DESIGN AND ESTIMATION ISSUES
IN COMPUTER EXPERIMENTS

BY

JAMES R. KOEHLER

TECHNICAL REPORT NO. 359
November 1990

PREPARED UNDER THE AUSPICES
OF
NATIONAL SCIENCE FOUNDATION GRANT DMS89–05874

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA
Design and Estimation Issues
In Computer Experiments

By

James R. Koehler

Technical Report No. 359
November 1990

Prepared Under the Auspices
Of
National Science Foundation Grant DMS89–05874

Also supported by National Science Foundation Grant DMS 901-1074.

Department of Statistics
Stanford University
Stanford, California
Abstract

Computer experiments are useful when the evaluations of a deterministic simulator is computationally expensive. A computer experiment consist of evaluating a design on the simulator and using the results to build a model that can be evaluated much faster than the original simulator. One approach to modeling deterministic simulators is the Kriging model which uses a two component model. The first component is a linear function while the second or lack-of-fit component is treated as a realization of a stationary Gaussian stochastic process. This model also provides a measure of error in the root mean square error.

This thesis investigate some of the design and estimation issues in the Kriging methodology as applied to computer experiments. An introduction is provided that describes some applications of computer experiments. The Kriging model along with optimal design criteria and estimation criteria are presented. Three separate studies investigate estimation, design, and error prediction issues.

The first study investigates the small sample behavior of the model parameter's maximum likelihood estimators. Estimates of the rate of convergence for these estimators are provided along with minimal sample sizes for adequate estimation. The study indicates that relatively small sample sizes may yield badly distributed maximum likelihood estimates.

The second study compares three computer experimental designs in terms of average squared error. The results indicate that the model estimates greatly influence predictions. The performance of designs that minimize the expected mean square error are not as robust in terms of estimation and prediction as the Faure design.
The third study investigates the root mean square error as a measure of error. The results suggest the root mean square error is very sensitive to the model parameters' estimates. For small sample sizes, the root mean square error is usually too large, sometimes by a factor of 6. However in some instances, the root mean square error is too low on average by a factor of 3.

Appendix B contains the User's Guide to PACE (Program for the Analysis of Computer Experiments). This interactive program is a valuable tool for both the experimenters and researchers of the Kriging methodology.
## Contents

Abstract ........................................... iv  
Acknowledgements .................................... vi  

1 Introduction to Computer Experiments ............ 1  
   1.1 Motivation ...................................... 1  
   1.2 Extending Response Surface Methods ............... 4  
   1.3 Outline ......................................... 5  

2 A Kriging Approach .................................... 7  
   2.1 The Kriging Model ................................ 8  
   2.2 Experimental Designs ............................. 14  
      2.2.1 Entropy Designs .............................. 16  
      2.2.2 Mean Square Error Designs .................... 18  
      2.2.3 Maximin and Minimax Designs ................. 22  
      2.2.4 Low Discrepancy Designs ..................... 28  
   2.3 Correlation Functions ............................ 30  
      2.3.1 Restrictions on $R(\cdot)$ .................... 33  
      2.3.2 Correlation Families ......................... 35  
   2.4 Correlation Function Estimation ................... 43  
      2.4.1 Maximum Likelihood .......................... 43  
      2.4.2 Cross-Validated Maximum Predictive Likelihood .. 45  
      2.4.3 Cross-validated Squared Bias ................. 47
# Table of Contents

3 Maximum Likelihood Estimation  
3.1 Estimation Methods ........................................... 48  
3.1.1 The use of NPSOL ........................................... 49  
3.1.2 Alternative Estimation Procedures ...................... 51  
3.2 Asymptotics of the Maximum Likelihood Estimates .......... 52  
3.3 Small Sample Behavior ........................................ 53  
3.3.1 Introduction ............................................... 53  
3.3.2 Methods ................................................... 54  
3.3.3 Results ................................................... 55  
3.3.4 Summary ................................................... 62  

4 A Design Comparison Study ...................................... 66  
4.1 Methods ....................................................... 67  
4.2 Results ....................................................... 68  
4.2.1 Two dimensions ............................................ 68  
4.2.2 Four dimensions ........................................... 72  
4.3 Summary ....................................................... 76  

5 The Root Mean Square Error ....................................... 77  
5.1 Introduction and Background .................................. 77  
5.2 Methods ....................................................... 79  
5.3 Results ....................................................... 82  
5.3.1 Nine Point Faure Design in Two Dimensions ............... 82  
5.3.2 Twenty-five point Faure design in Four Dimensions ... 89  
5.4 Summary ....................................................... 91  

6 Conclusion ..................................................... 94  
6.1 Summary ..................................................... 94  
6.2 Future Work .................................................. 96  

A The Best Linear Unbiased Predictor .............................. 103
B  PACE User’s Guide  107
   B.1 Overview  ........................................  107
   B.2 Running PACE ......................................  109
      B.2.1 Initialization  .................................  109
      B.2.2 Compute Optimal Designs .......................  110
      B.2.3 Evaluate a Correlation Function ...............  111
      B.2.4 Compute Optimum Correlation Parameters .....  111
      B.2.5 Prediction  ..................................  112
      B.2.6 Main Effects  ................................  113
   B.3 PACE-II .............................................  113
      B.3.1 Initialization  .................................  113
      B.3.2 Prediction  ..................................  114
   B.4 Fortran Subroutines  ...............................  114
   B.5 Future Improvements  ..............................  119

Bibliography  121
List of Tables

3.1 Between realization standard error of $\mu_{m1}$, $\sigma_n(\mu_{m1})$, for $p = 3$ .......................... 62
3.2 Between realization standard error of $\sigma^2_{m1}$, $\sigma_n(\sigma^2_{m1})$, for $p = 3$ .................. 63
3.3 Between-realization coefficient-of-variation of $\theta_{m1}$, $\sigma_n(\theta_{m1})/\theta$, for $p = 3$ and the Gaussian correlation function. ........................................ 63

