Sequential Design of Computer Experiments for Robust Parameter Design

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Jeffrey S. Lehman, M.S.

* * * * *

The Ohio State University

2002

Dissertation Committee:

Dr. William I. Notz, Co-Adviser

Dr. Thomas J. Santner, Co-Adviser

Dr. Angela M. Dean

Approved by

Co-Adviser

Co-Adviser
Department of Statistics

ABSTRACT

Many physical systems can be modeled mathematically so that "responses" are computable at arbitrary "experimental" inputs using numerical methods implemented by a complex computer code. In some cases, such computer codes allow us to conduct analogs of physical experiments that would not be possible due to the complexity of the required physical system, cost of the physical experiment, or time constraints. In a computer experiment, a response, $y(\boldsymbol{x})$, usually deterministic, is computed for each set of input variables, $\boldsymbol{x}$, according to some experimental design strategy. Then, as in physical experiments, the relationship between $\boldsymbol{x}$, the inputs, and $y(\boldsymbol{x})$, the outputs, is studied.

We are concerned with the design of computer experiments when there are two types of inputs: control variables, $\boldsymbol{x}_c$, and environmental variables, $\boldsymbol{x}_e$. Control variables are set by a product designer and environmental variables are those that are not controlled in the field but have some probability distribution characterizing a population of interest. Our interest is in the mean response $\mu(\boldsymbol{x}_c) = \mathrm{E}[y(\boldsymbol{x}_c, \boldsymbol{X}_e)]$ as a function of the control variables, where the expectation is taken over the distribution of the environmental variables. The goal is to find a robust choice of control variables. We review different methods of defining robustness and focus on finding a set of control variables at which the response is insensitive to the value of the environmental variables. Such a choice ensures that the mean response is insensitive to perturbations

of the nominal environmental variable distribution. We present a sequential strategy to select the inputs at which to observe the response so as to determine a robust setting of the control variables. Our solution is Bayesian; the prior takes the response as a draw from a stationary Gaussian stochastic process. The idea of the sequential algorithm is to compute the "improvement" over the current optimal robust setting for each untested site given the previous information; the design selects the next site to maximize an expected improvement.

Dedicated to Amy

ACKNOWLEDGMENTS

I would first like to thank my advisers, Professor Thomas Santner and Professor William Notz, for the support, friendship, and guidance that they have given me throughout my graduate study. The teaching and ideas that they shared with me were of tremendous help in my professional development and their kindness, flexibility and friendship were very meaningful to me personally.

The motivation for this research stems from the work of Brian Williams of the RAND Corporation, and a project that I worked on with my co-advisers and Kevin Ong, Professor Donald Bartel and others at Cornell University. I would like to thank Brian for helping me as I began my research in computer experiments and for the advice he has given me over the past several years. And I would like to thank Kevin for his patience in answering all of my questions.

I would also like to thank the faculty members of this department for creating such a great learning environment, and for the fine education that many of them have provided me over the years. In particular, I want to thank Professor Angela Dean for her excellence in teaching and for serving as a member of my dissertation committee, and Professor Douglas Wolfe for his kindness, help, and for always having an open door. They are both exemplary professors, and I am truly grateful for the effort that they have put into my education and the time that they have shared with me.

Thanks to my family and parents for the support they have given me throughout my life. I would not be here were it not for their efforts and sacrifices.

Finally, I want to express my deepest gratitude to my wife Amy for her love, for all the support she has given me in my education, for all the wonderful times we have spent together in the past years, and for all the hopes and dreams of many more wonderful times in the years to come.

VITA

June 18, 1975	Born - Washington, D.C.
1997	B.S. Mathematics, Millersville University
1999	M.S. Statistics, The Ohio State University
1997-present	Fellow, Graduate Teaching Associate and Graduate Research Associate, The Ohio State University
2001	Stat Analyst Intern, Battelle Memorial Institute, Columbus, Ohio

PUBLICATIONS

Research Publications

J.S. Lehman, D.A. Wolfe, A.M. Dean and B.A. Hartlaub, "Rank-based procedures for analysis of factorial effects." *Recent Developments in Design of Experiments and Related Topics*, 35-64, 2001.

FIELDS OF STUDY

Major Field: Statistics

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Computer experiments are a new breed of experiment in the computer dependent world of the 21^{st} century. Classically, in order to study physical processes, a physical experiment was necessary. The complexity of the physical system sometimes made such experiments prohibitive, if not impossible, due to time constraints, physical constraints, and/or financial constraints. Today, complex systems that lend themselves to mathematical modeling can be studied via computer codes that simulate the physical situation. These codes, often finite element models in engineering applications, are able to compute *responses* at arbitrary inputs using numerical methods and/or simulations run to the point of no simulation error. In other words, the process of interest is governed by a mathematical model, which is solved by a computer code that produces a response given the system inputs. Using the computer code, we are able to perform what we call a *computer experiment* on the process of interest by submitting arbitrary *inputs* (the experimental factors) to the code to obtain one or more *responses*. This allows us to study the effect of various input variables, much like a physical experiment.

Computer codes have been used to study phenomena in a number of scientific areas. For example, Chang et al. (1999a) used computer codes to model proximal

bone stress shielding and implant relative motion as a function of the design of a hip prosthesis. Ye (1998) discusses using a computer model to simulate the cooling system of an injection molding process. Other application areas include the design and modelling of integrated circuits, controlled nuclear fusion devices, thermal energy storage, and chemical kinetics (see Sacks, Welch, Mitchell and Wynn (1989b) and Currin et al. (1991)).

In many ways computer experiments and physical experiments have similar features. In both cases *data* are collected and analyzed to answer research hypotheses, and to determine the relationship between the input variables and the response. Unlike physical experiments, computer experiments are deterministic (i.e. rerunning the experiment with the same inputs produces the same output). Thus, random error, an important component of physical experiments, is not present in a computer experiment, making the use of replication irrelevant to computer experiments. This deterministic character of computer experiments must be kept in mind when developing models and generating designs for computer experimental data.

At first glance, the role of statistics in computer experiments may not be apparent. A deterministic and computable response exists, making it appear that exhaustive sampling of the input space (essentially collecting infinite data) would allow researchers to answer any question of interest. Unfortunately, this is impossible in many computer experiments. The code can be very time consuming to run, often taking longer than 5 hours to compute a single response, and it can involve high-dimensional inputs. Both of these properties make a "large" number of code runs, relative to the number of inputs, infeasible. Thus, statistical approaches have concentrated on building a "cheap" predictor (*modeling*) of the deterministic function based

on a small training sample (*design*) of computed function values. As a fast surrogate, the computationally inexpensive predictor is then used to predict the response at untried inputs, and to approximately achieve research goals of interest.

Consider the problem of predicting a *deterministic* function based on a small training sample of computed function values. The first questions are what model should be fit to the data, and what type of design should comprise the training sample? This chapter presents the relevant background material and the statistical approaches proposed in the computer experiments literature to answer these questions. We discuss the analysis of computer experimental data, review some of the methods of input site selection (design) involved in computer experiments, differentiate the types of variables that typically occur in computer experiments, and describe the experimental goals typically associated with each type. Koehler and Owen (1996), Sacks, Welch, Mitchell, and Wynn (1989b), and the forthcoming book by Santner, Williams, and Notz (2003) are excellent references on the design and analysis of computer experiments.

In the remaining chapters, we propose solutions to several analysis and design problems for computer experiments. Chapter 2 presents a small simulation study comparing the prediction accuracy of several types of predictors proposed for computer experiments. In Chapter 3 we propose a strategy for the constrained optimization of *two* computer codes when a subset of the input variables are uncontrollable but vary according to some probability distribution. Chapter 4 introduces exploratory data analysis methods for investigating the robustness of a design to variations in the assumed distribution of the environmental variables. Finally, Chapter 5 proposes a sequential design strategy for determining a "robust" set of control variables when a

subset of the input variables are uncontrollable but vary according to some probability distribution.

1.1 Modeling

The modeling of computer experimental data is best viewed from the Bayesian perspective with statistical models representing the prior beliefs about the uncertain relationship between the inputs and the response. We denote the deterministic function (computer code) as $y(\cdot)$ with inputs $\boldsymbol{x} \in \boldsymbol{\mathcal{X}} \subset \mathbb{R}^p$. One of the first goals in computer experiments is prediction of $y(\cdot)$ at untried inputs in $\mathcal{X}$. Best linear unbiased prediction is one means of constructing a predictor of $y(\cdot)$. Sacks, Welch, Mitchell and Wynn (1989b) and Sacks, Schiller and Welch (1989a) discuss this method of prediction in the context of computer experiments. From the Bayesian perspective, prediction is accomplished by combining the prior information about the deterministic function with information gathered from a set of training data. Currin, Mitchell, Morris and Ylvisaker (1991), Koehler and Owen (1996) , and Neal (1999) present the Bayesian interpretation of response prediction for this setting. In the following sections we outline the approaches to modeling of computer experimental data. We begin our discussion by defining the best linear unbiased predictor and stating its form.

1.1.1 Best Linear Unbiased Prediction

Best linear unbiased prediction constructs a predictor of the computer code based on a set of training data with a known joint distribution. In computer experiments this distribution arises by treating the deterministic function as a realization of a stochastic process (or random function), $Y(\boldsymbol{x})$. The model that is typically used in

computer experiments is

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

where $Z(\boldsymbol{x})$ is a random process assumed to have mean 0, variance σ^2, and correlation function $R(\boldsymbol{x}_1, \boldsymbol{x}_2)$, so that the covariance between $Z(\boldsymbol{x}_1)$ and $Z(\boldsymbol{x}_2)$ is

$$\text{Cov}[Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)] = \sigma^2 R(\boldsymbol{x}_1, \boldsymbol{x}_2).$$

The correlation function $R(\cdot, \cdot)$ and its properties are important in determining the smoothness of the sample paths of $Z(\cdot)$ as we will see in Section 1.1.3. The regression term $\boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{\beta}$ allows for a global trend with $\boldsymbol{f}^\top(\boldsymbol{x})$ a k-vector of known regression functions and $\boldsymbol{\beta} \in \mathbb{R}^k$ a vector of unknown regression parameters.

In the analysis stage, the interest is in predicting the random variable $Y(\boldsymbol{x}_0)$, for some untried $\boldsymbol{x}_0 \in \boldsymbol{\mathcal{X}}$, based on the data $\boldsymbol{Y}^n = (Y(\boldsymbol{x}_1), ..., Y(\boldsymbol{x}_n))^\top$. As in kriging, the class of predictors is restricted to linear unbiased predictors of the form

$$\hat{Y}(\boldsymbol{x}_0) = c^\top(\boldsymbol{x}_0)\boldsymbol{Y}^n$$

(see Cressie (1993), Chapter 3). To this end, we obtain the best linear unbiased predictor (BLUP) by choosing the vector $c(\boldsymbol{x}_0)$ to minimize the mean squared error of prediction,

$$MSE[\hat{Y}(\boldsymbol{x}_0)] = E[(\boldsymbol{c}^\top(\boldsymbol{x}_0)\boldsymbol{Y}^n - Y(\boldsymbol{x}_0))^2] \tag{1.2}$$

subject to the unbiasedness constraint

$$E[\boldsymbol{c}^\top(\boldsymbol{x}_0)\boldsymbol{Y}^n] = E[Y(\boldsymbol{x}_0)]. \tag{1.3}$$

Let $\boldsymbol{F} = [\boldsymbol{f}(\boldsymbol{x}_1), ..., \boldsymbol{f}(\boldsymbol{x}_n)]^\top$ be the $n \times k$ design matrix of $\boldsymbol{Y}^n$ (assume $\boldsymbol{F}$ has full column rank k), $\boldsymbol{R} = \{R(\boldsymbol{x}_i, \boldsymbol{x}_j)\}$ for $i, j \in \{1, ..., n\}$ be the $n \times n$ correlation matrix

of $\boldsymbol{Y}^n$, and $\boldsymbol{r}(\boldsymbol{x}_0) = (R(\boldsymbol{x}_1, \boldsymbol{x}_0), ..., R(\boldsymbol{x}_n, \boldsymbol{x}_0))^\top$ be the $n \times 1$ vector of correlations of $\boldsymbol{Y}^n$ with $Y(\boldsymbol{x}_0)$. With these definitions, the unbiasedness constraint (1.3) becomes

$$\boldsymbol{c}^\top(\boldsymbol{x}_0)\boldsymbol{F}\boldsymbol{\beta} = \boldsymbol{f}^\top(\boldsymbol{x}_0)\boldsymbol{\beta} \quad \forall\ \boldsymbol{\beta} \in \mathbb{R}^k \quad \Longleftrightarrow \quad \boldsymbol{F}^\top\boldsymbol{c}(\boldsymbol{x}_0) = \boldsymbol{f}(\boldsymbol{x}_0). \tag{1.4}$$

Then, using (1.4) in (1.2) we have,

$$MSE[\hat{Y}(\boldsymbol{x}_0)] = \sigma^2[1 + \boldsymbol{c}^\top(\boldsymbol{x}_0)\boldsymbol{R}\boldsymbol{c}(\boldsymbol{x}_0) - 2\boldsymbol{c}^\top(\boldsymbol{x}_0)\boldsymbol{r}(\boldsymbol{x}_0)]. \tag{1.5}$$

To perform the constrained minimization, introduce Lagrange multipliers, $\boldsymbol{\lambda}(\boldsymbol{x}_0)$, for the unbiasedness constraint, take derivatives with respect to $\boldsymbol{c}(\boldsymbol{x}_0)$ and $\boldsymbol{\lambda}(\boldsymbol{x}_0)$, and set the derivative equal to zero to see that the coefficient vector, $\boldsymbol{c}(\boldsymbol{x}_0)$, must satisfy

$$\begin{pmatrix} 0 & \boldsymbol{F}^\top \\ \boldsymbol{F} & \boldsymbol{R} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}(\boldsymbol{x}_0) \\ \boldsymbol{c}(\boldsymbol{x}_0) \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{x}_0) \\ \boldsymbol{r}(\boldsymbol{x}_0) \end{pmatrix}. \tag{1.6}$$

Inverting the partitioned matrix on the left hand side and solving for $\boldsymbol{c}(\boldsymbol{x}_0)$, the BLUP becomes

$$\hat{Y}(\boldsymbol{x}_0) = \boldsymbol{f}^\top(\boldsymbol{x}_0)\hat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x}_0)\boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}}), \tag{1.7}$$

where $\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{Y}^n$, the usual generalized least-squares estimate of $\boldsymbol{\beta}$. Substituting back into (1.5), the MSE of the BLUP can be shown to be

$$MSE[\hat{Y}(\boldsymbol{x}_0)] = \sigma^2 \left[1 \ - \ (\boldsymbol{f}^\top(\boldsymbol{x}_0)\ \boldsymbol{r}^\top(\boldsymbol{x}_0)) \begin{pmatrix} 0 & \boldsymbol{F}^\top \\ \boldsymbol{F} & \boldsymbol{R} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{f}(\boldsymbol{x}_0) \\ \boldsymbol{r}(\boldsymbol{x}_0) \end{pmatrix}\right], \tag{1.8}$$

or, the more convenient form,

$$\begin{aligned} MSE[\hat{Y}(\boldsymbol{x}_0)] = \sigma^2 \ \big[\ \ 1 \ - \ \boldsymbol{r}_0^\top\boldsymbol{R}^{-1}\boldsymbol{r}_0 \quad + \\ (\boldsymbol{f}_0^\top - \boldsymbol{r}_0^\top\boldsymbol{R}^{-1}\boldsymbol{F})(\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}(\boldsymbol{f}_0^\top - \boldsymbol{r}_0^\top\boldsymbol{R}^{-1}\boldsymbol{F})^\top\big], \end{aligned} \tag{1.9}$$

where $\boldsymbol{r}_0 = \boldsymbol{r}(\boldsymbol{x}_0)$ and $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$.

There are two important things to note about the formula (1.7) for the BLUP and its MSE, (1.9). First, the BLUP is an interpolating predictor, i.e., it satisfies $\hat{Y}(\boldsymbol{x}_i) =$

$Y(\boldsymbol{x}_i)$ for $i \in \{1, ..., n\}$, a desirable property in light of the deterministic nature of the computer code; we *know* the value of $y(\cdot)$ at $\boldsymbol{x}_i$, so our predictor should reflect this knowledge. To see the interpolating character of $\hat{Y}(\boldsymbol{x}_i)$, note that $\boldsymbol{r}(\boldsymbol{x}_i)\boldsymbol{R}^{-1}$ is a unit vector with 1 in the i^{th} position and 0 everywhere else. Using this fact it is easy to show that $\hat{Y}(\boldsymbol{x}_i) = Y(\boldsymbol{x}_i)$, and $MSE[\hat{Y}(\boldsymbol{x}_i)] = 0$ for $i \in \{1, ..., n\}$.

Second, the formula (1.7) for the BLUP assumes that $R(\cdot)$, the correlation function, is *known*, which is typically not the case. Usually a parametric family of correlation functions is specified and the parameters of the family are estimated and substituted into the above equations to produce the BLUP. We call the resulting predictor the EBLUP (empirical best linear unbiased predictor) to reflect that this is an empirical calculation based on the estimates of the correlation parameters. Section 1.1.3 presents several common parametric families of correlation functions, and Section 1.1.4 discusses several approaches to the estimation of the correlation parameters.

1.1.2 Bayesian Prediction and Posterior Distributions

The Bayesian perspective provides a more general development of prediction for the computer experiments setting. Recall that, in general, the best MSE predictor of $Y(\boldsymbol{x}_0)$ given $\boldsymbol{Y}^n$ is the conditional expectation $E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n]$. Letting $\hat{Y}(\boldsymbol{x}_0)$ be an

arbitrary predictor based on $\boldsymbol{Y}^n$ we have

$$\begin{aligned}
MSE(\hat{Y}(\boldsymbol{x}_0)) &= E\left[(\hat{Y}(\boldsymbol{x}_0) - Y(\boldsymbol{x}_0))^2\right] \\
&= E\left[(\hat{Y}(\boldsymbol{x}_0) - E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n] + E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n] - Y(\boldsymbol{x}_0))^2\right] \\
&= E\left[(\hat{Y}(\boldsymbol{x}_0) - E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n])^2\right] + MSE(E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n]) + \\
&\quad 2E\left[(\hat{Y}(\boldsymbol{x}_0) - E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n])(E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n] - Y(\boldsymbol{x}_0))\right] \\
&\geq MSE(E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n]) + \\
&\quad 2\ E\left[(\hat{Y}(\boldsymbol{x}_0) - E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n])\ (E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n] - Y(\boldsymbol{x}_0))\right] \\
&= MSE(E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n]) + 0,
\end{aligned}$$

where the final equality holds by conditioning on $\boldsymbol{Y}^n$, and noting that given $\boldsymbol{Y}^n$ the second term is 0. Thus, $E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n]$ has smaller MSE than any other predictor that depends on $\boldsymbol{Y}^n$. Currin, Mitchell, Morris and Ylvisaker (1991) propose using this posterior mean of $Y(\boldsymbol{x}_0)$ given $\boldsymbol{Y}^n$ as the prediction function. As we will see later, the BLUP in Equation (1.7) is the posterior mean for a specific choice of the prior distributions.

By calculating not only the posterior mean of $Y(\boldsymbol{x}_0)$ given the data $\boldsymbol{Y}^n$, but also the complete posterior distribution $[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n]$, we can obtain much more information. As a simple example, suppose the parameters $R(\cdot)$, $\boldsymbol{\beta}$, and σ^2 in model (1.1) are known, and that $Z(\cdot)$ is a Gaussian stochastic process. We obtain the posterior distribution $[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n]$ by noting that

$$\begin{pmatrix} Y(\boldsymbol{x}_0) \\ \boldsymbol{Y}^n \end{pmatrix} \sim \mathcal{N}_{n+1}\left(\begin{pmatrix} \boldsymbol{f}^\top(\boldsymbol{x}_0) \\ \boldsymbol{F} \end{pmatrix}\boldsymbol{\beta}, \sigma^2 \begin{pmatrix} 1 & \boldsymbol{r}^\top(\boldsymbol{x}_0) \\ \boldsymbol{r}(\boldsymbol{x}_0) & \boldsymbol{R} \end{pmatrix}\right),$$

where $\boldsymbol{r}(\boldsymbol{x}_0)$, $\boldsymbol{F}$, and $\boldsymbol{R}$ are the quantities described in Section 1.1.1. Then, applying Theorem B.0.7 from the Appendix we have

$$[Y(\boldsymbol{x}_0)|\boldsymbol{Y}^n] \quad \sim \quad \mathcal{N}_1\left(\boldsymbol{f}^\top(\boldsymbol{x}_0)\boldsymbol{\beta} + \boldsymbol{r}^\top(\boldsymbol{x}_0)(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\beta}),\ \sigma^2[1 - \boldsymbol{r}^\top(\boldsymbol{x}_0)\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}_0)]\right),$$

which gives

$$E[Y(\boldsymbol{x}_0) \mid \boldsymbol{Y}^n] = \boldsymbol{f}^\top(\boldsymbol{x}_0)\boldsymbol{\beta} + \boldsymbol{r}^\top(\boldsymbol{x}_0)\boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\beta}) \tag{1.10}$$

Of course $R(\cdot)$ (or at least its parameters), $\boldsymbol{\beta}$ and σ^2 are rarely known in practice. One method of handling unknown parameters is to estimate them via maximum likelihood (or some other means) and plug in the estimated values wherever necessary. If $R(\cdot)$ is known and we estimate $\boldsymbol{\beta}$ by the typical generalized least squares estimate, $\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{Y}^n$, substituting into (1.10) gives the BLUP (1.7). The fully Bayesian solution to the problem of unknown parameters is to assign prior distributions to the parameters and integrate out those parameters to obtain the posterior distribution, and/or the posterior mean. As an example, suppose now that σ^2 and $R(\cdot)$ are known, and $\boldsymbol{\beta}$ has the non-informative prior, $[\boldsymbol{\beta}] \propto 1$, then it can be shown that the posterior distribution $[Y(\boldsymbol{x}_0) \mid \boldsymbol{Y}^n]$ is Gaussian with mean given by (1.7) and variance given by (1.8) (see Handcock and Stein (1993)).

If, in addition, we place Jeffrey's prior on σ^2 so that $[\boldsymbol{\beta}, \sigma^2] \propto \frac{1}{\sigma^2}$, the predictive distribution of $Y(\boldsymbol{x}_0)$ turns out to be a shifted t distribution with $n - k$ degrees of freedom (recall k is the dimension of $\boldsymbol{\beta}$). O'Hagan (1992) also establishes a shifted t distribution for the posterior of $Y(\boldsymbol{x}_0)$ (see Lemma B.0.1), after placing a Gaussian prior on the distribution of $\boldsymbol{\beta}$ given σ^2 and an inverse chi-square prior on σ^2.

A more extensive treatment of assigning priors to the parameters of this model (still assuming $R(\cdot)$ is known) can be found in Williams (2000b) and in Santner et

al. (2003) where the following four priors, corresponding to informative and non-informative choices for the terms $[\boldsymbol{\beta}|\sigma^2]$ and $[\sigma^2]$, are considered. They show that if

	$[\sigma^2]$	
$[\boldsymbol{\beta} \mid \sigma^2]$	$c_0 \times \chi^{-2}_{\nu_0}$	$1/\sigma^2$
$N(\boldsymbol{b}_0, \sigma^2 \boldsymbol{V}_0)$	(1)	(2)
1	(3)	(4)

Table 1.1: Four $[\boldsymbol{\beta}, \sigma^2]$ priors corresponding to informative and non-informative choices for $[\boldsymbol{\beta} \mid \sigma^2]$ and $[\sigma^2]$.

$Z(\cdot)$ in model (1.1) is a Gaussian process and if the parameters $\boldsymbol{\beta}, \sigma^2$ have one of the priors corresponding to the four products (1) - (4) in Table 1.1, then the posterior distribution of $Y(\boldsymbol{x}_0)$ given $\boldsymbol{Y}^n$ is a scaled and shifted t distribution.

As before, these calculations assume that the correlation function is known. Where this is not the case, the usual approach is to choose a parametric form for $R(\cdot)$ (eg. (1.12)), estimate the parameters of the chosen parametric form, and use the estimated $R(\cdot)$ as the known $R(\cdot)$ in constructing the required distributions and predictors. We call the results *empirical* distributions and predictors. In Section 1.1.4 we discuss several methods of estimating the parameters of a chosen parametric correlation function.

If we assume that $R(\cdot)$ has a parametric form that depends on the unknown parameters $\boldsymbol{\gamma}$, the final piece of a fully Bayesian treatment is to assign a prior distribution to $\boldsymbol{\gamma}$ and integrate out $\boldsymbol{\gamma}$ to obtain the posterior mean of $Y(\boldsymbol{x}_0)$

$$E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}_n] = \int E[Y(\boldsymbol{x}_0)|\boldsymbol{Y}_n, \boldsymbol{\gamma}] p(\boldsymbol{\gamma}|\boldsymbol{Y}^n) d\boldsymbol{\gamma}$$

or the posterior distribution of $Y(\boldsymbol{x}_0)$

$$[Y(\boldsymbol{x}_0)|\boldsymbol{Y}_n] = \int [Y(\boldsymbol{x}_0)|\boldsymbol{Y}_n, \boldsymbol{\gamma}] p(\boldsymbol{\gamma}|\boldsymbol{Y}^n) d\boldsymbol{\gamma}.$$

Of course this integration is non-trivial, typically high-dimensional (due to $\boldsymbol{\gamma}$ being high-dimensional), and is often prohibitive. So, the empirical Bayes procedure, discussed above, is often used.

1.1.3 Parametric Correlation Functions

The derivations in the previous section have assumed that the correlation function, $R(\cdot)$, and its parameters, which we will denote by $\boldsymbol{\gamma}$, are known. To be a valid correlation function $R(\cdot)$ must have $R(\mathbf{0}) = 1$ and be a positive semidefinite function, i.e. it must satisfy

$$\sum_{j,k=1}^{n} c_j c_k R(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \tag{1.11}$$

for all finite n, all $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, and all real $c_1, ..., c_n$. We restrict ourselves to stationary correlation functions which further satisfy $R(\boldsymbol{x}_1, \boldsymbol{x}_2) = R(\boldsymbol{x}_1 - \boldsymbol{x}_2)$. This assumption allows us to "learn" about the process based on a single realization of the process by assuming that the correlation function depends only on the distance and direction between two points, and not on the location of those points (see Stein (1999) Section 2.1 or Cressie (1993) Section 2.3).

Appendix A summarizes the properties of stochastic processes, and the relationship that these properties have with the correlation function $R(\cdot)$. The primary concept to note is that the properties of $R(\cdot)$ determine the smoothness of the sample paths of the random process $Z(\cdot)$. This is important since we are using $Z(\cdot)$ as a model for the computer code, and properties of $Z(\cdot)$ should reflect our prior beliefs about the smoothness of the computer code output.

This thesis will focus on two *product type* correlation functions: the Power Exponential and the Matérn correlation function. In the following we assume that $\boldsymbol{x} \in \mathbb{R}^p$.

Definition 1.1.1 (Power Exponential). The product *power exponential* correlation function has the form

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{p} \exp\big(- \theta_i |x_{1,i} - x_{2,i}|^{\alpha_i}\big), \tag{1.12}$$

where $\theta_i > 0$ and $0 < \alpha_i \leq 2$ for $i \in \{1, ..., p\}$.

We call θ_i the scale parameter for the i^{th} coordinate. As θ_i increases the dependence between fixed input sites decreases since $R(\cdot)$ decreases. If $\alpha_i = 2$, then the $Z(\cdot)$ process from model (1.1) is infinitely mean square differentiable in direction i, and if all $\alpha_i = 2$ then the sample paths of $Z(\cdot)$ are almost surely infinitely differentiable (see Appendix A).

Definition 1.1.2 (Matérn Correlation Function). The product *Matérn* correlation function has the form

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{p} \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{2\sqrt{\nu}|x_{1,i} - x_{2,i}|}{\theta_i} \right)^{\nu} \mathrm{K}_{\nu} \left(\frac{2\sqrt{\nu}|x_{1,i} - x_{2,i}|}{\theta_i} \right), \tag{1.13}$$

where $\nu > 0$, $\theta_i > 0$ and $\mathrm{K}_{\nu}(\cdot)$ is the modified Bessel function of order ν (see Stein (1999) Section 2.7).

The attractive property of this correlation function is that it has a parameter that controls the smoothness of the Z process. For the *power exponential*, either sample paths of $Z(\cdot)$ are infinitely differentiable ($\alpha_i = 2$ for all i), or they are not differentiable at all ($\alpha_i < 2$). For the Matérn correlation function the parameter ν controls the smoothness of $Z(\cdot)$ in that $Z(\cdot)$ is m times *mean-square* differentiable if

and only if $\nu > m$ (see Stein (1999) Section 2.7), and if $Z(\cdot)$ is Gaussian, its sample paths are *almost surely* m times differentiable if $\nu > m$ (see Cramér and Leadbetter (1967) Secs. 9.2-9.5). Note that the power exponential correlation function with $\alpha_1 = \cdots = \alpha_p = 2$ and $1/\theta_i^2$ in place of θ_i in (1.12) is the limiting case of the Matérn correlation function (1.13) as $\nu \longrightarrow \infty$

There are, of course, a large number of functions satisfying (1.11). For additional examples of correlation functions used to model output of computer experiments see Koehler and Owen (1996), Sacks et al. (1989a), and Currin et al. (1991).

1.1.4 Estimation of Correlation Parameters

Once a parametric correlation function is chosen, estimation of the parameters of that function becomes necessary to compute the equations and formulas in Sections 1.1.1 and 1.1.2. As mentioned before, we typically estimate the parameters, $\boldsymbol{\gamma}$, of the correlation function, and plug the estimates into the prediction formulas as known values, thereby producing empirical best linear unbiased predictors (EBLUPs). In general, different estimates of $\boldsymbol{\gamma}$ will produce different EBLUPs. Sacks et al. (1989a) and Currin et al. (1991) suggest estimation of $\boldsymbol{\gamma}$ via maximum likelihood, Stein (1999 Chapter 6) and Cressie (1993 Chapter 2) present estimation of $\boldsymbol{\gamma}$ via restricted maximum likelihood, and Handcock and Stein (1993) discuss a posterior mode method of estimation. A fourth means of estimation of $\boldsymbol{\gamma}$ is cross validation. In Chapter 2 we present a small simulation study comparing the predictive ability of the maximum likelihood, restricted maximum likelihood, and cross validation EBLUPs.

Maximum Likelihood Estimation

Let $\boldsymbol{Y}^n = (Y(\boldsymbol{x}_1), ..., Y(\boldsymbol{x}_n))^\top$ be the vector of output obtained from running the computer code at the input sites $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$. Then, using Model (1.1) and assuming that $Z(\cdot)$ is a Gaussian process, we obtain

$$\boldsymbol{Y}^n \sim \mathcal{N}_n\left(\boldsymbol{F}\boldsymbol{\beta}, \sigma^2\boldsymbol{R}\right)$$

where $\boldsymbol{F} = (\boldsymbol{f}(\boldsymbol{x}_1), ..., \boldsymbol{f}(\boldsymbol{x}_n))^\top$ is the $n \times k$ regression matrix, $\boldsymbol{\beta}$ is the $k \times 1$ vector of unknown regression parameters, and the $n \times n$ matrix $\boldsymbol{R}$ is the correlation matrix of $\boldsymbol{Y}^n$ so that the $(i,j)^{th}$ entry of $\boldsymbol{R}$ is $\text{Corr}[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)]$. Note that this quantity depends on the correlation function chosen and the values of its parameters. So, the parameters of the model are $\boldsymbol{\beta}$, σ^2 and the parameters of the correlation function, $R(\cdot)$, which we will denote as $\boldsymbol{\gamma}$. The likelihood function is:

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma}, \boldsymbol{Y}^n) &= \tfrac{1}{(2\pi)^{n/2}|\sigma^2\boldsymbol{R}|^{1/2}} \times \\ &\quad \exp\left[-\tfrac{1}{2}(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\beta})^\top(\sigma^2\boldsymbol{R})^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\boldsymbol{\beta})\right]. \end{aligned} \tag{1.14}$$

Given $\boldsymbol{\gamma}$, the MLE's of $\boldsymbol{\beta}$ and σ^2 are

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^\top\boldsymbol{R}^{-1}\boldsymbol{Y}^n, \tag{1.15}$$

and

$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top\boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{1.16}$$

Substituting these formulas into (1.14), maximizing (1.14) to obtain the MLE, $\hat{\boldsymbol{\gamma}}$, of $\boldsymbol{\gamma}$ is equivalent to minimizing $|\boldsymbol{R}|^{1/n}\hat{\sigma}^2$, which is a function only of the correlation parameters $\boldsymbol{\gamma}$, and the data $\boldsymbol{Y}^n$, as desired. We then substitute $\hat{\boldsymbol{\gamma}}$ into (1.15) and (1.16) to obtain the MLE's of $\boldsymbol{\beta}$ and σ^2, respectively.

Welch et al. (1992) discuss an algorithm for calculating the maximum likelihood estimate of $\boldsymbol{\gamma}$ when the input space is high-dimensional (i.e. p is large). For p large, the number of correlation parameters for both the Matérn and power exponential correlation functions is large and maximizing (1.14) becomes very difficult. Welch et al. describe an iterative procedure to reduce the number of correlation parameters by screening out "unimportant" variables and allowing only "important" variables to have their own correlation parameter.

Restricted Maximum Likelihood

The restricted maximum likelihood (REML) approach to estimation of $\boldsymbol{\gamma}$ attempts to reduce the bias present in the maximum likelihood estimate of the variance of the process. REML estimators are found by maximizing the likelihood of a set of error contrasts of the data. Stein (1999) and Harville (1974) state that the REML estimates of $(\sigma^2, \boldsymbol{\gamma})$ maximize

$$\begin{aligned} \ell(\sigma^2, \boldsymbol{\gamma}|\boldsymbol{Y}^n) \;\; \propto \;\; & -\tfrac{1}{2}[(n-k)\log(\sigma^2) \;+\; \log|\boldsymbol{R}| \;+\; \log|\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F}| \;+ \\ & (\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})/\sigma^2]. \end{aligned} \tag{1.17}$$

Maximizing (1.17) over σ^2, for fixed $\boldsymbol{\gamma}$, gives

$$\tilde{\sigma^2} = \frac{1}{n-k}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{1.18}$$

Thus, the REML estimate of $\boldsymbol{\gamma}$ is obtained by maximizing

$$-\frac{1}{2}[(n-k)\log(\tilde{\sigma^2}) \;+\; \log|\boldsymbol{R}| \;+\; \log|\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F}|] \tag{1.19}$$

which is a function of $\boldsymbol{Y}^n$ and $\boldsymbol{\gamma}$. The REML estimate of σ^2 is then obtained by substitution of $\hat{\boldsymbol{\gamma}}$ into (1.18). Harville (1974) gives a Bayesian justification for REML, by showing that when a noninformative prior is placed on $\boldsymbol{\beta}$, the posterior density of $(\sigma^2, \boldsymbol{\gamma})$ is proportional to the restricted log likelihood in (1.17).

Estimation Via Cross-Validation

Leave-one-out cross validation has been studied extensively in the standard regression setting, and offers a heuristic prediction-oriented alternative to likelihood based estimation in the computer experiments setting. The idea of cross-validation is to remove the i^{th} observation and use the BLUP based on the remaining $n-1$ observations to predict the value for the i^{th} observation. The cross-validation choice for the correlation parameters is the one that makes the predicted value of $y(\boldsymbol{x}_i)$ closest to the true value, averaging over all n observations. More formally, to estimate $\boldsymbol{\gamma}$ we minimize the quantity

$$f(\boldsymbol{\gamma}) = \sum_{i=1}^{n} (\hat{Y}_{-i}(\boldsymbol{x}_i) - y(\boldsymbol{x}_i))^2, \tag{1.20}$$

where $\hat{Y}_{-i}(\boldsymbol{x}_i)$ is the BLUP (equation (1.7)) of $y(\boldsymbol{x}_i)$ based on the $n-1$ observations obtained by removing $y(\boldsymbol{x}_i)$ from $(y(\boldsymbol{x}_1), ..., y(\boldsymbol{x}_n))$. This strategy of estimation is investigated more closely in Chapter 2, where it is compared to maximum likelihood and restricted maximum likelihood estimation.

Posterior Mode Estimation

A final, and fully Bayesian, method of estimating $\boldsymbol{\gamma}$ results from calculation of the posterior distribution of $\boldsymbol{\gamma}$ given the data $\boldsymbol{Y}^n$ and the prior distribution of $\boldsymbol{\gamma}$, $[\boldsymbol{\gamma}]$. An empirical Bayes estimator of $\boldsymbol{\gamma}$ can be obtained as the posterior mode of this distribution. The posterior mode of $\boldsymbol{\gamma}$ is the value that maximizes

$$[\boldsymbol{\gamma}|\boldsymbol{Y}^n] = \frac{[\boldsymbol{Y}^n|\boldsymbol{\gamma}][\boldsymbol{\gamma}]}{[\boldsymbol{Y}^n]}.$$

Handcock and Stein (1993) show that if $[\boldsymbol{\beta}, \sigma^2] \propto \frac{1}{\sigma^2}$ (the non-informative choice of prior corresponding to (4) in Table 1.1) then the posterior distribution of $\boldsymbol{\gamma}$ satisfies

$$p(\boldsymbol{\gamma} \mid \boldsymbol{Y}^n) \propto p(\boldsymbol{\gamma})|\boldsymbol{R}|^{-1/2}|\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F}|^{-1/2}[\tilde{\sigma^2}]^{-(n-k)/2}, \tag{1.21}$$

where $p(\boldsymbol{\gamma})$ is the prior density of $\boldsymbol{\gamma}$, $\hat{\boldsymbol{\beta}}$ and $\tilde{\sigma^2}$ are defined in (1.15) and (1.18), and $\boldsymbol{R}$ and $\boldsymbol{F}$ are as defined in (1.14). The posterior mode of $\boldsymbol{\gamma}$ is the value of $\boldsymbol{\gamma}$ that maximizes (1.21). Note that if we assume $p(\boldsymbol{\gamma}) \propto 1$, the posterior mode of $\boldsymbol{\gamma}$ is equivalent to the REML estimator of $\boldsymbol{\gamma}$ because maximizing (1.21) is equivalent to maximizing the REML likelihood (1.19) (see Harville (1974)) .

1.1.5 Multivariate Modeling

Consider an application where multiple computer codes are available producing g related or competing responses on the same input space. For example, in the design of total hip replacements Chang et al. (1999a) discuss two competing responses, bone stress shielding and implant toggling.

An important issue is how to model this multi-response computer experiment data. One option is independent univariate modeling of each response. However, for the case where the multiple responses are related, intuitively the data can be better represented by a model that allows for some association between the outputs, $Y_i(\cdot)$ for $i = 1, ..., g$. We model the multiple outputs $Y_i(\cdot)$ as in Model (1.1) so that

$$Y_i(\boldsymbol{x}) = \boldsymbol{f}_i^\top(\boldsymbol{x})\boldsymbol{\beta} + Z_i(\boldsymbol{x})$$

where $Z_i(\boldsymbol{x})$ is a mean zero, stationary Gaussian stochastic process with variance σ_i^2 and correlation function $R_i(\cdot)$. To complete the model, we need to specify the cross-correlation functions $R_{ij}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \mathrm{Corr}[Z_i(\boldsymbol{x}_1), Z_j(\boldsymbol{x}_2)]$, so as to guarantee a

positive definite joint covariance structure for $Z_1(\cdot), ..., Z_g(\cdot)$. Note that if $R_{ij}(\cdot) = 0$ for all $i \neq j$ we have independent univariate modeling of each response.

The specification of a joint covariance structure that gives positive definite joint covariances is not a trivial problem. The literature on cokriging, for example Sun (1998), Stein and Corsten (1991) and Phillips et al. (1997), has modeled this type of data using parametric cross-covariance functions, and restricting the parameters to values that lead to positive definite joint covariance structures for the data sampled. For the case where there are *two* responses of interest, $Y_1(\cdot)$ and $Y_2(\cdot)$, an autoregressive model relating $Z_1(\cdot)$ to $Z_2(\cdot)$ is constructed by setting

$$Z_2(\boldsymbol{x}) = rZ_1(\boldsymbol{x}) + Z_\delta(\boldsymbol{x}), \tag{1.22}$$

where $Z_\delta(\boldsymbol{x})$ is a mean zero Gaussian stochastic process that is independent of $Z_1(\cdot)$ and has variance σ_δ^2 and correlation function $R_\delta(\cdot)$. From (1.22) the cross-correlation function, $R_{12}(\cdot)$, can be derived upon specification of the marginal correlation functions $R_1(\cdot)$ and $R_\delta(\cdot)$ and the parameters of the model. Kennedy and O'Hagan (2000) propose this model and present conditions which lead to positive definite joint covariance structures. This model is used in Williams, Santner, and Notz (2000c) and in Chapter 3.

Another model in the same spirit is that of VerHoef and Barry (1998) who discuss the construction of valid cross correlation functions by modeling the spatial data as a moving average over a white noise random process. This is potentially the most promising method for assuring that a multi-response model has a valid correlation structure, and can lead to a large variety of joint covariance structures.

1.2 Design

In the previous section we looked at prediction of the computer code based on n observations of the code. Here, we discuss how to select the input sites at which to observe the response (run the code). This is the design question for computer experiments. Of course, the large subject of experimental design cannot be covered in a single section, so we restrict our discussion to the designs often used in computer experiments.

When choosing experimental designs for computer experiments it is important to recall some of the distinguishing properties of computer experiments. First, responses are deterministic so that repeated observations at the same input will produce identical responses. This suggests that replication in computer experiments is not necessary and inefficient. Second, the relationship between the inputs and the response may be different in different regions of the input space and a priori we do not know which regions may contain features (such as extrema) that are of interest. This suggests that all regions of the input space should be sampled. For these reasons, one intuitive choice for the design of a computer experiment is a *"space-filling"* (or exploratory) design. Such designs are called *space-filling* because they attempt to spread observations "evenly" to cover the full range of the input space, generally without replication. This allows the researcher to gather information about the relationship between the inputs and the response for all regions of the input space. Also, by covering the full range of the input space, *space-filling* designs can (hopefully) lead to good prediction over the entire input space, which is typically a primary goal in computer experiments.

Latin hypercube designs and distance based designs are two types of space-filling designs. Latin hypercube sampling (LHS) designs spread out the observations so

that the resulting design has nice one-dimensional projection properties. There is a large body of work on LHS designs, their properties, and extensions. Section 1.2.1 discusses LHS designs and some of the proposed extensions of LHS designs. Distance based designs, on the other hand, explicitly take into account the distances between selected input sites in a candidate design or distances from selected input sites in a design to untried input sites. They spread points out by preventing points in the selected design from being too "close" (with respect to some measure of distance) to one another or from being too "far" from any untried sites. Section 1.2.2 presents several distance based design criteria, and discusses combinations of LHS designs with distance based properties. Koehler and Owen (1996) and Santner et. al. (2003) give overviews of these types of designs along with examples of each, and the computer program ACED (Welch 1985) is useful in generating both Latin hypercube designs and distance based designs.

Several other types of designs are also useful in computer experiments. When the experimental goal is optimization of the response, sequential design strategies are intuitively appealing. In Section 1.2.3 we present some of the sequential designs proposed in the literature. When the goal is minimizing mean-squared error at a given (set of) point(s), integrated mean square error designs (IMSE) and maximum mean square error (MMSE) designs are proposed for computer experiments by Sacks, Welch, Mitchell and Wynn (1989a). Mitchell, Morris and Ylvisaker (1994) discuss the relationship between distance based designs and D-optimal, G-optimal or A-optimal designs. Generating IMSE, MMSE, D-optimal, G-optimal, and A-optimal designs in this setting can be problematic because the intersite correlations need to be known before collecting any data (remember that in a typical computer experiment

we estimate the correlation parameters from the data). Under a certain asymptotic setup where the intersite correlations become progressively weaker, Mitchell et al. show that D-optimal and G-optimal designs are equivalent to certain distance based designs.

1.2.1 Latin Hypercube Sampling

Latin hypercube samples were first proposed by McKay, Beckman and Conover (1979) as an alternative to simple random sampling and stratified sampling when the interest is in estimating the mean, variance, and/or distribution function of an output $y = h(\boldsymbol{X})$, where $\boldsymbol{X} \sim F(\boldsymbol{x})$. Similar to stratified sampling, Latin hypercube samples attempt to ensure that all regions of the input space are sampled (at least marginally) by dividing the range of each of the inputs into n bins, each with equal probability, and then randomly sampling from each of these bins.

In more detail, to select an n-point Latin hypercube sample of $\boldsymbol{X} = (X_1, ..., X_p)$ we start by assuming the input $\boldsymbol{X} = (X_1, ..., X_p)$ has independent components with $X_j \sim F_j$. Divide the range of each X_j into n strata of equal marginal probability and sample once from each stratum, denoting the sample by x_{ji} for $i = \{1, ..., n\}$. These values form the X_j component of the sample, $j = \{1, ..., p\}$. Then, we match the various components randomly to form the n vectors $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_n$.

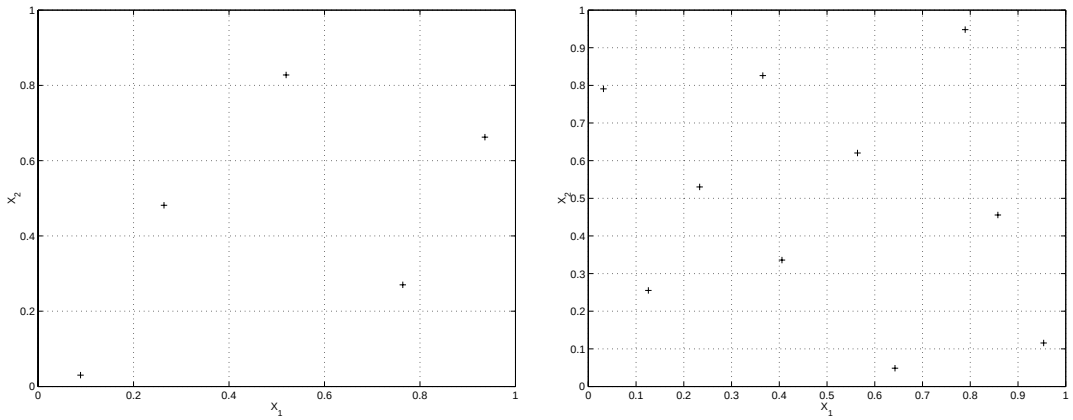
Stein (1987) describes the selection of a Latin hypercube in the following manner. Let $\boldsymbol{\Pi}$ be an $n \times p$ matrix, where each column is an independent random permutation of $\{1, 2, ..., n\}$, and let $\epsilon_{jk}, (j = 1, ..., n; k = 1, ..., p)$ be np i.i.d. Uniform(0,1) random variables. The k^{th} component of the j^{th} sample value, X_{jk} is defined as

$$X_{jk} = F_k^{-1}(\frac{1}{n}(\Pi_{jk} - 1 + \epsilon_{jk})). \tag{1.23}$$

McKay et al. (1978) demonstrate that Latin hypercube sampling (LHS) improves upon simple random sampling when $h(\cdot)$ has certain monotonicity properties ($h(\boldsymbol{X})$ is monotonic in each of its arguments). In particular, their interest is in estimating various properties, such as the mean and distribution function, of $h(\boldsymbol{X})$. They show that the estimates obtained from the Latin hypercube samples are more precise than those obtained from simple random sampling when $h(\cdot)$ has the desired properties. Stein (1987) investigates the asymptotic properties of Latin hypercube sampling and shows that as long as n is large compared to p, Latin hypercube sampling gives an estimator of the mean of $h(\boldsymbol{X})$ with lower variance than simple random sampling for any function $h(\boldsymbol{X})$ that has a finite second moment and provided some main effect is present.

