Gaussian distribution with mean μ_t and variance $z_i+\lambda_y$ as described by the heteroscedastic GP predictive equations (4.13)-(4.14).

The log posterior in Equation (A.1) is:

$$\mathcal{L} = -\frac{\ln(\lambda_y+z_i)}{2} - \frac{\ln(\lambda_z)}{2} - \ln(z_i) - \frac{(\mu_z-\ln(z_i))^2}{2\lambda_z} - \frac{(\mu_t-t_i)^2}{2\lambda_y+2z_i},$$

where $p(z_i|z_{-i}) = \ln\mathcal{N}(z_i|\mu_z = \exp(E[z_i]+\text{Var}[z_i]/2),\lambda_z = (\exp(\text{Var}[z_i])-1)\exp(2E[z_i]+\text{Var}[z_i])$ the Log Normal posterior whose moments are determined by the variance GP utilising hyper-parameters and training set obtained at the previous iteration. The posterior $p(t_i|t_{-i},z_{-i},z_i) = \mathcal{N}(t_i|\mu_t,z_i+\lambda_y)$ where λ_y the variance of the noise-free values.

Setting the derivative to zero:

$$\frac{2(\mu_t-t_i)^2}{(2\lambda_y+2z_i)^2} - \frac{1}{2\lambda_y+2z_i} - \frac{1}{z_i} + \frac{\mu_z-\ln(z_i)}{\lambda_z z_i} = 0. \tag{A.2}$$

This cannot be solved analytically with respect to z_i. Rather than employing numerical optimisation methods to solve Equation (A.2) we can approximate $p(z_i|z_{-i})$ by a Gaussian in which case Equation A.2 is a cubic equation with exactly one real root:.

$$\frac{(t_i-\mu_t)^2-z_i-\lambda_y}{(z_i+\lambda_y)^2} = \frac{2z_i-2\mu_z}{\lambda_z} \tag{A.3}$$

However the expression for z_i then no longer guarantees positive values. By the intermediate value theorem we know there will be at least one real root for a cubic equation with real coefficients. In experiments we have confirmed there is exactly one real root for this cubic.

Further, by setting the variance on the noise values to $\lambda_z = 2(z_i+\lambda_y)^2$ and the mean $\mu_z = z_i^\tau$,

i.e. the value obtained at the previous step, Equation (A.3) simplifies to:

$$z_i^{\tau+1} = \frac{1}{2}\left((t_i - \mu_t)^2 - \lambda_y + z_i^{\tau}\right) \tag{A.4}$$

which is very similar to Equation (4.10) used in the Kersting approach except for the sign of the variance on noise-free values λ_y. If we assume this value is negligible, we have obtained the Kersting approach as described in Section 4.3. In fact the derivation is exact if we assume the noise-free targets are known, i.e. $\lambda_y = 0$, which is an assumption mentioned in Section 4 of Kersting et al. (2007) to justify the sampling step of the algorithm (see Section 4.3.4 - Equation (4.9)).

A.2 Correcting bias in sample log variance

The log transformation of the sample variance introduces a bias in the estimation. In Cox and Solomon (2003) the mean and variance of the log variance distribution are given but as there are typos, we rederive here the proof.

Assuming the observations are normally distributed, the distribution of the sample variance is a Chi square distribution, which is a special case of a Gamma distribution:

$$s^2 \sim \Gamma\left(\frac{n_i - 1}{2}, \frac{2\sigma^2}{n-1}\right),$$

where n the number of observations and σ^2 the true variance.

The derivation requires the following theorem:

Theorem A.2.1. *If X is Gamma distributed with $X \sim \text{Gamma}(k,\theta)$, the mean and variance of the natural log transformation are $\text{E}(\log X) = \psi(k) + \log(\theta)$ and $\text{Var}(\log X) = \psi_2(k)$ where ψ and ψ_2 the digamma and trigamma functions respectively.*