4.1 $\overline{ASE}$ and expected IMSE for $p = 2$ and $n = 9$ designs with the Gaussian correlation family. ...................................................... 71
4.2 $\overline{ASE}$ and expected IMSE for $p = 2$ and $n = 9$ designs with the exponential correlation family. ...................................................... 71
4.3 $\overline{ASE}$ and expected IMSE for $p = 2$ and $n = 9$ designs with the Bessel correlation family. ...................................................... 72
4.4 $\overline{ASE}$ and expected IMSE for $p = 4$ and $n = 25$ designs. ............................................. 75

5.1 Maximum likelihood estimates and Q-Q coefficients from four realizations with the Bessel correlation function and $\theta = (2, 2)$. .................. 84
5.2 Maximum likelihood estimates and Q-Q coefficients from four realizations with the Bessel correlation function and $\theta = (2, 4)$. .................. 84
5.3 Mean and median Q-Q plot coefficients for the Bessel correlation function ........................................ 86
5.4 Mean and median Q-Q plot coefficients for the exponential correlation function ........................................ 89
5.5 Mean and median Q-Q plot coefficients for the Gaussian correlation function ........................................ 89
5.6 Mean and median Q-Q plot coefficients for various correlation functions in four dimensions ................................. 91
List of Figures

2.1 A \( n = 3 \) prediction example. ........................................ 11
2.2 (a) Maximum entropy designs for \( p = 2, n = 1–16 \), and the Gaussian
correlation function with \( \theta = (.5,.5) \). ............................ 19
2.2 (b) Maximum entropy designs for \( p = 2, n = 1–16 \), and the Gaussian
correlation function with \( \theta = (2,2) \). ............................... 20
2.2 (c) Maximum entropy designs for \( p = 2, n = 1–16 \), and the Gaussian
correlation function with \( \theta = (10,10) \). .............................. 21
2.3 (a) Minimum integrated mean square error designs for \( p = 2, n =
1–9 \), and the Gaussian correlation function with \( \theta = (.5,.5) \). .... 23
2.3 (b) Minimum integrated mean square error designs for \( p = 2, n =
1–16 \), and the Gaussian correlation function with \( \theta = (2,2) \). .... 24
2.3 (c) Minimum integrated mean square error designs for \( p = 2, n =
1–16 \), and the Gaussian correlation function with \( \theta = (10,10) \). ... 25
2.4 (a) Minimax and (b) maximin designs for \( n = 6 \) and \( p = 2 \) with
Euclidean distance. .......................................................... 27
2.5 Marginal Projection of a \( n = 25 \) and \( p = 5 \) Faure design. .......... 31
2.6 The effects of \( \theta \) on prediction. ...................................... 32
2.7 The product correlation rule. .............................................. 34
2.8 Realizations for the cubic correlation function \( (\rho, \gamma) = (a) (.15,.03),
(b) (.45,.20), (c) (.70,.50), and (d) (.95,.90) \). ......................... 37
2.9 Realizations for the exponential correlation function with \( \theta = (a)
0.5, (b) 2.0 (c) 5.0, and (d) 20.0. ...................................... 38
2.10 Realizations for the Gaussian correlation function with $\theta = (a) 0.5$, (b) 2.0, (c) 10.0, and (d) 50.0. .............................. 40

2.11 Realizations for the Bessel correlation function with $\nu = 2.5$ and $\theta = (a) 2.0$, (b) 4.0, (c) 10.0, and (d) 25.0. .............................. 42

3.1 Sample paths of $\mu_{ml}$ for $p = 2 - 5$ as $n$ increases. .............................. 56
3.2 Sample paths of $\sigma^2_{ml}$ for $p = 2 - 5$ as $n$ increases. .............................. 58
3.3 Sample paths of $\theta_{ml}$ for $p = 2$ and $n = 5, \ldots, 25$. .............................. 59

4.1 $n = 9$ and $p = 2$ designs: "0" = $D_M$, "+" = $D_F$, and "*" = $D_I$. .............................. 69
4.2 Pairwise marginal projections of $n = 25$ and $p = 2$ designs: "+" = $D_F$ and "*" = $D_I$. .............................. 74

5.1 A $n = 2$ example with (a) the realization and (b) the corresponding Q-Q plot. .............................. 80
5.2 Q-Q plots from 4 realizations with the Bessel correlation function and $\theta = (2, 2)$. .............................. 83
5.3 Q-Q plots from 4 realizations with the Bessel correlation function and $\theta = (2, 4)$. .............................. 85
5.4 Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the Bessel realizations in $p = 2$. .............................. 87
5.5 Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the exponential realizations in $p = 2$. .............................. 88
5.6 Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the Gaussian realizations in $p = 2$. .............................. 90
5.7 Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the realizations in $p = 4$. .............................. 92

6.1 (a) An example of a response ($Y$) and three predictors ($\hat{Y}_3, \hat{Y}_3', \hat{Y}_6$). .............................. 99
6.1 (b) An example of a derivative ($Y'$) and three predictors ($\hat{Y}_3', \hat{Y}_3'', \hat{Y}_6'$). .............................. 99
Chapter 1

Introduction to Computer Experiments

Researchers in many fields have benefited from the advancement of high-speed computers. One way that computers have helped researchers is in using deterministic simulators to solve complicated systems of equations or high dimensional functions. Typically, these functions are based on physical models that are too difficult to solve by hand. Computer experiments are useful when an evaluation of these functions is very computationally expensive.

A computer experimental design consists of a set of sample points to be evaluated through the simulator. The observations from this design are then used to develop a computationally-inexpensive surrogate model to the simulator. This new model is used to approximate the simulator cheaply and efficiently, to investigate the behavior of the function, or to optimize some aspect of the function.

1.1 Motivation

The scope of computer experiments applications is very wide, from manufacturing semiconductors and automobile parts to environmental monitoring. Computer experiments can be performed in any field where a deterministic simulator has been
developed based upon physical models. The inputs to the simulator might be manufacturing settings, physical descriptions or unknown constants. Typically these simulators return many response variables of importance. Computer experiments are efficient methods of extracting information about the unknown function and providing an approximating model that can be inexpensively evaluated. Some fields of application are described below.