For computer experiments, the initial goal is typically prediction of the response at untried inputs, $\boldsymbol{x}$. So, the inputs are often taken to have uniform distributions over their domain. For the case $p = 2$, Figure 1.1 displays a five point LHS and a ten point LHS when the marginal distributions for both X_1 and X_2 are uniform. Note the one-dimensional projection property of these designs; if we project the points onto either the horizontal or the vertical axes, we see the full range of both x_1 and x_2 are represented since each bin of length $\frac{1}{n}$ contains one point.

A relatively large literature on LHS has been written since it was introduced by McKay, Beckman and Conover. In addition to the result mentioned above, Stein (1987) also presents a method for producing Latin hypercube samples when the components of $\boldsymbol{X}$ are not independent but have some joint distribution. Owen (1992) extends Stein's work to prove a central limit theorem for LHS, proving that



5 Point LHS 10 Point LHS

Figure 1.1: Example of Latin hypercube sampling designs

$\bar{h} = \frac{1}{N}\sum_{i=1}^{N} h(\boldsymbol{x}_i)$ is asymptotically normal even when $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ is a Latin hypercube sample. Handcock (1991) introduces *Cascading Latin Hypercube* designs, whose construction is motivated by the desire to not only estimate the scale and smoothness parameters involved in the correlation function, but also the overall trend parameters in the model. Cascading Latin hypercubes attempt to cover the input space with some sites "clustered together". The sites that are "clustered together" help to estimate the parameters of the correlation function (local trend), while those that cover the input space help to estimate the global trend (the $\boldsymbol{\beta}'s$ in Model (1.1)). Figure 1.2.1 shows an example of a 2-stage Cascading Latin hypercube design of size 15. In the first stage we choose a 5 point LHS (denoted as the o's in the figure), and in the second stage we choose a 3-point Latin hypercube in the 3×3 grid surrounding each of the sites from the first stage. Note that the final design (denoted as the +'s) is a Latin hypercube design clustering sets of three points close together. The design can be easily generalized to ℓ stages if desired.

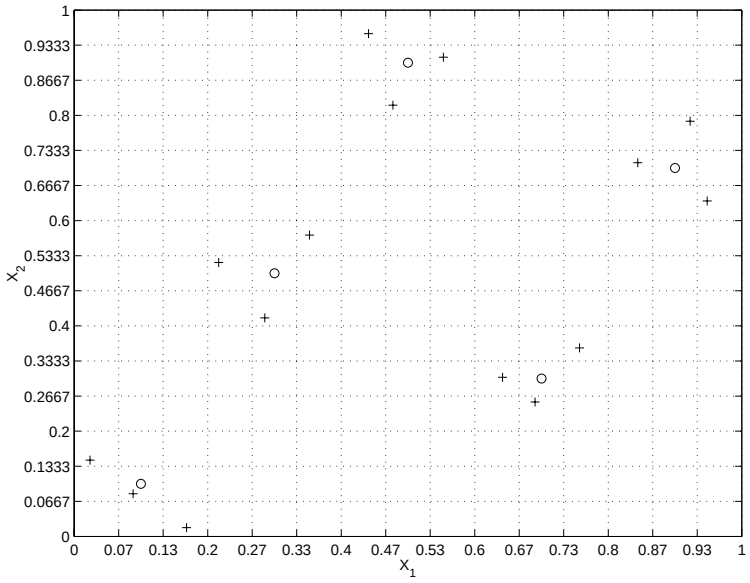


Figure 1.2: Example of a Cascading Latin hypercube design

Another extension of LHS designs are orthogonal array based Latin hypercubes found in Tang (1993) and Hoshino and Takemura (2000). These designs make it possible to stratify an m-variate margin as opposed to the usual univariate margin in simple LHS designs. Ye (1998) discusses orthogonal column Latin hypercubes that have the property that estimates of linear effects of all factors are uncorrelated with each other.

1.2.2 Distance-Based Designs

Distance based designs try to spread out the input sites more explicitly. These types of designs attempt to minimize or maximize properties of the distance between pairs of design points or distances between design points and unobserved sites in the input space. Johnson, Moore and Ylvisaker (1990) develop the setup for maximin

and minimax distance designs. Let the input space be $\mathcal{X} \subset \mathbb{R}^p$ and $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ be a distance measure on $\mathcal{X}$. For example,

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left[\sum_{i=1}^{p} (x_{1,i} - x_{2,i})^2 \right]^{1/2}$$

is the Euclidean distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Let $S_n = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \mathcal{X}$ be a potential n point design. We say that S_n^0 is a *maximin distance design* if

$$\max_{S_n} \min_{\boldsymbol{x}, \boldsymbol{x}' \in S_n} d(\boldsymbol{x}, \boldsymbol{x}') \;=\; \min_{\boldsymbol{x}, \boldsymbol{x}' \in S_n^0} d(\boldsymbol{x}, \boldsymbol{x}') \;=\; d^0, \tag{1.24}$$

and we say that S_n^0 is a *minimax distance design* if

$$\min_{S_n} \max_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}, S_n) \;=\; \max_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}, S_n^0) \tag{1.25}$$

where $d(\boldsymbol{x}, S_n) = \min_{\boldsymbol{x}_0 \in S_n} d(\boldsymbol{x}, \boldsymbol{x}_0)$. Thus, maximin distance designs attempt to spread points out by maximizing the minimum distance between any two points in the existing design. The intuition is that no two points should be too close to each other. On the other hand, minimax distance designs attempt to spread points so that any point not in the existing design is not too far away from a point that is in the existing design. The intuition is that points not in the design should be as close as possible to a point that is in the design.

There are many other criteria for distance based designs, such as maximizing the *average* distance between all pairs of design points. The software program ACED (Welch 1985) can determine various types of distance based designs. Figure 1.3 displays two distance based designs generated by ACED. The design on the right is an eight point design on $(0, 1) \times (0, 1)$ that maximizes the average distance between all pairs of design points, and the design on the left is a six point maximin distance design on $(0, 1) \times (0, 1)$.

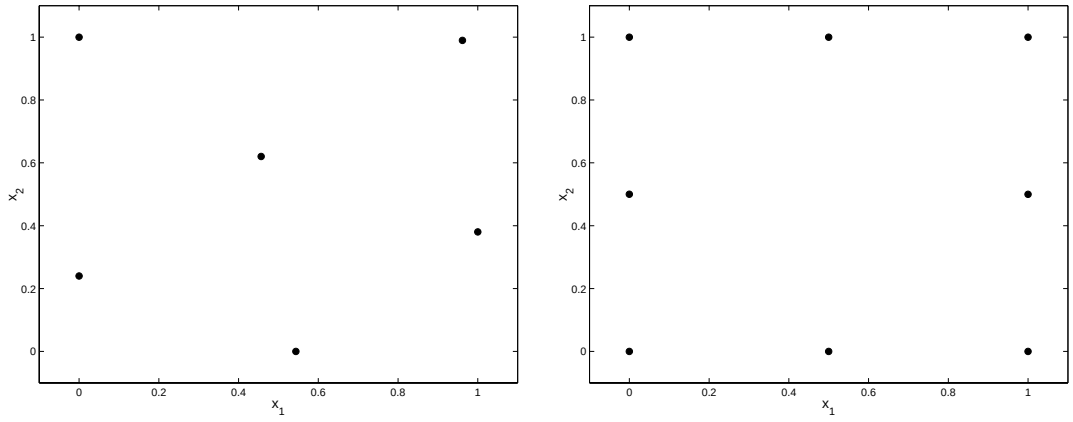


Figure 1.3: Example of 6-point maximin distance design (left panel), and 8-point distance based design that maximizes the average distance between all pairs of design points (right panel).

In some sense both LHS designs and distance based designs can miss the mark in terms of distributing the observations evenly over the input space. For example, in the pictures above we see two distance based designs that concentrate most of their observations on the outer regions of the input space, missing much of the interior of the region. Likewise, LHS designs can potentially put all the observations on the main diagonal (or anti-diagonal) of the input region. Morris and Mitchell (1995) recognize that both Latin hypercube designs, with their nice projection properties, and distance based designs have advantages for computer experiments. They propose combining the two designs by searching for a distance based design within the class of Latin hypercube designs. The program ACED (Welch (1985)) can generate distance based designs within the class of Latin hypercube designs that are centered on the marginal bins. Figure 1.4 displays the two distance based designs corresponding to Figure 1.3 with the further restriction that the design be a Latin hypercube design.

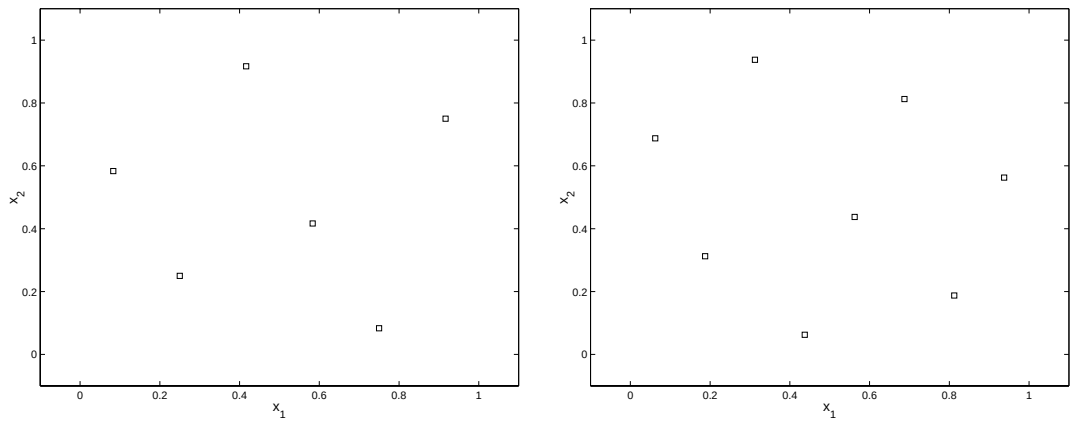


Figure 1.4: Example of 6-point LHS maximin distance design (left panel), and 8-point LHS distance based design that maximizes the average distance between all pairs of design points (right panel).

Johnson et al. (1990) point out that distance based designs are commonly non-unique and that finding designs satisfying (1.24) or (1.25) can be very difficult when $\mathcal{X}$ is infinite. Given this, they restrict themselves to the case where $\mathcal{X}$ is finite, limiting competing designs to a finite number. Morris and Mitchell (1995), on the other hand, present a simulated annealing approach to searching for optimal distance based designs. Trosset (1999a) also attempts to handle the difficulty of computing distance based designs by using an approximation that allows use of conventional nonlinear programming algorithms.

1.2.3 Sequential Designs

In some applications, one of the primary goals of a computer experiment is to find the values of the input variables that produce the "optimum" (in some sense) response. For example, the goal might be to find the $\boldsymbol{x}$ that minimizes $y(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{X}$. This seems like a standard optimization problem, and one that could be solved by

one of the many mathematical optimization routines widely available. The problem is that essentially all mathematical programming algorithms require a large number of function evaluations to find optima and this is not feasible due to the computing costs of many runs of the computer code. In recent years, statistical algorithms have been proposed for finding the "optimum" setting for the inputs. The basic approach of these algorithms is to produce a fast predictor (via statistical modeling) of the code based on a small training sample of computed responses (see Section 1.1), decide if the predictions are sufficiently "accurate", and, if so, the fast predictor is optimized over the input space. However, if the predictions are not sufficiently "accurate", more runs of the code (observations) are taken at chosen input values. This is the perfect setup for sequential design in that two questions are asked: "Have we sampled enough?", and "If not, where do we sample next?".

One of the first strategies addressing this problem was proposed by Welch and Sacks (1991) and Bernardo et al. (1992). The steps of their method are as follows:

1. Postulate an approximating model for the computer code.

2. Plan an initial experiment of n sets of $\boldsymbol{x}$ vectors and run the code at these input vectors.

3. Use the data gathered in Step 2 to fit the model.

4. Check the accuracy of prediction of the model.

5. If the model is not sufficiently accurate, choose a subregion of the input space in which the optimum appears to be and go back to Step 2 restricting the experiment to the chosen subregion.

6. If the model is sufficiently accurate, optimize the objective function using the fitted model in place of the computer code.

The approximating model in Step 1 is Model (1.1), and the correlation function that these authors use is the power exponential function given in (1.12). The prediction accuracy of the model is assessed by cross validation. Welch and Sacks apply this algorithm to the design of a voltage-shifter circuit and Bernardo et al. present several examples involving the design of manufacturable integrated circuits. Trosset and Torczon (1999) and Torczon and Trosset (1998) introduce a similar algorithm that chooses an initial grid, adds points until a minimizer is confirmed on that grid, and then refines the grid and repeats the procedure.

Jones, Schonlau and Welch (1998) present a criterion based sequential algorithm and discuss choosing the next input site at which to observe the code so as to maximize the *expected improvement*, which is computed as follows. Assume that n runs of the computer code have produced observations $(y^{(1)}, ..., y^{(n)})$. Let $f_{\min} = \min\{y^{(1)}, ..., y^{(n)}\}$ and define the expected improvement at the untried input $\boldsymbol{x}$ as $I(\boldsymbol{x}) = E[\max\{f_{\min} - Y(\boldsymbol{x}), 0\}|\boldsymbol{Y}^n]$. Note that the random variable in this calculation is $Y(\boldsymbol{x})$, since we are uncertain of the function's value at $\boldsymbol{x}$. Assuming that $Y(\cdot)$ is a Gaussian stochastic process, we can compute the distribution of $Y(\boldsymbol{x})$ given the data and the parameters. Using this distribution, and the definition of expected improvement, Jones et al. begin their algorithm by fitting Model (1.1), using the power exponential correlation function, to a set of initial points from a "space-filling"design. Then, they maximize the expected improvement to find the $(n+1)^{\text{st}}$ input site for the computer code. If the maximum expected improvement is very small, the algorithm stops and the model is used as a surrogate for the computer code in an optimization

routine. Otherwise, the computer code is run at the chosen $(n+1)^{\text{st}}$ site, and the procedure is repeated.

In Williams, Santner and Notz (2000a) a modification of this algorithm is investigated for the case where the inputs consist of both environmental variables, $\boldsymbol{x}_e$, and control variables, $\boldsymbol{x}_c$ (also see Welch, Yu, Kang and Sacks (1990) for this setting). Control variables are those variables that can be set by the product designer and environmental variables cannot be controlled but have values that follow some probability distribution representing variation in these variables for the population of interest (see Section 1.3.1 for more details of control and environmental variables). In this setting, Williams et al. (2000a) define the objective function as the mean of the computer code taken over the distribution of the environmental variables. The procedure of Jones et al. calls for direct observation of the objective function, which is infeasible in this setting since it would require too many evaluations of the code (one for each environmental variable value). Williams et al. (2000a) present an algorithm better suited for this situation.

Williams, Santner, and Notz (2000c) study the problem of constrained optimization. Specifically, they deal with the setting where two responses (perhaps related or competing) are computed on the same input space. One of the responses is considered the *objective* function, and the other is considered the *constraint* function. The goal is to optimize the *objective* function subject to an upper bound on the *constraint* function. In Chapter 3 we improve this algorithm, and further investigate the revised algorithm on a variety of test problems.

We test the sequential algorithm on a variety of test problems because evaluating the relative merits of a sequential design strategy on a single problem may be inadequate for evaluating strategies and can lead to biases. Trosset and Padula (2000) suggest measuring performance by replication. They propose not only presenting the algorithm with multiple problems, but also starting the algorithm at different starting points for the same problem. For sequential design of computer experiments, the notion of solving multiple problems by an algorithm is useful. If the algorithm, presented with a large number of random problems, manages to solve all of these problems (or at least a large proportion of them), then we might put more confidence in the merits of that algorithm. In Chapter 2 we use the proposal of Trosset and Padula (2000) and Trosset (1999b) for generating random problems in the computer experiments setting, and in Chapter 3 a parametric method of generating random problems is presented.

1.3 Experimental Goals

As in physical experiments, computer experiments can involve a large variety of experimental goals. As mentioned above, prediction of the computer code at untried inputs is one of these goals. We would like to predict $y(\boldsymbol{x})$ "well" for all $\boldsymbol{x} \in \mathcal{X}$ based on a set of training data $\{(\boldsymbol{x}_1, y(\boldsymbol{x}_1)), ..., (\boldsymbol{x}_n, y(\boldsymbol{x}_n))\}$. However, many other experimental goals are of interest in this setting. For example, interest may be in identifying important inputs (i.e., those that produce large variation in the response) and screening out non-important ones. Kleijnen and Helton (1999) propose using simple scatterplots to identify important factors (this is perhaps more useful when a large number of experimental runs are available), and Welch et al. (1992), Jones et

al. (1998), and Mrawira et al. (1999) discuss an approach for estimating the relative importance of each input in a computer model by using the predictor of $y(\boldsymbol{x})$ to calculate "main effects" and "interaction" effects of the various inputs. The recent book by Saltelli et al. (2000) presents a thorough explanation of sensitivity analysis, the study of the relationship between the inputs of a system and the outputs of that system.

For computer experiments, the experimental goal often depends on the nature of the input variables. Input variables for computer experiments fall into several different classes or types, each affecting the output of the computer code $y(\cdot)$. In this section we consider three classes of variables that are encountered in the computer experiments setting: control variables, environmental variables, and model variables; and we discuss the experimental goals that are common for each of these.

1.3.1 Control and Environmental Variable Inputs

Control variables, or manufacturing variables as they are sometimes called, are variables that can be set by the researcher or product designer. When the output $y(\cdot)$ is a performance measure that depends on control variable values, we often attempt to "design" the product or system by setting the control variable so that $y(\cdot)$ is optimal. For example, in Chapter 4 the design of an acetabular cup for a hip replacement is considered. The ultimate goal is to achieve optimal fixture/seating of the cup in the acetabulum. The design of the cup is characterized by two control variables: equatorial diameter of the cup and eccentricity of the cup. Determining the best setting of these variables is important in promoting optimal seating of the cup in

the acetabulum, and ultimately, optimal performance of the hip implant system. We denote control variables as $\boldsymbol{x}_c$.

Environmental variables describe the specific environment under which the product is used and cannot be controlled by the researcher or product designer. The environment can change over time, from location to location, and from subject to subject. We typically think of environmental variables as being random with some known or unknown distribution over the population of interest. For example, in the acetabular cup example described above (and in Chapter 4), patient bone properties, patient loading of the hip and surgical reaming of the bone cup interface are several environmental variables that may also affect the proper fixation of the acetabular cup into the acetabulum. We denote environmental variables as $\boldsymbol{x}_e$.

When the inputs consist only of environmental variables, researchers may be interested in describing how the distribution of the inputs propogates to the distribution of the outputs, a topic referred to as uncertainty analysis. Suppose $\boldsymbol{X}_e \sim F(\boldsymbol{x}_e)$, we would like to describe the distribution of $y(\boldsymbol{X}_e)$ that is induced by $F(\boldsymbol{x}_e)$. Typically, certain aspects of this distribution are investigated. For example, Haylock and O'Hagan (1996) describe a Bayesian approach to uncertainty analysis in estimating the quantity

$$K = \int_{\mathcal{X}} y(\boldsymbol{x}_e) dF(\boldsymbol{x}_e).$$

If $y(\boldsymbol{x}_e)$ is known, an estimate of K can be obtained by sampling from F and performing a Monte Carlo analysis. For computer experiments this is infeasible since it requires a very large number of runs of the code. Taking the Bayesian viewpoint described in Section 1.1.2, we can obtain the posterior distribution of K given a set of data $\boldsymbol{Y}^n$. Of course, other quantities relating to the distribution of $y(\boldsymbol{X}_e)$ are

of interest as well. O'Hagan, Kennedy and Oakley (1998) discuss estimation of the *uncertainty distribution* of $y(\boldsymbol{X}_e)$

$$G(c) = P(y(\boldsymbol{X}_e) \leq c) = \int_{\mathcal{X}_e} I(y(\boldsymbol{x}_e) \leq c) dF(\boldsymbol{x}_e).$$

Often, computer experiment inputs consist of both control and environmental variables, as is the case of the design of an acetabular cup in Chapter 4. For this situation, interest is in the distribution of the random variable $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$, a distribution that depends on $\boldsymbol{x}_c$ and on the distribution of $\boldsymbol{X}_e$. Again, attention is typically given to certain aspects of this distribution. For example, in Williams, Santner and Notz (2000a) sequential designs are introduced for the purpose of optimizing (minimizing or maximizing) the quantity

$$\mu(\boldsymbol{x}_c) = \int_{\mathcal{X}_e} y(\boldsymbol{x}_c, \boldsymbol{x}_e) dF(\boldsymbol{x}_e).$$

Williams, Santner, and Notz (2000c) describes sequential designs for constrained optimization of $\mu_1(\boldsymbol{x}_c)$ subject to a constraint on $\mu_2(\boldsymbol{x}_c)$ when

$$\mu_i(\boldsymbol{x}_c) = \int_{\mathcal{X}_e} y_i(\boldsymbol{x}_c, \boldsymbol{x}_e) dF(\boldsymbol{x}_e)$$

correspond to two (possibly related) computer codes $y_1(\boldsymbol{x}_c, \boldsymbol{X}_e)$ and $y_2(\boldsymbol{x}_c, \boldsymbol{X}_e)$. In Chapter 3 we describe this setting and modify the algorithm of Williams et al. (2000c).

A second important aspect of the distribution of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ is the variance

$$\sigma^2(\boldsymbol{x}_c) = \mathrm{Var}[y(\boldsymbol{x}_c, \boldsymbol{X}_e)].$$

Finding $\boldsymbol{x}_c$ that has optimal $\mu(\boldsymbol{x}_c)$ subject to "small" $\sigma^2(\boldsymbol{x}_c)$ may also be of interest in many applications. In the quality control literature it is well known that simply optimizing $\mu(\boldsymbol{x}_c)$ may lead to unacceptably large variability of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ across the

distribution of $\boldsymbol{X}_e$. So, instead of optimizing $\mu(\boldsymbol{x}_c)$, it may be more desirable to minimize $\sigma^2(\boldsymbol{x}_c)$ subject to a constraint ("target") on $\mu(\boldsymbol{x}_c)$. This may be called a "robust" choice of $\boldsymbol{x}_c$. In Chapter 5 we discuss this approach, and consider several other robustness formulations relevant to the situation where there are both control and environmental variable inputs.

1.3.2 Model Variables

In addition to control and environmental variables, a third type of variable is sometimes found in computer experiments. We call these "model variables" or "model parameters". As previously noted, a computer experiment is based on a mathematical model that describes some physical process. The code implementing the mathematical model may involve parameters affecting the output of the code. These must be specified by the user of the code, and we call these model variables, or, as they are sometimes called, tuning parameters. For example, Kennedy and O'Hagan (2001) describe a computer code that models the movement of a drug through various compartments of the body. This code allows the consequences of a given dose regime to be explored. However, to use the code for a particular drug it is necessary to specify rates of movement, from one compartment of the body to another, for that drug. In this example, we would consider the dose regime as the control variables, and the rates of movement as the model variables.

When model variables are unknown, attempts can be made to adjust them so that the code "matches" the physical process. In other words, we would like to calibrate the code so that the observed physical data for input $\boldsymbol{x}$ fits, as closely as possible, the output of the code for input $\boldsymbol{x}$. Note that here we are assuming that it is possible

to collect physical data. Cox, Park and Singer (1996) discuss matching the computer code to the "real" experimental data by setting the model variables of the computer code so that the squared difference between the computer code output and the experimental data is small. Kennedy and O'Hagan (2001) present a Bayesian approach to calibration which attempts to account for not only the uncertainty associated with the model parameters, but also

1. Computer Model Inadequacy: the computer code may not give the true value of the real physical process.

2. Residual Variability: the real process won't always take the same value at the same input.

3. Parametric Variability: some of the inputs are uncontrolled (eg. environmental variables).

4. Observation error: the physical data may be observed with observation error.

5. Code uncertainty: the output of the code is unknown before we actually run the code.

Thus, attempts are made to control all possible components of variability.

CHAPTER 2

A COMPARISON OF THE SMALL SAMPLE PROPERTIES OF SEVERAL EMPIRICAL PREDICTORS

Gaussian stochastic process models are often used in the analysis of computer experimental data. Best viewed from the Bayesian perspective, the deterministic computer code, $y(\cdot)$, is treated as a realization of a stochastic process (or random function), $Y(\cdot)$ (see Chapter 1). The random function model used in most computer experiments is

$$Y(\boldsymbol{x}) = \boldsymbol{f}^{\top}(\boldsymbol{x})\boldsymbol{\beta} + Z(\boldsymbol{x}) \tag{2.1}$$

where $Z(\boldsymbol{x})$ is a stationary stochastic process assumed to have mean 0, variance σ^2 and correlation function $R(\cdot)$ that depends on unknown parameters denoted as $\boldsymbol{\gamma}$. Thus, the model parameters are $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{\gamma}$.

The prediction of the computer code at untried inputs is a basic requirement at the heart of all statistical procedures used to analyze data from computer experiments. As seen in Chapter 1, the best linear unbiased predictor (BLUP) of $Y(\boldsymbol{x}_0)$ can be derived assuming that the correlation parameters, $\boldsymbol{\gamma}$, are known. In practice, $\boldsymbol{\gamma}$ is typically unknown but can be estimated. The estimate of $\boldsymbol{\gamma}$ is then plugged into the prediction equations as if it were a *known* value, producing an empirical best linear unbiased

predictor (EBLUP). This makes "good" estimation of $\boldsymbol{\gamma}$ an important component of accurate prediction. In this chapter we will compare the prediction accuracy of three methods of estimating the correlation parameters: restricted maximum likelihood, maximum likelihood, and cross-validation. A simulation study is performed to make this comparison.

2.1 Modeling and Estimation

In Model (2.1), $Z(\boldsymbol{x})$ is a *stationary* stochastic process assumed to have mean 0, variance σ^2 and correlation function $R(\boldsymbol{x}_1 - \boldsymbol{x}_2)$, so that the correlation between $Z(\boldsymbol{x}_1)$ and $Z(\boldsymbol{x}_2)$ is $\text{Corr}[Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)] = R(\boldsymbol{x}_1 - \boldsymbol{x}_2)$. The regression term $\boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{\beta}$ allows for a global (nonstationary) trend with $\boldsymbol{f}^\top(\boldsymbol{x})$ a k-vector of known regression functions and $\boldsymbol{\beta} \in \mathbb{R}^k$ a vector of unknown regression parameters.

For computer experiments, restricted maximum likelihood (REML) and maximum likelihood (ML) are both estimation procedures that depend on the additional assumption that $Z(\cdot)$ is a *Gaussian* stochastic process. This assumption allows calculation of the likelihood as a function of $\boldsymbol{\beta}$, σ^2, and the vector of correlation parameters, $\boldsymbol{\gamma}$, and calculation of the restricted likelihood as a function of σ^2 and $\boldsymbol{\gamma}$. The ML and REML estimates of these parameters are obtained by maximizing the likelihood and restricted likelihood, respectively, where both likelihoods are given below.

Let $\boldsymbol{Y}^n = (Y(\boldsymbol{x}_1), ..., Y(\boldsymbol{x}_n))^\top$ be the vector of output obtained from running the computer code at the n input sites $(\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$. It can be shown (see Section 1.1.4) that given $\boldsymbol{\gamma}$, the ML estimate of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^\top \boldsymbol{R}^{-1} \boldsymbol{Y}^n, \tag{2.2}$$

where $\boldsymbol{F} = (\boldsymbol{f}(\boldsymbol{x}_1), ..., \boldsymbol{f}(\boldsymbol{x}_n))^\top$ is the $n \times k$ regression matrix, and $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{\gamma})$ is the $n \times n$ correlation matrix of $\boldsymbol{Y}^n$ so that the $(i,j)^{th}$ entry of $\boldsymbol{R}$ is computed as $R(\boldsymbol{x}_i - \boldsymbol{x}_j)$. Note that $\boldsymbol{R}$ depends on the correlation parameters, $\boldsymbol{\gamma}$, through the correlation function $R(\cdot)$. The ML estimate of σ^2 is

$$\hat{\sigma^2} = \frac{1}{n}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{2.3}$$

To obtain the ML estimate of $\boldsymbol{\gamma}$, we minimize $|\boldsymbol{R}|^{1/n}\hat{\sigma^2}$, which is a function of only the correlation parameters, $\boldsymbol{\gamma}$, and the data $\boldsymbol{Y}^n$. The REML estimate of $\boldsymbol{\gamma}$ is obtained by maximizing

$$-\frac{1}{2}[(n-k)\log(\tilde{\sigma}^2) \ + \ \log|\boldsymbol{R}| \ + \ \log|\boldsymbol{F}^\top \boldsymbol{R}^{-1}\boldsymbol{F}|], \tag{2.4}$$

where

$$\tilde{\sigma}^2 = \frac{1}{n-k}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{2.5}$$

Note that (2.4) is also a function of only $\boldsymbol{Y}^n$ and $\boldsymbol{\gamma}$. Section 1.1.4 in Chapter 1 contains additional details of these calculations.

Cross validation (XVAL) is a method of estimating $\boldsymbol{\gamma}$ that does not depend on any distributional assumptions. It is a prediction based criterion that depends only on the BLUP of $Y(\boldsymbol{x}_0)$ for any $\boldsymbol{x}_0 \in \mathcal{X}$. As shown in Section 1.1.1 of Chapter 1, the formula for the BLUP based on the data, $\boldsymbol{Y}^n$, is

$$\hat{Y}(\boldsymbol{x}_0) = \boldsymbol{f}^\top(\boldsymbol{x}_0)\hat{\boldsymbol{\beta}} + \boldsymbol{r}^\top(\boldsymbol{x}_0)\boldsymbol{R}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}}), \tag{2.6}$$

where $\hat{\boldsymbol{\beta}}$ is defined by (2.2), and $\boldsymbol{r}(\boldsymbol{x}_0)$ is the $n \times 1$ vector of correlations of $\boldsymbol{Y}^n$ with $Y(\boldsymbol{x}_0)$ so that $\boldsymbol{r}(\boldsymbol{x}_0) = (R(\boldsymbol{x}_1 - \boldsymbol{x}_0), ..., R(\boldsymbol{x}_n - \boldsymbol{x}_0))^\top$. The XVAL estimation procedure removes the i^{th} observation and uses the BLUP based on the remaining

$n-1$ observations to predict the value for the i^{th} observation. The "best" correlation parameters, $\boldsymbol{\gamma}$, are chosen as those that make the predicted value closest to the true value, averaging over all n observations. Formally, we choose $\boldsymbol{\gamma}$ to minimize

$$\sum_{i=1}^{n}(\hat{Y}_{-i}(\boldsymbol{x}_i) - y(\boldsymbol{x}_i))^2, \tag{2.7}$$

where $\hat{Y}_{-i}(\boldsymbol{x}_i)$ is the BLUP of $Y(\boldsymbol{x}_i)$ based on the the $n-1$ observations obtained by removing $y(\boldsymbol{x}_i)$ from $(y(\boldsymbol{x}_1), ..., y(\boldsymbol{x}_n))$. Note that this formula only depends on the data $\boldsymbol{Y}^n$ and the value of the correlation parameters, $\boldsymbol{\gamma}$.

We will compare the REML, ML and XVAL estimation procedures by generating random surfaces, and fitting Model (2.1) to a small training sample for each surface and for each estimation method. Since prediction is often the primary interest in computer experiments, the criterion of primary interest is the mean squared error of prediction of the true surface over a grid of sites in the input space.

2.2 Simulation Study

2.2.1 Generating Random Surfaces

To compare the three methods of estimation, a method of generating random problems is necessary. In the computer experiments setting, a problem consists of a response, $y(\cdot)$, defined on $\mathcal{X} \subset \mathbb{R}^p$. Sampling $y(\cdot)$ at a set of n inputs (the training sites) and fitting Model (2.1) to the data, we obtain the predictor. We generate random responses using the *krigifier* of Trosset and Padula (2000) and Trosset (1999). The krigifier generates a response by simulating from a stochastic process observed at m sites and interpolating the result using (2.6) as the true $y(\cdot)$. The steps are as follows:

1. *Specify* an underlying trend, $\boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{\beta}$, a correlation function, $R(\cdot)$, and a process variance, σ^2, corresponding to the stationary Gaussian process, $Z(\cdot)$, in Model (2.1).

2. Choose a finite number of points, $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$ at which to observe $Z(\cdot)$.

3. Generate $z_1, ..., z_m$, the values of $Z(\cdot)$ at $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$.

4. Interpolate $\boldsymbol{z}_m = (z_1, ..., z_m)^\top$ using the BLUP in Equation (2.6) (with the *known* correlation function) to obtain the noise term, $z(\boldsymbol{x}_0)$ for any $\boldsymbol{x}_0 \in \mathbb{R}^p$

5. Add the trend, $\boldsymbol{f}^\top(\boldsymbol{x}_0)\boldsymbol{\beta}$, and noise term, $z(\boldsymbol{x}_0)$, to produce the objective function, $y(\boldsymbol{x}_0)$.

Using the krigifier, we are able to generate a large number of random problems which can be used to evaluate the three estimation procedures.

For the simulations presented here we let $p = 2$, $\mathcal{X} = (0,1) \times (0,1)$, $\boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{\beta} = \beta_0 = 100$, $\sigma^2 = 1$, and the correlation function corresponding to the stationary Gaussian process be the *Matérn* correlation function or the *power exponential* correlation function. The Matérn correlation is

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{p} \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{2\sqrt{\nu}|x_{1,i} - x_{2,i}|}{\theta_i} \right)^{\nu} \mathrm{K}_\nu \left(\frac{2\sqrt{\nu}|x_{1,i} - x_{2,i}|}{\theta_i} \right), \qquad (2.8)$$

where $\theta_i > 0$, $\nu > 0$, and $\mathrm{K}_\nu(\cdot)$ is the modified Bessel function of order ν (see Stein (1999) Section 2.7). The parameter ν controls the smoothness of realizations of the Gaussian process with realizations being *almost surely* m times differentiable if $\nu > m$ (see Section 1.1.3). The *power exponential* correlation function is

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{p} e^{-\theta_i (x_{1i} - x_{2i})^{\alpha_i}}, \qquad (2.9)$$

where $\theta_i > 0$, $0 < \alpha_i \leq 2$. Note that with $\alpha_1 = \alpha_2 = 2$ and $1/\theta_i^2$ in place of θ_i in (2.9) the power exponential correlation function is the limiting case of (2.8) as $\nu \to \infty$. Thus, for large ν (for numerical purposes $\nu \geq 50$ is assumed large enough to be the limiting case of the Matérn) the generated surfaces could have likewise been simulated using the power exponential correlation function. Using the steps in the previous paragraph, we set $\theta_1 = 1/\sqrt{8}$, $\theta_2 = 1/\sqrt{15}$ and generate 50 random "true" surfaces for each of $\nu = 5$, $\nu = 10$, and $\nu = 50$ ($\nu = 50$ is roughly equivalent to the power exponential function with $\alpha_1 = \alpha_2 = 2$, $\theta_1 = 8$, and $\theta_2 = 15$) by defining an 11×11 equispaced grid in $\mathcal{X}$ (step 2 above), sampling from a multivariate normal with mean 100 and covariance matrix computed from (2.8) for each pair of points in the grid (step 3 above), and computing the BLUP for any $\boldsymbol{x} \in \mathcal{X}$. Figure 2.1 displays two draws from this class of surfaces. Some corresponding REML EBLUPs are displayed in Figure 2.2.

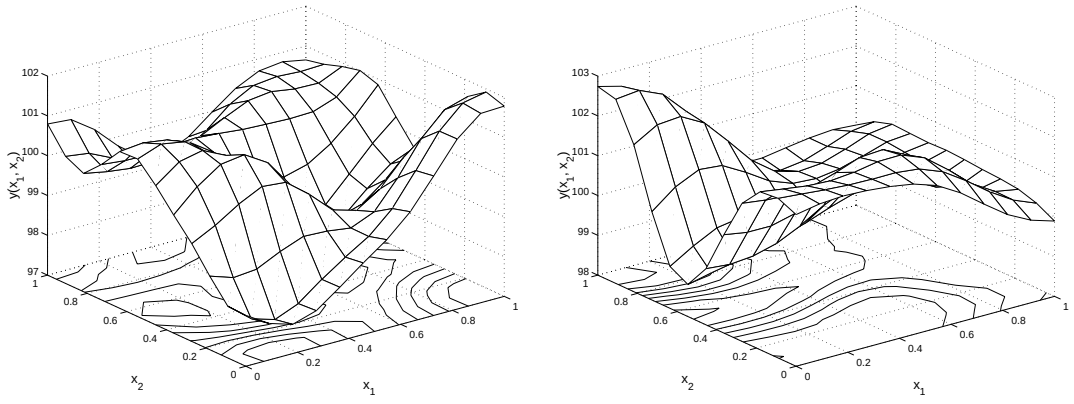


Figure 2.1: Examples of two surfaces generated using the krigifier.

2.2.2 Estimation and Prediction Details

For each of the 150 surfaces we compute the true response, $y(\cdot)$, at a set of 20 training sites, $\boldsymbol{x}_i \in (0,1) \times (0,1)$, $i = 1, ..., 20$, chosen as a Latin hypercube sample that maximizes the minimum inter-point distance (see Conover et al. (1979)). This type of design tends to spread observations throughout the input space. From the computed values, we form $\boldsymbol{y}^n = (y(\boldsymbol{x}_1), ..., y(\boldsymbol{x}_{20}))$, and fit Model (2.1) (i.e. estimate the parameters of the model) using each of REML, ML and XVAL, and the two correlation functions, the power exponential and Matérn. In addition, a cubic polynomial model was fit to $\boldsymbol{y}^n$ using ordinary least squares. Thus, for each of the 150 true surfaces a total of seven predicted surfaces were produced.

Each predictor was evaluated over an equispaced 625-point grid on $\mathcal{X}$, and the mean-squared error of prediction was calculated by comparing each predicted value to the true value, squaring their difference and taking the mean of the 625 squared differences. As an example, the REML predicted surfaces using the power exponential correlation function corresponding to the true surfaces shown in Figure 2.1 are displayed in Figure 2.2.

For those predicted surfaces corresponding to Model (2.1), the optimization of the likelihood estimates $\alpha_1 = \alpha_2 = 2$ and $\nu = 50$ if the log likelihood or restricted log likelihood for these values is within 1 (corresponding to a change of 2 in the -2*log-likelihood) of the respective maximum likelihood achieved by allowing these parameters to vary over their full ranges. For the cross validation estimation method, we set $\alpha_1 = \alpha_2 = 2$ and $\nu = 50$ if the minimum criterion achieved with this setting is within 10% of the minimum achievable by allowing these parameters to vary over their full ranges. This places a penalty on choosing a less-parsimonious model (i.e.

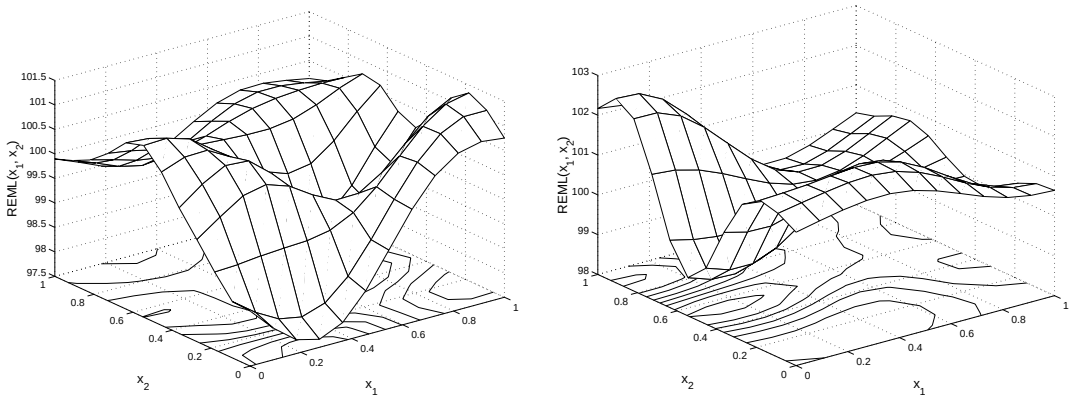


Figure 2.2: Examples of predicted surface corresponding to true surfaces in Figure 2.1. Predicted surfaces are based on REML estimation of the power exponential correlation function parameters. Parameter estimates for the left panel are $\theta_1 = 7.91, \theta_2 = 20.37, \alpha_1 = \alpha_2 = 2$, and for the right panel are $\theta_1 = 9.26, \theta_2 = 5.92, \alpha_1 = \alpha_2 = 2$.

one where $\nu < 50$ for the Matérn correlation function and $\alpha_i \neq 2$ for some i for the power exponential correlation function).

2.3 Results

The θ_1 and θ_2 correlation parameter estimates corresponding to predicted surfaces arising from Model (2.1) with the Matérn correlation function are displayed in Figure 2.3. Recall that the "true" surfaces were generated using the Matérn correlation function with $\theta_1 = 1/\sqrt{8} = .3535$ (vertical line in left panel), $\theta_2 = 1/\sqrt{15} = .2582$ (vertical line in right panel), and values of 5, 10, and 50 for ν. Note that the ML and REML estimation methods appear to be generally on target, while estimation via XVAL appears to overestimate both θ_1 and θ_2. Estimates above 8 have been truncated in the boxplots.

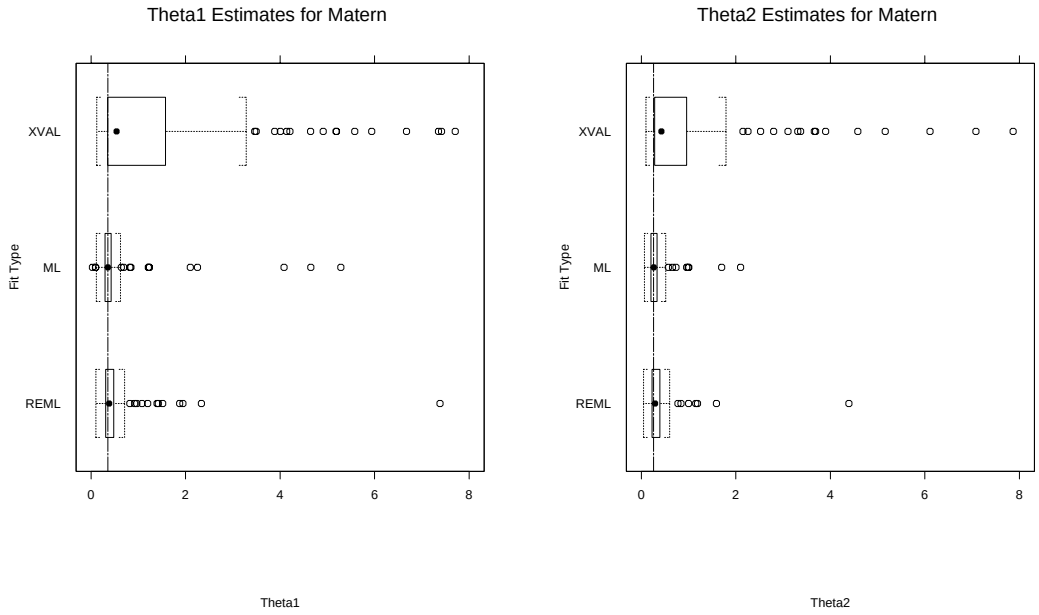


Figure 2.3: Boxplots of estimates of θ_1 (left panel) and θ_2 (right panel) for the Matérn correlation function. The true values $\theta_1 = .3535$ and $\theta_2 = .2582$ are indicated by the vertical lines in the figures.

Figure 2.4 displays boxplots of the θ_1 and θ_2 estimates of the power exponential correlation function parameters for the 50 true surfaces generated using the Matérn correlation function with $\theta_1 = .3535$, $\theta_2 = .2582$, and $\nu = 50$. Recall that this is equivalent to the power exponential correlation function with $\theta_1 = 8$, $\theta_2 = 15$, and $\alpha_1 = \alpha_2 = 2$. It appears that θ_1 and θ_2 are slightly underestimated by both ML and REML, and more severely underestimated by the XVAL estimation procedure. Estimates above 50 have been truncated in the boxplots.

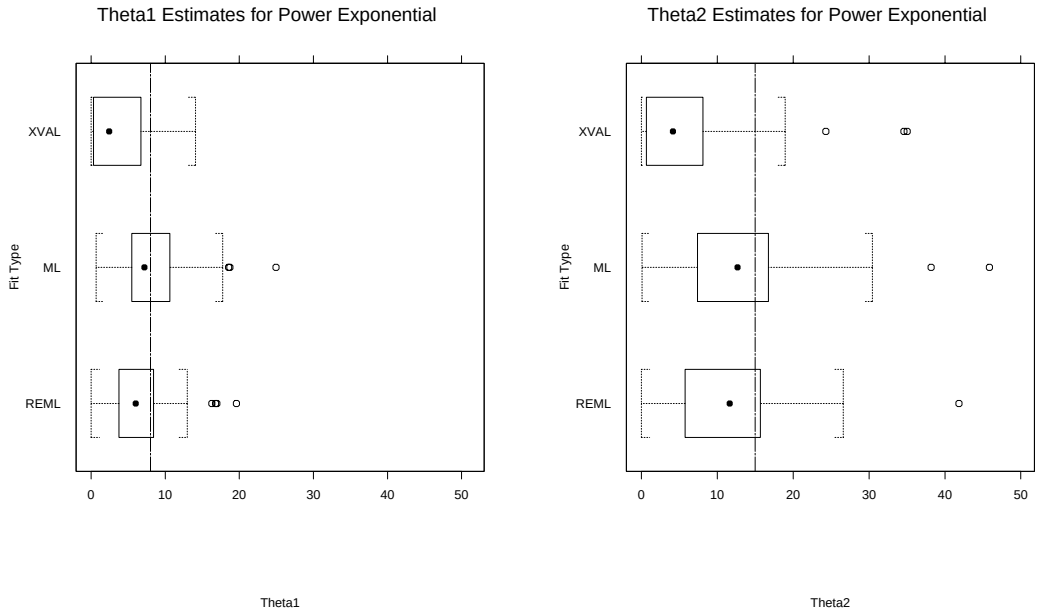


Figure 2.4: Boxplots of estimates of θ_1 (left panel) and θ_2 (right panel) for the power exponential correlation function. The true values $\theta_1 = 8$ and $\theta_2 = 15$ are indicated by the vertical lines in the figures.