Proof. The parametrisation of the Gamma distribution used is:

$$\text{Gamma}(k,\theta) = \frac{1}{\theta^k \Gamma(k)} x^{k-1} e^{-x/\theta}.$$

Let $\Phi(s)$ the moment generating function (Papoulis and Pillai, 2002) of log transformed X :

$$\begin{aligned}
\Phi(s) &= \int_{-\infty}^{+\infty} \text{Gamma}(k,\theta)\, e^{s\log(x)} \mathrm{d}x \\
&= \frac{1}{\theta^k \Gamma(k)} \int x^{k+s-1} e^{-x/\theta} \mathrm{d}x \qquad \textit{Gamma Integral} \\
&= \frac{1}{\theta^k \Gamma(k)} \Gamma(k+s)\, \theta^{k+s} \\
&= \frac{\Gamma(k+s)}{\Gamma(k)} \theta^s.
\end{aligned}$$

To get the first two moments of the distribution, the first two derivatives are calculated:

$$\begin{aligned}
\frac{\partial \Phi(s)}{\partial s} &= \frac{1}{\Gamma(k)} \left[\frac{\partial \Gamma(k+s)}{\partial s} \theta^s + \Gamma(b+s)\theta^s \log\theta\right] \\
\frac{\partial^2 \Phi(s)}{\partial s^2} &= \frac{1}{\Gamma(k)} \left[\frac{\partial^2 \Gamma(k+s)}{\partial s^2} \theta^s + 2\frac{\partial \Gamma(k+s)}{\partial s} \theta^s \log\theta + \Gamma(k+s)\theta^s (log\theta)^2\right]
\end{aligned}$$

Setting $s = 0$, the central moments are obtained:

$$\left.\frac{\partial \Phi(s)}{\partial s}\right|_{s=0} = \psi(k) + \log\theta$$
$$\left.\frac{\partial^2 \Phi(s)}{\partial s^2}\right|_{s=0} = \frac{1}{\Gamma(k)}\left[\frac{\partial^2 \Gamma(k+s)}{\partial s^2} + 2\frac{\partial \Gamma(k+s)}{\partial s}\log\theta + \Gamma(k+s)\theta^s(log\theta)^2\right]$$

Finally the first two moments are

$$E(\log X) = \left.\frac{\partial \Phi(s)}{\partial s}\right|_{s=0} = \psi(k) + \log\theta$$
$$Var(\log X) = \left.\frac{\partial^2 \Phi(s)}{\partial s^2}\right|_{s=0} - \left(\left.\frac{\partial \Phi(s)}{\partial s}\right|_{s=0}\right)^2 = \psi_2(k).$$

In the above the definitions of the digamma and trigamma functions are utilised:

$$\psi(x) = \frac{\partial \log\Gamma(x)}{\partial x}$$
$$\psi_2(x) = \frac{\partial^2 \log\Gamma(x)}{\partial x^2}$$

□

Given theorem A.2.1, the mean and variance of the log sample variance is :

$$\mathrm{E}(\log s^2) = \psi\left(\frac{n-1}{2}\right) + \log 2 + \log\sigma^2 - \log(n-1)$$
$$\mathrm{Var}(\log s^2) = \psi_2\left(\frac{n-1}{2}\right)$$

Therefore the bias corrected sample variance estimate is:

$$\log\sigma^2 = \mathrm{E}(\log s^2) - \psi\left(\frac{n-1}{2}\right) - \log 2 + \log(n-1)$$

Approximations of the digamma and trigamma functions are possible through truncated series expansions though we do not utilise them (Cox and Solomon, 2003).

A.3 Heteroscedastic Prior GP Derivation

We assume the noise process has zero mean and is independent given the design point. Given these conditions, the distribution of the sample mean $\hat{\mu}_i$ is (Hayter, 2002):

$$p(\hat{\mu}_i|\mu_i) = N(\hat{\mu}_i|\mu_i, \frac{\sigma^2(x_i)}{n_i}),$$

where n_i the number of replicate observations, $\sigma^2(x_i)$ the true variance at location x_i and μ_i the true mean.

Due to the independence of the noise we can write the likelihood in matrix form for all observations $1 \ldots N$:

$$p(\hat{\mu}|\mu) = N(\hat{\mu}|\mu, RP^{-1}),$$

where $R = \mathrm{diag}(\sigma^2(x_i))_{i=1}^N$ and $P = \mathrm{diag}(n_i)_{i=1}^N$.