The input variables for the above work are generally device sizes, metal concentrations, implant doses and gate oxide temperatures. The multiple responses are threshold voltages, subthreshold slopes, saturation currents and linear transconductance although the output variables of concern depend on the technology under investigation. The engineers use the physical/numerical simulators to assist them in optimizing process, device, and circuit design before the costly step of building prototype devices. They are also concerned with minimizing transmitted variability as this can significantly reduce the performance of the devices and hence reduce yield. For example, Welch et al [75], Currin et al [15] and Sacks et al [57] discuss the use of simulators to investigate the effect of transistor dimensions on the asynchronous of two clocks. They want to find the combination of transistor widths that produce zero clock skews with very small transmitted variability due to uncontrollable manufacturing variability in the transistors.

TIMS, a simulator developed by T. Osswald and C. L. Tucker III, helps in optimizing a compression mold filling process for manufacturing automobiles [13]. In this process a sheet of molding compound is cut and placed in a heated mold. The mold is slowly closed and a constant force is applied during the curing reaction.
CHAPTER 1. INTRODUCTION TO COMPUTER EXPERIMENTS

The controlling variables of the process are the geometry and thickness of the part, the compound viscosity, shape and location within the charge, and the mold closing speed. The simulator then predicts the position of the flow front as a function of time.

Miller and Frenklach [39] discuss the use of computers to solve systems of differential equations describing chemical kinetic models. In their work, the inputs to the simulator are vectors of possibly unknown combustion rate constants and the outputs are induction-delay times and concentrations of chemical species at specified reaction times. The objectives of their investigations are to find values of the rate constants that agree with experimental data and to find the most important rate constant to the process. Sacks et al [56] explore some of the design issues and applications to this field.

TWOLAYER, a thermal energy storage model developed by Alan Solomon and his colleagues at the Oak Ridge National Laboratory, simulates heat transfer through a wall containing two layers of different phase change material. Currin et al utilize TWOLAYER in a computer experiment. The inputs into TWOLAYER are the layers dimensions, the thermal properties of the materials and the characteristics of the heat source. The object of interest was finding the configuration of the input variables that produce the highest value of a heat storage utility index.

FOAM [3] models the transport of polycyclic aromatic hydrocarbon spills in streams using structure activity relationships. Bartell et al [4] modified this model to predict the fate of anthracene when introduced into ponds. This model tracts the "evaporation and dissolution of anthracene from a surface slick of synthetic oil, volatilization and photolytic degradation of dissolved anthracene, sorption to suspended particulate matter and sediments and accumulation by pond biota" [4]. They used Monte Carlo error analyses to assess the effect of the uncertainty in model parameters on their results.
1.2 Extending Response Surface Methods

Response surface methods have been tried as a first approach for building approximating models to deterministic simulators [1, 2, 62, 75]. Box-Behnken and central composite surface designs were tried in fitting quadratic polynomial response surfaces integrated circuit simulators. [1, 2, 62]. These approximating models were then used to investigate the simulator by either evaluation of thousands of points in the design space or by optimization a loss function. Hence, feasibility regions were discovered and the input variables were ranked. Generally, these investigations were also interested in the sensitivity of the input variables to the responses. Welch et al [75] used a design with a noise factor array to directly assess transmitted variability while Sharifzadeh et al [62] measured one direction derivative at each design point to directly model the gradient of the response surface.

Quadratic polynomials are often used in response surface models since they are second order Taylor series approximations to the true function. If the function is well behaved or if the design space is small enough, this type of model should be adequate. However, in most applications the simulator function is much more complicated and polynomial models are too inflexible to take advantage of the precise results obtained from computer experiments. For instance, Sharifzadeh et al [62] found cross-validated $R^2$ values from .25 to .88 for the gradient of certain response variables in VLSI devices while the response cross-validated $R^2$s were around .98. Their work indicates that the quadratic response surfaces may erroneously appear to be adequately modeled when using squared error.

Additionally, experimental design methods concerning replication, blocking, and randomization are not appropriate in the deterministic setting of computer simulators. Traditional designs generally evaluate each input variable at only a few unique settings. For instance, in a central composite design each input variable has only five unique settings out of the $n$ design points. Many of the simulation runs are wasted if only a few of the input variables are significantly affecting the response variable. Therefore, designs specifically related to computer experiments are needed to efficiently and accurately model deterministic simulators.
1.3 Outline

This paper studies some of the design and estimation issues in modeling deterministic simulators. A deterministic simulator evaluates the function with no random error since evaluation of the same code with the same inputs yields the same output. However, these simulators may have bias problems due to programming errors or lack of sophistication in the underlying models. There has been no attempt in this paper to identify simulator biases nor to calibrate simulators.

New techniques are needed to model computer experiments since no random error exists in deterministic simulators. An alternative approach to modeling responses from deterministic simulators is introduced and discussed in Chapter 2. The first component of this model consists of linear terms while the second component models the systematic bias of the first component. This systematic bias is expressed through a correlation function. Families of correlation functions are presented and methods of estimating the parameters of these families are discussed. Further, different criteria used in generating experimental designs useful for computer experiments are also introduced. Examples are also given in Chapter 2.

Chapter 3 discusses in more depth the maximum likelihood estimators of the model parameters. The workings and usefulness of a nonlinear program for solving these parameters is presented. Since the design space is compact, consistency of the maximum likelihood estimates is not guaranteed by ergodic theory. A simulation study that investigates the small sample properties of these estimators in 2 through 5 dimensions is reviewed. The results indicate minimal sample sizes needed for the distribution of these estimators to be well behaved. Alternatively, saturation sample sizes are given. These are sample sizes where additional design points do not significantly improve the distribution of the estimators. An alarming result indicates that the maximum likelihood estimators arbitrarily set some of the parameters to zero when the sample size is small. This leads to false conclusions regarding the relationship between the input and response variables. Minimal sample sizes are suggested to minimize this tendency.
Chapter 4 presents the results of a comparative design study. This study compares the performance of the maximin, Faure and optimal integrated mean square error designs when the underlying correlation function changes. Specifically, nine point designs in two dimensions and twenty-five point designs in four dimensions are compared in terms of the average mean square error over an evaluation grid. The results suggest that the optimal integrated mean square error design is the best when all of the input variables are important. However, if one or more of the input variables have no effect on the response variable the Faure design can be significantly better. Further, the optimal integrated mean square error design can have problems with calculating the maximum likelihood estimators. The maximin design fares poorly compared to the other designs.