Table 2.1 contains summary statistics for the MSE of prediction for the three factors: fit type, model type, and true correlation function. The factor *fit type* corresponds to the three estimation techniques (ML, REML, and XVAL). The *model type* factor has three levels corresponding to the three types of models used: the regression based cubic polynomial model, and Model 2.1 with the power exponential correlation function or with the Matérn correlation function. The *true correlation function* has three levels and corresponds to the three "true" values of ν that were used in generating the "true" responses. Figure 2.5 displays boxplots of the MSE of prediction for each of the predicted surfaces. In general, the REML and ML estimation methods

perform better then XVAL for all combinations of the above factors, and the XVAL based predictor can lead to very poor prediction for some cases (note the outliers). The cubic polynomial predictors, in general, do not perform as well as any of the predictors corresponding to Model (2.1).

True	Matérn Correlation Function Fits						
ν	Fit Type	Min	Q1	Median	Mean	Q3	Max
5	REML	0.0472	0.0947	0.1292	0.1755	0.2181	0.5088
	ML	0.0483	0.0930	0.1358	0.1820	0.2239	0.6351
	XVAL	0.0559	0.1030	0.1758	0.3300	0.4161	1.8124
10	REML	0.0362	0.0833	0.1149	0.1251	0.1473	0.4521
	ML	0.0371	0.0819	0.1119	0.1174	0.1459	0.3711
	XVAL	0.0433	0.1109	0.1876	0.5482	0.7834	4.4537
50	REML	0.0262	0.0617	0.0809	0.0931	0.1079	0.2570
	ML	0.0220	0.0613	0.0815	0.1047	0.1093	0.6978
	XVAL	0.0219	0.0790	0.1218	0.2769	0.2994	2.3363
True	Power Exponential Correlation Function Fits						
ν	Fit Type	Min	Q1	Median	Mean	Q3	Max
5	REML	0.0472	0.1078	0.1537	0.1876	0.2597	0.4880
	ML	0.0483	0.1073	0.1439	0.1747	0.2108	0.4227
	XVAL	0.0559	0.1138	0.1980	0.3216	0.3494	1.8123
10	REML	0.0370	0.0901	0.1155	0.1377	0.1589	0.4520
	ML	0.0373	0.0874	0.1104	0.1219	0.1496	0.3712
	XVAL	0.0433	0.1110	0.1702	0.3582	0.3808	2.4338
50	REML	0.0262	0.0623	0.0864	0.0998	0.1148	0.2542
	ML	0.0221	0.0619	0.0824	0.1050	0.1093	0.6301
	XVAL	0.0219	0.0901	0.1597	0.2862	0.3171	2.3363
True	Cubic Polynomial Regression Fits						
ν		Min	Q1	Median	Mean	Q3	Max
5		0.1699	0.3748	0.4258	0.5585	0.7825	1.1507
10		0.0770	0.2906	0.4247	0.4905	0.6166	1.3390
50		0.1304	0.2595	0.3745	0.4951	0.6668	1.6230

Table 2.1: Summary statistics for the MSE of prediction.

Comparing the ML and REML estimation procedures, there is no clear winner. REML is generally thought to be superior to ML (see Stein (1999)), however, in these examples the choice is not clear. Figure 2.6 plots the prediction MSE for the

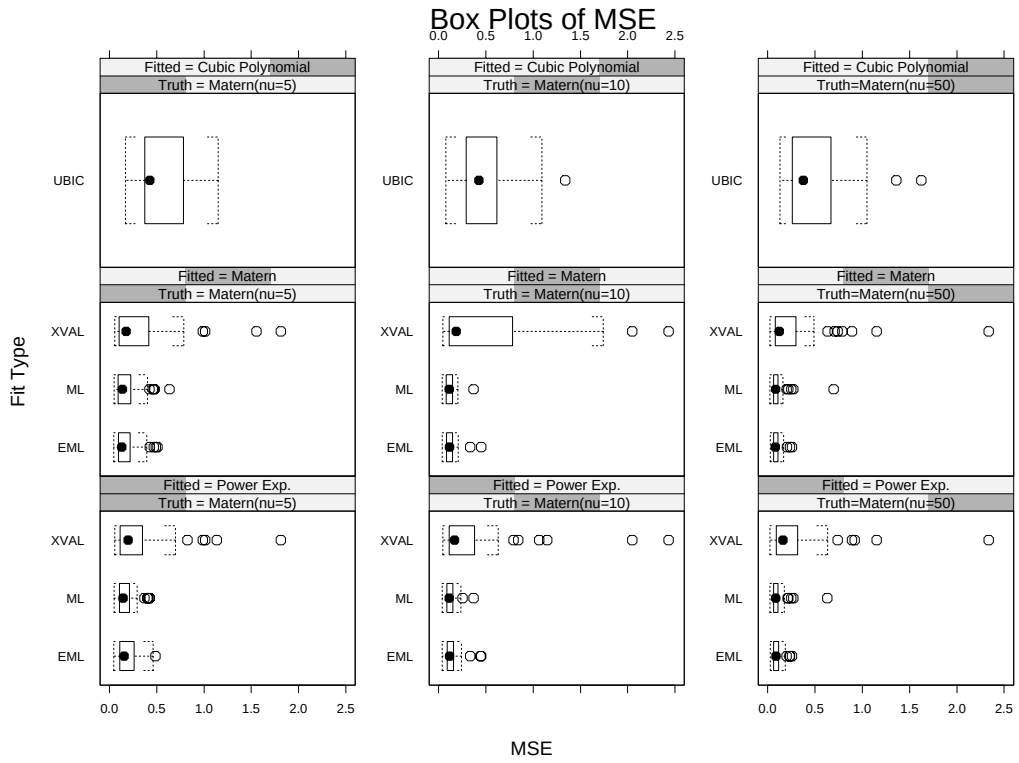


Figure 2.5: Boxplots of MSE of prediction on 625-point equispaced grid for each combination of factors.

REML procedure *paired* with that for the corresponding ML procedure. For the power exponential correlation function fits, it appears that more of the points fall below the diagonal lines, indicating that the MSE for REML is more often larger than the MSE for ML. Table 2.2 displays p-values for simple signed rank tests comparing the MSE for REML and ML for each subplot found in Figure 2.6. In general, it appears that the ML based estimators have smaller MSE (the Z-statistic is always negative), with significantly smaller MSE for surfaces fit with the power exponential correlation function and generated using the Matérn correlation function with $\nu = 5$.

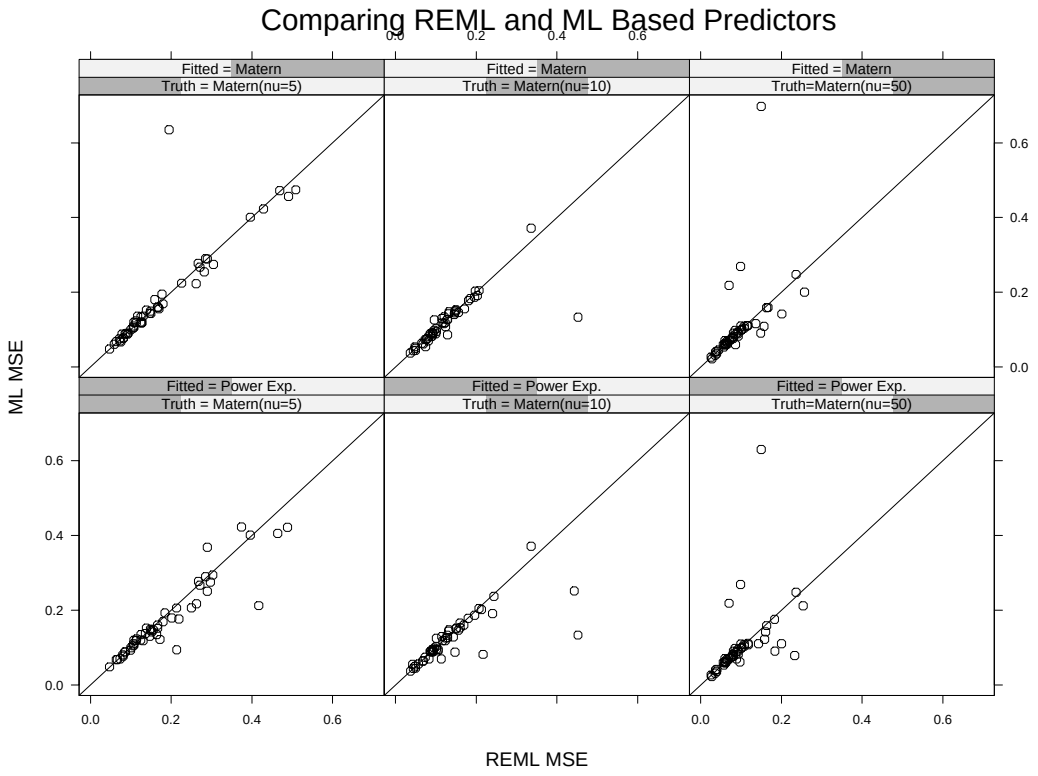


Figure 2.6: Scatter Plot of MSE for REML and ML EBLUP estimation procedures for each combination of fit type and true correlation function.

Figure 2.7 plots the MSE of prediction for the Matérn based predictor with the corresponding power exponential based predictor. The two correlation functions perform similarly in many of the cases with the Matérn based predictor appearing to have lower MSE in some cases. Due to the many ties in the MSE values, no formal tests were performed for these comparisons.

Fitted $R(\cdot)$	True $R(\cdot)$	Z Stat	P-value
Matérn	Matérn($\nu = 5$)	-0.4827	0.629
Matérn	Matérn($\nu = 10$)	-1.7955	0.073
Matérn	Matérn($\nu = 50$)	-1.4866	0.137
Power Exp.	Matérn($\nu = 5$)	-2.4133	0.016
Power Exp.	Matérn($\nu = 10$)	-1.9596	0.050
Power Exp.	Matérn($\nu = 50$)	-1.8148	0.070

Table 2.2: Table of p-values for two-sided signed rank test comparing MSE's of ML and REML based predictors.

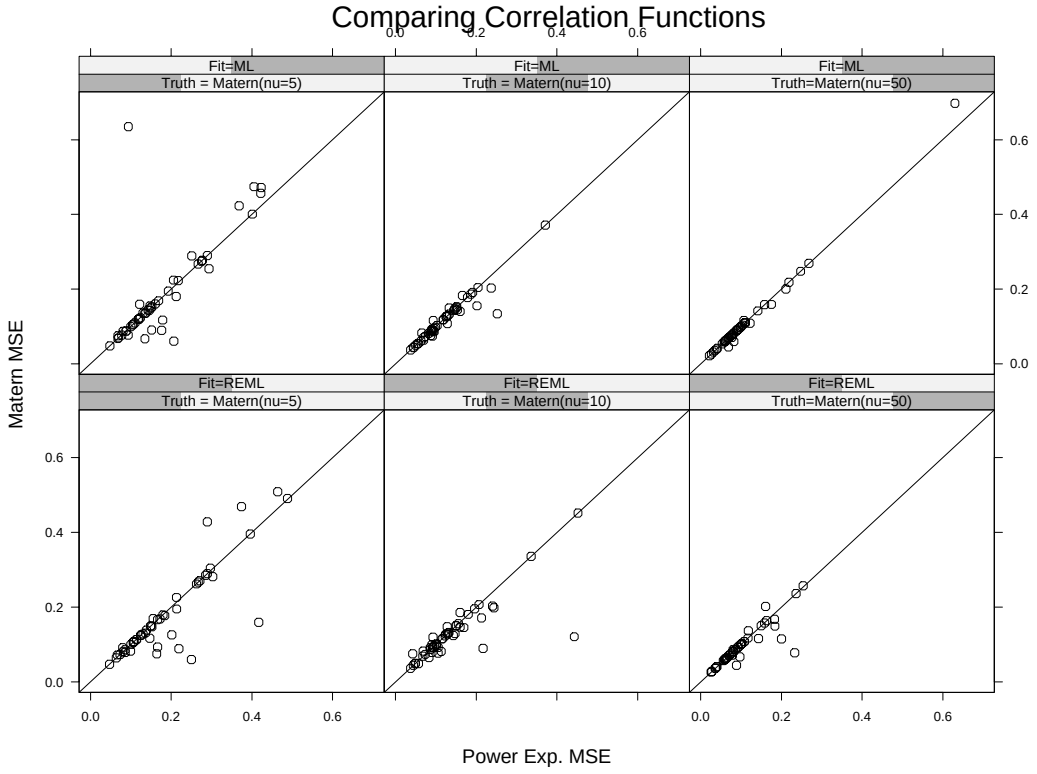


Figure 2.7: Scatter plot of MSE for Matérn and power exponential based predictors for each combination of estimation method and true correlation function.

2.4 Discussion

In the examples above we fit the predictive models based on a 20-point Latin hypercube sampling (LHS) design that maximizes the minimum inter-point distance. For this design and the "true" surfaces considered here, comparing classical response surface models, such as cubic polynomial regression models, to the stochastic process models that are typically used in computer experiments, we find that the predictors arising from stochastic process models perform significantly better. Of course, other choices of designs and other choices of design size may be worthy of further investigation for these predictors. For example, the cascading LHS of Handcock (1991), with their potential for local and global estimation (see Chapter 1), or an LHS design with different distance properties, may lead to better prediction of the "true" surface.

In addition, further investigation for more and different classes of random surfaces can lead to a clearer picture of the merits of each predictor. For example, the *krigifier*, which produced the "true" random surfaces for these comparisons, required specification of the number of inputs, p, the "true" correlation function, $R(\cdot)$, and the "true" trend term, $\boldsymbol{f}(\cdot)$. In our simulations, we chose $p = 2$, made a single choice for the form of the "true" correlation function and three choices for the values of the parameters of that correlation function corresponding to surfaces with different smoothness properties. Increasing p, choosing a different form for $R(\cdot)$, such as the *cubic* correlation function (see Currin et al. (1991)), or choosing different values of the parameters in (2.8) can lead to a class of random surfaces that have different smoothness (local) properties. Also, choosing different trend terms $\boldsymbol{f}(\cdot)$ will yield random surfaces having different global properties. For each class of random surfaces (or each

means of generating random surfaces), predictors can be evaluated via replication as seen in the simulation above.

For stochastic process models corresponding to (2.1), the results above do not suggest an overwhelming advantage, and perhaps no advantage at all, of using the Matérn correlation function versus the power exponential correlation function for these types of models. Theoretically, the Matérn correlation function is more attractive since it includes a parameter (ν) that controls the smoothness of realizations of the stochastic process $Z(\cdot)$ in Model (2.1). However, practically, the power exponential correlation function appears to perform almost as well (if not as well) in terms of prediction for the problems presented above. For these problems, the smallest value of ν was $\nu = 5$, which still leads to a reasonably smooth surface. Perhaps the lesson is that unless you know the true surface is not very smooth, the power exponential correlation function is adequate. In addition, the power exponential has the advantage that it is faster to compute than the Matérn correlation function.

Similarly, estimation via restricted maximum likelihood does not in general appear to have an advantage over estimation via maximum likelihood. However, REML estimation does manage to avoid, for the most part, situations where the predictor fails drastically (see Figure 2.6). It may prove informative to investigate more closely those surfaces where the ML based predictors performed poorly. These typically corresponded to instances where the likelihood was "flat" in the correlation parameters and optimization attempted to push the parameters to large values. Perhaps, as in log linear modeling, indicators can be developed that suggest numerical problems with the maximization of the likelihood. More research in this area is necessary to answer this question.

CHAPTER 3

A MODIFICATION OF THE WILLIAMS, SANTNER, AND NOTZ ALGORITHM FOR CONSTRAINED OPTIMIZATION

This chapter discusses an improved version of the algorithm proposed by Williams, Santner, and Notz (2000c) for optimizing the mean of one computer code (the *objective* function) subject to constraints on the mean of a second computer code (the *constraint* function). This chapter assumes that there are two types of inputs, $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{x}_e)$, where $\boldsymbol{x}_c$ is a set of control variables and $\boldsymbol{x}_e$ is a set of environmental variables. Control variables can be set by the product designer, and environmental variables have values that follow some probability distribution representing variation in these variables for a population of interest. For example, in the hip prosthesis problem of Chang et al. (1999a), the control variables specify the geometry of the implant and the environmental variables account for variability in patient bone properties and level of activity. The means of the objective and constraint functions are computed with respect to the distribution of the environmental variables, so that the means depend only on the value of the control variables. We then determine the "optimal" settings of the control variables for the population of interest.

There are many algorithms in the mathematical literature that study constrained optimization; however, such algorithms are prohibitive in a computer experiment

because they require many function evaluations. The statistical approach to optimization of a computer code produces a fast predictor of the code, and uses a traditional optimization algorithm to optimize the predictor. For example, Bernardo et al. (1992) implemented an algorithm for response minimization that sequentially focuses on the region of the input variable space where the optimum appears to be located. Jones, Schonlau and Welch (1998) and Williams, Santner and Notz (2000a) examine criterion-based sequential strategies for minimization of a single response.

Here, we investigate several modifications to the constrained optimization algorithm of Williams, Santner, and Notz (2000c). They assume that both of the responses are observed at the *same* sites not only in the initial design, but throughout the algorithm. This means that once the algorithm decides on the next site at which to observe the response, both computer codes are run for that input. We refine the improvement criterion to: (1) better accomodate situations where the location of the global optimum is an "easily" describable part of the feasible region and thus it is inefficient to take as many additional observations on the constraint function as on the objective function, and (2) improve the current guess at the constrained optimum. The algorithm proposed here allows the responses to be observed on *different numbers* of sites and at *different sites*. At each step it chooses which of the two responses to observe and the next site at which to observe that response.

We also examine the question of sample size for the initial design. In computer experiments, approximately 10 observations per input dimension has been proposed as a rule of thumb (see Jones et al. (1998) p. 473). We provide guidelines for the number of initial design sites for the algorithm in this paper, which may be useful in other sequential settings.

In Section 3.1 we outline the setup for the Bayesian modeling of the responses. Section 3.2 presents the expected improvement algorithm based on these models. Section 3.3 contains some random examples which use a class of closed-form test functions to allow the optimum found by the algorithm to be checked with the true constrained optimum. In Section 3.4, we discuss the implications of these results, as well as areas for future research.

3.1 Modeling

For $i = 1, 2$ we model the true response $y_i(\cdot)$ by the random function

$$Y_i(\boldsymbol{x}) = \boldsymbol{f}_i^\top(\boldsymbol{x})\boldsymbol{\beta}_i + Z_i(\boldsymbol{x}) \tag{3.1}$$

where $Z_i(\cdot)$ is a covariance stationary Gaussian stochastic process having mean zero, correlation function $R_i(\cdot)$, and unknown variance $\tau_i^2 > 0$. The linear model $\boldsymbol{f}_i^\top(\cdot)\boldsymbol{\beta}_i$ represents the global (nonstationary) mean of the Y_i process with $\boldsymbol{f}_i(\cdot)$ a k_i-vector of known regression functions and $\boldsymbol{\beta}_i \in \mathbb{R}^{k_i}$ a vector of unknown regression parameters. The model is completed by specifying a positive definite, joint covariance structure for $Y_1(\cdot)$ and $Y_2(\cdot)$. In the following, we take $\mathrm{Cov}(Y_1(\boldsymbol{x}_1), Y_2(\boldsymbol{x}_2)) = \tau_1\tau_2 R_{12}(\boldsymbol{x}_1 - \boldsymbol{x}_2)$, where $R_{12}(\cdot)$ is called the cross-correlation function.

Specifying a valid correlation structure for $(Y_1(\cdot), Y_2(\cdot))$ via (R_1, R_2, R_{12}) is non trivial and most frequently these three functions are assumed to belong to a given parametric family of known correlation functions (eg. see Ver Hoef and Barry (1998) for some examples). We assume that the three correlation functions $R_1(\cdot), R_2(\cdot)$ and $R_{12}(\cdot)$ depend on the parameter vector $\boldsymbol{\xi}$ which is allowed to take any value for which the covariance structure of the joint process $\boldsymbol{Y}(\boldsymbol{x}) = (Y_1(\boldsymbol{x}), Y_2(\boldsymbol{x})), \boldsymbol{x} \in \mathcal{X}$ is positive

definite. In Section 3.3, we consider a specific spatial autoregressive model for the processes and a parametric family of correlation functions: the power exponential.

The approach is Bayesian, it assumes a non-informative prior for the parameters $(\boldsymbol{\beta}, \tau_1^2)$

$$[\boldsymbol{\beta}, \tau_1^2] \propto \frac{1}{\tau_1^2}$$

and the calculations take the parameters $\boldsymbol{\gamma} = (\tau_2^2, \boldsymbol{\xi})$ that appear in the correlation of $(Y_1(\cdot), Y_2(\cdot))$ to be known. In the algorithm below we follow an empirical Bayes strategy, whereby we set $\boldsymbol{\gamma}$ equal to its posterior mode and substitute these values wherever necessary.

3.2 The Minimization Algorithm

Define the control and environmental variable portions of $\boldsymbol{x}$ as $\boldsymbol{x}_c \in \mathcal{X}_c$ and $\boldsymbol{x}_e \in \mathcal{X}_e$ (so $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{x}_e)$), and assume that the joint distribution of the environmental variables is discrete on $\{\boldsymbol{x}_{e,j}\}_{j=1}^{n_e}$ with weights $\{w_j\}_{j=1}^{n_e}$. Denote the mean response calculated over the distribution of the environmental variables by

$$\mu_i(\boldsymbol{x}_c) = E_{\boldsymbol{X}_e}[y_i(\boldsymbol{x}_c, \boldsymbol{X}_e)] = \sum_{j=1}^{n_e} w_j y_i(\boldsymbol{x}_c, \boldsymbol{x}_{e,j}) \quad \text{for } i = 1, 2.$$

The goal of this experiment is to identify the settings of the control variable, $\boldsymbol{x}_c$, that minimize the mean of the first response, $\mu_1(\boldsymbol{x}_c)$, subject to a constraint on the mean of the second response, $\mu_2(\boldsymbol{x}_c)$. In symbols, the objective is to find $\boldsymbol{x}_c^*$ satisfying

$$\boldsymbol{x}_c^* = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\operatorname{argmin}}\ \mu_1(\boldsymbol{x}_c) \quad \text{subject to} \quad \mu_2(\boldsymbol{x}_c^*) \leq U.$$

We let $M_i(\boldsymbol{x}_c) = \sum_{j=1}^{n_e} w_j Y_i(\boldsymbol{x}_c, \boldsymbol{x}_{e,j})$ be the random variable associated with $\mu_i(\cdot)$. In the following sections, we propose a sequential design algorithm for finding $\boldsymbol{x}_c^*$.

3.2.1 Overview

The starting point for the minimization algorithm involves choosing initial designs for both the $Y_1(\cdot)$ and $Y_2(\cdot)$ processes. Denote the initial design for $Y_1(\cdot)$ as $\boldsymbol{S}_{n_1} = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_{n_1}\}$, and the initial design for $Y_2(\cdot)$ as $\boldsymbol{T}_{n_2} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_{n_2}\}$. This notation allows for different numbers of runs to be taken on each process and at different sites in the input space. Let $\boldsymbol{Y}_1^{n_1}$ and $\boldsymbol{Y}_2^{n_2}$ represent the vector of responses associated with the initial designs $\boldsymbol{S}_{n_1}$ and $\boldsymbol{T}_{n_2}$, respectively. Denote the control variable portion of $\boldsymbol{S}_{n_1}$ by $\boldsymbol{S}_{n_1}^C = \{\boldsymbol{s}_{c,1}, ..., \boldsymbol{s}_{c,n_1}\}$ and set $\boldsymbol{M}_i^{n_1} = [M_i(\boldsymbol{s}_{c,1}), ..., M_i(\boldsymbol{s}_{c,n_1})]^\top$, the vector of values for the objective and constraint functions associated with $\boldsymbol{S}_{n_1}^C$. Define $M_1^{min,c}$ $= \min\{M_1(\boldsymbol{s}_{c,i}) \ : \ \boldsymbol{s}_{c,i}$ such that $M_2(\boldsymbol{s}_{c,i}) - t_{n_1+n_2-k,.95}\sqrt{Var(M_2(\boldsymbol{s}_{c,i}))} \leq U\}$, where $k = k_1 + k_2$ and $t_{n_1+n_2-k,.95}$ is the upper 95th percentile of a t-distribution with (n_1+n_2-k) degrees of freedom. In words, $M_1^{min,c}$ is the minimum of $M_1(\cdot)$ at control sites previously computed on the $Y_1(\cdot)$ process that appear to be in or "close" to the feasible region. Define the improvement at a potential new control variable site $\boldsymbol{x}_c$ as:

$$I(\boldsymbol{x}_c) = \max(0, M_1^{min,c} - M_1(\boldsymbol{x}_c))\ \chi(M_2(\boldsymbol{x}_c) \leq U) \tag{3.2}$$

where $\chi(A)$ is 1 if event A occurs and 0 otherwise. Williams et al. (2000c) use M_1^{min} $= \min\{M(\boldsymbol{s}_{c,i})\}$ rather than $M_1^{min,c}$ in (3.2) and thus fails to restrict the minimum to those values of $\boldsymbol{s}_{c,i}$ that appear to be in the feasible region. This can cause the following problem. Consider the hypothetical $\mu_1(\cdot)$ and $\mu_2(\cdot)$ pictured in Figure 3.1. Suppose that the goal is to minimize $\mu_1(x_c)$ subject to the constraint $\mu_2(x_c) < -8$, which restricts x_c to the interval $(0.1101, 0.4762)$. Also, suppose that the current input data are the points denoted by $*$'s, and that M_1^{min} and $M_1(x_c)$ are as shown

in the figure. Then, we have for $x_c \in (0.1101, 0.4762)$, $\max(0, M_1^{min} - M_1(x_c)) = 0$ since $M_1^{min} << M_1(x_c)$, and for $x_c \notin (0.1101, 0.4762)$, $\chi(M_2(x_c) \leq -8) = 0$. Thus, $I(x_c) \approx 0$ for all $x_c \in (0, 1)$, and there is no improvement for any x_c. This problem is avoided by using $M_1^{min,c}$ instead of M_1^{min} in (3.2).

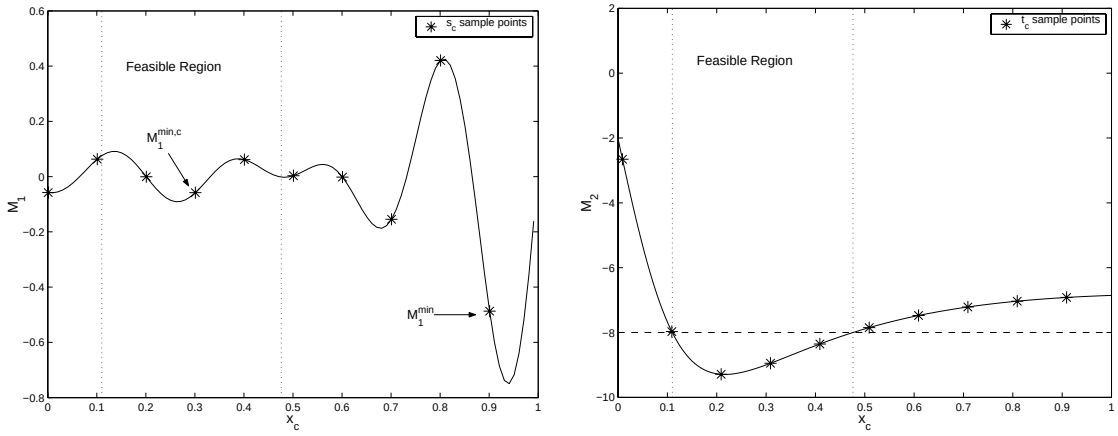


Figure 3.1: Hypothetical $M_1(x_c)$ (left panel) and $M_2(x_c)$ (right panel).

The proposed algorithm is:

1. Choose the initial set of design points on which to observe the $Y_1(\cdot)$ process, $\boldsymbol{S}_{n_1} = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_{n_1}\}$ and the $Y_2(\cdot)$ process, $\boldsymbol{T}_{n_2} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_{n_2}\}$.

2. Estimate the covariance parameter vector by $\hat{\boldsymbol{\gamma}}$, the mode of the posterior density of $\boldsymbol{\gamma}$ given $(\boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2})$.

3. Choose the next control variable site, $\boldsymbol{x}_c^*$, to maximize the posterior expected improvement given the current data and $\hat{\boldsymbol{\gamma}}$, i.e.,

$$\boldsymbol{x}_c^* = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\operatorname{argmin}} E\{I(\boldsymbol{x}_c) \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \hat{\boldsymbol{\gamma}}\} \tag{3.3}$$

where $I(\boldsymbol{x}_c)$ is given by (3.2).

4. Choose the process and the environmental variable site corresponding to the control site $\boldsymbol{x}_c^*$ as follows. Let $\boldsymbol{Y}_e^j = [Y_j(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{Y}_1^{n_1\top}, \boldsymbol{Y}_2^{n_2\top}]^\top$, and define the following for $j = 1, 2$:

$$\begin{aligned} MSE_j(\boldsymbol{x}_e) \;=\; & E\{(\hat{M_1^j}(\boldsymbol{x}_c^*) - M_1(\boldsymbol{x}_c^*))^2 \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \hat{\boldsymbol{\gamma}}\} \qquad + \\ & P\left[M_2(\boldsymbol{x}_c^*) > U \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \hat{\boldsymbol{\gamma}}\right] E\{(\hat{M_2^j}(\boldsymbol{x}_c^*) - M_2(\boldsymbol{x}_c^*))^2 \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \hat{\boldsymbol{\gamma}}\}. \end{aligned} \tag{3.4}$$

where $\hat{M_1^j}(\boldsymbol{x}_c^*)$ and $\hat{M_2^j}(\boldsymbol{x}_c^*)$ are the posterior means of $M_1(\boldsymbol{x}_c^*)$ and $M_2(\boldsymbol{x}_c^*)$ given the vector $\boldsymbol{Y}_e^j$ and $\boldsymbol{\gamma}$. We choose $\boldsymbol{x}_{e,j}^*$ as the following:

$$\boldsymbol{x}_{e,j}^* = \underset{\boldsymbol{x}_e \in \mathcal{X}_e}{\operatorname{argmin}}\, MSE_j(\boldsymbol{x}_e).$$

The process (either $j = 1$ or $j = 2$) and the $\boldsymbol{x}_e$ for the next run of the experiment are $Y_j(\cdot)$ and $\boldsymbol{x}_{e,j}^*$ where $j = 1$ if $MSE_1(\boldsymbol{x}_{e,1}^*) \leq MSE_2(\boldsymbol{x}_{e,2}^*)$ and $j = 2$ otherwise.

5. If the stopping criterion is not met, then we calculate $y_j(\cdot)$ at the new point $(\boldsymbol{x}_c^*, \boldsymbol{x}_{e,j}^*)$, add that point to the corresponding initial design, set $n_j = n_j + 1$ and go to Step 2. If the criterion is met, the global minimizer is set to be the minimizer of the empirical BLUP of $M_1(\cdot)$ subject to the empirical BLUP of $M_2(\cdot)$ satisfying the constraint. Several stopping criteria are discussed in the examples.

The intuition of this algorithm is as follows. In Step 3 we choose the next control variable site $\boldsymbol{x}_c$ to maximize the expected improvement in $M_1(\cdot)$ over the minimum of $M_1(\cdot)$ for control variable sites already observed that appear to satisfy the constraint. In Step 4 we choose the next environmental variable site $\boldsymbol{x}_e$ to minimize a weighted

sum of the mean-squared prediction error of $M_1(\boldsymbol{x}_c^*)$ and $M_2(\boldsymbol{x}_c^*)$. The weight for the MSE of $M_1(\boldsymbol{x}_c^*)$ is 1 since we always want to make this MSE as small as possible. The weight for the MSE of $M_2(\boldsymbol{x}_c^*)$ is the probability that $\boldsymbol{x}_c^*$ is outside the feasible region. If this probability is small (i.e. we are confident that $\boldsymbol{x}_c^*$ is in the feasible region), then there is no need to make the MSE of prediction for the $M_2(\cdot)$ process any smaller, and only the $M_1(\cdot)$ prediction error is of concern.

3.2.2 Details

Step 1: Choosing the Initial Design

How best to choose $\boldsymbol{S}_{n_1}$ and $\boldsymbol{T}_{n_2}$ is a difficult problem. A simple approach is to generate a space filling design (eg. Latin hypercube) for $\boldsymbol{S}_{n_1}$ and set $\boldsymbol{T}_{n_2} = \boldsymbol{S}_{n_1}$. This would observe both $y_1(\cdot)$ and $y_2(\cdot)$ at the same sites and allow all points to contribute to the estimation of the $R_{12}(\cdot)$ parameters. In Section 3.3 we propose an initial choice of design so that both $\boldsymbol{S}_{n_1}$ and $\boldsymbol{T}_{n_2}$ are Latin hypercubes that have only half of their points in common.

Step 2: Maximizing the Posterior of $\boldsymbol{\gamma}$ given $\boldsymbol{Y}_1^{n_1}$ and $\boldsymbol{Y}_2^{n_2}$

The probability density function of the posterior distribution of $\boldsymbol{\gamma}$ given $\boldsymbol{Y}_1^{n_1}$ and $\boldsymbol{Y}_2^{n_2}$ is

$$p(\boldsymbol{\gamma} \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}) \propto p(\boldsymbol{\gamma}) |\boldsymbol{V}_{22}|^{-1/2} |\boldsymbol{F}^\top \boldsymbol{V}_{22}^{-1} \boldsymbol{F}|^{-1/2} [\hat{\tau_1^2}]^{-(n_1+n_2-k)/2}, \tag{3.5}$$

where $p(\boldsymbol{\gamma})$ is a prior distribution on the correlation parameters in $\boldsymbol{\gamma}$. The matrices $\boldsymbol{F}$ and $\boldsymbol{V}_{22}$ are defined in the following sections, and $\hat{\tau}_1^2$ is found in (3.10).

Step 3: Selection of Control Variables

We need to obtain a formula for the posterior expected improvement given in (3.3). We use iterated expectation and Monte Carlo for this calculation. Let $\boldsymbol{Y}_c = (\boldsymbol{M}_1^{n_1\top}, \boldsymbol{M}_2^{n_1\top}, \boldsymbol{Y}_1^{n_1\top}, \boldsymbol{Y}_2^{n_2\top})^\top$ (recall that $\boldsymbol{M}_i^{n_1} = [M_i(\boldsymbol{s}_{c,1}), ..., M_i(\boldsymbol{s}_{c,n_1})]$), and write (3.3) as an iterated expectation

$$E\{I(\boldsymbol{x}_c) \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \boldsymbol{\gamma}\} = E_{\boldsymbol{M}_1^{n_1}, \boldsymbol{M}_2^{n_1} | \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \boldsymbol{\gamma}}\{E[I(\boldsymbol{x}_c) \mid \boldsymbol{Y}_c, \boldsymbol{\gamma}]\}. \tag{3.6}$$

To evaluate the inner expectation, $E[I(\boldsymbol{x}_c) \mid \boldsymbol{Y}_c, \boldsymbol{\gamma}]$, we need the distribution of $[M_1(\boldsymbol{x}_c), M_2(\boldsymbol{x}_c)]$ given $[\boldsymbol{Y}_c, \boldsymbol{\gamma}]$, since given these, $I(\boldsymbol{x}_c)$ depends only on $M_1(\boldsymbol{x}_c)$ and $M_2(\boldsymbol{x}_c)$. For $i \in \{1, 2\}$, define $\boldsymbol{Y}_i^{n_e} = [Y_i(\boldsymbol{x}_c, \boldsymbol{x}_{e,1}), ..., Y_i(\boldsymbol{x}_c, \boldsymbol{x}_{e,n_e})]^\top$ to be the $n_e \times 1$ vector of observations from process i evaluated at control site $\boldsymbol{x}_c$ paired with each of the n_e support points for the environmental variable. Let $\boldsymbol{Y}_i^{n_1 n_e} = (Y_i(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,1}), ..., Y_i(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,n_e}), ..., Y_i(\boldsymbol{s}_{c,n_1}, \boldsymbol{x}_{e,1}), ..., Y_i(\boldsymbol{s}_{c,n_1}, \boldsymbol{x}_{e,n_e}))^\top$ be the $n_e n_1 \times 1$ vector of observations on the Y_i process evaluated at each of the n_1 control points present in $\boldsymbol{S}_{n_1}^C$, paired with each of the support points for the environmental variable.

The joint distribution of the vector $[\boldsymbol{Y}_1^{n_e}, \boldsymbol{Y}_2^{n_e}, \boldsymbol{Y}_1^{n_1 n_e}, \boldsymbol{Y}_2^{n_1 n_e}, \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}]$ given $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top)^\top$, τ_1^2, and $\boldsymbol{\gamma}$ is multivariate normal with mean $(\boldsymbol{F}_c^\top, \boldsymbol{F}_M^\top)^\top \boldsymbol{\beta}$ and variance-covariance matrix $\tau_1^2((\boldsymbol{C}_{pq}))$ for $p, q \in \{1, 2, 3, 4, 5, 6\}$, where the components are defined next. Let $\boldsymbol{F}_i^{n_e} = [\boldsymbol{f}_i(\boldsymbol{x}_c, \boldsymbol{x}_{e,1}), ..., \boldsymbol{f}_i(\boldsymbol{x}_c, \boldsymbol{x}_{e,n_e})]^\top$, $\boldsymbol{F}_i^{n_1 n_e} = [\boldsymbol{f}_i(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,1}), ..., \boldsymbol{f}_i(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,n_e}), ..., \boldsymbol{f}_i(\boldsymbol{s}_{c,n_1}, \boldsymbol{x}_{e,n_e})]^\top$, $\boldsymbol{F}_1^{n_1} = [\boldsymbol{f}_1(\boldsymbol{s}_1), \ ... \ \boldsymbol{f}_1(\boldsymbol{s}_{n_1})]^\top$, and $\boldsymbol{F}_2^{n_2} = [\boldsymbol{f}_2(\boldsymbol{t}_1), ..., \boldsymbol{f}_2(\boldsymbol{t}_{n_2})]^\top$ be the regression matrices for $\boldsymbol{Y}_i^{n_e}$, $\boldsymbol{Y}_i^{n_1 n_e}$, $\boldsymbol{Y}_1^{n_1}$, and $\boldsymbol{Y}_2^{n_2}$, respectively. Then let

$$\boldsymbol{F}_c = \begin{pmatrix} \boldsymbol{F}_1^{n_e} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_2^{n_e} \end{pmatrix}, \ \boldsymbol{F}_M = \begin{pmatrix} \boldsymbol{F}_1^{n_1 n_e} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_2^{n_1 n_e} \\ & \boldsymbol{F} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{F} = \begin{pmatrix} \boldsymbol{F}_1^{n_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_2^{n_2} \end{pmatrix}$$

where $\mathbf{0}$ is a matrix of zeroes of the appropriate size. The indices $p, q \in \{1, ..., 6\}$ for the covariance matrices $\boldsymbol{C}_{pq}$ correspond to the six components $\boldsymbol{Y}_1^{n_e}$, $\boldsymbol{Y}_2^{n_e}$, $\boldsymbol{Y}_1^{n_1 n_e}$, $\boldsymbol{Y}_2^{n_1 n_e}$, $\boldsymbol{Y}_1^{n_1}$, $\boldsymbol{Y}_2^{n_2}$ in this order, so that, for example, $\text{Cov}[\boldsymbol{Y}_2^{n_e}, \boldsymbol{Y}_1^{n_1 n_e}] = \tau_1^2 C_{23}$.

We apply a linear transformation to $[\boldsymbol{Y}_1^{n_e}, \boldsymbol{Y}_2^{n_e}, \boldsymbol{Y}_1^{n_1 n_e}, \boldsymbol{Y}_2^{n_1 n_e}, \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}]$ to obtain $[M_1(\boldsymbol{x}_c),\, M_2(\boldsymbol{x}_c),\, \boldsymbol{M}_1^{n_1},\, \boldsymbol{M}_2^{n_1},\, \boldsymbol{Y}_1^{n_1},\, \boldsymbol{Y}_2^{n_2}]$. Because Gaussian random vectors remain Gaussian under linear transformations, this vector has a Gaussian distribution with mean $\begin{pmatrix} \overline{\boldsymbol{F}_c} \\ \overline{\boldsymbol{F}_M} \end{pmatrix} \boldsymbol{\beta}$, and variance-covariance matrix $\tau_1^2((\boldsymbol{\Sigma}_{c,jk}))$ for $j, k \in \{1, 2\}$, where $\overline{\boldsymbol{F}_c} = (\boldsymbol{I}_2 \otimes \boldsymbol{w}^\top)\boldsymbol{F}_c$, $\boldsymbol{W}_{n_1} = \boldsymbol{I}_{n_1} \otimes \boldsymbol{w}$,

$$\overline{\boldsymbol{F}_M} = \begin{pmatrix} \boldsymbol{W}_{n_1}^\top \boldsymbol{F}_1^{n_1 n_e} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{W}_{n_1}^\top \boldsymbol{F}_2^{n_1 n_e} \\ & \boldsymbol{F} \end{pmatrix}, \quad \boldsymbol{\Sigma}_{c,11} = \boldsymbol{w}^\top \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \cdot & \boldsymbol{C}_{22} \end{pmatrix} \boldsymbol{w},$$

$$\boldsymbol{\Sigma}_{c,12} = \boldsymbol{w}^\top \begin{pmatrix} \boldsymbol{C}_{13}\boldsymbol{W}_{n_1} & \boldsymbol{C}_{14}\boldsymbol{W}_{n_1} & \boldsymbol{C}_{15} & \boldsymbol{C}_{16} \\ \boldsymbol{C}_{23}\boldsymbol{W}_{n_1} & \boldsymbol{C}_{24}\boldsymbol{W}_{n_1} & \boldsymbol{C}_{25} & \boldsymbol{C}_{26} \end{pmatrix},$$

$$\boldsymbol{\Sigma}_{c,22} = \begin{pmatrix} \boldsymbol{W}_{n_1}^\top \boldsymbol{C}_{33} \boldsymbol{W}_{n_1} & \boldsymbol{W}_{n_1}^\top \boldsymbol{C}_{34} \boldsymbol{W}_{n_1} & \boldsymbol{W}_{n_1}^\top \boldsymbol{C}_{35} & \boldsymbol{W}_{n_1}^\top \boldsymbol{C}_{36} \\ \cdot & \boldsymbol{W}_{n_1}^\top \boldsymbol{C}_{44} \boldsymbol{W}_{n_1} & \boldsymbol{W}_{n_1}^\top \boldsymbol{C}_{45} & \boldsymbol{W}_{n_1}^\top \boldsymbol{C}_{46} \\ \cdot & \cdot & \boldsymbol{C}_{55} & \boldsymbol{C}_{56} \\ \cdot & \cdot & \cdot & \boldsymbol{C}_{66} \end{pmatrix}$$

and $\boldsymbol{\Sigma}_{c,21} = \boldsymbol{\Sigma}_{c,12}^\top$. Here, $\boldsymbol{I}_r$ denotes the $r \times r$ identity matrix, $\boldsymbol{w} = (w_1, ..., w_{n_e})^\top$ is the vector of weights defining the distribution of the environmental variables, and $\otimes$ denotes the Kronecker product operator. The dot entries in the covariance matrices are defined by symmetry.

Let $\mathcal{T}_q(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$ denote the q-variate t distribution with location shift $\boldsymbol{\mu}$, scale matrix $\boldsymbol{\Sigma}$, and ν degrees of freedom (see Definition B.0.8). To obtain the posterior distribution of $(M_1(\boldsymbol{x}_c), M_2(\boldsymbol{x}_c))$, given $\boldsymbol{Y}_c$ and $\boldsymbol{\gamma}$, we apply Lemma B.0.1 in Appendix B (see O'Hagan (1992)) and find

$$[M_1(\boldsymbol{x}_c), M_2(\boldsymbol{x}_c) \mid \boldsymbol{Y}_c, \boldsymbol{\gamma}] \;\sim\; \mathcal{T}_2(\boldsymbol{m}_c,\; \hat{\tau}_{1,c}^2 \boldsymbol{R}_c,\; 3n_1 + n_2 - k), \tag{3.7}$$

where

$$\boldsymbol{m}_c = \overline{\boldsymbol{F}_c}\hat{\boldsymbol{\beta}}_c \; + \; \boldsymbol{\Sigma}_{c,12}\boldsymbol{\Sigma}_{c,22}^{-1}(\boldsymbol{Y}_c - \overline{\boldsymbol{F}_M}\hat{\boldsymbol{\beta}}_c),$$

$$\hat{\boldsymbol{\beta}}_c = (\overline{\boldsymbol{F}_M}^\top \boldsymbol{\Sigma}_{c,22}^{-1}\overline{\boldsymbol{F}_M})^{-1}\overline{\boldsymbol{F}_M}^\top \boldsymbol{\Sigma}_{c,22}^{-1}\boldsymbol{Y}_c,$$

$$\hat{\tau}_{1,c}^2 = \frac{\boldsymbol{Y}_c^\top \boldsymbol{\Sigma}_{c,22}^{-1}\boldsymbol{Y}_c - \hat{\boldsymbol{\beta}}_c^\top(\overline{\boldsymbol{F}_M}^\top \boldsymbol{\Sigma}_{c,22}^{-1}\overline{\boldsymbol{F}_M})\hat{\boldsymbol{\beta}}_c}{3n_1+n_2-k},$$

and

$$\begin{aligned}\boldsymbol{R}_c \;\; = \;\; & \boldsymbol{\Sigma}_{c,11} - \boldsymbol{\Sigma}_{c,12}\boldsymbol{\Sigma}_{c,22}^{-1}\boldsymbol{\Sigma}_{c,12}^\top \;\; + \\ & (\overline{\boldsymbol{F}_c} - \boldsymbol{\Sigma}_{c,12}\boldsymbol{\Sigma}_{c,22}^{-1}\overline{\boldsymbol{F}_M})(\overline{\boldsymbol{F}_M}^\top \boldsymbol{\Sigma}_{c,22}^{-1}\overline{\boldsymbol{F}_M})^{-1}(\overline{\boldsymbol{F}_c} - \boldsymbol{\Sigma}_{c,12}\boldsymbol{\Sigma}_{c,22}^{-1}\overline{\boldsymbol{F}_M})^\top.\end{aligned}$$

Using this distribution we can calculate the inner conditional expectation in (3.6). Let $\nu = 3n_1 + n_2 - k$, $\hat{r} = \frac{R_{c,12}}{\sqrt{R_{c,11}R_{c,22}}}$, $U_1 = \frac{M_1^{min,c} - m_{c,1}}{\sqrt{\hat{\tau}_{1,c}^2 R_{c,11}}}$, $U_2 = \frac{U - m_{c,2}}{\sqrt{\hat{\tau}_{1,c}^2 R_{c,22}}}$, $C(z) = \sqrt{\frac{\nu}{\nu-2}}\; t_{\nu-2}\left(z\sqrt{\frac{\nu-2}{\nu}}\right)$, and $\zeta_{\hat{r}}^2(z) = (1-\hat{r^2})\frac{z^2+\nu}{\nu-1}$, where $t_\kappa(\cdot)$ denotes the standard t density function with κ degrees of freedom, and apply a linear transformation with Lemma B.0.2 to compute

$$\begin{aligned}E[I(\boldsymbol{x}_c) \mid \boldsymbol{Y}_c, \boldsymbol{\gamma}] \;\; = \;\; & \sqrt{\hat{\tau}_{1,c}^2 R_{c,11}} \;\; \Big[\; U_1 \; T_{2,\hat{r}}(U_1, U_2, \nu) \;\; + \\ & C(U_1)\; T_{\nu-1}\left(\tfrac{U_2 - \hat{r}U_1}{\zeta_{\hat{r}}(U_1)}\right) + \hat{r}\; C(U_2)\; T_{\nu-1}\left(\tfrac{U_1 - \hat{r}U_2}{\zeta_{\hat{r}}(U_2)}\right)\Big]\end{aligned} \tag{3.8}$$

where $T_{2,\hat{r}}(\cdot,\cdot,\kappa)$ is the joint CDF of the bivariate t with κ degrees of freedom, location vector $\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix} = \mathbf{0}_2$ and scale matrix $\begin{pmatrix}1 & \hat{r}\\ \hat{r} & 1\end{pmatrix}$, i.e. the $\mathcal{T}_2(\mathbf{0}_2, \begin{pmatrix}1 & \hat{r}\\ \hat{r} & 1\end{pmatrix}, \kappa)$ distribution, and $T_\kappa(\cdot)$ is the CDF of the univariate t distribution with κ degrees of freedom.