Our zero mean GP prior is:

$$p(\mu) = N(\mu|0, K).$$

The marginal observation density can then be calculated:

$$p(\hat{\mu}) = \int p(\hat{\mu}|\mu)p(\mu)d\mu = \int N(\hat{\mu}|\mu, RP^{-1})N(\mu|0,K)d\mu = N(\hat{\mu}|0, C_\mu = K + RP^{-1}). \quad \text{(A.5)}$$

The last step stems from applying the identity (2.115) (Bishop, 2007), that is given by

$$p(y) = \int p(y|x)p(x)dx = \int N(y|Ax+b, L^{-1})N(x|\mu, \Lambda^{-1})dx = N(y|A\mu+b, L^{-1}+A\Lambda^{-1}A^T).$$

We get the result of Equation (A.5) by setting $A = I$, $b = 0$, $L^{-1} = RP^{-1}$ and $\mu = 0$, $\Lambda^{-1} = K$. This result can also be obtained directly by noticing that the distribution of $\hat{\mu}$ is the sum of two normal distributions, $p(\hat{\mu}) = p(\mu) + N(0, RP^{-1})$, which are independent and therefore their variances add.

We can use Equation (A.5) to now derive the predictive distribution by conditioning on the known observations. We can partition the joint distribution by the unobserved sites $\hat{\mu}_*$ and the observed sites $\hat{\mu}$. Use Equation (A.5) we can write this partitioned joint distribution as:

$$p(\hat{\mu}_*, \hat{\mu}) = N\left(\begin{bmatrix} \hat{\mu}_* \\ \hat{\mu} \end{bmatrix} | \mathbf{0}, \begin{bmatrix} K(x_*,x_*) + R(x_*)P(x_*)^{-1} & K(x_*,x) \\ K(x_*,x)^T & K(x,x) + R(x)P(x)^{-1} \end{bmatrix}\right),$$

where x_* and x the unobserved and observed design sites respectively. $R(x_*)$ and $P(x_*)$ the variance and number of replicates at the unknown sites. For the off-diagonal terms, $R(x, x_*) = 0$ due to the independence of noise (R diagonal).

To make predictions of the sample mean $\hat{\mu}_*$, we condition on the known sites $\hat{\mu}$ (this can be done by completing the square - see page 86 of (Bishop, 2007) and equations (2.81) (2.82)).

$$\begin{aligned} p(\hat{\mu}_*|\hat{\mu}) = N(\quad & K(x_*,x)^T(K(x,x) + R(x)P(x)^{-1})^{-1}\hat{\mu}, \\ & K(x_*,x_*) + R(x_*)P(x_*)^{-1} + K(x_*,x)^T(K(x,x) + R(x)P(x)^{-1})^{-1}K(x_*,x)). \end{aligned}$$

We note that the sample mean $\hat{\mu}$ coincides with a single observation when the number of replicates is 1. Thus we can set $P(x_*) = I$ to obtain the predictive equations for single replicate test sites.

A.4 Derivation of likelihood for the Joint Model

In this section, the likelihood of the joint model described in Section 4.5 is derived. The joint likelihood of the sample mean $\hat{\mu}$ and sample variance s^2 is:

$$\begin{aligned} p(\hat{\mu}, s^2|\mathbf{X}, \theta, \beta) &= \int p(\hat{\mu}, s^2, \mu|\mathbf{X}, \theta, \beta)d\mu \\ &= \int p(\hat{\mu}, s^2|\mu, \mathbf{X}, \theta, \beta)p(\mu|\theta)d\mu \\ &= p(s^2|\mathbf{X}, \beta)\int p(\hat{\mu}|\mu, \theta, \beta, \mathbf{X})p(\mu|\theta)d\mu \\ &= \left(\prod_{i=1}^{N} p(s_i^2|x_i, \beta)\right)\int p(\hat{\mu}|\mu, \theta, \beta, \mathbf{X})p(\mu)d\mu \\ &= \left(\prod_{i=1}^{N} p(s_i^2|x_i, \beta)\right) N(\hat{\mu}|0, K_\theta + R_\beta P^{-1}) \end{aligned} \quad \text{(A.6)}$$

The last equality follows from Section A.3 Equation (A.5).