The model presented in Chapter 2 has an added benefit of providing estimates of the prediction errors. These are dependent on the underlying covariance function. If the underlying covariance function is known, the true errors from the model when divided by the prediction errors will be distributed as a standard normal variable. However, in practice the parameters of the covariance function need to be estimated as discussed in Chapters 2 and 3. The effect of using these estimators is investigated in Chapter 5. The results suggest that, for small sample sizes, this error predictor is not reliable with a tendency to under estimate the true prediction error. Alternatively, in some situations the error predictions are consistently over estimated.

Chapter 6 summarizes the work presented in the other chapters and suggests topics for future work. In particular, extensions of this methodology to modeling the gradient of the response are presented. A study to test designs with gradient information in regard to predicting both the response and the gradient is proposed.
Chapter 2

A Kriging Approach

Computer experiments are performed when the evaluation of a computer simulator is computationally expensive. Generally these simulators solve complex systems of equations which can be thought of as multidimensional functions. The purpose of a computer experiment is to evaluate a computer simulator at a design, or set of input conditions, and to use the resulting responses to build a surrogate model. This new model can then be used to evaluate the unknown function, at other input values, computationally cheaper than the original simulator.

Computer experiments are unique in that the data obtained has no measurement or random error other than computational round-off error. If the same computer evaluates the same function at the same input conditions, the same output is observed. Therefore, experimental designs with replicated observations are inefficient. Additionally, polynomial models are too inflexible to take advantage of the precise results obtained from computer experiments in describing complex functions. One approach to modeling simulator output [56, 75, 57] is based on a spatial model adapted from the geo-statistical Kriging model [37, 31, 14, 53]. This approach treats the bias, or systematic departure of the response surface from a polynomial model, as the realization of a stationary random function. This model can have exact predictions at the observed responses and predicts with increasing error as the prediction point move away from all the design points.

This chapter introduces the Kriging (or Bayesian) approach to modeling the
response surfaces of computer experiments. Several experimental design criteria are presented and discussed as well as the effect of using different correlation functions on prediction and error analysis.

2.1 The Kriging Model

The Kriging approach uses a two component model. The first component consisting of a general linear model while the second (or lack of fit) component is treated as the realization of a stationary Gaussian random function. Define \( S = [0, 1]^p \) to be the design space and let \( x \in S \) be a scaled \( p \)-dimensional vector of input values. The Kriging approach models the associated response as

\[
Y(x) = \sum_{j=1}^{k} \beta_j f_j(x) + Z(x)
\]  

(2.1)

where the \( f_j \)'s are known fixed functions, the \( \beta_j \)'s are unknown coefficients to be estimated and \( Z(x) \) is a stationary Gaussian random function with \( \text{E}[Z(x)] = 0 \) and covariance

\[
\text{Cov}[Z(x_i), Z(x_j)] = \sigma^2 R(x_j - x_i).
\]  

(2.2)

For any point \( x \in S \), the simulator output \( Y(x) \) at that point is distributed Gaussian with mean \( \sum \beta_j f_j(x) \) and variance \( \sigma^2 \). The linear component models the drift in the response, while the systematic lack-of-fit (or bias) is modeled by the second component. \( Z(x) \) is equal to the difference between \( Y(x) \) and \( \sum \beta_j f_j(x) \) and hence is the bias in the model. If \( Y(x) \) is smooth, then this error or \( Z(x) \) should also be smooth. Hence \( Z(x + \delta) \) should be close to \( Z(x) \) for small \( \delta \). The correlation between \( Z(x) \) and \( Z(x + \delta) \) is \( R(\delta) \). The smoothness and other properties of \( Z(\cdot) \) is controlled by \( R(\cdot) \).

Let design \( D = \{x_i, i = 1, \ldots, n\} \subseteq S \) yield responses \( y_D = \{y(x_1), \ldots, y(x_n)\} \) and consider a linear predictor

\[
\hat{y}(x_0) = \lambda'(x_0) y_D
\]  

(2.3)

of an unobserved point \( x_0 \) in the design space \( (S) \). The Kriging approach [37] treats \( \hat{y}(x_0) \) as a random variable by using a frequentist viewpoint that substitutes \( Y_D \) for
CHAPTER 2. A KRIGING APPROACH

\( y_D \) where

\[ Y'_D = (Y(x_1), \ldots, Y(x_n)). \]  \hspace{1cm} (2.4)

The best linear unbiased predictor (BLUP) finds the \( \lambda(x_0) \) that minimizes

\[ \text{MSE}[\hat{Y}(x_0)] = E[\lambda'Y_D - Y(x_0)]^2 \] \hspace{1cm} (2.5)

subject to the unbiased condition

\[ E[\lambda'Y_D] = E[Y(x_0)]. \] \hspace{1cm} (2.6)