Finally, calculation of the unconditional posterior expected improvement is accomplished by integrating (3.8) using Monte Carlo. We generate N random samples from the distribution of $[\boldsymbol{M}_1^{n_1}, \boldsymbol{M}_2^{n_1}]$ given $\boldsymbol{Y}_1^{n_1}$, $\boldsymbol{Y}_2^{n_2}$, and $\boldsymbol{\gamma}$. For each of the N samples we calculate $M_1^{min,c}$ and compute (3.8). The posterior expected improvement is

the average of these N quantities. In order to accomplish this we need to find the posterior distribution of $[\boldsymbol{M}_1^{n_1}, \boldsymbol{M}_2^{n_1}]$ given $\boldsymbol{Y}_1^{n_1}$, $\boldsymbol{Y}_2^{n_2}$, and $\boldsymbol{\gamma}$.

As before, we know that, given $(\boldsymbol{\beta}, \tau_1^2, \boldsymbol{\gamma})$, the random vector $\boldsymbol{Y}_c$ has a joint Gaussian distribution with mean $\overline{\boldsymbol{F}_M}\boldsymbol{\beta}$ and covariance matrix $\tau_1^2\boldsymbol{\Sigma}_{c,22}$. To partition these matrices into the components associated with $[\boldsymbol{M}_1^{n_1}, \boldsymbol{M}_2^{n_1}]$ and the data vector $\boldsymbol{Y}^d = [\boldsymbol{Y}_1^{n_1\top}, \boldsymbol{Y}_2^{n_2\top}]^\top$ we define the following:

$$\overline{\boldsymbol{F}_{M,1}} = (\boldsymbol{I}_{n_1} \otimes \boldsymbol{w}^\top) \begin{pmatrix} \boldsymbol{F}_1^{n_1 n_e} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_2^{n_1 n_e} \end{pmatrix}, \quad \boldsymbol{V}_{12} = (\boldsymbol{I}_{n_1} \otimes \boldsymbol{w}^\top) \begin{pmatrix} \boldsymbol{C}_{35} & \boldsymbol{C}_{36} \\ \boldsymbol{C}_{45} & \boldsymbol{C}_{46} \end{pmatrix},$$

$$\boldsymbol{V}_{11} = (\boldsymbol{I}_{n_1} \otimes \boldsymbol{w}^\top) \begin{pmatrix} \boldsymbol{C}_{33} & \boldsymbol{C}_{34} \\ \cdot & \boldsymbol{C}_{44} \end{pmatrix} (\boldsymbol{I}_{n_1} \otimes \boldsymbol{w}), \quad \boldsymbol{V}_{22} = \begin{pmatrix} \boldsymbol{C}_{55} & \boldsymbol{C}_{56} \\ \cdot & \boldsymbol{C}_{66} \end{pmatrix}.$$

Again applying Lemma B.0.1 we see that $[\boldsymbol{M}_1^{n_1}, \boldsymbol{M}_2^{n_1}]$ given $\boldsymbol{Y}_1^{n_1}$, $\boldsymbol{Y}_2^{n_2}$, and $\boldsymbol{\gamma}$ is $2n_1$-variate t:

$$[\boldsymbol{M}_1^{n_1}, \boldsymbol{M}_2^{n_1} \mid \boldsymbol{Y}^d, \boldsymbol{\gamma}] \sim \mathcal{T}_{2n_1}(\boldsymbol{m}, \hat{\tau}_1^2\boldsymbol{R}, n_1 + n_2 - k), \tag{3.9}$$

where $\boldsymbol{m} = \overline{\boldsymbol{F}_{M,1}}\hat{\boldsymbol{\beta}} + \boldsymbol{V}_{12}\boldsymbol{V}_{22}^{-1}(\boldsymbol{Y}^d - \boldsymbol{F}\hat{\boldsymbol{\beta}})$, $\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^\top\boldsymbol{V}_{22}^{-1}\boldsymbol{Y}^d$,

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{V}_{11} - \boldsymbol{V}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{V}_{12}^\top + \\ &\quad (\overline{\boldsymbol{F}_{M,1}} - \boldsymbol{V}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})(\boldsymbol{F}^\top\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^{-1}(\overline{\boldsymbol{F}_{M,1}} - \boldsymbol{V}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^\top, \end{aligned}$$

and

$$\hat{\tau}_1^2 = [\boldsymbol{Y}^{d\top}\boldsymbol{V}_{22}^{-1}\boldsymbol{Y}^d - \hat{\boldsymbol{\beta}}^\top(\boldsymbol{F}^\top\boldsymbol{V}_{22}^{-1}\boldsymbol{F})\hat{\boldsymbol{\beta}}]/(n_1 + n_2 - k). \tag{3.10}$$

Step 4: Selection of Environmental Variables

To calculate (3.4) we start by fixing $i, j \in \{1, 2\}$ and letting $J_i^j(\boldsymbol{x}_e) = (\hat{M}_i^j(\boldsymbol{x}_c^*) - M_i(\boldsymbol{x}_c^*))^2$. We have

$$E\{J_i^j(\boldsymbol{x}_e) \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \boldsymbol{\gamma}\} = E_{Y_j(\boldsymbol{x}_c^*, \boldsymbol{x}_e)|\boldsymbol{Y}^d, \boldsymbol{\gamma}}\{E[J_i^j(\boldsymbol{x}_e) \mid \boldsymbol{Y}_e^j, \boldsymbol{\gamma}]\} \tag{3.11}$$

where $\boldsymbol{Y}_e^j$ =[$Y_j(\boldsymbol{x}_c^*, \boldsymbol{x}_e)$, $\boldsymbol{Y}_1^{n_1\top}$, $\boldsymbol{Y}_2^{n_2\top}$]. To calculate the inner expectation, we first obtain the joint distribution of the random variables $\boldsymbol{Y}_1^{n_e}(\boldsymbol{x}_c^*)$, $\boldsymbol{Y}_2^{n_e}(\boldsymbol{x}_c^*)$ and $\boldsymbol{Y}_e^j$ where $\boldsymbol{Y}_i^{n_e}(\boldsymbol{x}_c^*) = (Y_i(\boldsymbol{x}_c^*, \boldsymbol{x}_{e,1}), ..., Y_i(\boldsymbol{x}_c^*, \boldsymbol{x}_{e,n_e}))^\top$. Define

$$\boldsymbol{F}_p^1 = \begin{pmatrix} \boldsymbol{f}_1^\top(\boldsymbol{x}_c^*, \boldsymbol{x}_e) & \boldsymbol{0} \\ \boldsymbol{F} & \end{pmatrix}, \quad \boldsymbol{F}_p^2 = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{f}_2^\top(\boldsymbol{x}_c^*, \boldsymbol{x}_e) \\ & \boldsymbol{F} \end{pmatrix}, \quad \boldsymbol{F}_c^* = \begin{pmatrix} \boldsymbol{F}_1^{n_e*} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_2^{n_e*} \end{pmatrix},$$

where $\boldsymbol{F}_i^{n_e*} = [\boldsymbol{f}_i(\boldsymbol{x}_c^*, \boldsymbol{x}_{e,1}), \ ... \ \boldsymbol{f}_i(\boldsymbol{x}_c^*, \boldsymbol{x}_{e,n_e})]^\top$ is $n_e \times k_i$. The joint distribution of $\boldsymbol{Y}_1(\boldsymbol{x}_c^*)$, $\boldsymbol{Y}_2(\boldsymbol{x}_c^*)$, $Y_j(\boldsymbol{x}_c^*, \boldsymbol{x}_e)$, $\boldsymbol{Y}_1^{n_1}$, $\boldsymbol{Y}_2^{n_2}$ is Gaussian with mean $\begin{pmatrix} \boldsymbol{F}_c^* \\ \boldsymbol{F}_p^j \end{pmatrix} \boldsymbol{\beta}$ and variance-covariance matrix $\tau^2((E_{pq}))$ for $p, q \in \{1, 2, 3, 4, 5\}$ where the indices p and q correspond to the five components $\boldsymbol{Y}_1(\boldsymbol{x}_c^*)$, $\boldsymbol{Y}_2(\boldsymbol{x}_c^*)$, $Y_j(\boldsymbol{x}_c^*, \boldsymbol{x}_e)$, $\boldsymbol{Y}_1^{n_1}$, $\boldsymbol{Y}_2^{n_2}$ so that, for example, Cov[$\boldsymbol{Y}_1(\boldsymbol{x}_c^*)$, $\boldsymbol{Y}_1^{n_1}$] = $\tau_1^2 E_{14}$.

Applying a linear transformation and Lemma B.0.1 with

$$\boldsymbol{\Sigma}_{e,11} = \boldsymbol{w}^\top \begin{pmatrix} E_{11} & E_{12} \\ \cdot & E_{22} \end{pmatrix} \boldsymbol{w}, \qquad \boldsymbol{\Sigma}_{e,12} = \boldsymbol{w}^\top \begin{pmatrix} E_{13} & E_{14} & E_{15} \\ E_{23} & E_{24} & E_{25} \end{pmatrix},$$

and $\boldsymbol{\Sigma}_{e,22}$ the 3×3 block matrix with elements E_{pq} for p, $q \in \{3, 4, 5\}$, the posterior distribution of $M_1(\boldsymbol{x}_c^*)$, $M_2(\boldsymbol{x}_c^*)$ given $\boldsymbol{Y}_e^j$ and $\boldsymbol{\gamma}$ is the scaled and shifted bivariate t:

$$[M_1(\boldsymbol{x}_c^*), M_2(\boldsymbol{x}_c^*) \mid \boldsymbol{Y}_e^j, \boldsymbol{\gamma}] \sim \mathcal{T}_2(\boldsymbol{m}_e, \ \hat{\tau}_{1,e}^2 \boldsymbol{R}_e, \ n_1 + n_2 + 1 - k), \tag{3.12}$$

where

$$\begin{aligned} \boldsymbol{m}_e &= \overline{\boldsymbol{F}_c^*}\hat{\boldsymbol{\beta}}_e + \boldsymbol{\Sigma}_{e,12}\boldsymbol{\Sigma}_{e,22}^{-1}(\boldsymbol{Y}_e^j - \boldsymbol{F}_p^j\hat{\boldsymbol{\beta}}_e), \\ \hat{\boldsymbol{\beta}}_e &= (\boldsymbol{F}_p^{j\top}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p^j)^{-1}\boldsymbol{F}_p^{j\top}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{Y}_e^j, \\ \hat{\tau}_{1,e}^2 &= \frac{\boldsymbol{Y}_e^{j\top}\boldsymbol{Q}_e\boldsymbol{Y}_e^j}{n_1 + n_2 + 1 - k}, \\ \boldsymbol{Q}_e &= \boldsymbol{\Sigma}_{e,22}^{-1} - \boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p^j(\boldsymbol{F}_p^{j\top}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p^j)^{-1}\boldsymbol{F}_p^{j\top}\boldsymbol{\Sigma}_{e,22}^{-1}, \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{R}_e &= \boldsymbol{\Sigma}_{e,11} - \boldsymbol{\Sigma}_{e,12}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{\Sigma}_{e,12}^\top \quad + \\ & (\overline{\boldsymbol{F}_c^*} - \boldsymbol{\Sigma}_{e,12}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p^j)(\boldsymbol{F}_p^{j\top}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p^j)^{-1}(\overline{\boldsymbol{F}_c^*} - \boldsymbol{\Sigma}_{e,12}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p^j)^\top. \end{aligned}$$

For $i \in \{1,2\}$, the posterior mean $\boldsymbol{m}_{e,i}$ is the best linear unbiased predictor of $M_i(\boldsymbol{x}_c^*)$ based on the data $\boldsymbol{Y}_e^j$, thus $E\{J_i^j(\boldsymbol{x}_e) \mid \boldsymbol{Y}_e^j, \boldsymbol{\gamma}\}$ is the posterior variance of $M_i(\boldsymbol{x}_c^*)$ for the distribution given in (3.12), i.e.,

$$E\{J_i^j(\boldsymbol{x}_e) \mid \boldsymbol{Y}_e^j, \boldsymbol{\gamma}\} = \left(\frac{n_1+n_2+1-k}{n_1+n_2-1-k}\right)\ \hat{\tau}_{1,e}^2\ R_{e,ii}. \tag{3.13}$$

To calculate the outer expectation in (3.11) we note that the formula for the inner expectation given in (3.13) is just a constant times a quadratic form in $\boldsymbol{Y}_e^j$, whose expectation can be evaluated using the well known formula for expectations of a quadratic form (see Theorem B.0.6 in Appendix B). The distribution of $Y_j(\boldsymbol{x}_c^*, \boldsymbol{x}_e)$ given $\boldsymbol{Y}^d$ and $\boldsymbol{\gamma}$ has mean $m = \boldsymbol{F}_{p,1}^j\hat{\boldsymbol{\beta}} + \boldsymbol{a}_{12}\boldsymbol{V}_{22}^{-1}(\boldsymbol{Y}^d - \boldsymbol{F}\hat{\boldsymbol{\beta}})$ where $\boldsymbol{a}_{12} = (E_{34}, E_{35})$. Setting $\boldsymbol{Z}_e^\top = (m, \boldsymbol{Y}^{d\top})$ and applying Theorem B.0.8, we compute the outer expectation in (3.11) as

$$E\{J_i^j(\boldsymbol{x}_e) \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \boldsymbol{\gamma}\} = \frac{R_{e,ii}}{n_1+n_2-1-k}\big(\boldsymbol{Z}_e^\top \boldsymbol{Q}_e \boldsymbol{Z}_e + \frac{n_1+n_2-k}{n_1+n_2-k-2}\ \hat{\tau}_1^2\big)$$

where $\hat{\boldsymbol{\beta}}$ is given below (3.9) and $\hat{\tau}_1^2$ is given in (3.10).

Finally, to calculate $P(M_2(\boldsymbol{x}_c^*) > U \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \boldsymbol{\gamma})$ we note that

$$[M_2(\boldsymbol{x}_c^*) \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \boldsymbol{\gamma}] \sim \mathcal{T}_1(m_p,\ \hat{\tau}_1^2 R_p,\ n_1+n_2-k)$$

where $m_p = \overline{\boldsymbol{F}_2^{n_e*}}\hat{\boldsymbol{\beta}} + \boldsymbol{w}_{12}\boldsymbol{V}_{22}^{-1}(\boldsymbol{Y}^d - \boldsymbol{F}\hat{\boldsymbol{\beta}})$,

$$\begin{aligned} R_p \ &= \ W_{11} - \boldsymbol{w}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{w}_{12}^\top \ \ + \\ &\quad (\overline{\boldsymbol{F}_2^{n_e*}} - \boldsymbol{w}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})(\boldsymbol{F}^\top\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^{-1}(\overline{\boldsymbol{F}_2^{n_e*}} - \boldsymbol{w}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^\top, \end{aligned}$$

$W_{11} = \boldsymbol{w}^\top E_{22}\boldsymbol{w}$, $\boldsymbol{w}_{12} = \boldsymbol{w}^\top\ (E_{24} \quad E_{25})$, and $\overline{\boldsymbol{F}_2^{n_e*}} = \boldsymbol{w}^\top(\boldsymbol{0}, \boldsymbol{F}_2^{n_e*})$. Using this distribution we obtain,

$$P(M_2(\boldsymbol{x}_c^*) > U \mid \boldsymbol{Y}_1^{n_1}, \boldsymbol{Y}_2^{n_2}, \boldsymbol{\gamma}) = \ 1 \ - \ T_{n_1+n_2-k}\left(\frac{U-m_p}{\sqrt{\hat{\tau}_1^2 R_p}}\right), \tag{3.14}$$

where $T_\kappa(\cdot)$ is the univariate T cdf with κ degrees of freedom.

3.3 Examples

The examples in this section will demonstrate the proposed algorithm, and investigate the effects of initial sample size and design. We fit the spatial autoregressive model proposed by Kennedy and O'Hagan (2000). This model specifies a joint distribution for $(Z_1(\cdot),Z_2(\cdot))$ by starting with independent processes $Z_1(\cdot)$ and $Z_\delta(\cdot)$ that are stationary Gaussian processes with mean zero, variances τ_1^2 and τ_δ^2, and correlation functions $R_1(\cdot)$ and $R_\delta(\cdot)$ respectively. Set

$$Z_2(\boldsymbol{x}) = rZ_1(\boldsymbol{x}) + Z_\delta(\boldsymbol{x}), \tag{3.15}$$

for $\boldsymbol{x} \in \mathcal{X}$. Then we can determine τ_2^2, $R_2(\cdot)$ and $R_{12}(\cdot)$ for this model: $\tau_2^2 = r^2\tau_1^2+\tau_\delta^2$, $R_2(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \text{Corr}[Z_2(\boldsymbol{x}_1), Z_2(\boldsymbol{x}_2)] = \big(r^2R_1(\boldsymbol{x}_1 - \boldsymbol{x}_2) + (\eta - r^2)R_\delta(\boldsymbol{x}_1 - \boldsymbol{x}_2)\big)/\eta$, and $R_{12}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \text{Corr}[Z_1(\boldsymbol{x}_1), Z_2(\boldsymbol{x}_2)] = rR_1(\boldsymbol{x}_1 - \boldsymbol{x}_2)/\eta$, where $\eta = \tau_2^2/\tau_1^2$.

In the following examples, we utilize the power exponential class of correlation functions for $R_1(\cdot)$ and $R_\delta(\cdot)$. For $h \in \{1, \delta\}$,

$$R_h(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{p} \exp\big(-\theta_i^h|x_{1,i} - x_{2,i}|^{\alpha_i^h}\big),$$

where $\theta_i^h > 0$ and $0 < \alpha_i^h \leq 2$. As θ_i^h increases, the dependence between fixed input sites decreases since $R_h(\cdot)$ decreases. If $\alpha_i^h = 2$, then the Z_h process is infinitely mean square differentiable and the sample paths are infinitely differentiable in the i-th direction. However, if $\alpha_i^h < 2$ then the process is mean square continuous but not differentiable in the i-th direction (see Cramér and Leadbetter (1967) secs. 9.2-9.5). For the power exponential class, the parameter vector is $\boldsymbol{\gamma} = (\tau_2^2,\, r, \theta_1^1,\, ...,\, \theta_p^1, \alpha_1^1,\, ... \alpha_p^1, \theta_1^\delta,\, ...,\, \theta_p^\delta, \alpha_1^\delta,\, ...,\, \alpha_p^\delta)$ or, reparametrizing, $\boldsymbol{\gamma} = (\eta, r, \theta_1^1,\, ...,\, \theta_p^1, \alpha_1^1,\, ...\,\, \alpha_p^1, \theta_1^\delta,\, ...,\, \theta_p^\delta, \alpha_1^\delta,\, ...,\, \alpha_p^\delta)$.

The Matérn family of correlation functions (see Stein (1999), Section 2.7) allows for a parameter that controls the smoothness (or mean square differentiability) of the process. In practice, we have found that this family does not add significant improvement to the accuracy of our algorithm, but merely increases the computing time needed in the estimation procedure (Step 2), and at times causes numerical problems in the necessary calculations. Of course, most of our examples involve fairly smooth functions. If the computer code was believed to produce output that was not so smooth (perhaps having only a single derivative) the Matérn family might be more reasonable.

3.3.1 Two-Dimensional Examples

The numerical examples study the characteristics of the algorithm when solving a sequence of randomly generated test problems. The input space is $(0,1) \times (0,1)$ with the first component being a control variable, and the second component being an environmental variable having a discrete uniform distribution on the 10 points $(0.05, 0.15, ..., 0.85, 0.95)$. The class of test functions for $y_1(\cdot)$ are draws from

$$y_1(x_c, x_e) = (x_c - \theta_1)(x_c - \theta_2)(x_c - x_e)\cos(\theta_3 x_c) + 0.1\sin(\frac{\theta_3}{2} x_e) \tag{3.16}$$

with θ_1 and θ_2 independent and uniformly distributed on (-1,1) and θ_3, independent of θ_1 and θ_2, and uniformly distributed on $(0, 8\pi)$. This is a very flexible class in that the responses have between 0 and 8 zeroes and multiple local optima. $y_1(\cdot)$ can be very "smooth" when θ_3 is small, and very "bumpy" when θ_3 is large. The functions in this class are continuous and infinitely differentiable, which may explain why the Matérn class of correlation functions offered no improvement over the power

exponential class. Figure 3.2 shows the wide variety of shapes that this class allows by displaying 9 $\mu_1(x_c)$'s corresponding to draws from (3.16).

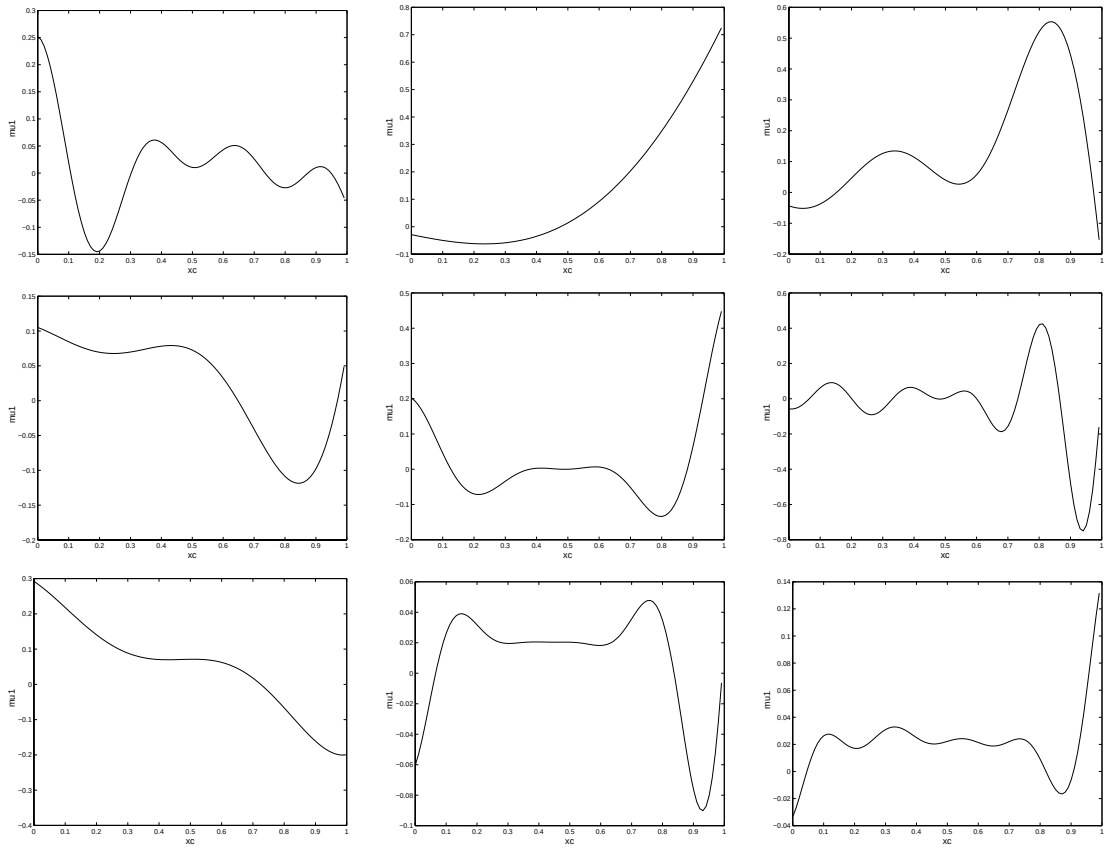


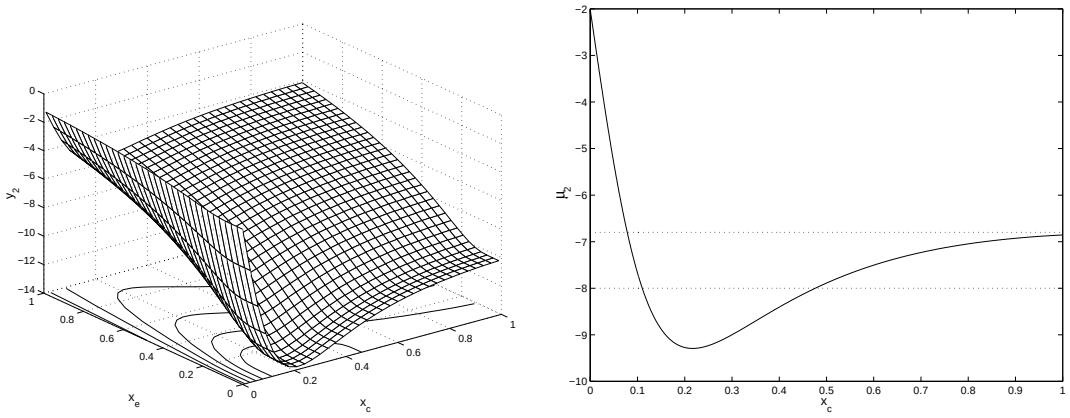
Figure 3.2: Nine $\mu_1(x_c)$ draws from (3.16)

The $y_2(\cdot)$ function is fixed for all runs to be

$$y_2(x_c, x_e) = -([1 - e^{\frac{-1}{2x_e}}] \times \frac{2300x_c^3 + 1900x_c^2 + 2092x_c + 60}{100x_c^3 + 500x_c^2 + 4x_c + 20}).$$

We choose the constraint U to be either -6.8 or -8, each with probability 1/2. If U = -6.8, then the feasible region for x_c is (0.0783, 1.0), and if U = -8, then the feasible

region for x_c is (0.1101, 0.4762). Figure 3.3 shows the true $y_2(\cdot)$ process, and the corresponding constraint function with horizontal lines drawn at both -8 and -6.8.



y_2 μ_2

Figure 3.3: True constraint function

For each of 25 draws of $y_1(\cdot)$ and U, we ran the proposed algorithm four times corresponding to two choices for the initial design size and two choices for the "type" of initial design. The two initial design sizes that we chose are $n_1 = n_2 = 18$ and $n_1 = n_2 = 10$. The Jones et al. (1998) rule of thumb of 10 observations per input dimension suggests $n_1 = n_2 = 20$. We investigate the effect of taking fewer observations on the accuracy of the answer produced by the algorithm and on the total number of observations required to obtain that answer. The two "types" of designs that we consider here are as follows;

(1) All $n_1 = n_2$ observations on $y_1(\cdot)$ and $y_2(\cdot)$ are taken at the *same* input sites, a Latin hypercube design that approximately maximizes the average distance between all pairs of points. (Generated using ACED of Welch (1985)).

(2) The n_1 sites for $y_1(\cdot)$ are chosen as in (1). The n_2 $(= n_1)$ sites for $y_2(\cdot)$ are chosen in the following manner: First we choose $n_2/2$ of the $y_1(\cdot)$ input sites so as to form a size $n_2/2$ Latin hypercube. These will be the common sites at which both $y_1(\cdot)$ and $y_2(\cdot)$ are observed. The remaining $n_2/2$ sites for $y_2(\cdot)$ are chosen by randomly pairing the p components of the remaining $n_2/2$ sites from the $y_1(\cdot)$ observations. For example, Figure 3.4 displays a *mixed* design with $n_1 = n_2 = 18$. The +'s are those sites where $y_1(\cdot)$ is observed and are chosen as a Latin hypercube design that maximizes the average distance between all pairs of points. The o's are those sites where $y_2(\cdot)$ is observed and are chosen using the individual components of the sites where $y_1(\cdot)$ is observed. As described above, we first choose 9 points from the $y_1(\cdot)$ sites to form a 9-point Latin hypercube on which to observe both $y_1(\cdot)$ and $y_2(\cdot)$. The remaining 9 sites for $y_2(\cdot)$ are obtained by randomly pairing the x_1 and x_2 components of the remaining 9 sites for $y_1(\cdot)$. Note that this construction allows the design for both $y_1(\cdot)$ and $y_2(\cdot)$ to be a Latin hypercube. We will refer to this design as the *mixed* design.

For the stopping criterion we follow Williams et al. (2000c) and stop the algorithm when both a moving average *and* a moving range of the expected improvements are "small". Let $\hat{I}_j$ be the observed expected improvement at iteration j. Then for $j \geq g$ let $A_j = (\hat{I}_j + ... + \hat{I}_{j-g+1})/g$ and $R_j = \max\{\hat{I}_j, ..., \hat{I}_{j-g+1}\} - \min\{\hat{I}_j, ..., \hat{I}_{j-g+1}\}$ be the moving average and range of the expected improvements for the previous g iterations of the algorithm. We stop the algorithm at the first $j \leq 45$ for which $A_j \leq 0.000005$ and $R_j \leq 0.00005$, or at $j = 45$. Note that this stopping rule is problem specific in that it depends on the *scale* of the responses for the moving average and range criterion

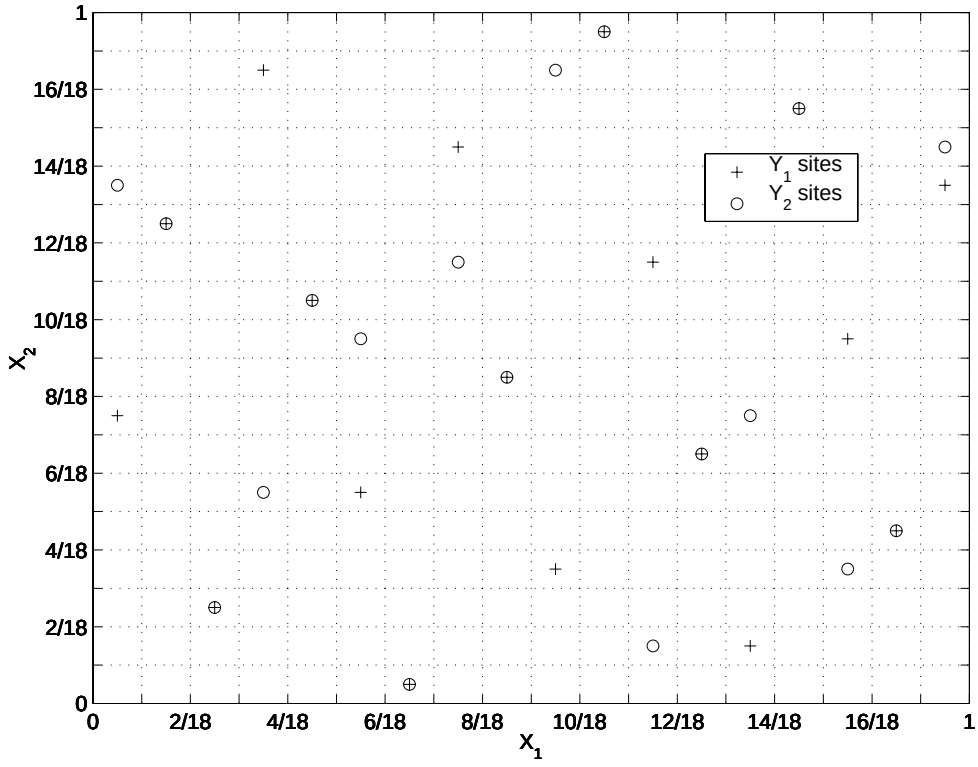


Figure 3.4: 18-Point mixed design

and on the *number of dimensions* in the input space for the maximum number of runs criterion.

3.3.2 An Illustration

We present one simulation run of our algorithm for illustrative purposes. The true objective and true constraint functions are shown as dashed lines in Figure 3.5. For this simulation we set $g = 5$, $N = 300$, and $U = -8$ (shown by the horizontal line in the right hand plot). The feasible region for x_c is (0.1101, 0.4762), and the constrained optimum (also the global optimum in this case) is $x_c = 0.22866$. The initial design for this example was the 10-point Latin hypercube where the two responses were observed

at the same sites. The algorithm added a total of 19 observations with 14 of those taken on $Y_1(\cdot)$ and 5 taken on $Y_2(\cdot)$. Figure 3.5 also shows the final predicted surfaces for $\mu_1(x_c)$ and $\mu_2(x_c)$ along with corresponding error bars and the true surface. As you can see, the boundary of the feasible region is closely approximated, and, within the feasible region, the objective function is predicted with virtually no visible difference from the true objective function. Figure 3.6 displays the control variable portions of the points where each response was observed.

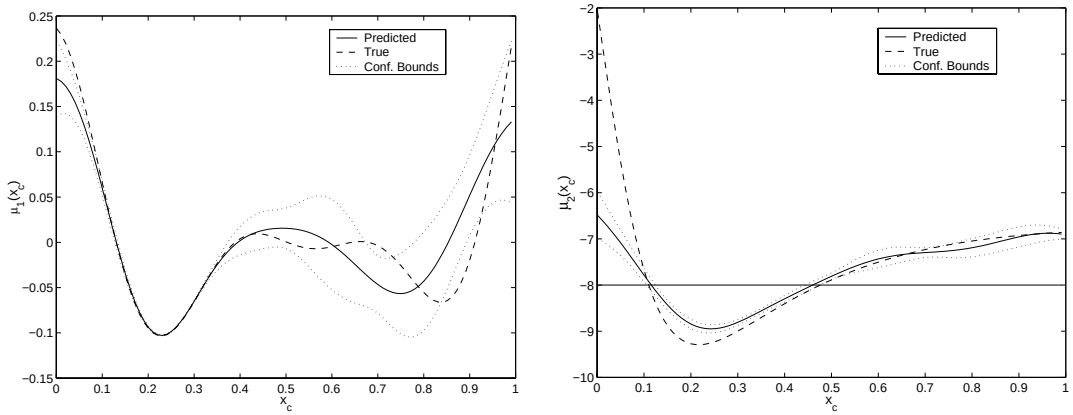


Figure 3.5: Final predictions for one draw of $y_1(\cdot)$ objective function

3.3.3 Results

Table 3.1 summarizes the 25 simulated examples solved by the algorithm (with $g = 5$ and $N = 300$). The column labeled *Size* corresponds to the initial sample size taken on each of $y_1(\cdot)$ and $y_2(\cdot)$. The column labeled *Type* refers to the two initial design types, corresponding to whether $y_1(\cdot)$ and $y_2(\cdot)$ are observed at the *same* sites or at a *mixed* set of sites. The table presents summary statistics taken over the 25

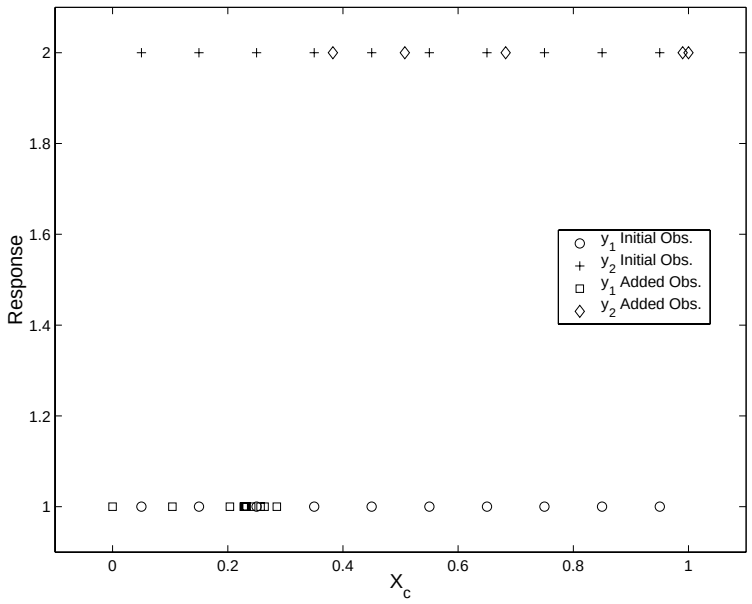


Figure 3.6: Control variable portions of the observation sites in the initial design ($\circ$ and $+$) and observation sites added by the sequential algorithm ($\square$ and $\diamond$) for the constrained optimization example.

draws (surfaces). The primary interest here is in minimizing the number of runs of the two computer codes, while being sure to find the constrained minimizer. In general, it appears that the algorithm observes the $y_1(\cdot)$ response more often then the $y_2(\cdot)$ response, and it manages to find the approximate optimal x_c value with the predicted optimal value being within 0.005 of the true optimal value for most algorithm runs. The algorithm runs where the predicted optimal value was further than 0.005 from the true optimal value correspond to situations where the constrained optimum was on the boundary of the feasible region. For these cases, the algorithm was stopped at the maximum number of iterations, not by the moving average/moving range stopping criterion. Continuing the algorithm further (i.e. until the moving average/moving

range criterion is met) will improve the estimation of the optimal value. Table 3.2 displays the summary of the 17 random situations where the moving average/moving range stopping criterion was used, removing those algorithm runs that stopped at 45 iterations for any of the four starting designs. Here, we see that in all four cases for each draw the predicted constrained optimum, x_c^* was within 0.005 of the true optimum, indicating that for small dimensions it is feasible to start the algorithm with less than 10 observations per dimension on each response.

Size	Type	y_1 Code Runs				y_2 Code Runs			
		Min	Median	Mean	Max	Min	Median	Mean	Max
18	Same	26.0	32.0	33.3	48.0	20.0	26.0	30.6	51.0
18	Mix	22.0	33.0	32.9	47.0	19.0	28.0	30.5	53.0
10	Same	19.0	28.0	27.9	43.0	11.0	19.0	25.3	46.0
10	Mix	18.0	26.0	28.2	39.0	12.0	22.0	25.2	46.0
Size	Type	Algorithm Iterations				$(x_c^* - x_c^{true})^2$			
		Min	Median	Mean	Max	Min	Median	Mean	Max
18	Same	10.0	29.0	27.9	45.0	0.19e-04	2.19e-04	4.24e-04	27.3e-04
18	Mix	10.0	23.0	27.4	45.0	0.01e-04	1.10e-04	6.48e-04	61.6e-04
10	Same	13.0	35.0	33.2	45.0	0.01e-04	2.43e-04	4.33e-04	20.6e-04
10	Mix	11.0	31.0	33.3	45.0	0.19e-04	2.34e-04	17.4e-04	304e-04

Table 3.1: Summary statistics for 25 random objectives

Figure 3.7 displays boxplots of the total number of computer code runs for algorithm runs that used the moving average stopping criterion. We see that the 10-point starting designs require fewer total computer runs than the 18-point starting designs. On the other hand, comparing the initial design that observes both responses at the same sites with the "mixed" design of the same size there doesn't appear to be an

Size	Type	y_1 Code Runs				y_2 Code Runs			
		Min	Median	Mean	Max	Min	Median	Mean	Max
18	Same	26.0	32.0	32.8	42.0	20.0	25.0	25.0	41.0
18	Mix	25.0	35.0	33.6	47.0	19.0	22.0	24.4	42.0
10	Same	19.0	29.0	29.4	43.0	11.0	18.0	18.3	26.0
10	Mix	18.0	30.0	29.0	39.0	12.0	19.0	18.8	30.0
Size	Type	Algorithm Iterations				$(x_c^* - x_c^{true})^2$			
		Min	Median	Mean	Max	Min	Median	Mean	Max
18	Same	10.0	20.0	21.8	37.0	0.57e-04	2.36e-04	4.91e-04	27.3e-04
18	Mix	10.0	20.0	21.9	41.0	0.01e-04	0.90e-04	5.37e-04	18.3e-04
10	Same	13.0	27.0	27.6	42.0	0.01e-04	2.43e-04	3.39e-04	11.5e-04
10	Mix	12.0	27.0	27.8	43.0	0.67e-04	4.08e-04	7.16e-04	25.3e-04

Table 3.2: Summary statistics for algorithm runs using moving average stopping criterion

advantage to either design *type*. Heuristically, it would seem that observing the responses at different sites might offer some improvement for prediction since this allows more "coverage" of the input space. On the other hand, it may be necessary to have the observations at the same sites so as to gather as much information as possible about how the two responses are related. In general, the "mixed" design is a broader class of designs as it includes the possibility of observing the two responses at all the same sites.

3.4 Discussion

This algorithm improves the sequential design algorithm of Williams et al. (2000c) by refining the improvement criterion and by allowing the algorithm to observe $y_1(\cdot)$ and $y_2(\cdot)$ at different sites. It is of greatest use when $y_1(\cdot)$ and $y_2(\cdot)$ are both computationally expensive and only a few calculations of $y_2(\cdot)$ are necessary to establish

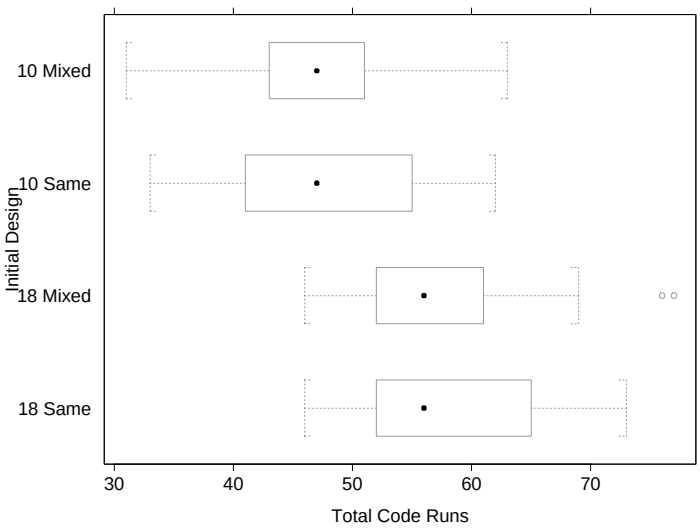


Figure 3.7: Boxplots of total number of computer code runs for those cases using the moving average stopping criterion.

that the optimum of $\mu_1(\cdot)$ is clearly within the $\mu_2(\cdot)$ feasible region. In such a case, all further computing effort is best spent identifying $\mu_1(\cdot)$.

The algorithm calls for several optimization routines. The first is in Step 2 where the posterior mode of $\boldsymbol{\gamma}$ must be determined. This optimization can be very time consuming. Williams et al. (2000c) suggest several approaches for decreasing the computation time involved in this step. One idea is to only update the $\boldsymbol{\gamma}$ estimates after groups of points have been added, particularly once a sufficiently large number of points have been computed. Further, as the algorithm nears the stopping criterion the $\boldsymbol{\gamma}$ estimates become more and more stable, and so an approach that skips Step 2 for larger groups of points might be appropriate. For the 25 $y_1(\cdot)$ draws in the

previous section, we update $\boldsymbol{\gamma}$ every other time after 25 $y_j(\cdot)$ values have been added, and every 5 times after 35 $y_j(\cdot)$ evaluations.

There are other ways for decreasing the computing time required to run the algorithm. In the examples of Section 3.3, we found that when the constrained optimum is on a boundary of the feasible region the algorithm took the maximum number of runs before stopping. In this case it makes intuitive sense to allow the algorithm to take an observation on both $y_1(\cdot)$ and $y_2(\cdot)$. When the next $\boldsymbol{x}_c$ chosen appears to be on the boundary of the feasible region, we should evaluate both of the computer codes at the next point and proceed with the algorithm. Perhaps by using $P(M_2(\boldsymbol{x}_c) \leq U \mid \boldsymbol{Y}^d)$, a criterion for deciding when $\boldsymbol{x}_c$ is on the boundary of the feasible region can be established.

The criterion in Section 3.3 is but one approach to stopping the algorithm. Another idea is to predict the global optimum at each stage and stop when the improvement in this prediction (and its standard error) are "small", or to stop when the change in the values for the global optimum are "small".

This algorithm can be extended to incorporate more complicated situations. First, measurement error can be added to the observations and accounted for in the algorithm. We considered the case where observations were measured without error, and so only one observation at each input site was taken. When measurement error is present, multiple observations need to be taken at a single input site to obtain information about the magnitude of the measurement error. In this situation it is not clear how to best form the initial design of the experiment. Second, other types of cokriging models or cross-correlation structures might prove more useful in modeling

the multivariate data. For example, Ver Hoef and Barry (1998) construct valid cross-variogram models by modeling the spatial data as a moving average of a white noise random process. Empirical work is needed to investigate if the Section 3.2 algorithm would be computationally feasible for such a $(Y_1(\cdot), Y_2(\cdot))$ model and its performance relative to that presented here.

A final area for further research is in the construction of the initial design for the two responses. First, how should observations be allocated to the initial design and the following sequential design strategy when there is a limited number of total runs allowed? In the examples above, we saw that starting designs with as few as 5 observations per input dimension can lead to accurate results, although the Jones et al. (1998) rule of 10 observations per input dimension, when feasible, seems a reasonable approach. Second, how should the points be chosen for the initial design? In Section 3.3 we present one heuristic for constructing an initial design for the two responses that has the property that the design for each $y_i(\cdot)$ is a Latin hypercube and each $y_i(\cdot)$ is observed the same number of times. For the case where we observe each process the same number of times, there are many designs that have the above property and there may be an optimal choice of one of these. Also, initial designs allowing for different numbers of observations on the two responses need to be investigated. For example, the fast code/slow code situation of Kennedy and O'Hagan (2000) is one example where this type of design may be a more efficient use of computing time.