The log likelihood can then be written:

$$\log p(\hat{\mu}, s^2|\mathbf{X}, \theta, \beta) = \left(\sum_{i=1}^{N} \log p(s_i^2|\beta, x_i)\right) + \log N(\hat{\mu}|0, K + RP^{-1}) = \left(\sum_{i=1}^{N} L_{si}\right) + L_N \quad \text{(A.7)}$$

where the latter term is a GP standard likelihood with the given covariance and the former can be expanded:

$$\begin{aligned}\log p(s_i^2|\beta,x_i) =& \frac{n_i-1}{2}\left(\log(n_i-1)-\log(2)-\log f_{\sigma^2}(x_i,\beta)\right)-\log\Gamma(\frac{n_i-1}{2})\\ &+\frac{n_i-3}{2}\log(s_i^2)-\frac{(n_i-1)s_i^2}{2f_{\sigma^2}(x_i,\beta)}.\end{aligned} \tag{A.8}$$

A.5 Proof of Fisher Information for Heteroscedastic Noise Models

The Fisher Information is defined as

$$\mathcal{F} = -\int\left(\frac{\partial^2}{\partial\theta^2}\log(L(\mathbf{X}|\theta)\right)L(\mathbf{X}|\theta)\mathrm{d}\mathbf{X},$$

where $L(\mathbf{X}|\theta)$ is the likelihood function.

For the heteroscedastic GP model $\theta=\{\beta,\theta_K\}$, i.e. the variance coefficients and the kernel hyperparameters. For parameters θ_j,θ_p the corresponding element in the FIM is:

$$\mathcal{F}_{jp} = -\int\int\left(\frac{\partial^2}{\partial\theta_j\theta_p}\log p(\hat{\mu},s^2|\theta)\right)p(\hat{\mu},s^2|\theta)\,\mathrm{d}\hat{\mu}\mathrm{d}s^2,$$

where we have omitted the dependency on the inputs $\mathbf{X}$.
The log likelihood term can be decomposed into two terms as shown in Equation (4.19), a sample variance term L_{si} and a Gaussian Process term L_N.

$$\begin{aligned}\mathcal{F}_{jp} &= -\int\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}\sum L_{si}\right]p(\hat{\mu},s^2|\theta)\,\mathrm{d}\hat{\mu}\mathrm{d}s^2-\int\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}L_N\right]p(\hat{\mu},s^2|\theta)\,\mathrm{d}\hat{\mu}\mathrm{d}s^2\\ &= -\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}\sum L_{si}\right]p(s|\beta)\,\mathrm{d}s^2\int p_\mu\,\mathrm{d}\hat{\mu}-\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}L_N\right]p_\mu\,\mathrm{d}\hat{\mu}\int p_s\mathrm{d}s^2\end{aligned}$$

where $p_\mu = N(\hat{\mu}|0,K_\theta+R_\beta P^{-1})$. Note $\int p_\mu\,\mathrm{d}\hat{\mu}=1$ and $\int p(s^2|\mathbf{X},\beta)\,\mathrm{d}s^2=1$ since they are density functions. Lastly we are able to separate the sample variance integrals to the individual s_i terms due to the noise independence assumption, i.e. $p(s^2|\mathbf{X},\beta)=\prod_{i=1}^N p(s_i^2|x_i,\beta)$.

$$\begin{aligned}\mathcal{F}_{jp} &= -\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}\sum_{i=1}^N L_{si}\right]\prod p(s_i^2|\sigma_\mu^2(x_i))\,\mathrm{d}s^2+F_N\\ &= -\sum_{i=1}^N\left(\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}L_{si}\right]p(s_i^2|x_i,\beta)\,\mathrm{d}s_i^2\int\prod_{j\neq i}^N p(s_i^2|x_i,\beta)\,\mathrm{d}s_j\right)+F_N\\ &= \sum_{i=1}^N F_{si}+F_N,\end{aligned} \tag{A.9}$$

where

$$\begin{aligned}F_{si} &= -\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}\log p(s_i^2|x_i,\beta)\right]p(s_i^2|x_i,\beta)\,\mathrm{d}s_i^2,\\ F_N &= -\int\left[\frac{\partial^2}{\partial\theta_j\theta_p}L_N\right]p_\mu\,\mathrm{d}\hat{\mu}.\end{aligned}$$