The BLUP of \( Y(x_0) \) is given by (see Appendix A)

\[ \hat{Y}(x_0) = f'(x_0)\hat{\beta} + v'_{x_0}V_D^{-1}(Y_D - F_D\hat{\beta}) \] \hspace{1cm} (2.7)

where

\[
\begin{align*}
  f'(x_0) &= (f_1(x_0), \ldots, f_k(x_0)) \\
  (F_D)_{ij} &= f_j(x_i) \\
  (V_D)_{ij} &= \text{Cov}[Z(x_i), Z(x_j)] \\
  v'_{x_0} &= (\text{Cov}[Z(x_0), Z(x_1)], \ldots, \text{Cov}[Z(x_0), Z(x_n)])
\end{align*}
\] \hspace{1cm} (2.8-2.11)

and

\[ \hat{\beta} = [F'V^{-1}F]^{-1}F'V^{-1}Y_D \] \hspace{1cm} (2.12)

is the generalized least squares estimate of \( \beta \). The mean square error of \( \hat{Y}(x_0) \) is

\[ \text{MSE}[\hat{Y}(x_0)] = \sigma^2 - (f'(x_0), v'_{x_0}) \begin{pmatrix} 0 & F_D' \\ F_D & V_D \end{pmatrix}^{-1} \begin{pmatrix} f(x_0) \\ v_{x_0} \end{pmatrix}. \] \hspace{1cm} (2.13)

The first component of equation (2.7) is the generalized least squares prediction at point \( x_0 \) given the design covariance matrix \( V_D \), while the second component "pulls" the generalized least squares response surface through the observed data points. The elasticity of the response surface "pull" is solely determined by the correlation function \( R(\cdot) \). The predictions at the design points are exactly the corresponding observations, and the mean square error equals zero. As a prediction
CHAPTER 2. A KRIGING APPROACH

point \((x_0)\) moves away from all of the design points, the second component of equation (2.7) goes to zero, if \(R(d) \to \infty\) as \(d \to \infty\), yielding the generalized least squares prediction, while the mean square error at that point goes to \(\sigma^2 + f'(x_0) [F'V^{-1}_D F]^{-1} f(x_0)\). In fact, these results are true in the wide sense if the Gaussian assumption is removed.

As an example, consider an experiment where \(n = 3\), \(p = 1\), \(\sigma^2 = .05\), \(R(d) = \exp\{-20d^2\}\) and \(D = \{.3, .5, .8\}\). The response of the unknown function at the design is \(y'_D = (.7, .3, .5)\). The dashed line of Figure 2.1 is the generalized least squares prediction surface for \(f(\cdot) \equiv 1\) where \(\hat{\beta} = .524\). The effect of the second component of equation (2.7) is to pull the dashed line through the observed design points as shown by the solid line. The shape of the surface or the amount of elasticity of the “pull” is determined by the vector \(v'_D V^{-1}_D\) as a function of \(x\) and therefore is completely determined by \(R(\cdot)\). The dotted lines are \(\pm 2\sqrt{\text{MSE}[\hat{Y}(x)]}\) 95% pointwise confidence envelopes around the prediction surface. The interpretation of these point-wise confidence envelopes is that for any point \(x_0\), if the unknown function is truly generated by a random function with constant mean and correlation function \(R(d) = \exp\{-20d^2\}\), then approximately 95% of the sample paths that go through the observed design points would be between these dotted lines at \((x_0)\). The predictions and confidence intervals can be very different for different \(\sigma^2\) and \(R(\cdot)\). The effect of different correlation functions is discussed in Section 2.3.

Clearly, the true function is not “generated” stochastically. The above model is used for prediction and to quantify the uncertainty of the prediction. This naturally leads to a Bayesian interpretation of this methodology.

A Bayesian Interpretation

An alternative to the frequentist interpretation of equation (2.1) is the Bayesian interpretation which uses the model as a way of quantifying the uncertainty of the unknown function. Diaconis [17] provides a historic survey of the use of stochastic processes as priors for one dimensional functions. Kimeldorf and Wahba [33] were the first to discuss the connection between Bayesian estimation using a Gaussian
Figure 2.1: A $n = 3$ prediction example.
prior distribution on $Y(\cdot)$ and penalized least squares smoothing. Wahba [74] applied this connection to Bayesian confidence intervals for cross-validated smoothing splines. Blight and Ott [7] use a stochastic process as a prior for estimating the bias in a one-dimensional regression problem while O'Hagan [46] presents a general Bayesian approach using a multidimensional Gaussian process as a Gaussian prior on $Y(\cdot)$. The Bayesian approach [15, 47] uses the same model (2.1) but has a different interpretation of the $\beta_j$'s. Here the $\beta_j$'s are random variables with prior distribution $\pi_j$. The effect of these prior distributions is to quantify the prior belief of the unknown function or to put a prior distribution on a large class of functions $G$. Hence hopefully the true function $y(\cdot) \in G$. The mixed convolution of the $\pi_j$'s and $\pi(Z)$ yield the prior distribution $\Pi(G)$ for subsets of functions $G \subset G$.

Once the data $Y_D = y_D$ has been observed, the posterior distribution $\Pi(G \mid Y_D)$ is calculated. The mean

$$\hat{Y}(x_0) = \int g(x_0)\Pi(g \mid Y_D = y_D)dg$$

and variance

$$\text{Var}(\hat{Y}(x_0) \mid Y_D = y_D) = \int (g(x_0) - \hat{Y}(x_0))^2\Pi(g \mid Y_D = y_D)dg$$

of the posterior distribution at each input point is then used as the predictor and as a measure of error, respectively at that point. In general, the Kriging and Bayesian approaches will lead to different estimators. However, if the prior distribution of $Z(\cdot)$ is Gaussian and if the prior distribution of the $\beta_j$'s are diffuse, then the two approaches yield identical estimators.