CHAPTER 4

EXPLORATORY METHODS OF ASSESSING ROBUSTNESS

In this chapter we present a prototype of an exploratory method for assessing robustness. The method is illustrated by an example of a computer experiment involving the design of an acetabular cup in a total hip replacement. Determining the optimal cup geometry (the design of the acetabular cup) that promotes the best "fixation" of the cup, or minimizes the incidence of cup loosening, is an important research question. Interfacial gaps between the acetabular cup and the periacetabular bone, the pelvic bone into which the cup is implanted, have been identified as a factor in cup loosening. The presence of these gaps inhibits bony ingrowth and allows particulate debris to penetrate the joint space, thereby inhibiting "fixation" of the acetabular cup.

Ong et al. (2002) conducted a computer experiment to analyze a class of acetabular cup designs using a three-dimensional finite element model of a cadaver pelvis undergoing cup insertion followed by peak gait joint loading. At the end of the loading cycle, the cups were evaluated for the amount of total and rim bone-implant apposition (total contact area and rim contact area), and the change in volume of the periprosthetic joint space (change in gap volume). These three responses are affected

by two control (engineering) variables and three environmental variables. The control variables, cup polar diameter and cup equatorial diameter, describe the geometry of the cup. The environmental variables, peak joint load magnitude, peak joint load direction and cup penetration into the acetabulum, are subject specific and have values that follow a probability distribution that characterizes some population of interest. The objective of the experiment was to determine the cup geometry (control variables) that minimizes the mean change in periprosthetic gap volume, maximizes the mean cup-acetabulum total contact area, and maximizes the mean cup-acetabulum rim contact area. The means are taken over the distribution of the environmental variables. Ideally, a single cup geometry will accomplish these goals, however, as with any multi-objective experiment, different goals often lead to different solutions.

To accomplish these goals, we fixed a nominal distribution for the environmental variables and performed a heuristic sequential optimization to determine the best setting of the control variables for each of the three responses. This was accomplished by minimizing or maximizing a predictor of each response based on a small training sample of computed responses. Since the optimal setting of the control variables depends on the assumed *nominal* distribution of the environmental variables, we also performed an exploratory robustness analysis by perturbing the environmental variable distributions and then determining the optimal cup geometry. In other words, we asked the question: is the cup geometry that is optimal for the nominal $\boldsymbol{X}_e$ distribution also optimal for other $\boldsymbol{X}_e$ distributions? Comparing the optimal cup geometry based on several alternative $\boldsymbol{X}_e$ distributions to that based on the nominal environmental variable distribution gives a sense of how robust the cup design is to misspecification of the environmental variable distribution. In Section 4.1 we describe

each input variable and the nominal distribution of the environmental variables. Section 4.2 discusses the heuristic sequential scheme. Section 4.3 explores the robustness of the chosen design to perturbations in the nominal distribution of the environmental variables.

4.1 Input Variables

The acetabular cup computer code requires five input variables to produce the three responses.

1. Cup equatorial diameter (control variable)
2. Cup eccentricity (control variable)
3. Displacement from nominal penetration (surgical/environmental variable)
4. Joint load magnitude (environmental variable)
5. Joint load direction (environmental variable)

As seen in Figure 4.1 the two control variables describe the geometry (design) of the acetabular cup. The cup *equatorial diameter* is restricted to values of 56 mm, 57 mm, 58 mm, 59 mm, and 60 mm. Cup *eccentricity*, the difference between the cup equatorial diameter and the cup polar diameter, is restricted to the values 0 mm, 1 mm, 2 mm, and 3 mm, so that the polar diameter is always less than or equal to the equatorial diameter.

The surgical variable *displacement*, which is treated as an environmental variable in these analyses, describes the variability of the insertion of the cup into the acetabulum. Ideally, the surgeon is able to exactly achieve nominal insertion, defined as a

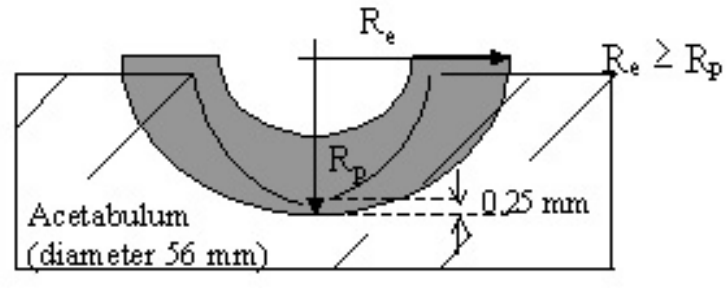


Figure 4.1: Cup geometry descriptors: polar (R_p) and equatorial (R_e) radii. Nominal insertion of cup allowed 0.25 mm penetration of cup into the acetabulum.

0.25 mm penetration of the cup dome into the acetabulum (see Figure 4.1), but in practice, perfect insertion is rarely achieved. The nominal distribution of displacement is assumed to be a discretized normal with mean 0 and standard deviation 0.3 mm (see Figure 4.3).

The final two environmental variables, peak *joint load magnitude* and peak *joint load direction*, (shown in Figure 4.2) have distributions that are taken from the Biomechanics literature (Bergmann et al. (1993), Kotzar et al. (1991), Crowninshield et al. (1978), and Pedersen et al. (1997)). The load directions in these studies ranged from a polar angle, the angle made by the load direction and the polar axis of the cup, of 27.51 degrees to 39.77 degrees. A discretized normal distribution was fit to the load directions to obtain a nominal distribution with mean polar angle of 34 degrees and standard deviation 2.45 degrees (see Figure 4.3). The load magnitudes ranged from 2.04 to 5.31 times body weight. We fit a discretized version of a constant times a chi-squared distribution to these magnitudes so that 1% of the distribution was less

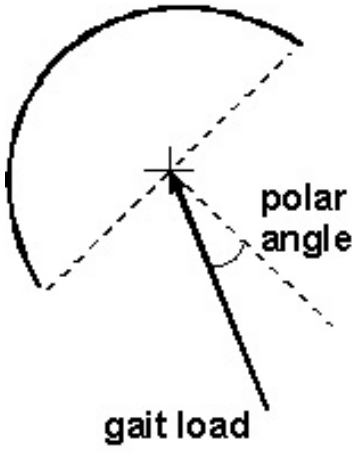


Figure 4.2: Environmental variables: joint load magnitude and load direction.

than 2.04 BW and 1% was above 5.31 BW, resulting in a constant of 0.0716 and 47.1 degrees of freedom (see Figure 4.3).

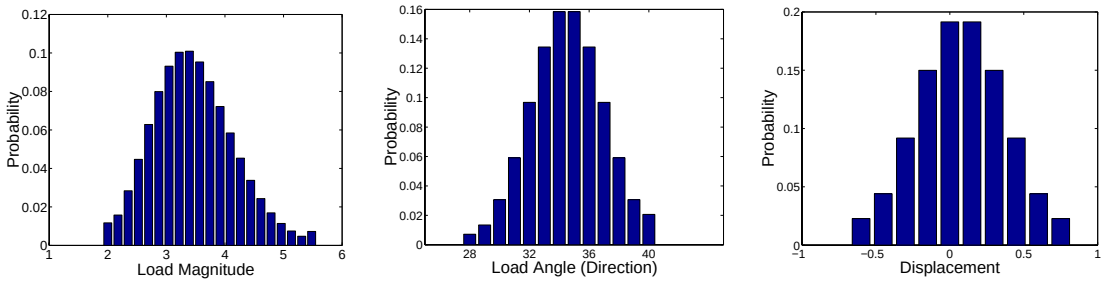


Figure 4.3: Plots of the discretized nominal environmental variable distributions.

4.2 Sequential Statistical Optimization

For each of the three responses, the following statistical optimization methodology was performed. Suppose $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is the selected response; it is a function of both control, $\boldsymbol{x}_c$, and environmental variables, $\boldsymbol{x}_e$. If $g(\cdot)$ is the distribution of the environmental variables that incorporates the variation in the environmental variables over the population of interest, the optimization goal is to minimize or maximize the mean of $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ over this distribution, i.e. to minimize or maximize

$$\mu(\boldsymbol{x}_c) = \int y(\boldsymbol{x}_c, \boldsymbol{x}_e) g(\boldsymbol{x}_e) d\boldsymbol{x}_e. \tag{4.1}$$

The finite element models used to compute the responses $y(\cdot)$ are sufficiently complicated so that up to 24 hours of CPU time are required for each run. This precludes the use of traditional numerical techniques to perform optimization. The statistical methodology involves constructing an inexpensive (rapidly computable) predictor $\hat{y}(\boldsymbol{x}_c, \boldsymbol{x}_e)$ of the true response $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$. Given $g(\boldsymbol{x}_e)$ and the predictor $\hat{y}(\boldsymbol{x}_c, \boldsymbol{x}_e)$, a predictor of the mean function can be defined as

$$\hat{\mu}(\boldsymbol{x}_c) = \int \hat{y}(\boldsymbol{x}_c, \boldsymbol{x}_e) g(\boldsymbol{x}_e) d\boldsymbol{x}_e,$$

and can be optimized using traditional numerical techniques.

The predictor $\hat{y}(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is determined from the computed response, $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$, at a set of "training" sites located in the control and environmental variable space. For this study, the training sites are chosen using a two stage procedure. In the first stage, 15 points are chosen as a Latin hypercube design that filled the 5-dimensional design $\times$ environmental space in that the points approximately maximize the average inter-point distance among all 15 sites. The second stage design consists of 10 additional

points chosen as in Stage 1 and concentrated in the region of the optimal $\boldsymbol{x}_c$ based on $\hat{y}(\cdot)$ computed from the first stage training data. For each of the three responses, the optimal $\boldsymbol{x}_c$ from Stage 1 appeared to be in the same region of the input space, making it possible to use a common Stage 2 design. The software program ACED (Welch (1985)) was used to generate these designs.

To construct the final predictor, $\hat{y}(\boldsymbol{x}_c, \boldsymbol{x}_e)$, we adopt the Bayesian viewpoint and model the true response as a stochastic process or random function. The prior model for each of the true responses, $y(\cdot)$, is the random function

$$Y(\boldsymbol{x}) = \beta_0 + Z(\boldsymbol{x}), \tag{4.2}$$

where $Z(\cdot)$ is a covariance stationary Gaussian stochastic process having mean zero, correlation function $R(\cdot)$, and unknown variance $\tau^2 > 0$. The model is completed by specifying a positive definite correlation function $R(\cdot)$. For this analysis we let

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{5} \exp\big(- \theta_i |x_{1,i} - x_{2,i}|^{\alpha_i}\big),$$

where $\theta_i > 0$ and $0 < \alpha_i \leq 2$. The empirical best linear unbiased predictor (EBLUP) of $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ based on our training sample is then used for prediction purposes. See Chapter 1 for more details of the modeling and analysis of computer experiments data.

4.2.1 Stage 1 Predictor

The control variable portion of the 15-point Stage 1 design is displayed in Figure 4.4. Based on this 15 point initial design and Model (4.2), the predictors of the mean response as functions of the two control variables, equatorial diameter and eccentricity, are constructed for each of the three responses and plotted in Figure 4.5.

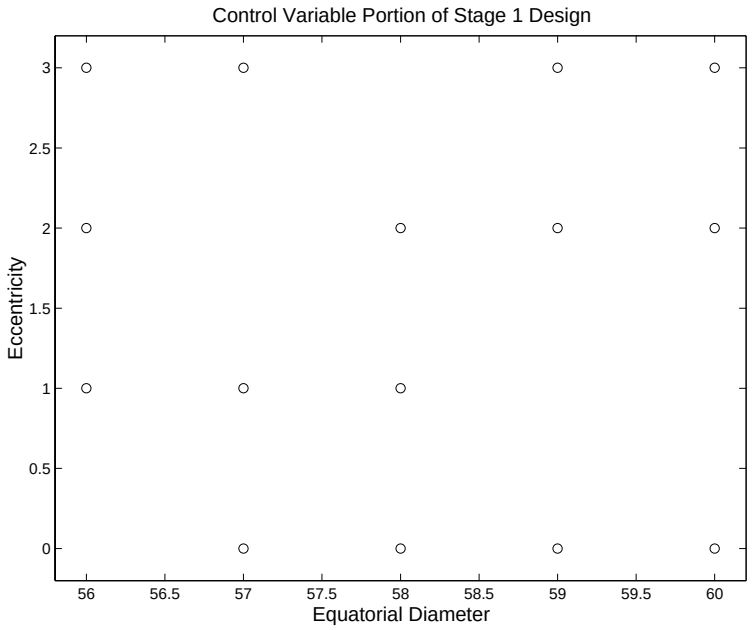


Figure 4.4: Plot of initial 15 point design projected into the control variable space.

The mean response is obtained by averaging over the nominal environmental variable distributions described in Section 4.1. Note that the 56, 57, or 58 mm hemispherical cups (i.e. equatorial diameter $= 56, 57$, or 58, and eccentricity $= 0$) minimize or are close to minimizing the mean change in gap volume (CGV), maximizing the mean total contact area (TCA), and maximizing the mean rim contact area (RCA). So, the 10 additional training sites were restricted to equatorial diameters of 56, 57, or 58 mm, and eccentricities of 0 or 1 mm.

4.2.2 Stage 2 Predictor and Optimal Cup Geometry

Combining the 15 original training data with the additional 10 points, Model (4.2) is refit to obtain the Stage 2 predictors of the mean response for all three outcomes. Figure 4.6 and the corresponding values in Table 4.1 show that the 58

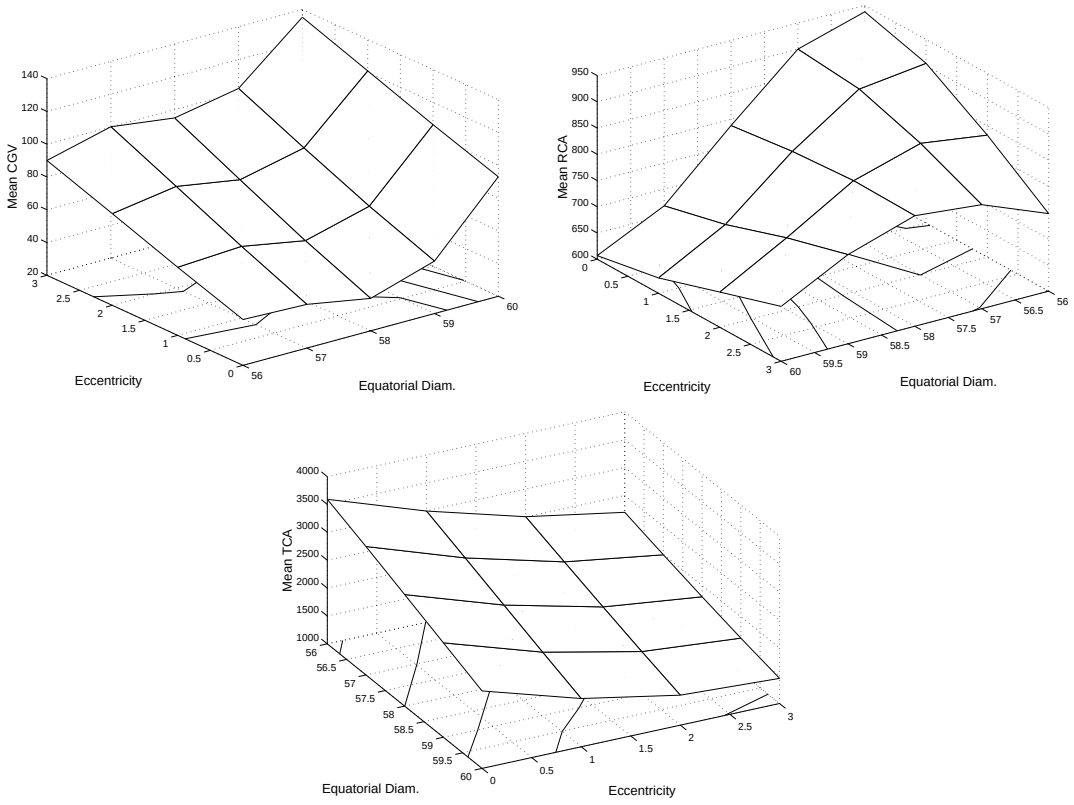


Figure 4.5: Plots of predicted mean responses (averaging over $\boldsymbol{X}_e$ distribution) based on the 15-point training data (Stage 1).

mm hemispherical design minimizes the mean change in gap volume, and the 56 mm hemispherical design maximizes both the rim contact area and the total contact area. Since minimizing the mean change in gap volume is the primary concern, and the 58 mm hemispherical cup is "close" to optimal for both rim contact area and total contact area, it is determined that, based on the nominal $\boldsymbol{X}_e$ distribution, the 58 mm hemispherical cup is the optimal cup geometry.

We can also use the predictor $\hat{y}(\cdot)$ to assess the effect of each variable on the response by calculation of main effects and interaction effects. Introduced by Welch et

Control Variables		Responses		
Equatorial Diameter	Eccentricity	Change in Gap Volume	Total Contact Area	Rim Contact Area
56	0	87.2212	**3909.75**	**1000.600**
56	1	74.5597	3305.28	887.391
56	2	76.2167	2564.83	817.638
56	3	82.8377	2153.42	709.957
57	0	55.9476	3675.73	919.637
57	1	64.9652	2511.70	892.695
57	2	70.9466	2046.51	892.432
57	3	92.0168	1797.38	881.530
58	0	**34.0771**	3193.07	806.240
58	1	59.4245	2173.17	805.483
58	2	66.4370	1871.50	818.326
58	3	101.8417	1679.85	821.672
59	0	44.5186	2495.75	662.704
59	1	73.5903	1859.57	704.248
59	2	78.3555	1671.41	751.999
59	3	121.7628	1543.18	777.702
60	0	86.9900	1932.77	579.316
60	1	109.6684	1705.57	606.922
60	2	111.4190	1628.96	651.382
60	3	153.3120	1490.21	700.159

Table 4.1: Predicted mean responses (averaging over $\boldsymbol{X}_e$ distribution) based on 25 point training data (Stage 2 results).

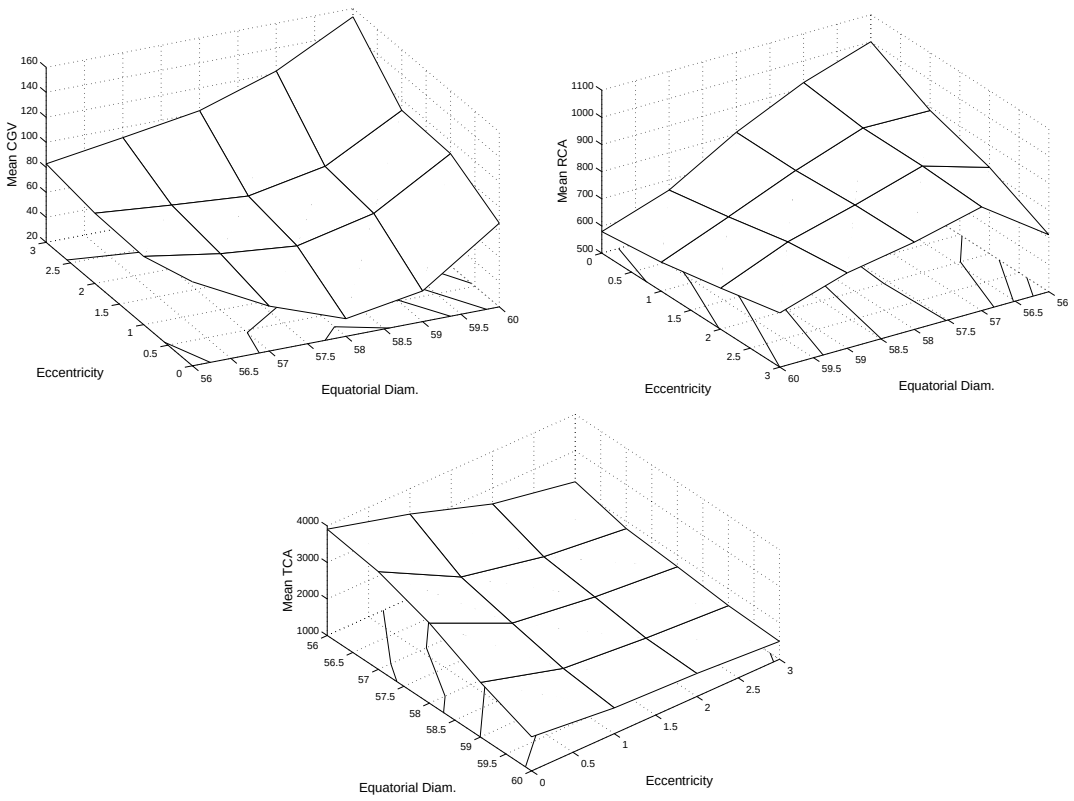


Figure 4.6: Plots of predicted mean responses (averaging over $\boldsymbol{X}_e$ distribution) based on the 25-point training data (Stage 2).

al. (1992) , and seen in Jones et al. (1998), Table 4.2 lists the percentage contribution of each input variable and each two-way interaction to total variability in each of the three responses. It appears that the main contributors to variability in each response are the control variables and their interaction. The environmental variables, on the other hand, are relatively "inactive," and do not seem to significantly affect the response.

Variable	Change in Gap Volume	Total Contact Area	Rim Contact Area
Load Magnitude (MAG)	1.76%	0.93%	0.22%
Load Direction (DIR)	0.01%	0.00%	0.00%
Displacement (DISP)	0.00%	0.80%	0.03%
Equ. Diameter (DIAM)	38.91%	40.31%	73.30%
Eccentricity (ECC)	39.97%	47.36%	0.32%
MAG * DIR	0.00%	0.00%	0.00%
MAG * DISP	0.00%	0.00%	0.00%
MAG * DIAM	3.51%	0.02%	0.30%
MAG * ECC	0.80%	0.09%	0.00%
DIR * DISP	0.00%	0.00%	0.00%
DIR * DIAM	0.47%	0.00%	0.00%
DIR * ECC	0.01%	0.00%	0.00%
DISP * DIAM	0.00%	0.25%	0.52%
DISP * ECC	0.00%	0.54%	0.00%
DIAM * ECC	14.51%	9.56%	25.26%

Table 4.2: Percentage contribution of main effects and interactions to the total variability in each response $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ based on 25-point predictors.

4.3 Robustness of Optima to Misspecification of $F(\boldsymbol{x}_e)$

The sequential optimization scheme and the results obtained in Section 4.2 are specific to the nominal environmental variable distributions described in Section 4.1. Now assume these distributions are not known with certainty; we wish to determine the sensitivity of the design of the acetabular cup to perturbations in the distribution of the environmental variables. In the above analysis, the environmental variables did not appear to be "active" for any of the three responses, contributing a small portion to the total variability in the response. This suggests that perturbing the distribution of the environmental variables will show little affect on the optimizing $\boldsymbol{x}_c$. In this section, we investigate the effect of perturbing the environmental variable distributions by answering three questions (for each of the three responses):

1. Does the optimal cup geometry change as we change the distribution of the environmental variables?

2. How much does the value of the mean response change at the nominal optima as we change the distribution of the environmental variables?

3. If the optimal cup geometry changes, by how much does the response for the new cup design beat the response for the old cup design?

To answer these questions we first describe the alternative environmental variable distributions that were considered.

4.3.1 Alternative Environmental Variable Distributions

The nominal distribution for *peak joint load magnitude* was a discretized constant times a chi square distribution. Prior information suggests that the minimum possible value for the load magnitude is approximately 1.8 times body weight (BW) and the maximum is approximately 5.5 times body weight. So, the alternative distributions for the peak joint load are restricted to this interval. Figure 4.7 displays the nominal distribution (solid line) along with several proposed alternative distributions. These alternative distributions have weights that follow a Beta(α, 5) distribution scaled to the interval $(1.8, 5.5)$. The entire class of distributions is obtained by allowing α to take values in $\{0.2, 0.25, ..., 0.75, 0.8\}$, corresponding to mean load magnitudes between 2.54 and 4.76. This gives a total of 13 alternative distributions for the peak joint load magnitude.

The nominal distribution for the *peak loading direction* (polar angle) was normal with a mean of 34 degrees and standard deviation of 2.45 degrees. To perturb this

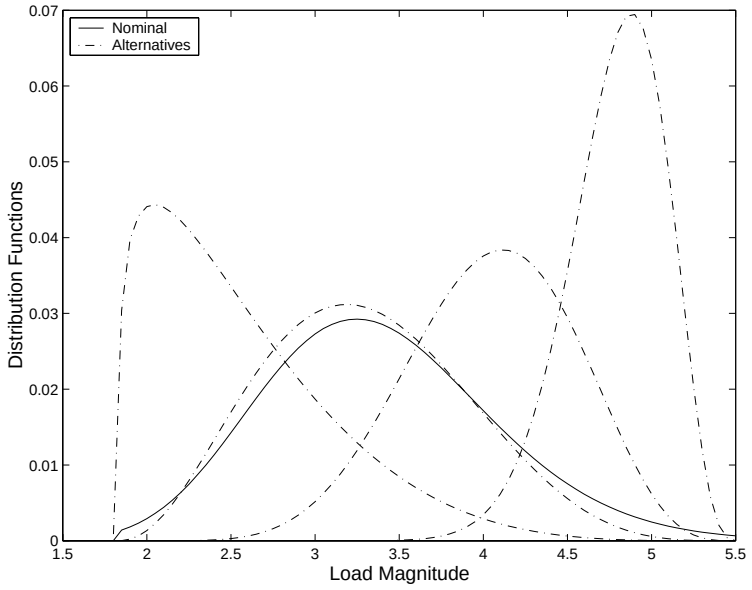


Figure 4.7: Nominal (−) and four alternative (− · −·) load magnitude distributions

distribution we allow the mean to take values in $\{29.1, 30.1, 31.1, ..., 39.1\}$ giving a total of 11 alternative distributions. Figure 4.8 displays the nominal distribution (solid line) of loading direction and the two extreme distributions (dashed line) corresponding to means of 29.1 and 39.1 degrees.

Finally, the nominal distribution for *displacement* (the deviation from the nominal cup penetration) was normal with a mean of 0 mm and a standard deviation of 0.3 mm. To perturb this distribution we allow the mean to take values in $\{-0.25, -0.20, ..., 0.20, 0.25\}$, again yielding 11 alternative distributions. Figure 4.9 displays the nominal distribution (solid line) and the two extreme distributions (dashed line) corresponding to means of −0.25 mm and 0.25 mm.

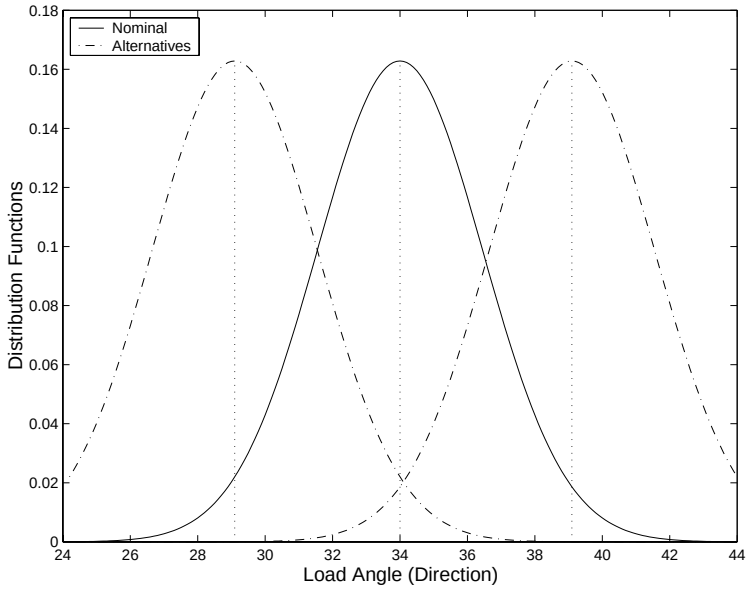


Figure 4.8: Nominal $(-)$ and the extreme alternative $(- \cdot -\cdot)$ load direction distributions

4.3.2 Results of Varying Environmental Distributions

Because some of the alternative distributions considered above contain support points outside of the initial parameter space (and thus outside of the 25-point training sample), we added 8 more training points with inputs located in the extreme regions of the input space. For each outcome, the predictor of $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ based on the 33 training sites is used to determine the optimal value of $\boldsymbol{x}_c$ (i.e. the optimal cup geometry) for each of the $1573 = 13 * 11 * 11$ (13 alternative distributions for load magnitude and 11 for both load direction and displacement) alternative environmental variable distributions. Table 4.3 lists the 33 training points (15 from Stage one, 10 from Stage 2 and 8 additional points) and their corresponding responses.

Control Variables		Evironmental Variables			Responses		
Diameter	Eccentricity	Load	Direction	Displacement	CGV	TCA	RCA
56	3	2.8710	30.8602	0.05037	76.2	2121.3	722.2
60	0	3.0453	34.4113	-0.38447	95.0	1799.3	574.6
58	0	2.5226	35.7834	0.29023	37.4	3134.1	821.2
59	0	3.5679	29.5069	0	42.3	2499.7	661.3
60	2	4.0905	31.6298	-0.10221	102.1	1662.7	648.2
56	1	2.3484	32.7152	-0.21837	55.2	3118.9	909.7
58	2	3.2195	38.4931	-0.29023	71.3	1762.7	814.3
57	1	3.2195	36.3702	-0.05037	66.1	2445.8	895.7
56	2	3.7421	35.2848	0.15732	82.5	2583.1	815.6
59	2	2.8710	33.1653	0.38447	79.9	1710.1	764.5
57	0	4.4390	33.5887	0.10221	47.8	3869.0	922.1
57	3	3.7421	32.2166	-0.55017	99.7	1796.9	911.0
60	3	3.3937	37.1398	0.21837	161.4	1465.4	703.0
58	1	3.9163	34.0000	0.55017	58.7	2739.2	813.4
59	3	4.7874	34.8347	-0.15732	112.0	1606.7	766.9
57	0	4.6132	33.0560	0.03770	50.6	3808.2	924.8
58	0	2.6968	31.4607	-0.11560	38.2	3095.2	810.7
56	1	3.5679	32.3475	-0.49346	83.0	3244.5	868.1
58	0	3.7421	33.6921	0.49346	32.3	3304.0	814.2
56	1	4.2647	35.6525	-0.03770	75.9	3460.5	865.2
58	1	3.2195	38.0299	0.11560	64.5	2195.6	810.2
56	0	3.0453	34.9440	0.20235	87.3	3860.6	1019.3
57	1	2.3484	34.3079	-0.20235	49.1	2223.0	900.5
57	0	3.3937	36.5393	-0.31093	58.9	3641.7	928.5
57	1	3.9163	29.9701	0.31093	73.5	2739.6	878.2
58	3	5.5722	43.5000	1.10000	143.6	1478.4	751.0
56	3	3.2539	33.6602	-1.10000	82.0	1880.8	772.7
58	1	4.5828	25.5000	-0.97590	89.8	1958.5	820.0
60	2	1.9789	26.6011	-0.02380	105.1	1375.8	582.1
60	0	1.8000	42.0325	0.17970	102.5	1246.1	526.6
56	0	4.0819	40.5174	-0.99930	40.5	4568.2	1019.3
59	3	5.8000	35.4695	-0.88430	131.8	1574.1	778.0
59	1	3.8502	43.5000	0.74970	61.9	2294.1	719.5

Table 4.3: Training data (33 points) from actabular cup computer code.

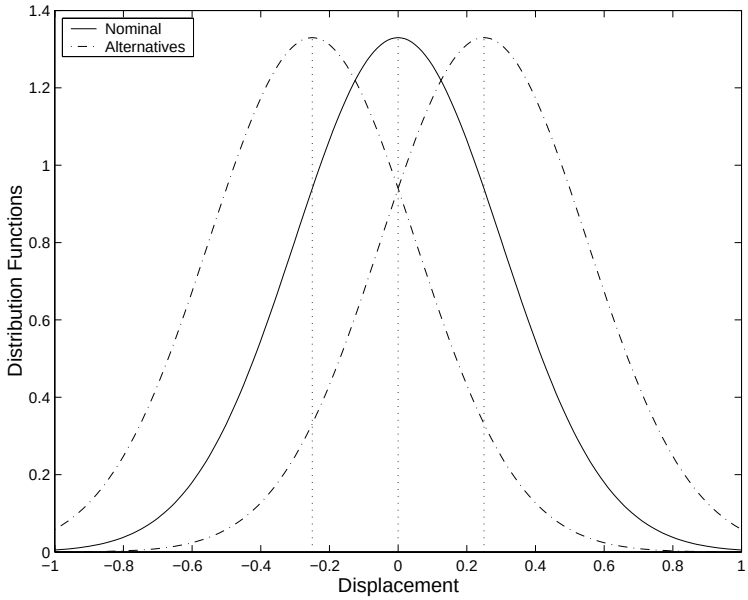


Figure 4.9: Nominal (−) and the extreme alternative (− · −·) displacement distributions

As in the case of the nominal distribution, for *every* alternative environmental distribution the maximum mean *total contact area* was attained for a 56 mm hemispherical cup. This is the ideal situation and indicates that the optimal cup design is robust to misspecification of the environmental variable distribution, as was indicated from the ANOVA results in Table 4.2 that showed little variation due to the environmental variables. Figure 4.10 displays the predicted maximum total contact area as a function of the three values governing the alternative $\boldsymbol{X}_e$ distributions: mean of the load magnitude distribution, mean of the load direction distribution, and mean of the displacement distribution. The maximum total contact area as a function of these means ranges from 3760 mm^2 to 4368 mm^2 with the maximum total contact area being 4024 mm^2 for the nominal distribution.

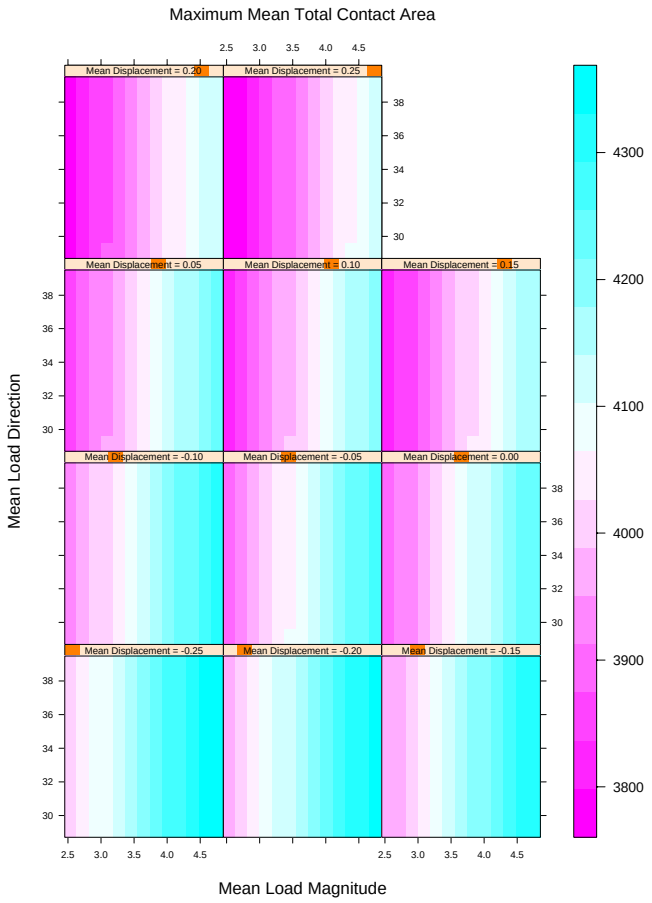


Figure 4.10: Predicted maximum total contact area as a function of the means of the alternative environmental variable distributions.

For the *rim contact area*, a 56 mm hemispherical cup maximized the mean rim contact area for the nominal distribution and for most (1320 out of 1573) of the alternative environmental distributions. For the remaining alternative distributions a 57 mm hemispherical cup maximized the mean rim contact area. The distributions for which the 57 mm hemispherical cup was optimal tended to place more weight on the higher end of the load magnitude range. Figure 4.11 displays the predicted maximum

rim contact area (left panel) and the equatorial diameter producing this maximum (right panel) as functions of the three values governing the alternative $\boldsymbol{X}_e$ distributions. The maximum mean rim contact area ranges from 910 mm^2 to 1083 mm^2 with the maximum mean rim contact area being 989 mm^2 for the nominal distribution. From Figure 4.11 we see that the distribution for *displacement* does not appear to affect the results, whereas, there is an obvious interaction effect between the *load direction* and *load magnitude* distributions, with smaller mean load directions (polar angles) and larger mean load magnitudes being associated with smaller maximum mean rim contact areas.

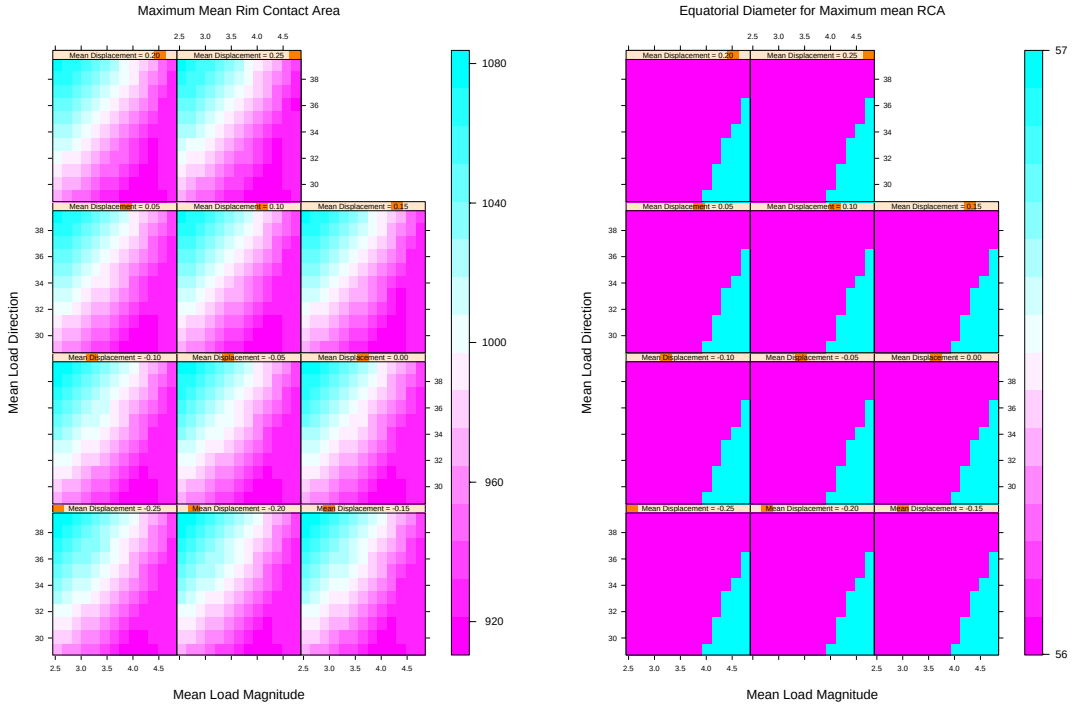


Figure 4.11: Predicted maximum rim contact area (left panel) and the equatorial diameter (right panel) that produces this maximum as functions of the means of the alternative environmental variable distributions.

The 58 mm hemispherical cup minimized the mean *change in gap volume* for the nominal environmental variable distribution and for most (1183 out of 1573) of the alternative distributions. A 56 mm hemispherical cup minimized the mean change in gap volume for the remaining 390 alternative distributions which again tended to place more weight on the higher end of the load magnitude range. Figure 4.12 displays the predicted minimum change in gap volume (left panel) and the equatorial diameter corresponding to this minimum (right panel) as functions of the three values governing the alternative $\boldsymbol{X}_e$ distributions. The range of the minimum mean change in gap volume was 36 mm^3 to 48 mm^3 with the minimum mean change in gap volume being 39 mm^3 for the nominal distribution. In Figure 4.12 we again see that the distribution for *displacement* appears to have little affect on the results, whereas, there is an interaction effect between the *load direction* and *load magnitude* distributions.

4.4 Discussion

In this case study we see the importance of considering the sensitivity of the optimal solution to the assumed nominal distribution of the environmental variables. We would like to choose the design of the acetabular cup so that the response is "insensitive" to the distribution of the environmental variables. We would call such a design a "robust" design. There are several reasonable methods of defining insensitivity, and hence robust design. We consider several below, and define the concept of robustness more formally in Chapter 5.

For the total contact area response the robust cup design is a 56 mm hemispherical cup, since for all alternative environmental variable distributions this cup geometry

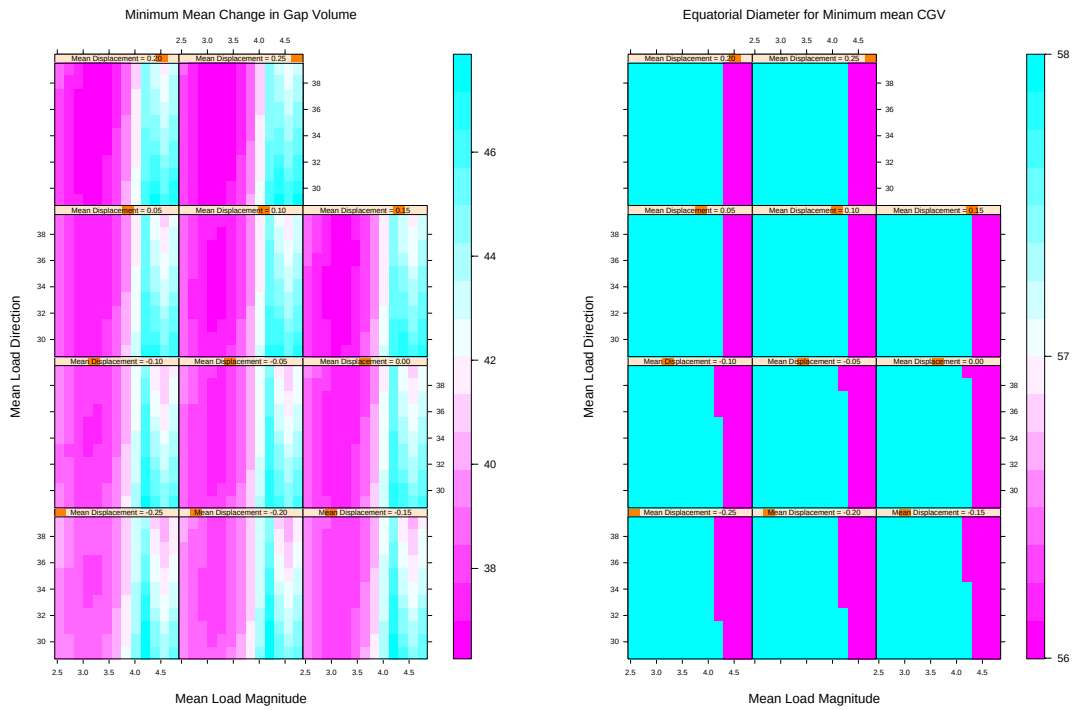


Figure 4.12: Predicted minimum change in gap volume (left panel) and the equatorial diameter producing this minimum (right panel) as functions of the means of the alternative environmental variable distributions.

minimizes the mean total contact area, averaging over the distribution of the environmental variables. However, for both the rim contact area and change in gap volume responses, the optimal cup geometry depends on the assumed distribution of the environmental variables. A robust choice of cup geometry is more difficult in this situation. Choosing the design that is optimal the majority of the time, or choosing the design that is never too suboptimal may be appropiate strategies. Chapter 5 discusses several robustness considerations and outlines some sequential strategies for finding robust control variable settings.

In addition to choosing a "robust" cup geometry, we must also consider the impact of choosing a given design on all *three* responses. In the above example, univariate modeling was used to compute the optimal setting for each response separately. This can and does lead to different choices for the cup design depending on which response is optimized. The decision between the 56, 57, and 58 mm hemispherical cups is difficult to make since we have competing objectives, minimizing change in gap volume and maximizing total contact area and rim contact area. The 56 mm hemispherical cup maximizes the total contact area regardless of the $\boldsymbol{X}_e$ distribution, and maximizes the rim contact area for most of the $\boldsymbol{X}_e$ distributions. However, the 58 mm hemispherical cup minimizes the change in gap volume for most $\boldsymbol{X}_e$ distributions. A more comprehensive approach that models all three responses together, and defines an optimization formulation so as to lead to a single choice of optimal cup design may be more appropriate. For example, in Chapter 3 the bivariate response setting was investigated, and the optimal control variable value was chosen so as to minimize the first response with a constraint on the second response. A similar approach may be useful in the three-response setting.

CHAPTER 5

OPTIMAL ROBUST PARAMETER DESIGN

We again consider the computer experiments setting where there are two types of input variables: control variables and environmental variables. Control variables are those that can be set by the product designer and environmental variables are uncontrollable but have values that follow some probability distribution. Williams (2000b), and Williams, Santner, and Notz (2000a) investigate this setting and present sequential algorithms for finding the setting of the control variables that minimizes (or maximizes) the mean response taken over the distribution of the environmental variables. This choice of control variables may be "sensitive" to the nominal (chosen) distribution of the environmental variables and "sensitive" to the value of the environmental variables. A more "robust" choice of control variables may be obtainable by considering other characteristics of the induced distribution (induced by the distribution of the environmental variables) of the response at given control variable values.

In this chapter we discuss the goal of finding a "robust" value for the control variables. We review different methods of defining robustness and focus on finding a set of control variables at which the response is "insensitive" to the value of the environmental variables. Such a choice ensures that the mean response is "insensitive"

to perturbations of the nominal environmental variable distribution. We present a sequential strategy to select the inputs at which to observe the response to determine a robust setting of the control variables.

5.1 Concepts of Robustness

Let $y(\cdot)$ denote the true response function, the computer code output, and denote the input by $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{x}_e)$ where $\boldsymbol{x}_c$ is the vector of control variables and $\boldsymbol{x}_e$ is the vector of environmental variables with $\boldsymbol{X}_e \sim F(\boldsymbol{x_e})$. Our goal is to identify a "robust" setting of $\boldsymbol{x}_c$ by examining the properties of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$, a random variable with a distribution that is induced by $\boldsymbol{X}_e$. Note that the distribution of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ will change as $\boldsymbol{x}_c$ changes and as the $\boldsymbol{X}_e$ distribution, $F(\cdot)$, changes. In this section we define several types of robustness and compare these concepts using a simple example.