The solution to the F_N integral is known and for a zero mean GP is $\frac{1}{2}\mathrm{tr}(\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_j}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_p})$ (Pázman, 2004). The F_{si} integral can be solved by rewriting the integral given the second order derivative

and the sample variance distribution:

$$\begin{aligned} F_{si} &= -\int \frac{\partial^2 \log p(s_i^2|\beta, x_i)}{\partial\beta_j\beta_p} p(s_i^2|\beta, x_i)\, \mathrm{d}s_i^2 \\ &= \frac{n_i - 1}{2}\frac{\partial^2 f}{\partial\beta_j\beta_p}\int p(s_i^2|\beta, x_i)\, \mathrm{d}s_i^2 \\ &- \frac{(n_i-1)}{2}\left[-\exp(-f)\frac{\partial f}{\partial\beta_j}\frac{\partial f}{\partial\beta_p} + \exp(-f)\frac{\partial^2 f}{\partial\beta_j\beta_p}\right]\int s_i^2 p(s_i^2|\beta, x_i)\, \mathrm{d}s_i^2 \end{aligned}$$

The integral can be analytically solved. For notational brevity let $f_{\sigma^2} = f_{\sigma^2}(x_i, \beta) = \exp(f)$.

$$\int s_i^2 p(s_i^2|\beta, x_i)\, \mathrm{d}s_i^2 = \frac{\frac{n_i-1}{2f_{\sigma^2}}^{\frac{n_i-1}{2}}}{\Gamma(\frac{n_i-1}{2})}\int s_i^2 (s_i^2)^{\frac{n_i-3}{2}} e^{-\frac{n_i-1}{2f_{\sigma^2}}s_i^2}\, \mathrm{d}s_i^2. \tag{A.10}$$

The last integral is the mean of Gamma distribution. Therefore the Gamma integral is:

$$\frac{2f_{\sigma^2}}{n_i-1}\frac{n_i-1}{2} = f_{\sigma^2}.$$

To conclude the Fisher information contribution of the sample variance term of the log likelihood F_{si} is:

$$\begin{aligned} F_{si} &= \frac{n_i-1}{2}\frac{\partial^2 f}{\partial\beta_j\beta_p} - \frac{(n_i-1)}{2}\left[-\exp(-f)\frac{\partial f}{\partial\beta_j}\frac{\partial f}{\partial\beta_p} + \exp(-f)\frac{\partial^2 f}{\partial\beta_j\beta_p}\right] f_{\sigma^2} \\ &= \frac{n_i-1}{2}\left(\frac{\partial^2 f}{\partial\beta_j\beta_p} - \frac{\partial^2 f}{\partial\beta_j\beta_p} + \frac{\partial f}{\partial\beta_j}\frac{\partial f}{\partial\beta_p}\right). \end{aligned}$$

The final result is:

$$\boxed{F_{si} = \frac{n_i-1}{2}\frac{\partial f}{\partial\beta_j}\frac{\partial f}{\partial\beta_p}.}$$

In the case of the fixed basis variance model $\frac{\partial f}{\partial\beta_j} = H^T(x_i)J_j$ and the F_{si} for parameters $\{\beta_j, \beta_p\}$:

$$F_{si} = \frac{n_i-1}{2}H^T(x_i)J_j H^T(x_i)J_p,$$

where J_j the zero vector with j^{th} element 1.

B Details of Methods Used

B.1 The bootstrap method

We use a method suggested in Efron and Tibshirani (1993) to determine the number of bootstrap samples required to estimate the standard error in Section 5.6.2. As usual, bootstrap is done by random sampling with replacement.

In particular we first estimate the bias $E_{bootstrap} - E_{data}$, where $E_{bootstrap}$ the mean value across all bootstrap samples and E_{data} the estimated value from data. If the bias / standard error ratio is less then 0.25, we judge we have enough samples in our bootstrap.

B.2 Data Preprocessing and Standardisation

In this section we describe the process of preprocessing data, which might often be undertaken prior to for example screening or more general emulation. This can take several forms. A very common preprocessing step is centring, which produces data with zero mean. If the range of variation is known *a priori* a simple linear transformation to the range [0,1] is often used. It might also be useful to standardise (sometimes called normalise) data to produce zero mean and unit variance. For multivariate data it can be useful to whiten (or sphere) the data to have zero mean and identity covariance, which for one variable is the same as standardisation. The linear transformation and normalisation processes are not equivalent since the latter is a probabilistic transformation using the first two moments of the observed data. This section is based on the MUCM Toolkit (World Wide Web electronic publication, Release 6, 2010) `ProcDataPreProcessing` page.