As an example, consider the case where the prior distribution of the vector of $\beta$'s is

$$\beta \sim N_k(b, \tau^2 \Sigma)$$

and the prior distribution of $Z(\cdot)$ is a stationary Gaussian distribution with expected value zero and covariance function (2.2). After the simulator function has been evaluated at the experimental design, the posterior distribution of $\beta$ is

$$\beta \mid Y_D \sim N_k(\hat{\beta}, \hat{\Sigma})$$
where
\[
\tilde{\beta} = \tilde{\Sigma} \left[ F'V_D^{-1}Y_D + \tau^{-2}\Sigma^{-1}b \right]
\] (2.18)
and
\[
\tilde{\Sigma} = \left[ F'V_D^{-1}F + \tau^{-2}\Sigma^{-1} \right]^{-1}
\] (2.19)
and the posterior distribution of \( Y(x_0) \) is
\[
Y(x_0) \mid Y_D \sim N \left( \psi'_{x_0} V_D^{-1}Y_D + c'_{x_0} \tilde{\beta} + \sigma^2 - \psi'_{x_0} V_D^{-1}v_{x_0} + c'_{x_0} \tilde{\Sigma}c_{x_0} \right)
\] (2.20)
where
\[
c'_{x_0} = f' - \psi'_{x_0} V_D^{-1}F.
\] (2.21)
Hence the posterior distribution is still Gaussian but it is no longer stationary. Now if \( \tau^2 \to \infty \) then
\[
\tilde{\beta} \to \hat{\beta}
\] (2.22)
\[
\tilde{\Sigma} \to \left[ F'V_D^{-1}F \right]^{-1}
\] (2.23)
and hence the posterior variance of \( Y(x_0) \) is
\[
\text{Var}(Y(x_0) \mid Y_D) = \sigma^2 - \psi'_{x_0} V_D^{-1}v_{x_0} + c'_{x_0} \left[ F'V_D^{-1}F \right]^{-1} c_{x_0}
\] (2.24)
\[
= \sigma^2 - \psi'_{x_0} V_D^{-1}v_{x_0} + f' \left[ F'V_D^{-1}F \right]^{-1} f - 2f' \left[ F'V_D^{-1}F \right]^{-1} F'V_D^{-1}v_{x_0} + \psi'_{x_0} V_D^{-1}F \left[ F'V_D^{-1}F \right]^{-1} F'V_D^{-1}v_{x_0}
\] (2.25)
\[
= \sigma^2 - \left[ -f' \left[ F'V_D^{-1}F \right]^{-1} f + 2f' \left[ F'V_D^{-1}F \right]^{-1} F'V_D^{-1}v_{x_0} \right] - \left[ \psi'_{x_0} (V_D^{-1} - V_D^{-1}F \left[ F'V_D^{-1}F \right]^{-1} F'V_D^{-1})v_{x_0} \right]
\] (2.26)
\[
= \sigma^2 - (f'(x_0), \psi'_{x_0}) \begin{pmatrix} 0 & F_D \\ F_D & V_D \end{pmatrix}^{-1} \begin{pmatrix} f(x_0) \\ v_{x_0} \end{pmatrix}
\] (2.27)
which is the same variance as the BLUP in the Kriging approach. Therefore, if \( Z(\cdot) \) has a Gaussian prior distribution and if the \( \beta \)'s have a diffuse prior, the Bayesian and the Kriging approaches yield identical estimators.
CHAPTER 2. A KRIGING APPROACH

Currin et al [15] provide a more in depth discussion of the Bayesian approach for the model with a fixed mean \( f \equiv 1 \). O'Hagan [47] discusses Bayes Linear Estimators (BLE) and their connection to (2.1) and (2.7). The Bayesian approach, which uses random functions as a method of quantifying the uncertainty of the unknown simulator function \( Y(\cdot) \), is more subjective than the Kriging or frequentist approach. While both approaches require prior knowledge or an objective method of estimating the covariance function, the Bayesian approach additionally requires knowledge of parameters of the prior distribution of \( \beta \) (\( b \) and \( \Sigma \)). For this reason, the Kriging results and Bayesian approach with diffuse prior distributions and the Gaussian assumption are of primary concern in this paper.

2.2 Experimental Designs

Selecting an experimental design, \( D \), is a key issue in building an efficient and informative model. Classical experimental designs [19, 32, 12] which use the ideas of blocking, replication and randomization are inappropriate for the deterministic response surfaces of computer simulators. While some experimental design theories [9, 69] do investigate the case where bias rather than solely variance plays a crucial role in the error of the fitted model, how good these designs are to the pure bias problem of computer experiments is unclear. Box and Draper [9] studied the effect on the scale of factorial designs by using a first order polynomial model when the true function is a quadratic polynomial. Box and Draper [10] extended the results to using a quadratic polynomial model when the true response surface is a cubic polynomial. They found that mean square error designs are close to all-bias designs. The all-bias designs could be found by scaling factorial designs inward. Steinberg [69] extended these ideas further by using a prior model proposed by Young [80] that puts prior distributions on the coefficients of a sufficiently large polynomial. However, model (2.1) is more flexible than high ordered polynomials and therefore better designs are needed.

Welch [76] proposed two algorithms for constructing approximate or exact minimum mean square error designs which attempt to protect against a large class of
bias. In the one-dimensional case, Sacks and Ylvisaker [58, 59] considered the design problem for estimating functionals of the output function when the bias is infinite dimensional and when there also may exist measurement error. A more general theory for the multidimensional case is presented by Ylvisaker [79].

McKay et al [38] were the first to consider designs for use with deterministic computer simulators. Their interest was in the propagation of input random error to the simulator output. They proposed Latin hypercube sampling as an alternative to simple random and stratified sampling. Latin hypercube sampling ensures that each of the input variables has \( n \) distinct values spread over \( n \) disjoint cells. In their examples, they showed that Latin hypercube sampling outperformed both simple random and stratified sampling in estimating the mean, variance and distribution function of the output variable. Stein [67] proves that asymptotically, the variance of expectations of a function of output variables is less using Latin Hypercube sampling than simple random sampling with the amount of reduction dependent on the degree of additivity of the function.