When $F(\cdot)$ is known, we typically focus attention on some aspect of the distribution of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$. For example, Williams (2000b) and Williams, Santner and Notz (2000a) propose sequential algorithms for minimizing the mean response

$$\mu_F(\boldsymbol{x}_c) = E_F[y(\boldsymbol{x}_c, \boldsymbol{X}_e)] \tag{5.1}$$

as a function of the control variables. However, as is well recognized in the quality control literature, simply minimizing (or maximizing) the mean $\mu_F(\boldsymbol{x}_c)$ can lead to a choice of $\boldsymbol{x}_c$ for which the variability in $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ across the range of $\boldsymbol{x}_e$ is unacceptably large. In other words, it may be possible to find a setting for $\boldsymbol{x}_c$ that minimizes $\mu_F(\boldsymbol{x}_c)$, but has widely fluctuating values for $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ across $\mathcal{X}_e$. Suppose there are other settings for the control variables that have a mean response that is "almost" as small and for which $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is relatively "insensitive" to the value of the environmental variables. In some sense, the "insensitive" alternative is a better choice for the

control variables. It performs consistently well for all elements of the environmental variable population, whereas simply minimizing the mean response may lead to great performance for certain elements of the environmental variable distribution and poor performance for others.

In addition, if $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is "sensitive" to the value of $\boldsymbol{x}_e$, then $\mu_F(\boldsymbol{x}_c)$ is "sensitive" to the assumed distribution of $\boldsymbol{X}_e$, which is problematic if the distribution is not known precisely. In this case, different environmental variable distributions can lead to very different choices for the optimal setting of the control variables. For example, suppose that $\boldsymbol{x}_c^F$ minimizes $\mu_F(\boldsymbol{x}_c)$ for some assumed nominal $\boldsymbol{X}_e$ distribution $F(\cdot)$. If the true distribution of $\boldsymbol{X}_e$ is $G \neq F$ and

$$\mu_G(\boldsymbol{x}_c^F) \gg \min_{\boldsymbol{x}_c} \mu_G(\boldsymbol{x}_c), \tag{5.2}$$

then $\boldsymbol{x}_c^F$ is actually an inferior setting of the control variables. In this case a "robust" choice of control variable is a value that comes "close" to minimizing $\mu_F(\boldsymbol{x}_c)$ for the nominal $\boldsymbol{X}_e$ distribution and avoids the situation in (5.2) for plausible alternative $\boldsymbol{X}_e$ distributions. An $\boldsymbol{x}_c$ value that accomplishes these objectives can be considered "robust" to misspecification of the environmental variable distribution.

More generally, suppose that $\boldsymbol{X}_e \sim G(\boldsymbol{x}_e)$ with $G(\cdot) \in \mathcal{G}$, where $\mathcal{G}$ is a class of distributions representing the set of plausible environmental variable distributions. A minimax approach to defining robustness (Huber (1981)) is to choose $\boldsymbol{x}_c$ to minimize

$$\max_{G \in \mathcal{G}} \quad \mu_G(\boldsymbol{x}_c). \tag{5.3}$$

The goal is to make $\mu_G(\boldsymbol{x}_c)$ small, and the minimax approach attempts to guard against the worst case scenario (a pessimistic view) among all environmental variable

distributions in $\mathcal{G}$. We will say that $\boldsymbol{x}_c^{\mathcal{G}}$ is $\mathcal{G}$-robust if

$$\max_{G \in \mathcal{G}} \quad \mu_G(\boldsymbol{x}_c^{\mathcal{G}}) = \min_{\boldsymbol{x}_c \in \mathcal{X}_c} \quad \max_{G \in \mathcal{G}} \quad \mu_G(\boldsymbol{x}_c). \tag{5.4}$$

This formulation requires specification of $\mathcal{G}$ and the minimax computation of the $\mathcal{G}$-robust design, both of which may be challenging.

Another robustness formulation, more Bayesian in nature, starts by placing a prior distribution $\pi(\cdot)$ on $G \in \mathcal{G}$ with $\mathcal{G}$ being a known set of environmental variable distributions. In this setting we find $\boldsymbol{x}_c$ that minimizes $\mu_G(\boldsymbol{x}_c)$ averaged over the set of distributions G in $\mathcal{G}$. In particular, suppose that the class of environmental variable distributions is parametrized by θ, so $\boldsymbol{X}_e \sim G_\theta(\boldsymbol{x}_e) \in \mathcal{G}$; then $\pi(\cdot)$ is a distribution on θ. We call $\boldsymbol{x}_c^{\pi}$ $\pi(\cdot)$-robust if

$$\mu_\pi(\boldsymbol{x}_c^{\pi}) \;=\; \min_{\boldsymbol{x}_c} \mu_\pi(\boldsymbol{x}_c),$$

where

$$\mu_\pi(\boldsymbol{x}_c) = \int_\theta \mu_{G_\theta}(\boldsymbol{x}_c)\pi(\theta)d\theta. \tag{5.5}$$

This formulation requires not only the specification of a class $\mathcal{G}$, as in the $\mathcal{G}$-robust setting, but also the specification of a prior on $\mathcal{G}$. However, computation of the $\pi(\cdot)$-robust design is typically easier than computation of the $\mathcal{G}$-robust design.

A final formulation of robustness returns to the direct analysis of the sensitivity of $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ to $\boldsymbol{x}_e$ and to the quality control notion of minimizing the variability of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$. This formulation requires only that a *nominal* $\boldsymbol{X}_e$ distribution $F(\cdot)$ be specified. Suppose that $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is relatively "flat" in $\boldsymbol{x}_e$. Then the value of the mean of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ will be relatively independent of the choice of $F(\cdot)$, and thus be robust to misspecification of $F(\cdot)$. If small values of $\mu_F(\boldsymbol{x}_c)$ are desirable, a robust value of

$\boldsymbol{x}_c$ minimizes $\mu_F(\boldsymbol{x}_c)$ among $\boldsymbol{x}_c$'s for which $y(\boldsymbol{x}_c, \boldsymbol{x}_e)$ is flat in $\boldsymbol{x}_e$. One measure of flatness is the variability of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$

$$\sigma_F^2(\boldsymbol{x}_c) = \mathrm{Var}_F[y(\boldsymbol{x}_c, \boldsymbol{X}_e)]. \tag{5.6}$$

However, this measure depends on the chosen $F(\cdot)$, a quantity that is not assumed to be known with certainty. As a more general measure of flatness we define

$$\sigma_G^2(\boldsymbol{x}_c) = \mathrm{Var}_G[y(\boldsymbol{x}_c, \boldsymbol{X}_e)], \tag{5.7}$$

where $G(\cdot)$ is a distribution on $\boldsymbol{X}_e$ selected by the user. For example, one could simply set $G = F$, or take $G(\cdot)$ to be a uniform or noninformative distribution on $\boldsymbol{X}_e$. In particular, if $\boldsymbol{X}_e$ is bounded on the hyper-rectangle $\prod_i [a_i, b_i]$ then we have

$$\sigma_G^2(\boldsymbol{x}_c) = \frac{1}{\prod_i (b_i - a_i)} \int y^2(\boldsymbol{x}_c, \boldsymbol{x}_e) d\boldsymbol{x}_e \; - \; \left(\frac{1}{\prod_i (b_i - a_i)} \int y(\boldsymbol{x}_c, \boldsymbol{x}_e) d\boldsymbol{x}_e \right)^2$$

as the "variance" of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ with respect to a uniform distribution.

In this setting, a robust value of $\boldsymbol{x}_c$ might minimize $\mu_F(\boldsymbol{x}_c)$ subject to an upper bound constraint on $\sigma_G^2(\boldsymbol{x}_c)$. We will define $\boldsymbol{x}_c^M$ to be M-robust if it minimizes

$$\begin{gathered} \mu_F(\boldsymbol{x}_c) \\ \text{subject to} \\ \sigma_G^2(\boldsymbol{x}_c) \leq \min_{\boldsymbol{x}_c^* \in \mathcal{X}_c} \sigma_G^2(\boldsymbol{x}_c^*) \; \times \; a \; + \; c, \end{gathered} \tag{5.8}$$

where $a \in \{0, [1, \infty)\}$ and $c \geq 0$. The constraint parameters a and c can be set to reflect the goal of the experiment. If an absolute bound on $\sigma_G^2(\cdot)$ is desired we let $a = 0$ and c be the desired bound. If a relative bound that requires $\sigma_G^2(\boldsymbol{x}_c^M)$ be close to the minimum of $\sigma_G^2(\cdot)$ we set $a \geq 1$ and $c \geq 0$. Note that a constraint involving a relative bound has the advantage that the feasible region is nonempty when $a \geq 1$ and

$c \geq 0$, whereas there need not exist $\boldsymbol{x}_c$ for which a chosen absolute bound ($a = 0$ and $c \geq 0$) is satisfied. Alternatively, perhaps more in keeping with the quality control concept of having a "target" mean, a robust $\boldsymbol{x}_c$ might minimize $\sigma^2_G(\boldsymbol{x}_c)$ subject to a constraint on $\mu_F(\boldsymbol{x}_c)$. We will define $\boldsymbol{x}_c^V$ to be V-robust if it minimizes

$$\begin{gathered} \sigma^2_G(\boldsymbol{x}_c) \\ \text{subject to} \\ \mu_F(\boldsymbol{x}_c) \leq c \quad \text{or} \quad \mu_F(\boldsymbol{x}_c) \leq \min_{\boldsymbol{x}_c \in \mathcal{X}_c} \mu_F(\boldsymbol{x}_c) \; + \; c, \end{gathered} \tag{5.9}$$

where the "target" for $\mu_F(\boldsymbol{x}_c)$ is to make it small. Of course, other constraint settings, such as restricting $\mu_F(\boldsymbol{x}_c)$ to an interval, could be used in this formulation. V robustness allows us to choose the $\boldsymbol{x}_c$ that minimizes $\sigma^2_G(\boldsymbol{x}_c)$ from a class of $\boldsymbol{x}_c$'s that satisfy the given constraints on $\mu_F(\boldsymbol{x}_c)$.

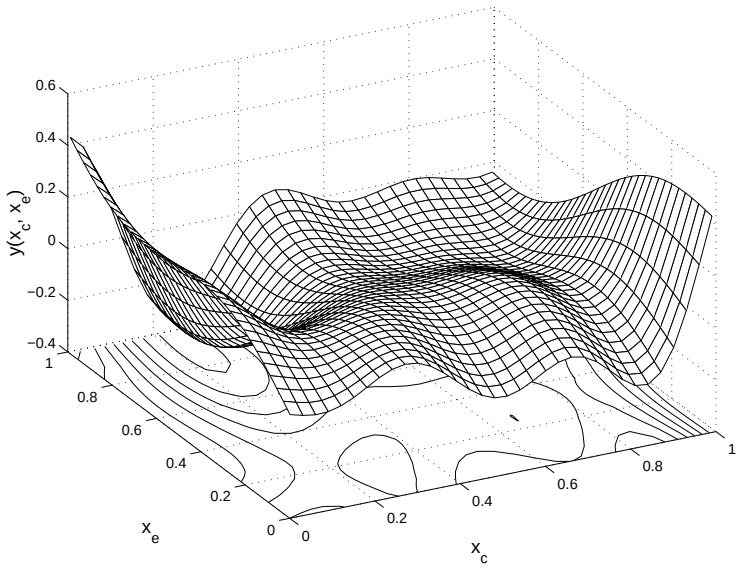


Figure 5.1: True $y(x_c, x_e)$ for robustness example.

We demonstrate these robustness definitions with a hypothetical example. Figure 5.1 plots the response $y(\cdot)$ that depends on a single control variable, $x_c \in (0,1)$, and a single environmental variable, $x_e \in (0,1)$. Figure 5.2 displays four environmental variable distributions that constitute the class $\mathcal{G}$ of $\boldsymbol{X}_e$ distributions considered; these distributions are admittedly different in character. To find the $\mathcal{G}$-robust de-

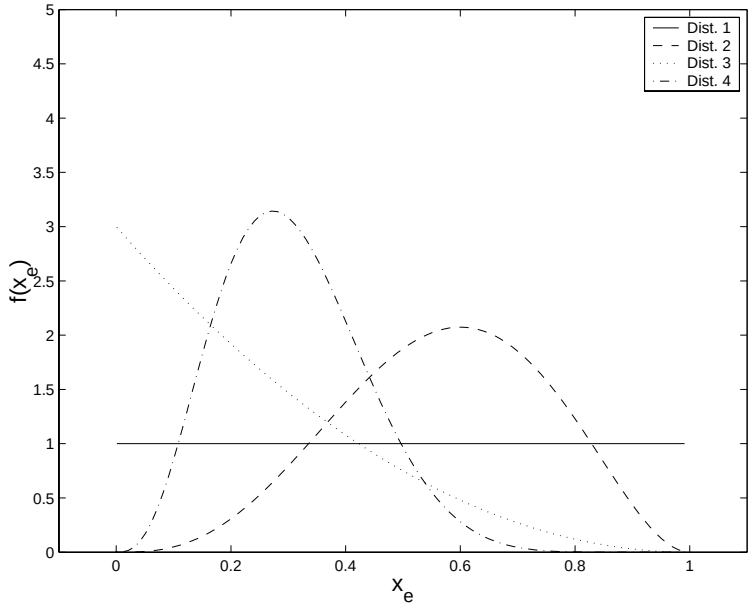


Figure 5.2: Class of four environmental variable distributions.

sign (Equation (5.4)) we plot $\max_{F \in \mathcal{G}} \mu_F(\boldsymbol{x}_c)$ in the left panel of Figure 5.3; $x_c \approx 0.82$ minimizes this quantity, and, thus, is the $\mathcal{G}$-robust design. The right panel of Figure 5.3 plots $\mu_\pi(\boldsymbol{x}_c)$ (Equation (5.5)), assuming that each of the four X_e distributions in Figure 5.2 are equally probable (i.e. $\pi(\cdot)$ is uniform). The optimal $\pi(\cdot)$-robust setting is also $x_c^\pi \approx 0.82$, however, $x_c \approx 0.22$ produces a value of $\mu_\pi(x_c)$ that is very close to optimal.

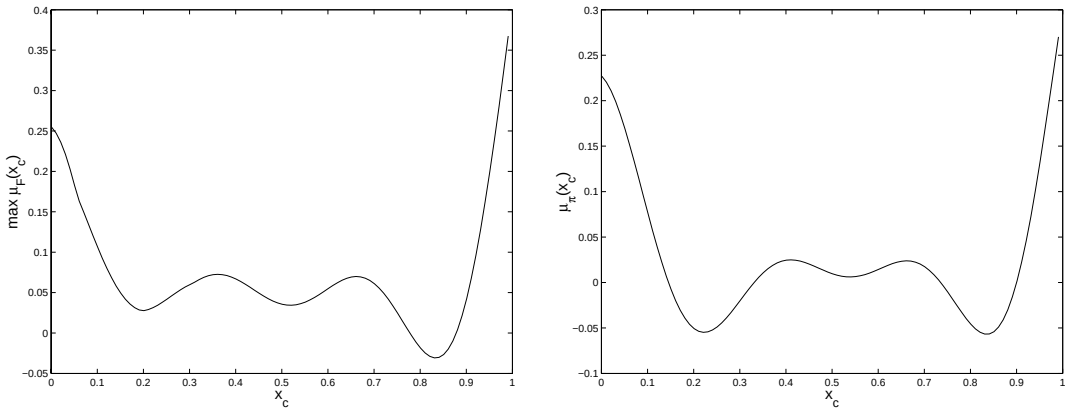


Figure 5.3: Plot of $\max_{F \in \mathcal{G}} \mu_F(\boldsymbol{x}_c)$ (left panel) and $\mu_\pi(x_c)$ (right panel) corresponding to the true response in Figure 5.1.

For M-robust and V-robust control variable values, Figure 5.4 plots $\mu_F(x_c)$ (Equation (5.1)) in the left hand panel and $\sigma_F^2(x_c)$ (Equation (5.7)) in the right hand panel for each of the four X_e distributions. We see clearly the dependence of the minimizers

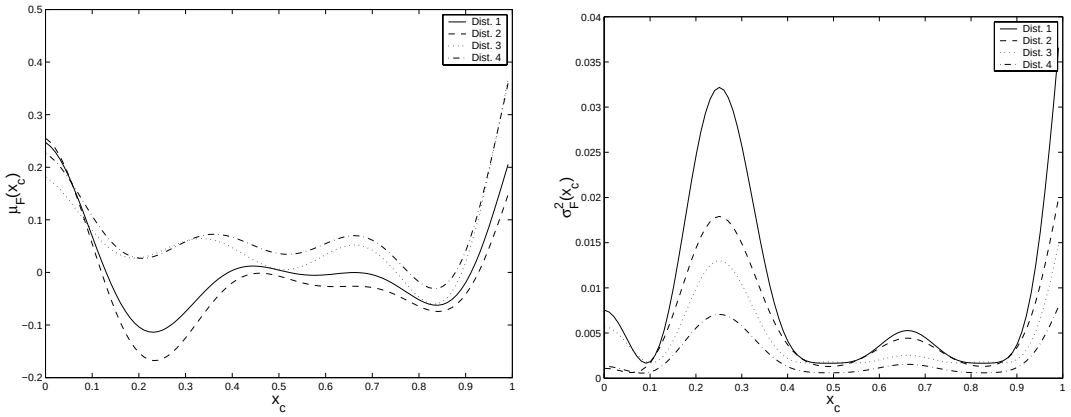


Figure 5.4: True $\mu_F(x_c)$ (left panel) and $\sigma_F^2(x_c)$ (right panel) for each environmental variable distribution.

of $\mu_F(\boldsymbol{x}_c)$ and $\sigma_F^2(x_c)$ on $F(\cdot)$. The value of x_c that minimizes $\mu_F(x_c)$ is $x_c \approx 0.22$ for distributions one and two, and $x_c \approx 0.82$ for distributions three and four. In addition, $x_c \approx 0.22$ is associated with "large" (relatively) $\sigma_F^2(x_c)$ for all four $\boldsymbol{X}_e$ distributions, whereas, $x_c \approx 0.82$ produces small values of $\sigma_F^2(x_c)$ for each $\boldsymbol{X}_e$ distribution. Setting G as a uniform distribution (Distribution 1) with $a = 0$ and $c = 0.01$ in (5.8), the optimal M-robust setting is $x_c \approx 0.82$, and setting $c = -0.05$ in the first contraint of (5.9), the V-robust setting is also $x_c \approx 0.82$. Of course, other constraint settings will lead to other M-robust and V-robust control variable values.

Finally, note that the value of $\mu_F(0.22)$ is relatively unstable in $F(\cdot)$, whereas $\mu_F(0.82)$ is relatively stable regardless of the X_e distribution. Figure 5.5 displays the quantity

$$v_\pi(\boldsymbol{x}_c) = \int_\theta (\mu_{F_\theta}(\boldsymbol{x}_c) - \mu_\pi(\boldsymbol{x}_c))^2 \pi(\theta) d\theta, \tag{5.10}$$

where $\pi(\cdot)$ is uniform over the 4 $\boldsymbol{X}_e$ distributions in Figure 5.2. Again we see that $x_c \approx 0.22$ may be a poor choice for a robust control variable value because it is associated with larger values of $v_\pi(\boldsymbol{x}_c)$. Intuitively, it seems that high values of $v_\pi(\cdot)$ are associated with high values of $\sigma_F^2(\boldsymbol{x}_c)$ (and low values with low values) since if we choose an $\boldsymbol{x}_c$ for which $\sigma_F^2(\boldsymbol{x}_c)$ is small, then the values for $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ are relatively stable (or "flat") across the support points for $\boldsymbol{X}_e$. Thus, changing the distribution of $\boldsymbol{X}_e$ should not significantly change the value for $\mu_F(\boldsymbol{x}_c)$, as indicated by small values of $v_\pi(\boldsymbol{x}_c)$. For discrete $\boldsymbol{X}_e$ and assuming $\pi(\cdot)$ is a Dirichlet distribution, this relationship can be established more rigorously, indicating that choosing $\boldsymbol{x}_c$ to make $\sigma_F^2(\boldsymbol{x}_c)$ small, will also ensure that the value of $\mu_F(\boldsymbol{x}_c)$ will change minimally as the distribution of the environmental variables changes.

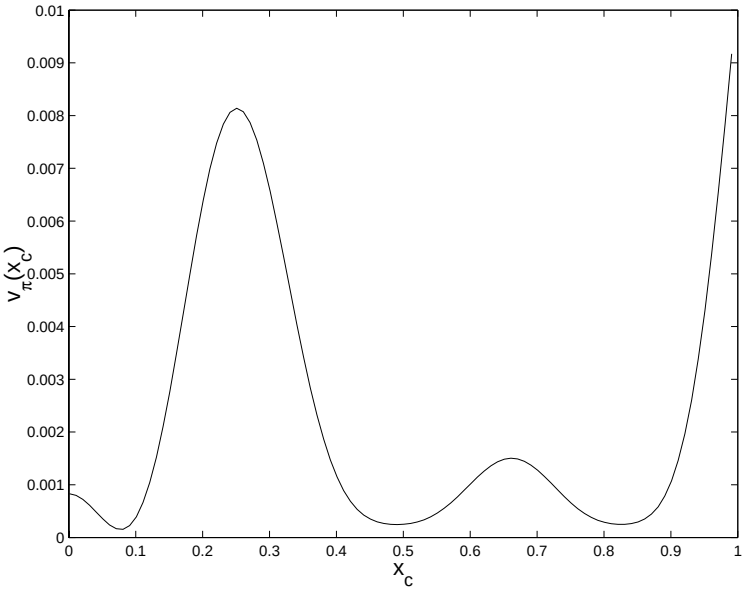


Figure 5.5: Plot of variability of $\mu_F(\boldsymbol{x}_c)$ for varying $F(\cdot)$.

In this simple example we see the importance of choosing robust control variables. If we had assumed that x_e is uniformly distributed on $(0, 1)$ and minimized $\mu_F(x_c)$ based on that assumption, we would choose an x_c for which $\sigma_F^2(x_c)$ is very large and $\mu_F(x_c)$ is very sensitive to $F(\cdot)$. Instead, we would like to find control variable values for which the response is most "stable" regardless of the $\boldsymbol{X}_e$ setting.

In the following sections, we propose algorithms that can be used to find M-robust and V-robust control variable values. In Section 5.2 we present the model for the response. Section 5.3 outlines the sequential algorithm proposed for finding these robust designs, and Section 5.4 presents examples illustrating the performance of these algorithms for several different experimental goals. Finally, Section 5.5 discusses several computational considerations for these algorithms and areas for future research.

5.2 Modeling

As before, let $y(\boldsymbol{x})$ be the true response function with input $\boldsymbol{x} \in \mathcal{X}$, and denote the control and environmental variable portions of the input by $\boldsymbol{x}_c \in \mathcal{X}_c$ and $\boldsymbol{x}_e \in \mathcal{X}_e$ so that $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{x}_e)$. The modeling approach taken here is Bayesian. The prior model for the true response $y(\cdot)$ is the random function

$$Y(\boldsymbol{x}) = \boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{\beta} + Z(\boldsymbol{x}), \tag{5.11}$$

where $Z(\cdot)$ is a covariance stationary Gaussian stochastic process having mean zero, positive definite correlation function $R(\cdot)$, and unknown variance $\tau^2 > 0$. The linear model $\boldsymbol{f}^\top(\cdot)\boldsymbol{\beta}$ represents the (nonstationary) global mean of the $Y(\cdot)$ process with $\boldsymbol{f}(\cdot)$ a k-vector of known regression functions and $\boldsymbol{\beta} \in \mathbb{R}^k$ a vector of unknown regression parameters. Note that there is no noise component of this model, reflecting the deterministic property of a computer experiment (running the code at the same inputs will produce the same output).

We complete specification of the model by assuming the non-informative prior

$$[\boldsymbol{\beta}, \tau^2] \propto \frac{1}{\tau^2}$$

for the parameters $(\boldsymbol{\beta}, \tau^2)$, and by choosing $R(\cdot)$ from a parametric family of known correlation functions. We assume that the correlation function $R(\cdot)$ depends on the parameter vector $\boldsymbol{\gamma}$ which is allowed to take any value for which the covariance structure of the process $Y(\boldsymbol{x})$, for $\boldsymbol{x} \in \mathcal{X}$, is positive definite. In the examples below we follow an empirical Bayes strategy and set $\boldsymbol{\gamma}$ equal to its posterior mode, and substitute these values in wherever necessary.

We will assume that the joint distribution of the environmental variables is discrete on $\{\boldsymbol{x}_{e,j}\}_{j=1}^{n_e}$ with probabilities $P_F\{\boldsymbol{X}_e = \boldsymbol{x}_{e,j}\} = w_j$ and $P_G\{\boldsymbol{X}_e = \boldsymbol{x}_{e,j}\} = \lambda_j$. Here

$F(\cdot)$ is the nominal distribution of $\boldsymbol{X}_e$, and $G(\cdot)$ is the $\boldsymbol{X}_e$ distribution associated with the desired measure of flatness $\sigma^2_G(\cdot)$. Note that we allow $G \neq F$. We define the following quantities

$$\mu_F(\boldsymbol{x}_c) = E_F[y(\boldsymbol{x}_c, \boldsymbol{X}_e)] = \sum_{j=1}^{n_e} w_j y(\boldsymbol{x}_c, \boldsymbol{x}_{e,j}) = \boldsymbol{w}^\top \boldsymbol{y}_{n_e}(\boldsymbol{x}_c), \tag{5.12}$$

and

$$\begin{aligned} \sigma^2_G(\boldsymbol{x}_c) &= \mathrm{Var}_G[y(\boldsymbol{x}_c, \boldsymbol{X}_e)] = \textstyle\sum_{j=1}^{n_e} \lambda_j (y(\boldsymbol{x}_c, \boldsymbol{x}_{e,j}) - \mu_G(\boldsymbol{x}_c))^2 \\ &= \boldsymbol{y}_{n_e}(\boldsymbol{x}_c)^\top (I_{n_e} - \mathbf{1}_{n_e}\boldsymbol{\lambda}^\top)^\top \mathrm{diag}(\boldsymbol{\lambda})(I_{n_e} - \mathbf{1}_{n_e}\boldsymbol{\lambda}^\top)\boldsymbol{y}_{n_e}(\boldsymbol{x}_c), \end{aligned} \tag{5.13}$$

where $\boldsymbol{w} = (w_1, ..., w_{n_e})^\top$, $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_{n_e})^\top$, $\boldsymbol{y}_{n_e}(\boldsymbol{x}_c) = (y(\boldsymbol{x}_c, \boldsymbol{x}_{e,1}), ..., y(\boldsymbol{x}_c, \boldsymbol{x}_{e,n_e}))^\top$, I_n is the $n \times n$ identity matrix, $\mathbf{1}_n$ is an $n \times 1$ vector of ones, and $\mathrm{diag}(\boldsymbol{\lambda})$ is an $n_e \times n_e$ diagonal matrix with $\boldsymbol{\lambda}$ down the diagonal. Note that these correspond to (5.1) and (5.7) in the previous section. Typically, G is taken as a uniform distribution on $\boldsymbol{X}_e$ so that $\lambda_j = 1/n_e$. The prior model in (5.11) induces the distribution of $M_F(\boldsymbol{x}_c) = \sum_{j=1}^{n_e} w_j Y(\boldsymbol{x}_c, \boldsymbol{x}_{e,j})$ for the mean $\mu_F(\cdot)$, and $V_G(\boldsymbol{x}_c) = \sum_{j=1}^{n_e} w_j (Y(\boldsymbol{x}_c, \boldsymbol{x}_{e,j}) - M_G(\boldsymbol{x}_c))^2$ for the "variance" (or flatness) $\sigma^2_G(\cdot)$. For the remainder of this chapter, we will suppress the dependence of $\sigma^2_G(\boldsymbol{x}_c)$ on $G(\cdot)$ and the dependence of $\mu_F(\boldsymbol{x}_c)$ on $F(\cdot)$, simply writing $\sigma^2(\boldsymbol{x}_c)$ as the measure of "flatness" and $\mu(\boldsymbol{x}_c)$ as the mean of $y(\boldsymbol{x}_c, \boldsymbol{X}_e)$ over $F(\cdot)$.

5.3 Sequential Algorithms

This section presents sequential strategies for determining M-robust and V-robust designs. Williams (2000b) and Williams, Santner and Notz (2000a) outline a sequential scheme for choosing input sites at which to run the computer code when the objective is to find an $\boldsymbol{x}_c^*$ satisfying

$$\boldsymbol{x}_c^* = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\mathrm{argmin}}\ \mu(\boldsymbol{x}_c).$$

Recalling the robustness motivations in Section 5.1 and the definitions of the M-robust and V-robust designs, we propose sequential algorithms for finding the M-robust and V-robust control variable settings.

The starting point for both algorithms involves choosing an n-point initial design for $y(\cdot)$, denoted as $\boldsymbol{S}_n = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_n\}$. Let $\boldsymbol{Y}_n = \big[Y(\boldsymbol{s}_1), ..., Y(\boldsymbol{s}_n)\big]^\top$ represent the vector of responses associated with the initial design sites in $\boldsymbol{S}_n$. An outline of the steps of the sequential algorithms are as follows:

1. Choose the initial set of design points $\boldsymbol{S}_n = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_n\}$ at which to observe $y(\cdot)$. We used ACED (Welch (1985)) to generate a maximin distance based design within the set of Latin Hypercube designs.

2. Estimate the covariance parameter vector by $\hat{\boldsymbol{\gamma}}$, the mode of the posterior density of $\boldsymbol{\gamma}$ given $\boldsymbol{Y}_n$. The posterior density function of $\boldsymbol{\gamma}$ given $\boldsymbol{Y}_n$ satisfies

$$p(\boldsymbol{\gamma}|\boldsymbol{Y}_n) \propto p(\boldsymbol{\gamma}) \ |\boldsymbol{R}_{33}|^{-1/2} \ \left|\boldsymbol{F}_n^\top \boldsymbol{R}_{33}^{-1} \boldsymbol{F}_n\right|^{-1} \ [\hat{\tau}^2]^{(n-k)/2}, \tag{5.14}$$

   where $p(\boldsymbol{\gamma})$ is a prior distribution for $\boldsymbol{\gamma}$. The matrices $\boldsymbol{F}_n$ and $\boldsymbol{R}_{33}$ are defined in the following sections, and $\hat{\tau}^2$ is defined in (5.24).

3. Choose the next control variable site $\boldsymbol{x}_c^*$ by an *improvement* criterion.

4. Choose the next environmental variable site $\boldsymbol{x}_e^*$, corresponding to $\boldsymbol{x}_c^*$, by a *precision* criterion.

5. Determine if the algorithm should be stopped. If not, set $\boldsymbol{S}_{n+1} = \boldsymbol{S}_n \bigcup (\boldsymbol{x}_c^*, \boldsymbol{x}_e^*)$, compute the response $y(\boldsymbol{x}_c^*, \boldsymbol{x}_e^*)$ and return to Step 2. If the stopping criterion is met, then the optimal robust setting for $\boldsymbol{x}_c$ is obtained using traditional

optimization techniques with the posterior means of $M(\cdot)$ and $V(\cdot)$ substituted for $\mu(\cdot)$ and $\sigma^2(\cdot)$.

The criterion for selecting the input site at which to take the next observation (Steps 3 and 4 above) will depend on the goal of the experiment. In Section 5.3.1 we present the criterion for adding points when the goal is to find the V-robust control variable values, and in Section 5.3.2 we present the criterion for adding points when the goal is to find the M-robust control variable value.

For both algorithms let the control variable portion of $\boldsymbol{S}_n$ be denoted by $\boldsymbol{S}_n^C = \{\boldsymbol{s}_{c,1}, ..., \boldsymbol{s}_{c,n}\}$. Assume that $\boldsymbol{X}_e$ is discrete on $\{\boldsymbol{x}_{e,j}\}_{j=1}^{n_e}$ with nominal weights $\{w_j\}_{j=1}^{n_e}$. For a given control variable value $\boldsymbol{x}_c$ we let $\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c) = [Y(\boldsymbol{x}_c, \boldsymbol{x}_{e,1}), ..., Y(\boldsymbol{x}_c, \boldsymbol{x}_{e,n_e})]^\top$ and define the random variables corresponding to (5.12) and (5.13) as

$$M(\boldsymbol{x}_c) = \boldsymbol{w}^\top \boldsymbol{Y}_{n_e}(\boldsymbol{x}_c), \tag{5.15}$$

and

$$V(\boldsymbol{x}_c) = \boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)^\top \boldsymbol{A} \boldsymbol{Y}_{n_e}(\boldsymbol{x}_c), \tag{5.16}$$

where $\boldsymbol{A} = (I_{n_e} - \mathbf{1}_{n_e}\boldsymbol{\lambda}^\top)\text{diag}(\boldsymbol{\lambda})(I_{n_e} - \mathbf{1}_{n_e}\boldsymbol{\lambda}^\top)^\top$ and $\boldsymbol{\lambda}$ is the vector of weights associated with the desired measure of "flatness". Finally, let $\boldsymbol{M}_n = [M(\boldsymbol{s}_{c,1}), ..., M(\boldsymbol{s}_{c,n})]^\top$ be the vector of values for the mean response associated with $\boldsymbol{S}_n^C$.

5.3.1 Algorithm Details for Finding V-Robust Designs

Recall that $\boldsymbol{x}_c^*$ is V-robust if it satisfies

$$\begin{gathered} \boldsymbol{x}_c^* = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\text{argmin}}\ \sigma^2(\boldsymbol{x}_c) \\ \text{subject to either} \\ \mu(\boldsymbol{x}_c^*) \leq \mu(\boldsymbol{x}_c^{min}) + c, \qquad \text{or} \qquad \mu(\boldsymbol{x}_c^*) \leq c, \end{gathered} \tag{5.17}$$

where $\boldsymbol{x}_c^{min} = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\text{argmin}}\ \mu(\boldsymbol{x}_c)$. In words, the goal is to find $\boldsymbol{x}_c^*$ that minimizes $\sigma^2(\cdot)$ subject to $\mu(\boldsymbol{x}_c^*)$ being "small" (i.e., close to $\min_{\boldsymbol{x}_c \in \mathcal{X}_c} \mu(\boldsymbol{x}_c)$ or satisfying a given constraint). The constraint equation in (5.17) and the constraint parameter c are chosen to reflect the research objective. If it is necessary to choose an $\boldsymbol{x}_c$ that is a global minimizer of $\mu(\cdot)$, then set $c = 0$ and use the left hand constraint equation. On the other hand, if it is sufficient to choose between values of $\boldsymbol{x}_c$ such that $\mu(\boldsymbol{x}_c) < 5$, for example, then set $c = 5$ and use the right hand constraint equation. To complete specification of the algorithm for finding the V-robust design we need to define the *improvement* criterion for selecting the next control variable value (Step 3), and the *precision* criterion for selecting the next environmental variable value (Step 4). The stopping criterion (Step 5) will be discussed in the examples in Section 5.4 and in Section 5.5.

Selection of Control Variables

We choose the next control variable site $\boldsymbol{x}_c^*$ to maximize

$$I(\boldsymbol{x}_c) = E[\max\{0, v_{min,f} - V(\boldsymbol{x}_c)\} \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}] \times P[\text{ constraint } |\boldsymbol{Y}_n, \boldsymbol{\gamma}], \tag{5.18}$$

where the constraint is either

$$M(\boldsymbol{x}_c) \ \leq \ M_{n,min} + c, \tag{5.19}$$

or

$$M(\boldsymbol{x}_c) \ \leq \ c. \tag{5.20}$$

The random variable $M_{n,min}$ is the minimum of $\boldsymbol{M}_n$, the vector of $M(\cdot)$ values at control variable sites in $\boldsymbol{S}_n^C$, and the *constant*, $v_{min,f}$, is the current best guess at the

constrained minimum of $\sigma^2(\cdot)$; $v_{min,f}$ is the *minimum* of the posterior expectation of $V(\cdot)$ for control variable values in $\boldsymbol{S}_n^C$ that are estimated to be in the *feasible* region. Formally, we let $M_{n,min} = \min\{M(\boldsymbol{s}_{c,i}) \ : \ 1 \leq i \leq n\}$, and define the constant $v_{min,f}$ to be the minimum of $E\big[V(\boldsymbol{x}_c)|\boldsymbol{Y}_n, \boldsymbol{\gamma}\big]$ for $\boldsymbol{x}_c \in \boldsymbol{C}_n$ where $\boldsymbol{C}_n = \{\boldsymbol{s}_{c,i} \in \boldsymbol{S}_n^C \ : \ M_{.025}(\boldsymbol{s}_{c,i}) \leq \text{constraint}\}$ and $M_{.025}(\boldsymbol{s}_{c,i})$ is the lower 2.5^{th} percentile of the posterior distribution of $M(\boldsymbol{s}_{c,i})$ given $\boldsymbol{Y}_n$ and $\boldsymbol{\gamma}$ (this posterior distribution is given in (5.31)). Thus, we choose $\boldsymbol{x}_c^*$ such that

$$\boldsymbol{x}_c^* = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\text{argmax}} \ I(\boldsymbol{x}_c).$$

The intuition behind this criterion is as follows. We choose the next control variable value site $\boldsymbol{x}_c^*$ to maximize the improvement in $V(\cdot)$ over the minimum of the posterior expected values of $V(\cdot)$ for control variable sites already observed that appear to satisfy the constraint. We multiply this improvement by the probability that the constraint is satisfied thus "downweighting" observations in the infeasible region of the control variable space. The calculations necessary for computing (5.18) are outlined next. Lemma B.0.1 in Appendix B will be used throughout.

We begin calculation of (5.18) with the joint distribution of $(\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c), \boldsymbol{Y}_{n,n_e}, \boldsymbol{Y}_n)$ given $\boldsymbol{\beta}$, τ^2, and $\boldsymbol{\gamma}$, where $\boldsymbol{Y}_{n,n_e} = (Y(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,1}), \ ..., \ Y(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,n_e}), \ ..., \ Y(\boldsymbol{s}_{c,n}, \boldsymbol{x}_{e,1}),...$ $Y(\boldsymbol{s}_{c,n}, \boldsymbol{x}_{e,n_e}))^\top$ is the $(n \cdot n_e) \times 1$ vector of responses at control sites in $\boldsymbol{S}_n^C$ paired with each support point for the environmental variables. From Model (5.11) the joint distribution of $[\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c), \boldsymbol{Y}_{n,n_e}, \boldsymbol{Y}_n]$ given $(\boldsymbol{\beta}, \tau^2, \boldsymbol{\gamma})$, is multivariate normal with mean $(\boldsymbol{F}_{n_e}^\top(\boldsymbol{x}_c), \boldsymbol{F}_{n,n_e}^\top, \boldsymbol{F}_n^\top)^\top \boldsymbol{\beta}$ and variance-covariance matrix $\tau^2((\boldsymbol{\Sigma}_{pq}))$ for $p, q \in \{1, 2, 3\}$, where the components are defined next. The vectors $\boldsymbol{F}_{n_e}(\boldsymbol{x}_c) = [\boldsymbol{f}(\boldsymbol{x}_c, \boldsymbol{x}_{e,1}),...,$ $\boldsymbol{f}(\boldsymbol{x}_c, \boldsymbol{x}_{e,n_e})]^\top$, $\boldsymbol{F}_{n,n_e} = [\boldsymbol{f}(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,1}), \ ...,\boldsymbol{f}(\boldsymbol{s}_{c,1}, \boldsymbol{x}_{e,n_e}), \ ... \ \boldsymbol{f}(\boldsymbol{s}_{c,n}, \boldsymbol{x}_{e,n_e})]^\top$, and $\boldsymbol{F}_n =$

$[\boldsymbol{f}(\boldsymbol{s}_1),\ ...,\ \boldsymbol{f}(\boldsymbol{s}_n)]^\top$ are the regression matrices for $\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)$, $\boldsymbol{Y}_{n,n_e}$, and $\boldsymbol{Y}_n$ respectively. The indices $p, q \in \{1, 2, 3\}$ for the covariance matrices $\boldsymbol{\Sigma}_{pq}$ correspond to the three components $\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)$, $\boldsymbol{Y}_{n,n_e}$, and $\boldsymbol{Y}_n$ in this order, so that, for example, $\text{Cov}[\boldsymbol{Y}_{n,n_e}, \boldsymbol{Y}_n] = \tau^2\boldsymbol{\Sigma}_{23}$.

Because Gaussian random vectors remain Gaussian under linear transformations we have

$$\begin{pmatrix} \boldsymbol{Y}_{n_e}(\boldsymbol{x}_c) \\ \boldsymbol{M}_n \\ \boldsymbol{Y}_n \end{pmatrix} |\boldsymbol{\beta}, \tau^2, \boldsymbol{\gamma} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{F}_{n_e}(\boldsymbol{x}_c) \\ \bar{\boldsymbol{F}}_{n,n_e} \\ \boldsymbol{F}_n \end{pmatrix} \boldsymbol{\beta}, \tau^2 \begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \boldsymbol{R}_{13} \\ \cdot & \boldsymbol{R}_{22} & \boldsymbol{R}_{23} \\ \cdot & \cdot & \boldsymbol{R}_{33} \end{pmatrix}\right), \quad (5.21)$$

where $\bar{\boldsymbol{F}}_{n,n_e} = (I_n \otimes \boldsymbol{w}^\top)\boldsymbol{F}_{n,n_e}$, $\boldsymbol{R}_{11} = \boldsymbol{\Sigma}_{11}$, $\boldsymbol{R}_{13} = \boldsymbol{\Sigma}_{13}$, $\boldsymbol{R}_{33} = \boldsymbol{\Sigma}_{33}$, $\boldsymbol{R}_{12}$=$\boldsymbol{\Sigma}_{12}(I_n \otimes \boldsymbol{w}^\top)^\top$, $\boldsymbol{R}_{23}$=$(I_n \otimes \boldsymbol{w}^\top)\boldsymbol{\Sigma}_{23}$, and $\boldsymbol{R}_{22}$=$(I_n \otimes \boldsymbol{w}^\top)\boldsymbol{\Sigma}_{22}(I_n \otimes \boldsymbol{w}^\top)^\top$.

To calculate $E[\max\{0, v_{min,f} - V(\boldsymbol{x}_c)\}|\boldsymbol{Y}_n, \boldsymbol{\gamma}]$ we first compute the constant $v_{min,f}$. From (5.16), $V(\boldsymbol{x}_c) = \boldsymbol{Y}_{n_e}^\top(\boldsymbol{x}_c)\boldsymbol{A}\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)$ is a quadratic form in $\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)$ for all $\boldsymbol{x}_c$. Using Lemma B.0.1, the posterior distribution of $\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)$ given $\boldsymbol{Y}_n$ and $\boldsymbol{\gamma}$ is

$$[\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)|\boldsymbol{Y}_n, \boldsymbol{\gamma}] \ \sim \ \mathcal{T}_{n_e}(\boldsymbol{m}, \hat{\tau}^2\boldsymbol{R}, n-k) \quad (5.22)$$

where $\mathcal{T}_q(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$ denotes the q-variate t distribution with location shift $\boldsymbol{\mu}$, scale matrix $\boldsymbol{\Sigma}$ and ν degrees of freedom (see Definition B.0.8),

$$\boldsymbol{m} = \boldsymbol{F}_{n_e}(\boldsymbol{x}_c)\hat{\boldsymbol{\beta}} \ + \ \boldsymbol{R}_{13}\boldsymbol{R}_{33}^{-1}[\boldsymbol{Y}_n - \boldsymbol{F}_n\hat{\boldsymbol{\beta}}],$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}_n^\top\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)^{-1}\boldsymbol{F}_n^\top\boldsymbol{R}_{33}^{-1}\boldsymbol{Y}_n, \quad (5.23)$$

$$\hat{\tau}^2 = \frac{\boldsymbol{Y}_n^\top\boldsymbol{R}_{33}^{-1}\boldsymbol{Y}_n \ - \ \hat{\boldsymbol{\beta}}^\top(\boldsymbol{F}_n^\top\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)\hat{\boldsymbol{\beta}}}{n-k}, \quad (5.24)$$

and

$$\begin{aligned} \boldsymbol{R} \ &= \ \boldsymbol{R}_{11} - \boldsymbol{R}_{13}\boldsymbol{R}_{33}^{-1}\boldsymbol{R}_{13}^\top \quad + \\ &(\boldsymbol{F}_{n_e}(\boldsymbol{x}_c) - \boldsymbol{R}_{13}\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)(\boldsymbol{F}_n^\top\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)^{-1}(\boldsymbol{F}_{n_e}(\boldsymbol{x}_c) - \boldsymbol{R}_{13}\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)^\top. \end{aligned}$$

Applying the well known formula for expectations of a quadratic form (see Theorem B.0.6) we obtain

$$\mathrm{E}[V(\boldsymbol{x}_c)|\boldsymbol{Y}_n,\boldsymbol{\gamma}] = \frac{n-k}{n-k-2}\mathrm{trace}[\hat{\tau}^2\boldsymbol{R}\boldsymbol{A}] + \boldsymbol{m}^\top\boldsymbol{A}\boldsymbol{m}. \tag{5.25}$$

We calculate (5.25) for each control variable value in $\boldsymbol{C}_n$, set $v_{min,f}$ as the minimum of these expected values, and compute $E[\max\{0, v_{min,f} - V(\boldsymbol{x}_c)\}|\boldsymbol{Y}_n,\boldsymbol{\gamma}]$ via Monte Carlo by generating N_v samples of $\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c)$ from distribution (5.22).

To sample from a multivariate t-distribution, $\boldsymbol{Y} \sim \mathcal{T}_{q_1}(\boldsymbol{\mu},\boldsymbol{\Sigma},\nu)$, we follow these steps (see Johnson and Kotz chapter 37 (1972)):

1. Sample x from a chi-square distribution with ν degrees of freedom.

2. Sample $\boldsymbol{Y}$ from a q_1-variate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\frac{\nu}{x}\boldsymbol{\Sigma}$.