B.2.1 Centring

It is often useful to remove the mean from a data set. In general the mean, $\mathrm{E}[x]$, will not be known and thus must be estimated and the centred data is given by: $x' = x - \mathrm{E}[x]$. Centring will often be

used if a zero mean GP is being used to build the emulator, although in general it would be better to include an explicit mean function in the emulator.

B.2.2 Linear transformations

To linearly transform the data region $x \in [c,d]$ to another domain $x' \in [a,b]$:

$$x' = \frac{x-c}{d-c}(b-a)+a.$$

In experimental design the convention is for $[a,b] = [0,1]$.

B.2.3 Standardising

If the domain of the design region is not known, samples from the design space can be used to rescale the data to have 0 mean, unit variance by using the process of standardisation. If on the other hand the design domain is known we can employ a linear rescaling.

The process involves estimating the mean $\mu = \mathrm{E}[x]$ and standard deviation of the data σ and applying the transformation $x' = \frac{x-\mu}{\sigma}$. It is possible to standardise each input / output separately which rescales the data, but does not render the outputs uncorrelated. This might be useful in situations where correlations or covariances are difficult to estimate, or where these relationships need to be preserved, so that individual inputs can still be distinguished.

B.2.4 Sphering / Whitening

For multivariate inputs and outputs it is possible to whiten the data, that is convert the data to zero mean, identity variance. The data sphering process involves estimating the mean $\mathrm{E}[x]$ and variance matrix of the data $\mathrm{Var}[x]$, computing the eigen decomposition $P\Delta P^T$ of $\mathrm{Var}[x]$ and applying the transformation $x' = P\Delta^{-1/2}P^T(x-\mathrm{E}[x])$.

B.3 Proof of Lemma 3.2.1.

In this appendix the proof for Lemma 3.2.1 is presented. As Lemma 3.2.1 is defined for a single factor, at this stage design points are considered univariate and the elementary effect of Equation (3.2) is computed for a single factor.

Proof. We first note that at the point x_i, the elementary effect $EE(x_i) = (Y(x_i+\Delta) - Y(x_i))/\Delta$ follows a normal distribution, i.e. $EE(x_i) \sim N(a, \frac{2\gamma}{\Delta^2})$. The independence of elementary effects $EE(x_1),\ldots,EE(x_R)$ follows directly from independence of observations of the Model (3.5) at different points.

The rest of the proof uses a classic decomposition of sums of squares of iid normal random variables. The mean of elementary effects $\mu = \frac{1}{R}\sum_{i=1}^{R} EE(x_i)$ follows a normal distribution $N(a, \frac{2\gamma}{R\Delta^2})$. To compute the distribution of the variance of elementary effects $\sigma^2 = \frac{1}{R-1}\sum_{i=1}^{R}(EE(x_i) - \mu)^2$, the following sum of squares is used

$$\sum_{i=1}^{R}\left(\frac{EE(x_i)-a}{\sqrt{\frac{2\gamma}{\Delta^2}}}\right)^2 = (R-1)\frac{\sigma^2}{\frac{2\gamma}{\Delta^2}} + R\frac{(\mu-a)^2}{\frac{2\gamma}{\Delta^2}}.$$

The left hand side of the above expression is a sum of squared independent standard normal random variables and thus it has a chi-squared distribution with R degrees of freedom χ^2_R. The quantity that interests us is the first summand on the right hand side above. By the independence of μ and σ^2, this quantity has a chi-squared distribution with $R-1$ degrees of freedom χ^2_{R-1}, that is, σ^2 follows a scaled chi-squared distribution $\sigma^2 \sim \frac{2\gamma}{(R-1)\Delta^2}\chi^2_{R-1}$. □