This section introduces five design optimality criteria for use with computer experiments: entropy, mean square error, minimax, maximin and star-discrepancy. Entropy designs maximize the amount of information expected for the design while mean square error designs minimize the expected mean square error. Both these designs require a priori knowledge of the correlation function \( R(\cdot) \). Alternatively, maximin and minimax designs do not require any previous knowledge of the correlation structure to construct. These designs optimally pack and cover the design space respectively. And finally, Faure designs which are based upon numerical sequence theory will be presented and discussed. Simple sequential designs, where the location of the \( n^{th} \) design point is determined after the first \( n - 1 \) points have been evaluated, will not be presented due to their tendencies to replicate [57]. However, sequential block strategies could be used where the above designs could be used as starting blocks. Depending upon the ultimate goal of the computer experiment, the first design block might be utilized to refine the design and reduce the design space. The design criteria described below are for the case of fixed design size \( n \).
2.2.1 Entropy Designs

Lindley [34] introduced a measurement, based upon Shannon’s entropy [61], on the amount of information provided by an experiment. This Bayesian measure uses the expected reduction in entropy as a design criterion. This criteria has been used in Box and Hill [11] and Borth [8] for model discrimination. Shewry and Wynn [63] showed that, if the design space $S$ is discrete, that minimizing the expected posterior entropy is equivalent to maximizing the prior entropy.

**Definition 1** A design $D_E$ is a **Maximum Entropy Design** if

$$E_Y[-\ln P(Y_{D_E})] = \min_D E_Y[-\ln P(Y_D)]$$

(2.28)

where $P(Y_D)$ is the density of $Y_D$.

In the Gaussian case, this is equivalent to finding a design that maximizes the determinant of the variance of $Y_D$. In the Gaussian prior case, where $\beta \sim N_k(\beta, \tau^2\Sigma)$, the determinant of the unconditioned covariance matrix is

$$|V_D + \tau^2 F\Sigma F'| = |V_D + \tau^2 F\Sigma F'F|$$

$$= \begin{vmatrix} V_D + \tau^2 F\Sigma F' & F \\ 0 & I \end{vmatrix}$$

$$= \begin{vmatrix} \begin{pmatrix} V_D & F \\ -\tau^2\Sigma F' & I \end{pmatrix} \begin{pmatrix} I & 0 \\ \tau^2\Sigma F' & I \end{pmatrix} \\ V_D & F \\ -\tau^2\Sigma F' & I \end{vmatrix}$$

$$= \begin{vmatrix} I & 0 \\ \tau^2\Sigma F'V_D^{-1} & -\tau^2\Sigma F' I \end{vmatrix} \begin{vmatrix} V_D & F \\ 0 & \tau^2\Sigma F'V_D^{-1}F + I \end{vmatrix}$$

$$= |V_D| |\tau^2\Sigma F'V_D^{-1}F + I|$$

$$= |V_D| \left| F'V_D^{-1}F + \tau^{-2}\Sigma^{-1} \right| \tau^2\Sigma |. \quad (2.29)$$

Since $\tau^2\Sigma$ is fixed, the maximum entropy criteria is equivalent to finding the design $D_E$ that maximizes

$$|V_D| \left| F'V_D^{-1}F + \tau^{-2}\Sigma^{-1} \right|. \quad (2.30)$$
If the prior distribution is diffuse, $\tau^2 \to \infty$, the maximum entropy criteria is equivalent to

$$|V_D| \left| F' V_D^{-1} F \right|$$

(2.31)

and if $\beta$ is treated as fixed, then the maximum entropy criteria is equivalent to

$$|V_D|$$

(2.32)

Shewry and Wynn [63, 64] applied this measure in designs for spatial models. Currin et al [15] and Mitchell and Scott [42] have applied the entropy measure to finding designs for computer experiments. By this measure, the amount of information in experimental design is dependent on the prior knowledge of $Z(\cdot)$ through $R(\cdot)$. In general, $R(\cdot)$ will not be known a priori. Additionally, these optimal designs are difficult to construct due to the required $n \times p$ dimensional optimization of the $n$ design point locations. Currin et al [15] describe an algorithm adopted from DETMAX [41] which successively removes and adds design points to improve the optimality criteria.

An algorithm using NPSOL [25] has been implemented in a fortran program called PACE for Program for the Analysis of Computer Experiments. See Appendix B for the user’s guide to PACE and Section 3.1.1 for a description of NPSOL. The NPSOL algorithm requires an initial design, the correlation family and its parameters. Currently PACE only deals with the constant linear term and maximizes the entropy criterion for the diffuse prior, $|V_D| |1' V_D^{-1} 1|$. The derivatives with respect to the $n \times p$ design point locations are also programmed into PACE to expedite the optimization. A 16-point design in 2 dimensions takes only 5-10 seconds on a DEC-3100 workstation.

Figure 2.2 shows the optimal entropy designs for $p = 2$, $n = 1, \ldots, 16$, $R(\delta) = \exp\{-\theta \sum d_i^2\}$ where $\theta = .5, 2, 10$. The entropy designs tend to spread the points out in the plane and favor the edge of the design space over the interior. For example, the $n = 16$ designs displayed in Figure 2.2 have 12 points on the edge and only 4 points in the interior. Further, most of the designs are similar across the different correlation functions although there are some differences. Generally,
the ratio of the edge to interior points are constant. The entropy criterion appears to be insensitive to changes in the location of the interior points.

### 2.2.2 Mean Square Error Designs

Box and Draper [9] proposed minimizing the normalized integrated mean square error (IMSE) of \( \hat{Y}(x) \) over \( S \). Welch [76] extended this measure to the case when the bias is more complicated. Sacks and Schiller [55] and Sacks et al [56] discuss in more detail IMSE designs for computer experiments.

**Definition 2** A design \( D_I \) is an **Integrated Mean Square Error (IMSE)** design if

\[
J(D_I) = \min_D J(D)
\]

where

\[
J(D) = \frac{1}{\sigma^2} \int_S E \left[ (Y(x) - \hat{Y}(x))^2 \right] dx.
\]  

(2.33)

\( J(D) \) is dependent on \( R(\cdot) \) through \( Y(x) \). For any design, \( J(D) \) can be expressed as

\[
J(D) = \sigma^2 - \text{trace} \left\{ \begin{bmatrix} 0 & F' \\ F & V_D \end{bmatrix}^{-1} \int \begin{bmatrix} f(x)f'(x) & f(x)v_x' \\ v_x f'(x) & v_x v_x' \end{bmatrix} dx \right\}
\]  

(2.34)

and, as pointed out by Sacks et al [56], if the elements of \( f(x) \) and \( V_x \) are products of functions of a single input variable, the multidimensional integral simplifies to products of one-dimensional integrals. As in the entropy design criteria, the minimization of \( J(D) \) is a optimization in \( n \times p \) dimensions and is also dependent on \( R(\cdot) \).