A formula for the posterior probability of (5.19) (the first constraint) is obtained via iterated expectations and Monte Carlo. We have

$$\begin{aligned} P[M(\boldsymbol{x}_c) \le M_{n,min} + c \mid \boldsymbol{Y}_n,\boldsymbol{\gamma}] \quad &= \\ E_{\boldsymbol{M}_n|\boldsymbol{Y}_n,\boldsymbol{\gamma}}\left[P\big(M(\boldsymbol{x}_c) \le M_{n,min} + c \mid \boldsymbol{M}_n,\boldsymbol{Y}_n,\boldsymbol{\gamma}\big)\right]. & \end{aligned} \tag{5.26}$$

To compute the inner expectation we apply a linear transformation and Lemma B.0.1 to distribution (5.21) to obtain

$$[M(\boldsymbol{x}_c)|\boldsymbol{Z}_{2n},\boldsymbol{\gamma}] \sim \mathcal{T}_1(\mu_{M,1}, \tilde{\tau}^2\sigma_{M,1}, 2n-k), \tag{5.27}$$

where $\boldsymbol{Z}_{2n} = (\boldsymbol{M}_n^\top, \boldsymbol{Y}_n^\top)^\top$. Letting $\boldsymbol{R}_{1,23} = (\boldsymbol{R}_{12}, \boldsymbol{R}_{13})$, $\boldsymbol{F}_{2n} = \begin{pmatrix} \bar{\boldsymbol{F}}_{n,n_e} \\ \boldsymbol{F}_n \end{pmatrix}$, and $\boldsymbol{R}_{2n} = \begin{pmatrix} \boldsymbol{R}_{22} & \boldsymbol{R}_{23} \\ \cdot & \boldsymbol{R}_{33} \end{pmatrix}$, we have

$$\mu_{M,1} = \boldsymbol{w}^\top\boldsymbol{F}_{n_e}(\boldsymbol{x}_c)\tilde{\boldsymbol{\beta}} + \boldsymbol{w}^\top\boldsymbol{R}_{1,23}(\boldsymbol{Z}_{2n} - \boldsymbol{F}_{2n}\tilde{\boldsymbol{\beta}}), \tag{5.28}$$

$$\tilde{\boldsymbol{\beta}} = (\boldsymbol{F}_{2n}^{\top}\boldsymbol{R}_{2n}^{-1}\boldsymbol{F}_{2n})^{-1}\boldsymbol{F}_{2n}^{\top}\boldsymbol{R}_{2n}^{-1}\boldsymbol{Z}_{2n},$$

$$\tilde{\tau}^2 = \frac{\boldsymbol{Z}_{2n}^{\top}\boldsymbol{R}_{2n}^{-1}\boldsymbol{Z}_{2n} - \tilde{\boldsymbol{\beta}}^{\top}(\boldsymbol{F}_{2n}^{\top}\boldsymbol{R}_{2n}^{-1}\boldsymbol{F}_{2n})\tilde{\boldsymbol{\beta}}}{2n-k},$$

and

$$\begin{aligned}\sigma_{M,1} &= \boldsymbol{w}^{\top}\boldsymbol{R}_{11}\boldsymbol{w} - \boldsymbol{w}^{\top}\boldsymbol{R}_{1,23}\boldsymbol{R}_{2n}^{-1}\boldsymbol{R}_{1,23}^{\top}\boldsymbol{w} \quad + \\ &\boldsymbol{w}^{\top}(\boldsymbol{F}_{n_e}(\boldsymbol{x}_c) - \boldsymbol{R}_{1,23}\boldsymbol{R}_{2n}^{-1}\boldsymbol{F}_{2n})(\boldsymbol{F}_{2n}^{\top}\boldsymbol{R}_{2n}^{-1}\boldsymbol{F}_{2n})^{-1}(\boldsymbol{F}_{n_e}(\boldsymbol{x}_c) - \boldsymbol{R}_{1,23}\boldsymbol{R}_{2n}^{-1}\boldsymbol{F}_{2n})^{\top}\boldsymbol{w}.\end{aligned} \tag{5.29}$$

Thus, the inner expectation is

$$\begin{aligned}P\big[M(\boldsymbol{x}_c) \leq M_{n,min} + c \mid \boldsymbol{M}_n, \boldsymbol{Y}_n, \boldsymbol{\gamma}\big] \quad &= \\ \mathrm{T}_{2n-k}\left[(M_{n,min} + c - \mu_{M,1})/\sqrt{\tilde{\tau}^2\sigma_{M,1}}\right],&\end{aligned} \tag{5.30}$$

where $T_\nu(\cdot)$ is the univariate t-distribution cdf with ν degrees of freedom. The outer expectation is obtained via Monte Carlo. We generate N_μ random samples from the distribution of $[\boldsymbol{M}_n|\boldsymbol{Y}_n, \boldsymbol{\gamma}]$, which is given next, and compute (5.30) for each sample. The value for (5.26) is then obtained as the average of these N_μ quantities.

The distribution of $[\boldsymbol{M}_n|\boldsymbol{Y}_n, \boldsymbol{\gamma}]$ is computed by applying Lemma B.0.1 to the $[\boldsymbol{M}_n, \boldsymbol{Y}_n]$ portion of distribution (5.21). We have

$$[\boldsymbol{M}_n|\boldsymbol{Y}_n, \boldsymbol{\gamma}] \sim \mathcal{T}_n(\boldsymbol{\mu}_{M_n}, \hat{\tau}^2\boldsymbol{\Sigma}_{M_n}, n-k), \tag{5.31}$$

where $\boldsymbol{\mu}_{M_n} = \bar{\boldsymbol{F}}_{n,n_e}\hat{\boldsymbol{\beta}} + \boldsymbol{R}_{23}\boldsymbol{R}_{33}^{-1}(\boldsymbol{Y}_n - \boldsymbol{F}_n\hat{\boldsymbol{\beta}})$, $\hat{\boldsymbol{\beta}}$ as in (5.23), $\hat{\tau}^2$ as in (5.24), and

$$\begin{aligned}\boldsymbol{\Sigma}_{M_n} &= \boldsymbol{R}_{22} - \boldsymbol{R}_{23}\boldsymbol{R}_{33}^{-1}\boldsymbol{R}_{23}^{\top} \quad + \\ &(\bar{\boldsymbol{F}}_{n,n_e} - \boldsymbol{R}_{23}\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)(\boldsymbol{F}_n^{\top}\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)^{-1}(\bar{\boldsymbol{F}}_{n,n_e} - \boldsymbol{R}_{23}\boldsymbol{R}_{33}^{-1}\boldsymbol{F}_n)^{\top}.\end{aligned}$$

The N_μ random samples from this multivariate t-distribution are generated using the steps above.

A formula for $P[M(\boldsymbol{x}_c) \leq c \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}]$, the posterior probability of (5.20) follows in a similar fashion. Applying a linear transformation to the distribution of $[\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c), \boldsymbol{Y}_n]$

given $(\boldsymbol{\beta}, \tau^2, \boldsymbol{\gamma})$, and then applying Lemma B.0.1 to the result gives

$$[M(\boldsymbol{x}_c)|\boldsymbol{Y}_n, \boldsymbol{\gamma}] \;\sim\; \mathcal{T}_1(\mu_{M,2}, \hat{\tau}^2\sigma_{M,2}, n-k),$$

where the formulas for $\mu_{M,2}$ and $\sigma_{M,2}$ are identical to formulas (5.28) and (5.29) for $\mu_{M,1}$ and $\sigma_{M,1}$ with $\tilde{\boldsymbol{\beta}}$ replaced by $\hat{\boldsymbol{\beta}}$, $\boldsymbol{R}_{1,23}$ replaced by $\boldsymbol{R}_{13}$, $\boldsymbol{Z}_{2n}$ replaced by $\boldsymbol{Y}_n$, $\boldsymbol{F}_{2n}$ replaced by $\boldsymbol{F}_n$, and $\boldsymbol{R}_{2n}$ replaced by $\boldsymbol{R}_{33}$. We obtain

$$P[M(\boldsymbol{x}_c) \le c \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}] = \mathrm{T}_{n-k}((c-\mu_{M,2})/\sqrt{\hat{\tau}^2\sigma_{M,2}}), \tag{5.32}$$

where $T_\nu(\cdot)$ is the univariate T cdf with ν degrees of freedom.

Selection of Environmental Variables

We select the next environmental variable site $\boldsymbol{x}_e^*$, corresponding to $\boldsymbol{x}_c^*$, to minimize the posterior mean-squared prediction error for $M(\boldsymbol{x}_c^*)$

$$E[(M(\boldsymbol{x}_c^*) - \hat{M}(\boldsymbol{x}_c^*))^2 \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}], \tag{5.33}$$

where $\hat{M}(\boldsymbol{x}_c^*)$ is the posterior mean of $M(\boldsymbol{x}_c^*)$ given $Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e)$, $\boldsymbol{Y}_n$, and $\boldsymbol{\gamma}$. A formula for (5.33) can be computed via iterated expectations

$$\begin{aligned} E[(M(\boldsymbol{x}_c^*) - \hat{M}(\boldsymbol{x}_c^*))^2|\boldsymbol{Y}_n] \quad &= \\ E_{Y(\boldsymbol{x}_c^*,\boldsymbol{x}_e)|\boldsymbol{Y}_n,\boldsymbol{\gamma}}&\Big\{E[(M(\boldsymbol{x}_c^*) - \hat{M}(\boldsymbol{x}_c^*))^2|Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{Y}_n, \boldsymbol{\gamma}]\Big\}. \end{aligned} \tag{5.34}$$

Define the following correlation matrices,

$$\boldsymbol{R}_{n_e} = \mathrm{Corr}(\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c^*), \boldsymbol{Y}_{n_e}(\boldsymbol{x}_c^*)), \quad \boldsymbol{r}_{n_e} = \mathrm{Corr}(Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{Y}_{n_e}(\boldsymbol{x}_c^*)),$$

$$\boldsymbol{R}_{n_e,n} = \mathrm{Corr}(\boldsymbol{Y}_{n_e}(\boldsymbol{x}_c^*), \boldsymbol{Y}_n), \qquad \boldsymbol{r}_n = \mathrm{Corr}(\boldsymbol{Y}_n, Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e)), \quad \text{and}$$

$$\boldsymbol{R}_n = \mathrm{Corr}(\boldsymbol{Y}_n, \boldsymbol{Y}_n).$$

Then, using the Gaussian assumption and a linear tranformation, given $\boldsymbol{\beta}$, τ^2 and $\boldsymbol{\gamma}$

$$\begin{pmatrix} M(\boldsymbol{x}_c^*) \\ Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e) \\ \boldsymbol{Y}_n \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \bar{\boldsymbol{F}}_{n_e}(\boldsymbol{x}_c^*) \\ \boldsymbol{f}^\top(\boldsymbol{x}_c^*, \boldsymbol{x}_e) \\ \boldsymbol{F}_n \end{pmatrix}\boldsymbol{\beta},\; \tau^2 \begin{pmatrix} \boldsymbol{w}^\top\boldsymbol{R}_{n_e}\boldsymbol{w} & \boldsymbol{w}^\top\boldsymbol{r}_{n_e} & \boldsymbol{w}^\top\boldsymbol{R}_{n_e,n} \\ \cdot & 1 & \boldsymbol{r}_n \\ \cdot & \cdot & \boldsymbol{R}_n \end{pmatrix}\right),$$

where $\bar{\boldsymbol{F}}_{n_e}(\boldsymbol{x}_c^*)$ and $\boldsymbol{F}_n$ are defined as before. Using Lemma B.0.1 and letting $E_{11} = \boldsymbol{w}^\top \boldsymbol{R}_{n_e} \boldsymbol{w}$, $\boldsymbol{E}_{12} = (\boldsymbol{w}^\top \boldsymbol{r}_{n_e}, \boldsymbol{w}^\top \boldsymbol{R}_{n_e,n})$, $\boldsymbol{E}_{22} = \begin{pmatrix} 1 & \boldsymbol{r}_n \\ \cdot & \boldsymbol{R}_n \end{pmatrix}$, $\boldsymbol{Z} = (Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{Y}_n^\top)^\top$, and $\boldsymbol{F}_e = (\boldsymbol{f}(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{F}_n^\top)^\top$ we obtain

$$[M(\boldsymbol{x}_c^*)|\boldsymbol{Z}, \boldsymbol{\gamma}] \quad \sim \quad \mathcal{T}(m_e, \ddot{\tau}^2 R_e, n+1-k),$$

where

$$m_e = \bar{\boldsymbol{F}}_{n_e}(\boldsymbol{x}_c^*)\ddot{\boldsymbol{\beta}} + \boldsymbol{E}_{12}\boldsymbol{E}_{22}^{-1}(\boldsymbol{Z} - \boldsymbol{F}_e\ddot{\boldsymbol{\beta}}),$$

$$\ddot{\boldsymbol{\beta}} = (\boldsymbol{F}_e^\top \boldsymbol{E}_{22}^{-1} \boldsymbol{F}_e)^{-1} \boldsymbol{F}_e^\top \boldsymbol{E}_{22}^{-1} \boldsymbol{Z},$$

$$\ddot{\tau}^2 = [\boldsymbol{Z}^\top \boldsymbol{Q} \boldsymbol{Z}]/(n+1-k),$$

$$\boldsymbol{Q} = \boldsymbol{E}_{22}^{-1} - \boldsymbol{E}_{22}^{-1}\boldsymbol{F}_e(\boldsymbol{F}_e^\top \boldsymbol{E}_{22}^{-1}\boldsymbol{F}_e)^{-1}\boldsymbol{F}_e^\top \boldsymbol{E}_{22}^{-1},$$

and

$$\begin{aligned} R_e \;=\; E_{11} \quad - \quad & \boldsymbol{E}_{12}\boldsymbol{E}_{22}^{-1}\boldsymbol{E}_{12}^\top \quad + \\ & (\bar{\boldsymbol{F}}_{n_e}(\boldsymbol{x}_c^*) - \boldsymbol{E}_{12}\boldsymbol{E}_{22}^{-1}\boldsymbol{F}_e)(\boldsymbol{F}_e^\top \boldsymbol{E}_{22}^{-1}\boldsymbol{F}_e)^{-1}(\bar{\boldsymbol{F}}_{n_e}(\boldsymbol{x}_c^*) - \boldsymbol{E}_{12}\boldsymbol{E}_{22}^{-1}\boldsymbol{F}_e)^\top. \end{aligned}$$

Since $\hat{M}(\boldsymbol{x}_c^*)$ is the posterior mean of $M(\boldsymbol{x}_c^*)$ given $\boldsymbol{Z}$, the inner expectation in (5.34) is the variance of this t-distribution, which is

$$E[(M(\boldsymbol{x}_c^*) - \hat{M}(\boldsymbol{x}_c^*))^2 | Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{Y}_n, \boldsymbol{\gamma}] = (\frac{n+1-k}{n-k-1})\; R_e\; \ddot{\tau}^2. \tag{5.35}$$

To calculate the outer expectation in (5.34) we note that given $\boldsymbol{Y}_n$ (5.35) is a constant, $(\frac{1}{n-k-1})R_e$, times a quadratic form in $\boldsymbol{Z}$, $\boldsymbol{Z}^\top \boldsymbol{Q} \boldsymbol{Z}$. Using Lemmas B.0.3 and B.0.4 in the Appendix with

$$\mu(\boldsymbol{x}_c^*, \boldsymbol{x}_e) = E[Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e)|\boldsymbol{Y}_n, \boldsymbol{\gamma}] = \boldsymbol{f}^\top(\boldsymbol{x}_c^*, \boldsymbol{x}_e)\hat{\boldsymbol{\beta}} + \boldsymbol{r}_n \boldsymbol{R}_n^{-1}(\boldsymbol{Y}_n - \boldsymbol{F}_n\hat{\boldsymbol{\beta}}),$$

and $\boldsymbol{m}_e = (\mu(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{Y}_n^\top)^\top$, we have

$$E[\boldsymbol{Z}^\top \boldsymbol{Q} \boldsymbol{Z}|\boldsymbol{Y}_n, \boldsymbol{\gamma}] = \boldsymbol{m}_e^\top \boldsymbol{Q} \boldsymbol{m}_e + \text{trace}[\frac{n-k}{n-k-2}\; \hat{\tau}^2 * 1]. \tag{5.36}$$

Thus the outer expectation in (5.34) is

$$E[(M(\boldsymbol{x}_c^*) - \hat{M}(\boldsymbol{x}_c^*))^2 | \boldsymbol{Y}_n, \boldsymbol{\gamma}] = (\frac{1}{n-k-1})\ R_e\ [\boldsymbol{m}_e^\top \boldsymbol{Q} \boldsymbol{m}_e + \frac{n-k}{n-k-2}\ \hat{\tau}^2]. \quad (5.37)$$

5.3.2 Algorithm Details for Finding M-Robust Designs

Recall that $\boldsymbol{x}_c^*$ is M-robust if it satisfies

$$\begin{gathered} \boldsymbol{x}_c^* = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\operatorname{argmin}}\ \mu(\boldsymbol{x}_c) \\ \text{subject to} \\ \sigma^2(\boldsymbol{x}_c^*) \leq a \times \sigma^2(\boldsymbol{x}_c^\bullet) + c, \end{gathered} \quad (5.38)$$

where $\boldsymbol{x}_c^\bullet = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\operatorname{argmin}}\ \sigma^2(\boldsymbol{x}_c)$, $a \in \{0, [1, \infty)\}$, and $c \geq 0$. In words, the goal is to find $\boldsymbol{x}_c^*$ that minimizes $\mu(\cdot)$ subject to $\sigma^2(\boldsymbol{x}_c^*)$ being close to $\min\limits_{\boldsymbol{x}_c \in \mathcal{X}_c}\ \sigma^2(\boldsymbol{x}_c)$ ($c \geq 0$ and $a \geq 1$) or satisfying a given constraint ($c > 0$ and $a = 0$). To complete specification of the algorithm for finding the M-robust design we need to define the *improvement* criterion for selecting the next control variable value (Step 3), and the *precision* criterion for selecting the next environmental variable value (Step 4). The stopping criterion (Step 5) will be discussed in the examples and in Section 5.5.

Selection of Control Variables

We choose the next control variable site $\boldsymbol{x}_c^*$ to maximize

$$I(\boldsymbol{x}_c) = E\big[\max\{0, M_{min,f} - M(\boldsymbol{x}_c)\} \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}\big] \times P[\text{ constraint } | \boldsymbol{Y}_n, \boldsymbol{\gamma}], \quad (5.39)$$

where the constraint is

$$V(\boldsymbol{x}_c)\ \leq\ a \times v_{n,min} + c. \quad (5.40)$$

Intuitively, the constant $v_{n,min}$ is the minimum of the posterior expected values of $V(\cdot)$ for control variable values in $\boldsymbol{S}_n^C$, and the random variable $M_{min,f}$ is the *minimum* of

$M(\cdot)$ at control variable values in $\boldsymbol{S}_n^c$ that appear to be in the *feasible* region. Formally, we let $v_{n,min} = \min\{E[V(\boldsymbol{s}_{c,i})|\boldsymbol{Y_n}, \boldsymbol{\gamma}] \ : \ 1 \leq i \leq n\}$, and $M_{min,f} = \min\{M(\boldsymbol{s}_{c,i}) \ : \ E[V(\boldsymbol{s}_{c,i})|\boldsymbol{Y_n}, \boldsymbol{\gamma}] \leq a \times v_{n,min} + c\}$. Thus, we choose $\boldsymbol{x}_c^*$ such that

$$\boldsymbol{x}_c^* = \underset{\boldsymbol{x}_c \in \mathcal{X}_c}{\operatorname{argmax}} \ I(\boldsymbol{x}_c).$$

The intuition behind this criterion is as follows. We choose the next control variable $\boldsymbol{x}_c^*$ to maximize the improvement in $M(\cdot)$ over the minimum of $M(\cdot)$ for control variable sites already observed that appear to satisfy the constraint. We multiply this improvement by the probability that the constraint is satisfied so as to "downweight" observations in the infeasible region of the control variable space. The calculations necessary to determine (5.39) are outlined below.

As in Williams (2000b, Chapter 3), $E[\max\{0, M_{min,f} - M(\boldsymbol{x}_c)\} \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}]$ is computed by iterated expectations and Monte Carlo. Recall $\boldsymbol{M}_n = [M(\boldsymbol{s}_{c,1}), ..., M(\boldsymbol{s}_{c,1})]^\top$ and write

$$\begin{aligned} E[\max\{0, M_{min,f} - M(\boldsymbol{x}_c)\} \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}] \quad = \\ E_{\boldsymbol{M}_n|\boldsymbol{Y}_n}\big[E\big(\max\{0, M_{min,f} - M(\boldsymbol{x}_c)\} \mid \boldsymbol{M}_n, \boldsymbol{Y}_n, \boldsymbol{\gamma}\big)\big]. \end{aligned} \tag{5.41}$$

The inner expectation is computed from distribution (5.27), the posterior distribution of $M(\boldsymbol{x}_c)$ given $\boldsymbol{M}_n$, $\boldsymbol{Y}_n$, and $\boldsymbol{\gamma}$. We obtain

$$\begin{aligned} E\big[\max\{0, \ M_{min,f} - M(\boldsymbol{x}_c)\} \mid \boldsymbol{M}_n, \boldsymbol{Y}_n, \boldsymbol{\gamma}\big] \quad = \\ (M_{min,f} - \mu_{M,1}) T_{2n-k}\left(\tfrac{M_{min,f} - \mu_{M,1}}{\sqrt{\tilde{\tau}^2}\sigma_{M,1}}\right) \quad + \\ \tfrac{1}{2n-k}\left[(2n-k)\sqrt{\tilde{\tau}^2}\sigma_{M,1} \ + \ \tfrac{(M_{min,f} - \mu_{M,1})^2}{\sqrt{\tilde{\tau}^2}\sigma_{M,1}}\right] t_{2n-k}\left(\tfrac{M_{min,f} - \mu_{M,1}}{\sqrt{\tilde{\tau}^2}\sigma_{M,1}}\right), \end{aligned} \tag{5.42}$$

where $T_\nu(\cdot)$ and $t_\nu(\cdot)$ denote the standard t cumulative distribution function and density function with ν degrees of freedom, and the values of $\mu_{M,1}$, $\sigma_{M,1}$, and $\tilde{\tau}^2$ are defined below equation (5.27).

The outer expectation is computed via Monte Carlo. A random sample of size N_c is taken from the posterior n-variate t distribution of $\boldsymbol{M}_n$ given $\boldsymbol{Y}_n$ and $\boldsymbol{\gamma}$ in (5.31). For each sample, we calculate $M_{min,f}$ and compute (5.42). The value of (5.41) is then obtained as the average of these N_c numbers. Note that the distribution in (5.31) does not depend on $\boldsymbol{x}_c$ so that the same Monte Carlo sample can be used in calculation of (5.41) for any $\boldsymbol{x}_c$.

Calculation of $P\big[V(\boldsymbol{x}_c) \leq a \times v_{n,min} + c \mid \boldsymbol{Y}_n, \boldsymbol{\gamma}\big]$ is also accomplished via Monte Carlo. First, the constant $v_{n,min} = \min\{E[V(\boldsymbol{s}_{c,i})|\boldsymbol{Y_n}, \boldsymbol{\gamma}] \ : \ 1 \leq i \leq n\}$ is computed using the well known formula for expectations of a quadratic form (see Theorem B.0.6 in the Appendix). Then a random sample of size N_p is selected from the distribution of $Y_{n_e}(\boldsymbol{x}_c)$ given $\boldsymbol{Y}_n$ and $\boldsymbol{\gamma}$ (see (5.22)), and the desired probability is calculated as the proportion of the N_p samples satisfying the constraint.

Selection of Environmental Variables

We select the next environmental variable site $\boldsymbol{x}_e^*$, corresponding to $\boldsymbol{x}_c^*$, to maximize the distance between $(\boldsymbol{x}_c^*, \boldsymbol{x}_e^*)$ and the point in $\boldsymbol{S}_n$ that is closest to $(\boldsymbol{x}_c^*, \boldsymbol{x}_e^*)$. In other words, let $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ be a distance measure and define

$$D(\boldsymbol{x}, \boldsymbol{S}_n) = \min\{d(\boldsymbol{x}, \boldsymbol{s}_{c,i}) \ : \ 1 \leq i \leq n\},$$

to be the distance from $\boldsymbol{x}$ to the closest point in $\boldsymbol{S}_n$. We choose $\boldsymbol{x}_e^*$ as

$$\boldsymbol{x}_e^* = \underset{\boldsymbol{x}_e \in \mathcal{X}_e}{\operatorname{argmax}} \ D[(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{S}_n].$$

Since maximin optimization problems are generally difficult to solve, we will restrict the search for $\boldsymbol{x}_e^*$ to a grid of points in $\mathcal{X}_e$ (a reasonable restriction because $\boldsymbol{X}_e$ is assumed to have discrete support).

5.4 Examples

The following examples illustrate the performance of the M-robust and V-robust sequential algorithms. All calculations are performed using Model (5.11) with $\boldsymbol{f}(\boldsymbol{x}) = 1$ and the product power exponential correlation function

$$R(\boldsymbol{h}) = \prod_{i=1}^{p} \exp\left(-\theta_i |h_i|^{\alpha_i}\right), \tag{5.43}$$

where $\theta_i > 0$, $0 < \alpha_i \leq 2$, so that $\boldsymbol{\gamma} = (\theta_1, ..., \theta_p, \alpha_1, ..., \alpha_p)$.

5.4.1 Simple V-Robust 2-D Example

We illustrate the algorithm for finding V-robust designs with a simple example. Consider the hypothetical $y(\cdot)$ shown in Figure 5.1 that depends on a single control variable and a single environmental variable, each on $(0, 1)$. We assume that $\boldsymbol{X}_e$ has a discretized uniform distribution on the 20 points $\{0.025, 0.075, ..., 0.975, 1\}$. For this example we wish to minimize $\sigma^2(x_c)$ subject to an absolute bound $\mu(x_c) \leq -0.08$.

The first step of the sequential algorithm involves choosing an initial set of design points at which to observe $y(\cdot)$. We use a space filling design, and, following the recommendations of Jones, Schonlau and Welch (1998), use 10 observations per input dimension. Figure 5.7 displays the 20-point maximin distance Latin hypercube design (+'s) used as the starting design for this example. $y(\cdot)$ is evaluated at each of the 20 points in the starting design, and the posterior mode of $\boldsymbol{\gamma}$ is obtained using Model (5.11). Figure 5.6 displays the true $\mu(x_c)$ (left panel) and $\sigma^2(x_c)$ (right panel) along with the posterior means of $M(x_c)$ and $V(x_c)$ given the data from the starting design and the posterior mode of $\boldsymbol{\gamma}$. Note that the constraint $\mu(x_c) \leq -0.08$ (dotted horizontal line in the left panel) restricts x_c to values in the approximate interval

(0.176, 0.30). For this constraint the optimal setting of x_c is on the lower boundary of the feasible region at $x_c = 0.176$. If no points are added to this design and the posterior means of $M(x_c)$ and $V(x_c)$ based on the initial design are optimized we set $x_c = 0.811$, a value of x_c that does not satisfy the constraint of interest but appears to do so in the initial predictions.

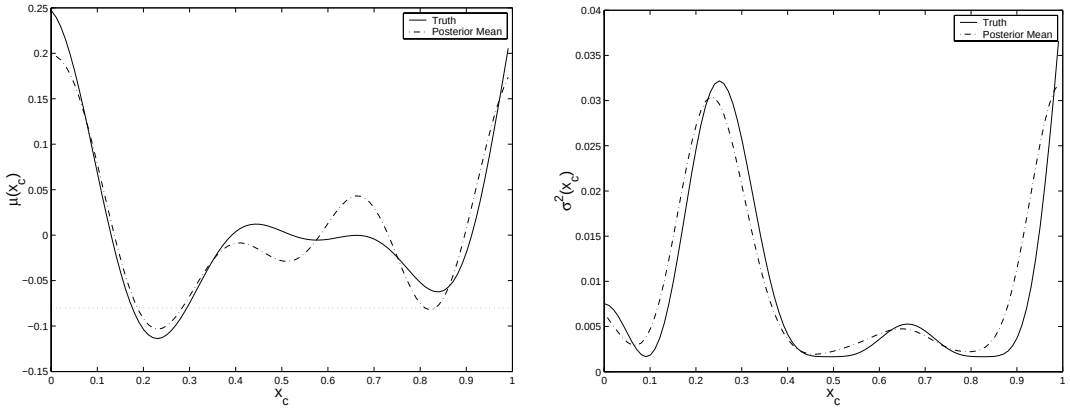


Figure 5.6: True $\mu(\cdot)$ (left panel) and $\sigma^2(\cdot)$ (right panel) and their posterior mean predictors based on the 20-point starting design for the V-robust example.

Using the V-robust sequential algorithm with $N_v = 500$, points are added to the starting design until we reach a predefined stopping criterion. In general, we would like to stop the algorithm when there is no longer any improvement in adding points and/or when prediction in the feasible region is "accurate". For this example, we choose to stop the algorithm when a moving average of the improvement criterion is "small". The definition of "small" is problem specific since it is relative to the values of $\sigma^2(\cdot)$. One means of defining a small improvement is to require that the improvement be a small fraction (eg. a thousandth) of the range for the posterior

expected value of $V(\cdot)$ at the current stage. Here, we stop the algorithm when a 5-point moving average of the improvement is less than 0.00001. Figure 5.7 displays the 18 points that the algorithm added, and Figure 5.8 displays the final posterior means of $M(\cdot)$ and $V(\cdot)$ given the combined 38 (20 initial plus 18) point set of training data. The V-robust optimal value based on the posterior means matches the true V-robust optimal value of $x_c = 0.176$.

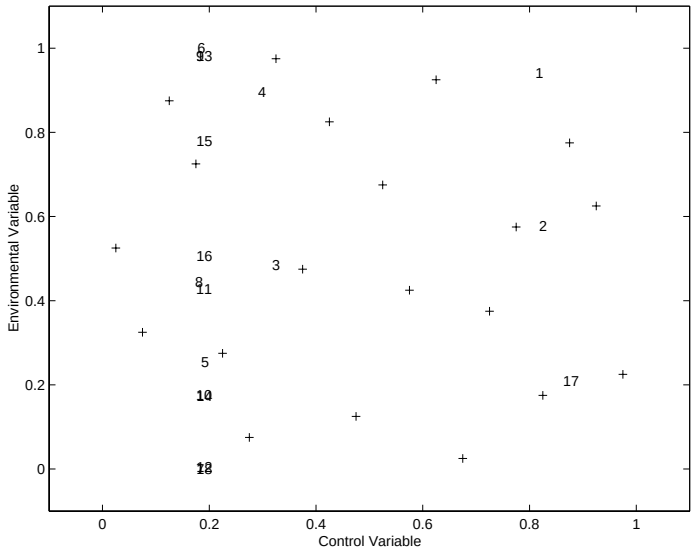


Figure 5.7: Locations of 38 points for the final design. +'s denote the 20 points in the initial design and the numbered sites are the sites added by the V-robust sequential algorithm in that order.

5.4.2 Simple M-Robust 2-D Example

We illustrate the M-robust sequential algorithm again using the hypothetical $y(\cdot)$ shown in Figure 5.1, and the same discretized uniform distribution on the points

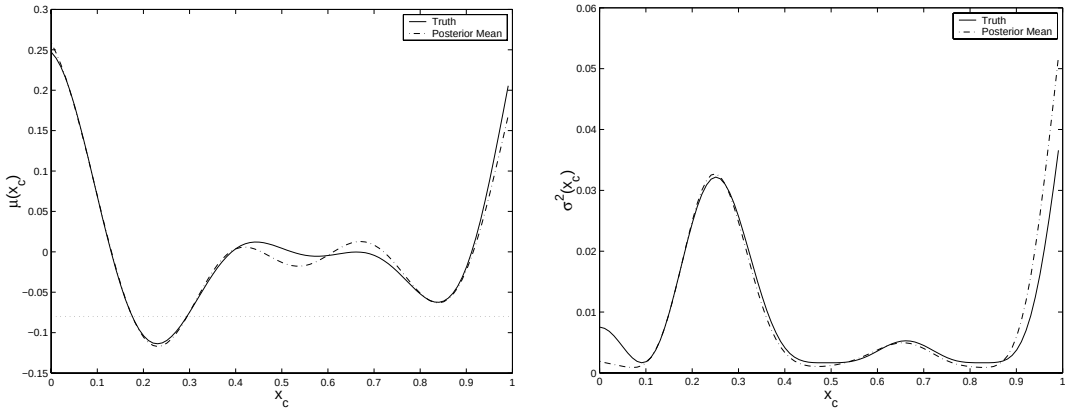


Figure 5.8: True $\mu(\cdot)$ and $\sigma^2(\cdot)$ and their posterior mean predictors based on the 38-point final design (20 points in initial design and 18 points added by the V-robust sequential algorithm).

$\{0.025, 0.075, ..., 0.975, 1\}$ for $\boldsymbol{X}_e$. The M-robust goal is to minimize $\mu(x_c)$ subject to a relative bound on $\sigma^2(x_c)$, $\sigma^2(x_c) \leq 1.1 \times \min_{x_c^* \in \mathcal{X}_c} \sigma^2(x_c^*) + 0.01$ (see Figure 5.9).

The M-robust algorithm is started using the same 20-point maximin distance Latin hypercube design as in Section 5.4.1 (see +'s in Figure 5.10). The posterior mode of $\boldsymbol{\gamma}$ is obtained using Model (5.11) and the initial 20 observations from the starting design. Figure 5.9 displays the true $\mu(x_c)$ (left panel) and $\sigma^2(x_c)$ (right panel) along with the posterior means of $M(x_c)$ and $V(x_c)$ given the data from the starting design and the estimate of $\boldsymbol{\gamma}$. The constraint $\sigma^2(x_c) \leq 1.1 \times \sigma^2(x_c^\bullet) + 0.01$ (dotted horizontal line in right panel) restricts the x_c feasible region to approximately $(0, 0.15) \cup (0.37, 0.95)$, and the value of the true M-robust control variable is $x_c = 0.838$. If no points are added to the initial design and the posterior means of $M(x_c)$ and $V(x_c)$ are used to find the M-robust x_c we set $x_c = 0.822$, the approximate true M-robust value.

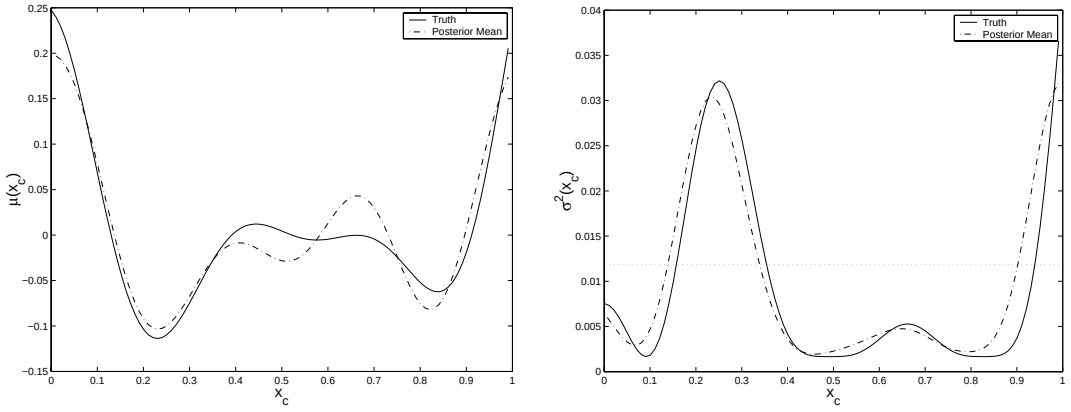


Figure 5.9: True $\mu(\cdot)$ and $\sigma^2(\cdot)$ and their initial posterior mean predictors based on the 20-point starting design for the M-robust example.

Using the M-robust sequential algorithm with $N_c = N_p = 500$, points are added to the starting design until a predefined stopping criterion is met. For this example we stop the algorithm when a 5-point moving average of the improvement criterion is "small", less than 0.00001. A "small" improvement is problem specific since it is relative to the range of the values of $\mu(\cdot)$, and may be set as a fraction of the range of the posterior mean of $M(\cdot)$. Figure 5.10 displays the 17 points that the algorithm added. Note that most of the additional points are added in the $x_c = 0.83$ region, the value of the *constrained* minimum of $\mu(x_c)$, while several are added around the global minimum of $\mu(x_c)$, $x_c = 0.23$. Figure 5.11 displays the final posterior means of $M(\cdot)$ and $V(\cdot)$ given the 37 point set of training data (20 initial plus 17 added). The M-robust optimal value based on the posterior means matches the true M-robust optimal value of $x_c = 0.838$.

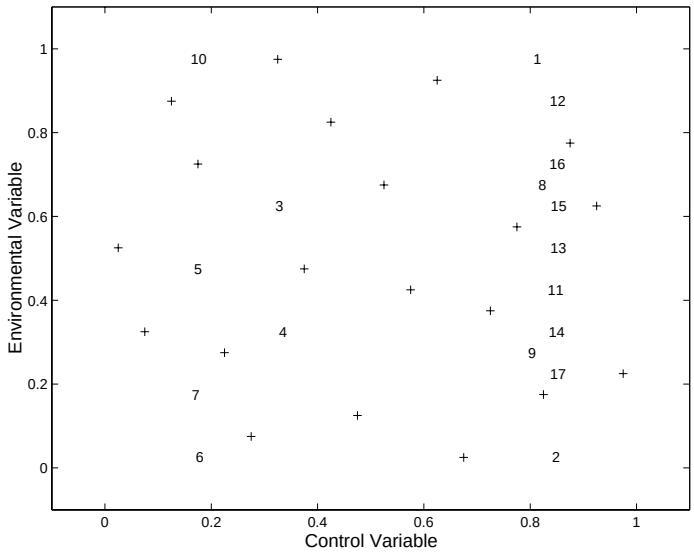


Figure 5.10: Locations of 37 points for the final design. +'s denote the 20 points in the initial design and the numbered sites are the sites added by the sequential algorithm in that order.

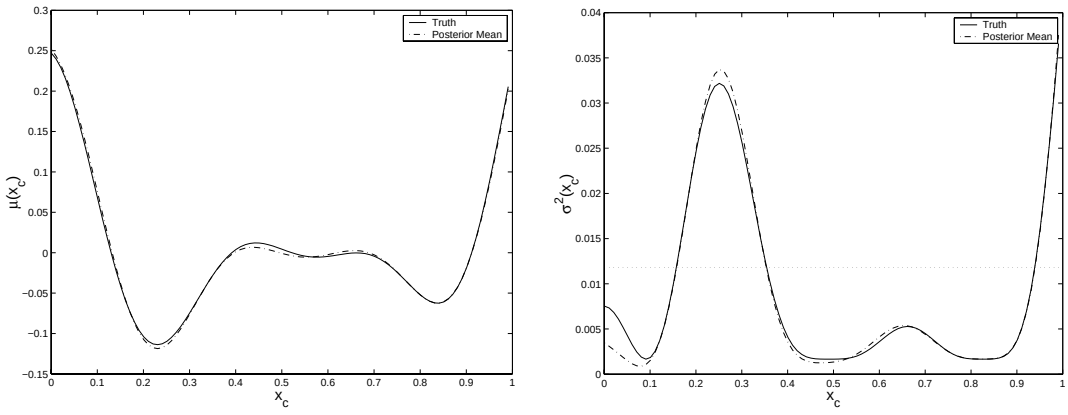


Figure 5.11: True $\mu(\cdot)$ and $\sigma^2(\cdot)$ and their final posterior mean predictors based on the 37-point final design (20 points in initial design and 17 points added by the M-robust sequential algorithm).

5.4.3 A 4-D Example

In this example, the Branin function of Dixon and Szego (1978), defined on $\mathcal{X} = [-5, 10] \times [0, 15]$ by

$$z(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10$$

is used in the specification of the true reponse. The true response function has four inputs and is set to be

$$y(x_1, x_2, x_3, x_4) = \frac{1}{30} z(x_1, x_2) z(x_3, x_4) + (x_1 - \pi)^2$$

with x_1 and x_2 being the control variables, $\boldsymbol{x}_c = (x_1, x_2)$, and x_3 and x_4 being the environmental variables, $\boldsymbol{x}_e = (x_3, x_4)$. Table 5.1 lists the assumed joint distribution of the environmental variables and Figure 5.12 displays the true $\mu(\boldsymbol{x}_c)$ (left panel) and the true $\sigma^2(\boldsymbol{x}_c)$ (right panel).

		x_3			
		-2	1	4	7
x_4	3.75	0.0375	0.0875	0.0875	0.0375
	7.5	0.0750	0.1750	0.1750	0.0750
	11.25	0.0375	0.0875	0.0875	0.0375

Table 5.1: Joint distribution of environmental variables.

We search for the M-robust control variable setting that minimizes $\mu(\boldsymbol{x}_c)$ subject to $\sigma^2(\boldsymbol{x}_c) < 10000$. The left panel of Figure 5.13 plots the $\boldsymbol{x}_c$ feasible region along with the global minimum of $\mu(\boldsymbol{x}_c)$, which occurs at the point $(\pi, 2.275)$ (denoted by $\triangle$ in the figure).

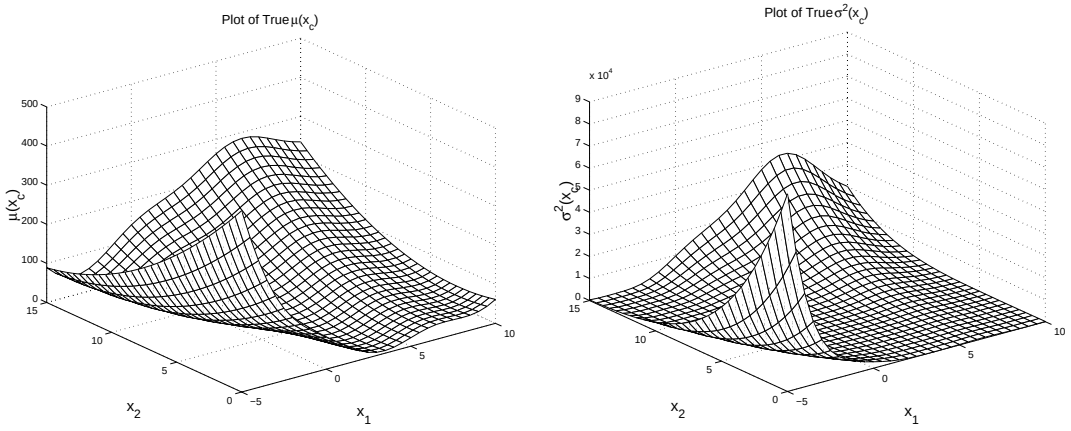


Figure 5.12: Plot of true $\mu(\boldsymbol{x}_c)$ (left panel) and true $\sigma^2(\boldsymbol{x}_c)$ (right panel) for 4-d example.

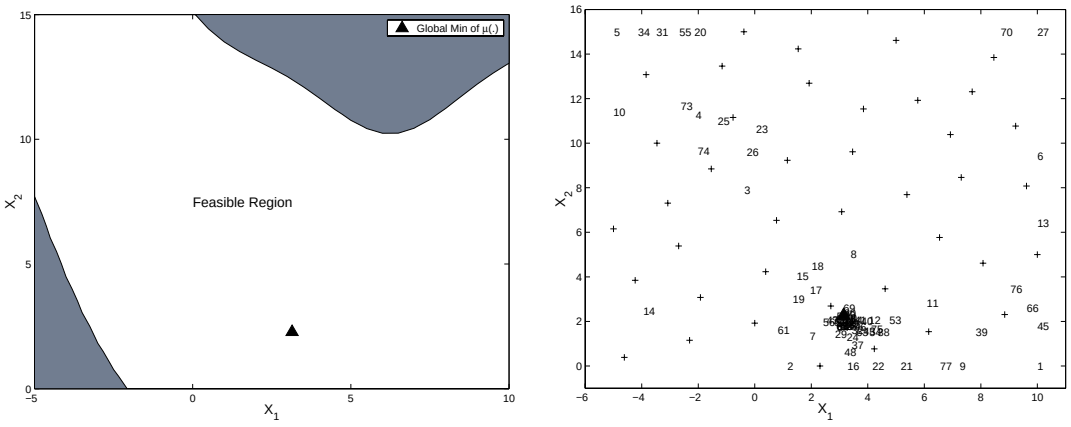


Figure 5.13: Plot of $\boldsymbol{x}_c$ feasible region (left panel) and 120 point final design projected into the control variable space (right panel). The +'s indicate initial design sites and the numbers indicate the additional design sites in the order they were chosen.

We start the M-robust sequential algorithm by computing the response on a 40-point (again 10 observations for each dimension) maximin Latin hypercube design, fitting Model (5.11), and plotting the posterior means of $M(\cdot)$ and $V(\cdot)$ based on

the initial design in Figure 5.14. Setting $N_c = N_p = 1000$ and using the M-robust criterion defined above, 80 points are added to the initial design. The right panel of Figure 5.13 plots the set of 120 (40 initial plus 80 added) final design sites projected onto the control variable space, and Figure 5.15 plots the posterior means of $M(\cdot)$ and $V(\cdot)$ based on the final 120 point design. Note the improvement in accuracy of the final predictors over the initial predictors, and note that the algorithm performs as desired by adding sites around the feasible region and slowly zeroing in on the true M-robust value of $(\pi, 2.275)$. Table 5.2 lists the value of the improvement and the predicted constrained minimizer as points are added. After 80 points are added, the final predicted M-robust value is $(3.15, 2.25)$.

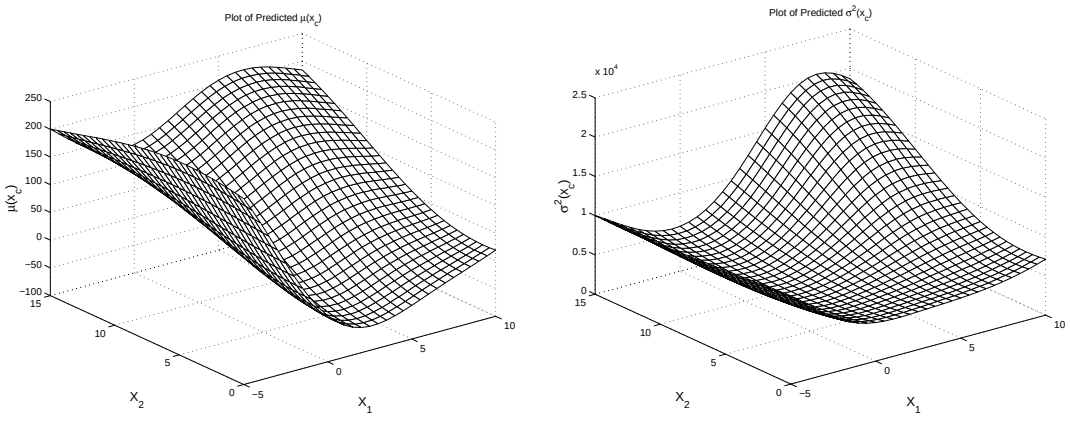


Figure 5.14: Posterior mean predictors of $\mu(\boldsymbol{x}_c)$ (left panel) and $\sigma^2(\boldsymbol{x}_c)$ (right panel) based on the initial 40 point design.