B.4 Screening Test function

We describe the function used in Section 3.1.3.2 to generate the simulated data. The function has 99 input variables, with one dummy variable (x_{99}). The effects are classified into linear, polynomial of order 2 or greater and step-linear:

$$f(x) = f^{Linear}(x_{1,10}) + f^{Poly}(x_{11,38}) + f^{Periodic}(x_{39,76}) + f^{Step}(x_{79,98})$$

where

$$\begin{aligned} f^{Linear}(x_{1,10}) &= 2 + \sum_{i=1}^{8} x_i + -x_9 - x_{10}, \\ f^{Poly}(x_{11,38}) &= 2(\prod_{i=11}^{12} x_i) - 1.5(\prod_{i=13}^{15} x_i) + 3(\prod_{i=16}^{19} x_i) + (\sum_{i=20}^{22} x_i^2) + x_{23}x_{24}^2 - x_{25}x_{26}^2 \\ &\quad - x_{27}x_{28} - x_{29}x_{30}^2 + 6x_{31}^3 x_{32}^2 x_{33}^{0.7} - 4x_{34}^2\sqrt{x_{35}} + \sqrt{x_{36}}x_{37}^2 - x_{38}^7, \end{aligned}$$

$$\begin{aligned} f^{Periodic}(x_{39,76}) &= 5sin(x_{39}x_{40}/5) + (5/2)sin(2x_{41}x_{42}/5) + (5/3)sin(3x_{43}x_{44}/5) \\ &\quad + (5/4)sin(4x_{45}x_{46}/5) + sin(x_{47}x_{48}/5) + (5/6)sin(6x_{49}x_{50}/5) \\ &\quad + (5/7)sin(7x_{51}x_{52}/5) + (5/8)sin(8x_{53}x_{54}/5) + (5/9)sin(9x_{55}x_{56}/5) \\ &\quad + (5/10)sin(10x_{57}x_{58}/5) + sin(13x_{59}) + sin(10x_{60}) + sin(7x_{61}) \\ &\quad + sin(4x_{62}) + sin(x_{63}) - cos(13x_{64}) - cos(10x_{65}) - cos(7x_{66}) \\ &\quad - cos(4x_{67}) - cos(x_{68}) + 2sin(2x_{69}x_{70})cos(2x_{71}x_{72}) + cos(0.1 \times 3.1472 \times 5x_{73}) \\ &\quad + sin(0.3 \times 6 \times 3.1472x_{74}) - cos(4x_{75}) + 8x_{78}x_{77}sin(3x_{76}) \end{aligned}$$

$$\begin{aligned} f^{Step}(x_{79,98}) &= H(x_{79} < 0.05, 3x_{79} + 0.1, 3x_{79}) + H(x_{80} < 0.1, 3x_{80} - 0.5, 3x_{80}) \\ &\quad - H(x_{81} < 0.15, 0.1, 0.5) + H(x_{82} < 0.2, 0.5x_{82} - 4, 5x_{82}) \\ &\quad - H(x_{83} < 0.25, x_{83} + 1, x_{83}) + H(x_{84} < 0.3, 3x_{84} + 0.1, 3x_{84}) \\ &\quad - H(x_{85} < 0.35, 2x_{85} - 1.5, 2x_{85})H(x_{86} < 0.4, 0.1, 0.5) \\ &\quad + H(x_{87} < 0.45, 0.5x_{87} - 4, 0.5x_{87}) - H(x_{88} < 0.5, x_{88} + 1, x_{88}) \\ &\quad + H(x_{89} < 0.05, x_{89} + 0.2, x_{89}) + H(x_{90} < 0.1, 2x_{90} - 0.2, 2x_{90}) \\ &\quad - H(x_{91} < 0.15, 0.1, 0.5) + H(x_{92} < 0.2, 0.66x_{92} - 4, 0.66x_{92}) \\ &\quad - H(x_{93} < 0.25, x_{93} - 0.2, x_{93})H(x_{94} < 0.05, 3x_{94} + 0.1, 3x_{94}) \\ &\quad + H(x_{95} < 0.1, 3x_{95} - 0.5, 3x_{95}) - H(x_{96} < 0.15, 0.1, 0.5) \\ &\quad + H(x_{97} < 0.2, 0.5x_{97} - 4, 0.5x_{97}) - H(x_{98} < 0.25, x_{98} + 1, x_{98}) \end{aligned}$$

where $H(a, b(x), c(x)) = \begin{cases} b(x) & \text{if } a \text{ is true} \\ c(x) & \text{otherwise} \end{cases}$.