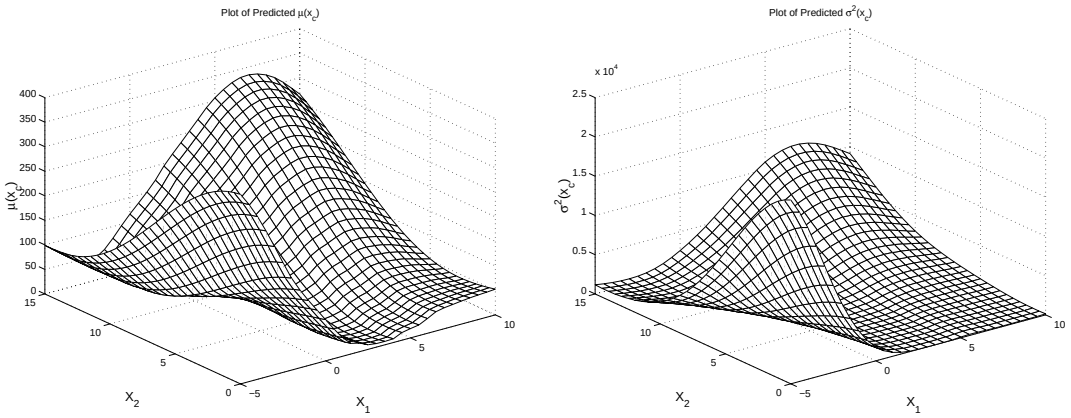


Figure 5.15: Posterior mean predictors of $\mu(\boldsymbol{x}_c)$ (left panel) and $\sigma^2(\boldsymbol{x}_c)$ (right panel) based on the final 120 point design.

# Points Added	Improvement	Predicted Minimizer
1	2.904673	(2.65, 0.00)
40	0.236627	(3.31, 2.02)
60	0.020322	(2.92, 1.98)
80	0.019231	(3.15, 2.25)

Table 5.2: Summary results for 4-D example

5.5 Discussion

The numerical optimization in Steps 2 and 3 of the algorithms can be computationally challenging. In Step 2, the algorithm calls for optimization of the posterior distribution of $\boldsymbol{\gamma}$ at each stage to obtain the posterior mode of $\boldsymbol{\gamma}$. Computational savings can be obtained by updating correlation parameter estimates only after groups

of points have been added to the existing design and/or by using the previous correlation parameter estimates as a starting point for the current stage's numerical optimization.

In Step 3, the algorithm calls for numerical optimization of the improvement to obtain the next control variable value at which to observe $y(\cdot)$. The improvement surface typically contains many local optima, making numerical optimization difficult without good starting values for the optimization. We obtain promising starting values by evaluating the improvement criterion on a grid of points in $\mathcal{X}_c$. This helps the optimization avoid local optima in the improvement surface.

In Step 4 of these algorithms we select the next environmental variable site at which to observe $y(\cdot)$. Two methods of selecting the next environmental variable site are presented. In the V-robust sequential design algorithm the next environmental variable site is chosen to minimize the MSE of prediction for $M(\boldsymbol{x}_c^*)$. However, in the M-robust sequential design algorithm we choose the next environmental variable value by maximizing the minimum distance between the current design and the next point. In other words, given the next control variable site $\boldsymbol{x}_c^*$ we choose $\boldsymbol{x}_e$ so that the distance between $(\boldsymbol{x}_c^*, \boldsymbol{x}_e)$ and any other point already in the design is maximized. The results of McKay et al. (1979) suggest that, for a fixed set of values of the control variables, the environmental variables should be chosen as a Latin hypercube design if we intend to compute a mean over the values of the environmental variables. Since we are only adding one observation at a time, the distance based criterion attempts to spread out observations much like a Latin hypercube. In addition the distance based choice of $\boldsymbol{x}_e$ has the advantage of ease of computation. For this reason, we suggest using the distance criterion to choose the next environmental variable site.

Comparing Figures 5.7 and 5.10 it appears that the final design obtained using the distance based choice of $\boldsymbol{x}_e$ manages to spread out the observations more uniformly.

As seen in the examples, the stopping criterion for the algorithm is problem specific. Generally, we want to stop the algorithm when our predictions are *accurate* and/or the improvement is small. For accurate prediction, an intuitive stopping criterion is based on the *leave one out* mean square prediction error being "small". For optimization, a stopping criterion based on the improvement is more appropriate. In general, the improvement at each stage decreases. However, due to updated correlation parameter estimates and additional information from the new observed response, it is possible to find improvements larger than previously observed. For this reason, we suggest stopping the optimization algorithm when a moving average of the improvement is "small". The actual value of a "small" improvement depends on the scale of the objective function. For the V-robust algorithm a "small" improvement may be determined as a small fraction of the range of the posterior expected values of $V(\cdot)$, and for the M-robust algorithm a "small" improvement may be a small fraction of the range of the posterior expected values of $M(\cdot)$.

Both the V-robust and M-robust sequential algorithms require a starting design at which the responses are calculated. In the examples above, maximin distance LHS designs were used as starting designs. However, any other *space filling* design may be appropriate for the goals presented above. Intuitively, since all points in the input space are equally likely to be the location of an optimum, all portions of the input space are equally important to observe in the initial stage. Thus, designs that spread out observations in order to "cover" the input space (i.e. space filling designs) seem

a natural choice. Generating space filling designs and determining which are best suited to computer experiments is an area of active research.

In addition to the location of the sites in the initial design we must also consider the number of sites to include in the initial design. Allocation of runs to the initial design and subsequent sequential design can be an important component of minimizing the number of runs of the code necessary to find the desired optimal value. Too many sites in the initial design will "waste" observations, while too few may lead to poor correlation parameter estimates and a larger number of sequential steps to find the optimum. In the examples, we use the Jones, Schonlau and Welch (1998) suggestion of 10 observations per input dimension, which is a reasonable rule of thumb.

In this chapter we have considered the case of a single deterministic response $y(\cdot)$ that depends on both control and environmental variables. Straightforward extensions of these concepts can be made for the case where the single response also involves measurement error. However, many computer experiments involve multiple related responses. For these situations, extensions of the above algorithms and concepts may be appropriate and are an open area of research.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

This dissertation has been concerned with the design and analysis of data from a computer experiment. The modeling approach we have taken is Bayesian with statistical models representing the prior beliefs about the relationship between the inputs and the response. We model the true response as a stochastic process or random function with a parametric correlation function having unknown parameters. Chapter 2 compares the predictive ability of stochastic process models using different parameter estimation procedures and standard regression based cubic polynomial models. For the design considered and the variety of "true" responses investigated, we show that the stochastic process models we propose, which use a simple mean structure and a complex correlation structure, have much higher predictive ability then simple polynomial regression models having a more complex mean structure and no correlation structure. Of course, model (1.1) allows a complex mean structure and a complex correlation structure, which may also prove useful, and perhaps preferrable, for prediction purposes. In addition, we show that REML or ML should be used to estimate the correlation parameters of the stochastic process model. Future research in this area involves comparing the predictive ability of the various models for different design strategies and for a larger variety of "true" surfaces.

The experimental design aspect of computer experiments must consider the scientific or engineering goals of the experimenters. For exploratory purposes, space filling designs, such as those used in Chapter 2, are intuitively (and empirically) appealing, while for optimization of computer responses or integrals of computer responses, sequential design strategies, such as those proposed in Chapters 3, 4, and 5, are appropriate.

These sequential design strategies involve the commonly occurring situation where the input variables consist of both control variables, which can be set by the product designer, and environmental variables, which are uncontrollable but vary according to some probability distribution. In this situation, we study the distribution of the response that is induced by the distribution of the environmental variables. We propose sequential experimental design strategies for the constrained optimization of various characteristics of this induced distribution. For univariate responses, interest is in finding values of the control variables that are robust to the environmental variable setting, and, for bivariate responses, interest is in optimizing the expectation of the first response subject to a constraint on the expectation of the second response, where the expectation is taken over the distribution of the environmental variables. We present motivations for these algorithms and examples of their performance in various settings.

The algorithms in Chapters 3 and 5 are highly computational and we suggest strategies for reducing the computational costs of these algorithms. The primary computational cost is in the estimation of the correlation parameters corresponding to the data model, particularly when there are a large number of inputs or a large

number of observations. One approach to reducing this cost is to update the correlation parameter estimates only after groups of points have been added to the existing design, especially as the number of observations increases and the parameter estimates remain fairly stable from one iteration to the next. If further reductions in computational costs are desired, simpler correlation structures that reduce the number of correlation parameters, such as isotropic versions of the power exponential and Matérn correlation functions, and/or Bayesian Markov chain methods for sampling from the posterior distribution of the correlation parameters (thereby avoiding their expensive estimation) may prove useful and are viable avenues for future research.

Another area for future research in computer experiments involves the combination of information from different sources, as may be the case where both data from a computer experiment and a "real" experiment are available. The multi-response models mentioned in Section 1.1.5 and other hierarchical models may be ideal for this situation since they allow for different data sources. In addition, hierarchical models are able to account for different observation scales, for multiple measurement errors, and can accommodate high-dimensional problems with huge amounts of data, making them suitable for those computer experiments where large amounts of data can be collected.

Finally, future work also involves investigation of computer experiments where the inputs consist of all three types of inputs introduced in Chapter 1: control variables, environmental variables, *and* model variables. The treatment of these model variables when describing the "optimal" or "robust" choice of control variables is an area of open research.

APPENDIX A

PROPERTIES OF RANDOM PROCESSES AND THEIR CORRELATION FUNCTIONS

The relationship between the properties of the correlation function and the properties of the corresponding random process is important to understand when trying to choose a parametric family of correlation functions. In the following, we give an overview of this relationship. For more details of the definitions and theorems presented, the reader is referred to Adler (1981 chapters 2 and 3) or Cramér and Leadbetter (1967 Chapters 4, 7 and 9).

Let Z be a random field (stochastic process) on $\mathbb{R}^p$. We need to be able to "learn" about Z based on partial information from a single sample path. To this end we will assume that Z is stationary, which essentially means that Z behaves similarly in different parts of the input space.

Definition A.0.1. A stochastic process $Z(\boldsymbol{x})$ is *strictly stationary* provided that for any $k \geq 1$, any $\boldsymbol{x}_1, ..., \boldsymbol{x}_k \in \mathbb{R}^p$ and any $\boldsymbol{h} \in \mathbb{R}^p$ we have

$$\mathcal{L}(Z(\boldsymbol{x}_1), ..., Z(\boldsymbol{x}_k)) = \mathcal{L}(Z(\boldsymbol{x}_1 + \boldsymbol{h}), ..., Z(\boldsymbol{x}_k + \boldsymbol{h})),$$

i.e., the distribution of $(Z(\boldsymbol{x}_1), ..., Z(\boldsymbol{x}_k))$ is equivalent to the distribution of $(Z(\boldsymbol{x}_1 + \boldsymbol{h}), ..., Z(\boldsymbol{x}_k + \boldsymbol{h}))$.

Definition A.0.2. A stochastic process $Z(\boldsymbol{x})$ is *weakly stationary* or *second order stationary* if it has second moments, $E[Z(\boldsymbol{x})]$ is independent of $\boldsymbol{x}$ and $\text{Cov}[Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)] = K(\boldsymbol{x}_2 - \boldsymbol{x}_1)$.

Definition A.0.3. A stochastic process $Z(\boldsymbol{x})$ is *mean square (MS) continuous at* $\boldsymbol{x}_0$ if for all $\{\boldsymbol{x}_n\}$ with $\boldsymbol{x}_n \to \boldsymbol{x}_0$ we have

$$E[(Z(\boldsymbol{x}_n) - Z(\boldsymbol{x}_0))^2] \to 0.$$

$Z(\boldsymbol{x})$ is *mean square continuous on* $\mathcal{X}$ if it is mean square continuous at each $\boldsymbol{x} \in \mathcal{X}$.

Definition A.0.4. A stochastic process $Z(\boldsymbol{x})$ is *mean square differentiable at* $\boldsymbol{x}_0 \in \mathcal{X}$ *in direction* j provided there is a value $Z_j^{(1)}(\boldsymbol{x}_0)$ such that for any sequence of real numbers $h_n \to 0$

$$E[(\frac{Z(\boldsymbol{x}_0 + h_n \boldsymbol{e}_j) - Z(\boldsymbol{x}_0)}{h_n} - Z_j^{(1)}(\boldsymbol{x}_0))^2] \to 0$$

where $\boldsymbol{e}_j$ is the unit vector with a 1 in position j and zeroes elsewhere. $Z(\boldsymbol{x})$ is *mean square differentiable on* $\mathcal{X}$ *in direction* j if it is mean square differentiable in direction j at every $\boldsymbol{x} \in \mathcal{X}$. The process $Z_j^{(1)}(\boldsymbol{x}_0)$ is called the *mean square derivative* of $Z(\boldsymbol{x})$ in direction j.

Note that the definitions of mean square continuity and mean square differentiability are not properties of the sample paths of the process. In other words, $Z(\cdot)$ being mean square differentiable does not imply that a realization of $Z(\cdot)$ will be differentiable. Realizations of $Z(\cdot)$ being continuous or differentiable are stronger conditions.

Definition A.0.5. A stochastic process $Z(\boldsymbol{x})$ is *almost surely continuous at* $\boldsymbol{x}_0$ if

$$P\{\omega : Z(\cdot, \omega) \text{ is continuous at } \boldsymbol{x}_0\} = 1$$

Definition A.0.6. A stochastic process $Z(\boldsymbol{x})$ is *almost surely differentiable at* $\boldsymbol{x}_0$ *in direction* j if

$$P\{\omega : Z(\cdot, \omega) \text{ is differentiable in direction } j \text{ at } \boldsymbol{x}_0\} = 1$$

So, $Z(\boldsymbol{x})$ is almost surely continuous (differentiable) if realizations of $Z(\boldsymbol{x})$ are continuous (differentiable) with probability 1. In this probability statement ω is the random variable and it denotes the realization of the stochastic process $Z(\cdot)$.

The properties described above are intimately related to the properties of the covariance function of the random field. In the following discussion assume that $Z(\cdot)$ is a weakly stationary process with mean zero, variance σ^2 and correlation function $R(\boldsymbol{h})$.

Theorem A.0.1. $Z(\cdot)$ is MS continuous on $\mathcal{X}$ if and only if $R(\boldsymbol{h})$ is continuous at $\boldsymbol{h} = 0$.

Proof. Fix $\{\boldsymbol{x}_n\}$ with $\boldsymbol{x}_n \to \boldsymbol{x}_0$ then

$$E[(Z(\boldsymbol{x}_n) - Z(\boldsymbol{x}_0))^2] = 2\sigma^2(1 - R(\boldsymbol{x}_n - \boldsymbol{x}_0))$$

which converges to 0 if and only if R is continuous at 0.

Theorem A.0.2. $Z(\cdot)$ is MS differentiable on $\mathcal{X}$ in direction j provided

$$\left.\frac{\partial^2 R(\boldsymbol{h})}{\partial h_j^2}\right|_{\boldsymbol{h}=0}$$

exists. The covariance function for the derivative process $Z_j^{(1)}(\boldsymbol{x}_0)$ is then $-\sigma^2 \frac{\partial^2 R(\boldsymbol{h})}{\partial h_j^2}$. This can be generalized to $Z(\cdot)$ being m-times mean-square differentiable in direction j provided that the $2m^{th}$ partial derivative of $R(\boldsymbol{h})$ in direction j exists.

Proof. see Adler (1981) Theorem 2.2.2.

Theorem A.0.3. Suppose $Z(\cdot)$ is a weakly stationary Gaussian process over an open set $\mathcal{X} \subset \mathbb{R}^p$ that has mean zero, variance σ^2 and continuous correlation function $R(\cdot)$. Then $Z(\cdot)$ is almost surely continuous on $\mathcal{X}$ provided one of the following hold:

1. There is a finite c, $0 < c < \infty$ and $\epsilon > 0$ such that

$$R(0) - R(\boldsymbol{h}) \leq \frac{c}{|log(||\boldsymbol{h}||)|^{1+\epsilon}}$$

   for all $\boldsymbol{h} \in \mathbb{R}^p$.

2. There is an $\epsilon > 0$ such that

$$\int_{\mathbb{R}^p} (\log(1 + ||\boldsymbol{w}||))^{1+\epsilon} f(\boldsymbol{w}) d\boldsymbol{w} < \infty$$

   where $R(\boldsymbol{h}) = \int_{\mathbb{R}^p} cos(\boldsymbol{w}^\top \boldsymbol{h}) f(\boldsymbol{w}) d\boldsymbol{w}$. The function $f(\boldsymbol{w})$ is called the *spectral density* corresponding to $R(\boldsymbol{h})$.

Proof. see Adler (1981) Theorem 3.4.1 and 3.4.3.

Theorem A.0.4. Suppose $Z(\cdot)$ is a weakly stationary Gaussian process over an open set $\mathcal{X} \subset \mathbb{R}^p$ that has mean zero, variance σ^2 and correlation function $R(\cdot)$ that has continuous second partial derivatives in direction j. Then $Z(\cdot)$ is almost surely differentiable on $\mathcal{X}$ in direction j provided one of the following holds:

1. There is a finite c, $0 < c < \infty$ and $\epsilon > 0$ such that

$$\frac{\partial^2 R(\boldsymbol{h})}{\partial h_j^2} - \frac{\partial^2 R(0)}{\partial h_j^2} \leq \frac{c}{|(log||\boldsymbol{h}||)|^{1+\epsilon}}$$

2. There is an $\epsilon > 0$ such that

$$\int_{\mathbb{R}^p} (\log(1 + ||\boldsymbol{w}||))^{1+\epsilon} w_j^2 f(\boldsymbol{w}) d\boldsymbol{w} < \infty$$

where $R(\boldsymbol{h}) = \int_{\mathbb{R}^p} cos(\boldsymbol{w}^\top \boldsymbol{h}) f(\boldsymbol{w}) d\boldsymbol{w}$. The function $f(\boldsymbol{w})$ is called the *spectral density* corresponding to $R(\boldsymbol{h})$.

Proof. The proof follows from Theorem A.0.3. Note that since the correlation function $R(\cdot)$ has continuous second partial derivatives in direction j, we know, by Theorem A.0.2, that the derivative process $Z_j^{(1)}(\boldsymbol{x})$ exists and has covariance function $-\sigma^2 \frac{\partial^2 R(\boldsymbol{h})}{\partial h_j^2}$. Conditions 1 or 2 guarantee that the derivative process $Z_j^{(1)}(\boldsymbol{x})$ is almost surely continuous by Theorem A.0.3, so, certainly $Z_j^{(1)}(\boldsymbol{x})$ almost surely exists.

Theorem A.0.5. Suppose $Z(\cdot)$ is a weakly stationary Gaussian process over an open set $\mathcal{X} \subset \mathbb{R}^p$ that has mean zero, variance σ^2 and correlation function $R(\cdot)$ that has continuous $2m$ partial derivatives in direction j. Then $Z(\cdot)$ is almost surely m-times differentiable on $\mathcal{X}$ in direction j provided one of the following holds:

1. There is a finite c, $0 < c < \infty$ and $\epsilon > 0$ such that

$$(-1)^m \frac{\partial^{(2m)} R(\boldsymbol{h})}{\partial h_j^{(2m)}}\bigg|_{\boldsymbol{h}=0} - (-1)^m \frac{\partial^{(2m)} R(\boldsymbol{h})}{\partial h_j^{(2m)}} \leq \frac{c}{|(log||\boldsymbol{h}||)|^{1+\epsilon}}$$

2. There is an $\epsilon > 0$ such that

$$\int_{\mathbb{R}^p} (\log(1 + ||\boldsymbol{w}||))^{1+\epsilon} w_j^{2m} f(\boldsymbol{w}) d\boldsymbol{w} < \infty$$

where $R(h) = \int_{\mathbb{R}^p} cos(\boldsymbol{w}^\top \boldsymbol{h}) f(\boldsymbol{w}) d\boldsymbol{w}$. The function $f(\boldsymbol{w})$ is called the *spectral density* corresponding to $R(\boldsymbol{h})$.

Proof. The proof follows in the same manner as Theorem A.0.4.

Applying these theorems to the stationary Power Exponential correlation function, defined for $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^p$ as

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{p} \exp\big(- \theta_i |x_{1,i} - x_{2,i}|^{\alpha_i}\big),$$

where $\theta_i > 0$ and $0 < \alpha_i \leq 2$, we note that if $\alpha_i = 2$, then the Z process is infinitely mean square differentiable in direction i, and if all $\alpha_i = 2$ then the sample paths are infinitely differentiable. We can demonstrate the infinitely mean square differentiable claim by simply noting that $R(\boldsymbol{h})$ is infinitely differentiable in the i^{th} direction and applying Theorem A.0.2. The infinite differentiability of the sample paths follows from noting that,

$$\begin{aligned} \int_{\mathbb{R}^p} \cos(\boldsymbol{w}^\top \boldsymbol{h}) \prod_{i=1}^{p} \frac{1}{2\sqrt{\pi\theta_i}} e^{-w_i^2/4\theta_i} dw_i &= \int_{\mathbb{R}^p} \prod_{i=1}^{p} \cos(w_i h_i) \frac{1}{2\sqrt{\pi\theta_i}} e^{-w_i^2/4\theta_i} dw_i \\ &= R(\boldsymbol{h}). \end{aligned}$$

Thus the spectral density is $f(\boldsymbol{w}) = \prod_{i=1}^{p} \frac{1}{2\sqrt{\pi\theta_i}} e^{-w_i^2/4\theta_i}$, and we can check that the condition of Theorem A.0.5 holds for all m. For any other values of α_i we know that the process is mean-square continuous, because $R(\boldsymbol{h})$ is continuous at 0, but is not mean-square differentiable in the i^{th} direction, since $R(\boldsymbol{h})$ is not differentiable in the i^{th} coordinate.

For the Matérn correlation function, defined for $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^p$ as

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{p} \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{2\sqrt{\nu}|x_{1,i} - x_{2,i}|}{\theta_i} \right)^{\nu} \mathrm{K}_\nu \left(\frac{2\sqrt{\nu}|x_{1,i} - x_{2,i}|}{\theta_i} \right), \qquad \text{(A.1)}$$

where $\nu > 0$, $\theta_i > 0$ and $\mathrm{K}_\nu(\cdot)$ is the modified Bessel function of order ν, we note that the parameter ν controls the smoothness of $Z(\cdot)$ in that $Z(\cdot)$ is m times mean-square differentiable if and only if $\nu > m$ (see Stein (1999) Section 2.7), and its sample paths are almost surely m times differentiable if $\nu > m$ (see Cramér and Leadbetter (1967)

Secs. 9.2-9.5). This follows from noting that the spectral density for the Matérn correlation function is a t-density with ν degrees of freedom (see Stein (1999) Section 2.7), and applying Theorem A.0.5.

So, what does all this mean to the stochastic process modeler? If we are sure that the function producing the response is very smooth, and even infinitely differentiable, then we should use the power exponential correlation function and set $\alpha_i = 2$ for all i. This highlights one of the objections to the power exponential class. Either the sample paths are assumed to be infinitely differentiable, or they are not differentiable at all. The Matérn family of correlation functions offers an alternative family of correlation functions which includes a parameter that controls the degree of smoothness of the sample paths, and thus seems more flexible when dealing with surfaces having unknown smoothness properties.

APPENDIX B

LEMMAS AND THEOREMS FOR POSTERIOR CALCULATIONS

The definitions, lemmas and theorems in this section are useful in deriving the formulas and equations in Chapters 1, 3, and 5.

Definition B.0.7. The density of the q-variate Normal distribution $\mathcal{N}_q(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given by:

$$f(\boldsymbol{x}) \;=\; \frac{1}{(2\pi)^{q/2}|\boldsymbol{\Sigma}|^{1/2}}\exp[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})],$$

for $\boldsymbol{x} \in \mathbb{R}^q$.

Definition B.0.8. The density of the q-variate t distribution $\mathcal{T}_q(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$ is given by:

$$f(\boldsymbol{x}) \;=\; \frac{\Gamma[(\nu+q)/2]}{|\boldsymbol{\Sigma}|^{1/2}(\nu\pi)^{q/2}\Gamma[\nu/2]}\left(1 \;+\; \frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)^{-(\nu+q)/2}$$

for $\boldsymbol{x} \in \mathbb{R}^q$. See Berger (1985) for several properties of this distribution (e.g. Mean $= \boldsymbol{\mu}$ if $\nu > 1$ and Covariance matrix $= \nu\boldsymbol{\Sigma}/(\nu-2)$ if $\nu > 2$).

Theorem B.0.6. Suppose $\boldsymbol{U}$ is a $q \times 1$ vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Let $Q(\boldsymbol{U}) = \boldsymbol{U}^\top \boldsymbol{A}\boldsymbol{U}$, where $\boldsymbol{A}$ is a known symmetric matrix, then

$$E[Q(\boldsymbol{U})] = \boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu} + \text{Trace}(\boldsymbol{A}\boldsymbol{\Sigma}).$$

Proof. The proof can be found in most standard books on linear models (see Seber (1977) for example).

In particular, for $\boldsymbol{U} \sim \mathcal{T}_q(\boldsymbol{\mu}, \boldsymbol{V}, \nu)$ we have $E[\boldsymbol{U}^\top \boldsymbol{A}\boldsymbol{U}] = \frac{\nu}{\nu-2}\text{Trace}(\boldsymbol{V}\boldsymbol{A}) + \boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu}$.

Theorem B.0.7. Suppose $\boldsymbol{U}_i$ for $i \in 1, 2$ denote $q_i \times 1$ random vectors having the Gaussian distribution

$$\begin{pmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_2 \end{pmatrix} \mid \boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}_{q_1+q_2}\left[\begin{pmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{pmatrix} \boldsymbol{\beta}, \sigma^2 \begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{pmatrix}\right],$$

where $\boldsymbol{\beta} \in \mathbb{R}^k$ and $\sigma^2 > 0$. Assuming that each of the elements of $\boldsymbol{F}_i$ and $\boldsymbol{R}_{ij}$ are known, each $\boldsymbol{F}_i$ has full column rank, and the correlation matrix is positive definite. Then

$$\boldsymbol{U}_1 | \boldsymbol{U}_2, \boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}_{q_1}\left(\boldsymbol{m}_{1|2}, \sigma^2 \boldsymbol{R}_{1|2}\right),$$

where $\boldsymbol{m}_{1|2} = \boldsymbol{F}_1\boldsymbol{\beta} + \boldsymbol{R}_{12}\boldsymbol{R}_{22}^{-1}(\boldsymbol{Y}_2 - \boldsymbol{F}_2\boldsymbol{\beta})$, and $\boldsymbol{R}_{1|2} = \boldsymbol{R}_{11} + \boldsymbol{R}_{12}\boldsymbol{R}_{22}^{-1}\boldsymbol{R}_{12}^\top$.

Proof. The proof can be found in most standard books on linear models (see Seber (1977) for example).

The following Lemma appears in O'Hagan (1992).

Lemma B.0.1. (O'Hagan (1992)) Suppose $\boldsymbol{U}_i$ for $i \in 1, 2$ denote $q_i \times 1$ random vectors having the Gaussian distribution

$$\begin{pmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_2 \end{pmatrix} \mid \boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}_{q_1+q_2}\left[\begin{pmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{pmatrix} \boldsymbol{\beta}, \sigma^2 \begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{pmatrix}\right],$$

where $\boldsymbol{\beta} \in \mathbb{R}^k$ and $\sigma^2 > 0$. Assuming that each of the elements of $\boldsymbol{F}_i$ and $\boldsymbol{R}_{ij}$ are known, each $\boldsymbol{F}_i$ has full column rank, the correlation matrix is positive definite and the parameter vector $\boldsymbol{\beta}, \sigma^2$ has the noninformative prior $[\boldsymbol{\beta}, \sigma^2] \propto 1/\sigma^2$, the posterior

distribution of $\boldsymbol{U}_1$ given $\boldsymbol{U}_2$ is q_1-variate t: $[\boldsymbol{U}_1 \mid \boldsymbol{U}_2] \sim \mathcal{T}_{q_1}(\boldsymbol{m}_{1|2}, \hat{\sigma^2}\boldsymbol{R}_{1|2}, q_2 - k)$ where:

$$\boldsymbol{m}_{1|2} = \boldsymbol{F}_1\hat{\boldsymbol{\beta}} + \boldsymbol{R}_{12}\boldsymbol{R}_{22}^{-1}(\boldsymbol{U}_2 - \boldsymbol{F}_2\hat{\boldsymbol{\beta}}),$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}_2^\top \boldsymbol{R}_{22}^{-1}\boldsymbol{F}_2)^{-1}\boldsymbol{F}_2^\top \boldsymbol{R}_{22}^{-1}\boldsymbol{U}_2,$$

$$\hat{\sigma^2} = \frac{\boldsymbol{U}_2^\top \boldsymbol{R}_{22}^{-1}\boldsymbol{U}_2 - \hat{\boldsymbol{\beta}}^\top(\boldsymbol{F}_2^\top \boldsymbol{R}_{22}^{-1}\boldsymbol{F}_2)\hat{\boldsymbol{\beta}}}{q_2 - k}$$

$$\begin{aligned}\boldsymbol{R}_{1|2} \;&=\; \boldsymbol{R}_{11} - \boldsymbol{R}_{12}\boldsymbol{R}_{22}^{-1}\boldsymbol{R}_{12}^\top + \\ &\quad (\boldsymbol{F}_1 - \boldsymbol{R}_{12}\boldsymbol{R}_{22}^{-1}\boldsymbol{F}_2)(\boldsymbol{F}_2^\top \boldsymbol{R}_{22}^{-1}\boldsymbol{F}_2)^{-1}(\boldsymbol{F}_1 - \boldsymbol{R}_{12}\boldsymbol{R}_{22}^{-1}\boldsymbol{F}_2)^\top.\end{aligned}$$

Lemma B.0.2. For any $(w_1, w_2) \in \Re^2$

$$\begin{aligned}\int_{-\infty}^{w_2}\int_{-\infty}^{w_1} z_1\; t_{2,r}(z_1, z_2, \nu)\; dz_1 dz_2 \;&=\; \\ -\big[C_\nu(w_1)T_{\nu-1}\big(\frac{w_2 - rw_1}{\zeta_{r,\nu}(w_1)}\big) \;&+\; rC_\nu(w_2)T_{\nu-1}\big(\frac{w_1 - rw_2}{\zeta_{r,\nu}(w_2)}\big).\end{aligned}$$

where $C_\nu(u) = \sqrt{\frac{\nu}{\nu-2}}\; t_{\nu-2}\left(u\sqrt{\frac{\nu-2}{\nu}}\right)$, $\zeta_{r,\nu}^2(u) = (1-r^2)\frac{u^2+\nu}{\nu-1}$, $T_{\nu-1}(\cdot)$ is the univariate t cdf and $t_{2,r}(\cdot,\cdot,\nu)$ is the joint density function of the bivariate t distribution given above with mean vector 0 and scale matrix $\begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$.

Proof. see Williams (2000b) Lemma B.2.2

Lemma B.0.3. Let $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ be $q_1 \times 1$ and $q_2 \times 1$ random vectors with $\boldsymbol{\mu}_{1|2}$ and $\boldsymbol{\Sigma}_{1|2}$ denoting the conditional mean vector and covariance matrix of $\boldsymbol{Y}_1$ given $\boldsymbol{Y}_2$, and let $\boldsymbol{m}^\top = (\boldsymbol{\mu}_{1|2}^\top, \boldsymbol{Y}_2^\top)$. If

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^\top & \boldsymbol{A}_{22} \end{pmatrix} \quad \text{and} \quad \boldsymbol{Q} = (\boldsymbol{Y}_1^\top, \boldsymbol{Y}_2^\top)\boldsymbol{A}\begin{pmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \end{pmatrix},$$

then

$$E[\boldsymbol{Q} \mid \boldsymbol{Y}_2] \;=\; \boldsymbol{m}^\top\boldsymbol{A}\boldsymbol{m} \;+\; \text{trace}[\boldsymbol{A}_{11}\boldsymbol{\Sigma}_{1|2}].$$

Proof. See Williams (2000b) Lemma B.1.6.

Lemma B.0.4. Consider the setting of Lemma B.0.3 and let $\boldsymbol{A}$ have the form

$$\boldsymbol{A} = \boldsymbol{\Sigma}^{-1} \; - \; \boldsymbol{\Sigma}^{-1}\boldsymbol{F}(\boldsymbol{F}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^{\top}\boldsymbol{\Sigma}^{-1},$$

where $\boldsymbol{\Sigma}$ is the variance-covariance matrix of the random vector $(\boldsymbol{Y}_1^{\top}, \boldsymbol{Y}_2^{\top})$ and $\boldsymbol{F}^{\top} = (\boldsymbol{F}_1^{\top}, \boldsymbol{F}_2^{\top})$. Assume that $\boldsymbol{\Sigma}$ is partitioned in the same way as $\boldsymbol{A}$ and that $\boldsymbol{\Sigma}_{22}$ is invertible. Then

$$\boldsymbol{A}_{11}^{-1} = \boldsymbol{\Sigma}_{11|2} \; + \; (\boldsymbol{F}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{F}_2)(\boldsymbol{F}_2^{\top}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{F}_2)^{-1}(\boldsymbol{F}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{F}_2)^{\top}.$$

where $\boldsymbol{\Sigma}_{11|2} = \boldsymbol{\Sigma}_{11} \; - \; \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{12}^{\top}$.

Proof. See Williams (2000b) Lemma B.1.7.

Theorem B.0.8. Let $\boldsymbol{Y}_e = (Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e), \boldsymbol{Y}^{d\top})^{\top}$ given $\boldsymbol{\beta}$, τ_1^2 and $\boldsymbol{\gamma}$ have a $q+1$-variate normal distribution with mean $\boldsymbol{F}_p\boldsymbol{\beta}$ and variance-covariance matrix $\tau_1^2\boldsymbol{\Sigma}_{e,22}$. Assume that $\boldsymbol{\beta} \in \mathbb{R}^k$ and let $\boldsymbol{Q}_e = \boldsymbol{\Sigma}_{e,22}^{-1}$ - $\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p \; (\boldsymbol{F}_p^{\top}\boldsymbol{\Sigma}_{e,22}^{-1}\boldsymbol{F}_p)^{-1}\boldsymbol{F}_p^{\top}\boldsymbol{\Sigma}_{e,22}^{-1}$ where

$$\boldsymbol{F}_p = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{F} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_{e,22} = \begin{pmatrix} a_{11} & \boldsymbol{a}_{12} \\ \boldsymbol{a}_{12}^{\top} & \boldsymbol{V}_{22} \end{pmatrix}.$$

Then

$$E[\boldsymbol{Y}_e^{\top}\boldsymbol{Q}_e\boldsymbol{Y}_e \mid \boldsymbol{Y}^d, \boldsymbol{\gamma}] \; = \; \big(\boldsymbol{M}_e^{\top}\boldsymbol{Q}_e\boldsymbol{M}_e + \frac{q-k}{q-k-2}\;\hat{\tau_1^2}\big)$$

where $m = \boldsymbol{f}\hat{\boldsymbol{\beta}} + \boldsymbol{a}_{12}\boldsymbol{V}_{22}^{-1}(\boldsymbol{Y}^d - \boldsymbol{F}\hat{\boldsymbol{\beta}})$, $\boldsymbol{M}_e^{\top} = (m, \boldsymbol{Y}^{d\top})$, $\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^{\top}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^{\top}\boldsymbol{V}_{22}^{-1}\boldsymbol{Y}^d$, and

$$\hat{\tau_1^2} = [\boldsymbol{Y}^{d\top}\boldsymbol{V}_{22}^{-1}\boldsymbol{Y}^d - \hat{\boldsymbol{\beta}}^{\top}(\boldsymbol{F}^{\top}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})\hat{\boldsymbol{\beta}}]/(q-k).$$

Proof. First apply Lemma B.0.1 to obtain the posterior mean and variance of $Y(\boldsymbol{x}_c^*, \boldsymbol{x}_e)$ given $(\boldsymbol{Y}^d, \boldsymbol{\gamma})$ as $m = \boldsymbol{f}\hat{\boldsymbol{\beta}} + \boldsymbol{a}_{12}\boldsymbol{V}_{22}^{-1}(\boldsymbol{Y}^d - \boldsymbol{F}\hat{\boldsymbol{\beta}})$ and $\boldsymbol{\Sigma}_{1|2} = \frac{q-k}{q-k-2}\,\hat{\tau}_1^2\boldsymbol{R}$ respectively, where $\boldsymbol{R} = a_{11} - \boldsymbol{a}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{a}_{12}^\top + (\boldsymbol{f} - \boldsymbol{a}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})(\boldsymbol{F}^\top\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^{-1}(\boldsymbol{f} - \boldsymbol{a}_{12}\boldsymbol{V}_{22}^{-1}\boldsymbol{F})^\top$. Then applying Lemma B.0.3 we get

$$E[\boldsymbol{Y}_e^\top\boldsymbol{Q}_e\boldsymbol{Y}_e \mid \boldsymbol{Y}^d, \boldsymbol{\gamma}] \;=\; \left(\boldsymbol{M}_e^\top\boldsymbol{Q}_e\boldsymbol{M}_e + \text{trace}\left(\frac{q-k}{q-k-2}\;\hat{\tau}_1^2\boldsymbol{R}\boldsymbol{Q}_{e,11}\right)\right. . \qquad \text{(B.1)}$$

And applying Lemma B.0.4 we have that $\boldsymbol{Q}_{e,11} = \boldsymbol{R}^{-1}$ which, upon substitution into (B.1), gives the result.

BIBLIOGRAPHY

Adler, R. J. (1981). *The Geometry of Random Fields*. J. Wiley, New York.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer–Verlag, New York.

Bergmann, G., Graichen, F. and Rohlmann, A. (1993). Hip Joint Loading During Walking and Running, Measured in Two Patients. *J. Biomech* **26**, 969–990.

Bernardo, M. C., Buck, R., Liu, L., Nazaret, W. A., Sacks, J. and Welch, W. J. (1992). Integrated circuit design optimization using a sequential strategy. *IEEE Transactions on Computer-Aided Design* **11**, 361–372.

Chang, P. B., Williams, B. J., Bhalla, K. S. B., Belknap, T. W., Santner, T. J., Notz, W. I. and Bartel, D. L. (1999b). Robust design and analysis of total joint replacements: Finite element model experiments with environmental variables. *Journal of Biomechanical Engineering* **1**, 1–2.

Chang, P. B., Williams, B. J., Notz, W. I., Santner, T. J. and Bartel, D. L. (1999a). Robust optimization of total joint replacements incorporating environmental variables. *Journal of Biomechanical Engineering* **121**, 304–310.

Cox, D. D., Park, J. S. and Singer, C. E. (1996). A statistical method for tuning a computer code to a data base. *Technical Report 96-3*. Department of Statistics, Rice University.

Cramér, H. and Leadbetter, M. R. (1967). *Stationary and Related Stochastic Processes*. J. Wiley, New York.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. J. Wiley, New York.

Crowninshield, R., Johnston, R., Andrews, J. and Brand, R. (1978). A Biomechanical Investigation of the Human Hip. *J. Biomech* **11**, 75–85.

Currin, C., Mitchell, T. J., Morris, M. D. and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association* **86**, 953–963.

Dixon, L. C. W. and Szego, G. P. (1978). The global optimisation problem: an introduction. In *Towards Global Optimisation*, Vol. 2 (L. C. W. Dixon and G. P. Szego (eds)), pp. 1–15, North Holland, Amsterdam.

Handcock, M. S. (1991). On cascading latin hypercube designs and additive models for experiments. *Commun. Statist.—Theory Meth.* **20**, 417–439.

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

Harville, D. A. (1974). Bayesian inference for variance components using only error contrasts. *Biometrika* **61**, 383–385.

Haylock, R. G. and O'Hagan, A. (1996). On inference for outputs of computationally expensive algorithms with uncertainty on the inputs. In *Bayesian Statistics*, Vol. 5 (J. Bernardo, J. Berger, A. Dawid and A. Smith (eds)), pp. 629–637, Oxford University Press.

Hoshino, N. and Takemura, A. (2000). On Reduction of Finite-Sample Variance by Extended Latin Hypercube Sampling. *Bernoulli* **6**, 1035–1050.

Huber, P. J. (1981). *Robust Statistics*. J. Wiley, New York.

Iman, R. L., Helton, J. C. and Campbell, J. E. (1981a). An Approach to Sensitivity Analysis of Computer Models: Part I - Introduction, Input Variable Selection and Preliminary Variable Selection. *Journal of Quality Technology* **13**(3), 174–183.

Iman, R. L., Helton, J. C. and Campbell, J. E. (1981b). An Approach to Sensitivity Analysis of Computer Models: Part II - Ranking of Input Variables, Response Surface Validation, Distribution Effect and Technique Synopsis. *Journal of Quality Technology* **13**(4), 232–240.

Johnson, M. E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26**, 131–148.

Jones, D. R., Schonlau, M. and Welch, W. J. (1998). Efficient global optimization of expensive black–box functions. *Journal of Global Optimization* **13**, 455–492.

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society B* **63**(3), 425–464.

Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika* **87**, 1–13.

Kleijnen, J. and Helton, J. (1999). Statistical analyses of scatterplots to identify important factors in large-scale simulations. *Reliability Engineering and System Safety* **65**, 187–197.

Koehler, J. R. and Owen, A. B. (1996). Computer experiments. In *Handbook of Statistics*, Vol. 13 (S. Ghosh and C. R. Rao (eds)), pp. 261–308, Elsevier Science B.V.

Kotzar, G., Davy, D., Goldberg, V., Heiple, K., Berilla, J., Jr, K. H., Brown, R. and Burstein, A. (1991). Telemetrized in Vivo Hip Joint Force Data: A Report on Two Patients After Total Hip Surgery. *J. Orthop. Res* **9**, 621–633.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245.

Mitchell, T., Morris, M. and Ylvisaker, D. (1994). Asymptotically optimum experimental designs for prediction of deterministic functions given derivative information. *Journal of Statistical Planning and Inference* **41**, 377–389.

Morris, M. and Mitchell, T. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* **43**, 381–402.

Morris, M. D., Mitchell, T. J. and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics* **35**, 243–255.

Mrawira, D., Welch, W. J., Schonlau, M. and Haas, R. (1999). Sensitivity analysis of computer models: the world bank HDM-III model. *Journal of Transportation Engineering* **125**, 421–428.

Neal (1999). Regression and classification using Gaussian process priors. *Bayesian Statistics* **6**, 475–501.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal* **7**, 308–313.

O'Hagan, A. (1992). Some Bayesian numerical analysis. In *Bayesian Statistics*, Vol. 4 (J. Bernardo, J. Berger, A. Dawid and A. Smith (eds)), pp. 345–363, Oxford University Press.

O'Hagan, A., Kennedy, M. C. and Oakley, J. E. (1998). Uncertainty analysis and other inference tools for complex computer codes. In *Bayesian Statistics*, Vol. 6 (J. Bernardo, J. Berger, A. Dawid and A. Smith (eds)), Oxford University Press.

Ong, K., Gunsallus, K., Williams, B., Lehman, J., Santner, T. and Bartel, D. (2002). Acetabular cup designs have more influence on mechanical stability then joint loading and surgical variations. 48th Annual Meeting of Orthopaedic Research Society.

Owen, A. B. (1992). A central limit theorem for latin hypercube sampling. *J.R. Statist. Soc. B* **54**, 541–551.

Pebesma, E. J. and Heuvelink, G. B. (1999). Latin Hypercube Sampling of Gaussian Random Fields. *Technometrics* **41**, 303–312.

Pedersen, D., Brand, R. and Davy, D. (1997). Pelvic Muscle and Acetabular Contact Forces During Gait. *J. Biomech* **30**, 959–965.

Phillips, D., Lee, E. H., Herstrom, A., Hogsett, W. and Tingey, D. (1997). Use of Auxiliary Data for Spatial Interpolation of Ozone Exposure in Southeastern Forests. *Environmetrics* **8**, 43–61.

Sacks, J., Schiller, S. B. and Welch, W. J. (1989a). Designs for computer experiments. *Technometrics* **31**, 41–47.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989b). Design and analysis of computer experiments. *Statistical Sciences* **4**, 409–423.

Saltelli, A. and Homma, T. (1992). Sensitivity Analysis for Model Output. *Computational Statistics and Data Analysis* **13**, 73–94.

Saltelli, A., Andres, T. and Homma, T. (1993). Sensitivity Analysis of Model Output. *Computational Statistics and Data Analysis* **15**, 211–238.

Saltelli, A., Chan, K. and Scott, E. (2000). *Sensitivity Analysis*. John Wiley & Sons, Chichester.

Saltelli, A., Tarantola, S. and Chan, K.-S. (1999). A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output. *Technometrics* **41**(1), 39–56.

Santner, T. J., Williams, B. J. and Notz, W. I. (2003). *Design and Analysis of Computer Experiments*. to be published by Springer Verlag, New York.

Seber, G. (1977). *Linear Regression Analysis*. J. Wiley, New York.

Stein, A. and Corsten, L. (1991). Universal Kriging and Cokriging as a Regression Procedure. *Biometrics* **47**, 575–587.

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

Stein, M. L. (1999). *Interpolation of Spatial Data: some theory for kriging*. Springer-Verlag, New York.

Sun, W. (1998). Comparison of a Cokriging Method with a Bayesian Alternative. *Environmetrics* **9**, 445–457.

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

Torczon, V. and Trosset, M. W. (1998). Using Approximations to Accelerate Engineering Design Optimization. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization: A Collection of Technical Papers, Part 2.

Trosset, M. W. (1999a). Approximate Maximin Distance Designs. *1999 Proceedings of the Section on Physical and Engineering Sciences* (), pp. 223–227, American Statistical Association.

Trosset, M. W. (1999b). The Krigifier: A Procedure for Generating Pseudorandom Nonlinear Objective Functions for Computational Experimentation. Interim Report 35, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center.

Trosset, M. W. and Padula, A. D. (2000). Designing and Analyzing Computational Experiments for Global Optimization. *Technical Report 00-25*. Department of Computational and Applied Mathematics, Rice University.

Trosset, M. W. and Torczon, V. (1999). Numerical Optimization Using Computer Experiments. *Technical Report 97-2*. Department of Computational and Applied Mathematics, Rice University.

VerHoef, J. M. and Barry, R. P. (1998). Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Statistical Planning and Inference* **69**, 275–294.

Welch, W. J. (1985). ACED: Algorithms for the construction of experimental designs. *The American Statistician* **39**, 146.

Welch, W. J. and Sacks, J. (1991). A system for quality improvement via computer experiments. *Commun. Statist.-Theory Methods* **20**, 477–495.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J. and Morris, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics* **34**, 15–25.

Welch, W. J., Yu, T.-K., Kang, S. M. and Sacks, J. (1990). Computer experiments for quality control by parameter design. *Journal of Quality Technology* **22**, 15–22.

Williams, B. J. (2000b). *Sequential Design of Computer Experiments to Minimize Integrated Response Functions*. PhD thesis. Ohio State University.

Williams, B. J., Santner, T. J. and Notz, W. I. (2000a). Sequential Design of Computer Experiments to Minimize Integrated Response Functions. *Statistica Sinica* **10**, 1133–1152.

Williams, B. J., Santner, T. J. and Notz, W. I. (2000c). Sequential design of computer experiments for constrained optimization of integrated response functions. *Technical Report 658*. Department of Statistics, The Ohio State University.

Ye, K. Q. (1998). Orthogonal column latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association* **93**, 1430–1439.

Zhang, S. (1998). *Prediction of Deterministic Functions with Applications in Computer Experiments*. PhD thesis. Ohio State University.