Sacks and Schiller [55] describe the use of an simulated annealing method for constructing IMSE designs for bounded and discrete design spaces. Sacks et al [57] use a quasi-Newton optimizer on a Cray X-MP48. They found that optimizing a \( n = 16, p = 6 \) design with \( \theta_1 = \cdots = \theta_6 = 2 \) took 11 minutes. NPSOL [25] has been used in solving the IMSE optimization for continuous \( S \). A similar optimization requires 13 minutes on a DEC3100. This routine has also been programmed into
Figure 2.2: (a) Maximum entropy designs for $p = 2$, $n = 1-16$, and the Gaussian correlation function with $\theta = (0.5, 0.5)$. 
Figure 2.2: (b) Maximum entropy designs for $p = 2$, $n = 1-16$, and the Gaussian correlation function with $\theta = (2, 2)$. 
Figure 2.2: (c) Maximum entropy designs for $p = 2$, $n = 1-16$, and the Gaussian correlation function with $\theta = (10,10)$. 
PACE (see Appendix B). Generally, these algorithms can find only local minima and therefore many random starts are required.

Since $J(D)$ is dependent on $R(\cdot)$, robust designs need to be found for general $R(\cdot)$. Sacks et al [56] found that for $n = 9, p = 2$ and $R(d) = \exp\{-\theta \sum_{j=1}^{2} d_j^2\}$ (see Section 2.3.2 for details on the Gaussian correlation function) the IMSE design for $\theta = 1$ is robust in terms of relative efficiency. However, this analysis used a quadratic polynomial model and the results may not extend to higher dimensions nor different linear model components. Sacks et al [57] used the optimal design for the Gaussian correlation function with $\theta = 2$ for design efficiency-robustness.

Figure 2.3 displays IMSE designs for $p = 2$ and $n = 1, \cdots, 9$ for $\theta = .5, 2, 10$. The designs, in general lie in the interior of $S$. For fixed design size $n$, the designs usually are similar geometrically for different $\theta$ values with the scale decreasing as $\theta$ increases. They have much symmetry for some values of $n$, particularly $n = 12$. Notice that for the case when $n = 5$ that the design only takes on three unique values for each of the input variables. These designs tend to have clumped projections onto lower dimension marginals of the input space. Good projections are needed if the true function is only dependent on a subset of the input variables. Design projections will be discussed more below and in Chapter 4.

### 2.2.3 Maximin and Minimax Designs

Johnson, Moore and Ylvisaker [30] developed the idea of minimax and maximin designs. These designs are dependent on a distance measure or metric. Let $d(\cdot, \cdot)$ be a metric on $S$. Hence for all $x_1, x_2, x_3 \in S$,

\[ d(x_1, x_2) = d(x_2, x_1) \]  \hspace{1cm} (2.35)

\[ d(x_1, x_2) \geq 0 \]  \hspace{1cm} (2.36)

\[ d(x_1, x_2) = 0 \iff x_1 = x_2 \]  \hspace{1cm} (2.37)

\[ d(x_1, x_2) \leq d(x_1, x_3) + d(x_3, x_2) \]  \hspace{1cm} (2.38)

**Definition 3** Design $D_{MI}$ is a **minimax distance design** if

\[ \min_D \max_{x \in S} d(x, D) = \max_{x \in S} d(x, D_{MI}) \]
Figure 2.3: (a) Minimum integrated mean square error designs for \( p = 2, n = 1-9 \), and the Gaussian correlation function with \( \theta = (.5,.5) \).
Figure 2.3: (b) Minimum integrated mean square error designs for $p = 2$, $n = 1$–$16$, and the Gaussian correlation function with $\theta = (2, 2)$. 
Figure 2.3: (c) Minimum integrated mean square error designs for $p = 2$, $n = 1-16$, and the Gaussian correlation function with $\theta = (10, 10)$. 
where

\[ d(x, D) = \min_{x_0 \in D} d(x, x_0). \]

Minimax distance designs ensure that all the sites in \( S \) are not too far from a design point. Consider placing a \( p \)-dimensional sphere with radius \( r \) around each design point. The idea of a minimax design is to place the \( n \) points so that the design space, \( S \), is covered by the spheres with minimal \( r \). As an illustration, consider the owner of a petroleum corporation who wants to open some franchise gas stations. He would like to locate the stations in the most convenient sites for the customers. A minimax strategy of placing the gas stations would ensure that no customer is too far from one of his company's stations.

Figure 2.4(a) shows a possible minimax design for \( p = 2 \) and \( n = 6 \) with \( d(\cdot, \cdot) \) Euclidean distance. The maximum distance to a design point is .318. For small \( n \), minimax designs will generally lie in the interior of \( S \).

**Definition 4** A design \( D_{MA} \) is a maximin distance design if

\[
\max_D \min_{x_1, x_2 \in D} d(x_1, x_2) = \min_{x_1, x_2 \in D_{MA}} d(x_1, x_2)
\]

Maximin designs pack the \( n \) design points, with their associated spheres, into the design space, \( S \), with maximum radius. Parts of the sphere may be out of \( S \) but the design points must be in \( S \). Analogous to the minimax illustration is the position of a owner of a franchise gas station. She wishes to minimize the competition from the other franchise owners by locating the competing stations as far away from hers as possible. A maximin strategy for placing the franchises would ensure that no two stations are too close to each other.

Figure 2.4(b) shows a maximin design for \( p = 2, n = 6 \) and \( d(\cdot, \cdot) \) Euclidean distance. For small \( n \), maximin designs will generally lie on the exterior of \( S \) and fill in the interior as \( n \) becomes large.

Both the minimax and maximin distance designs are not dependent on \( R(\cdot) \). However Johnson, Moore and Ylvisaker [30] have shown conditions of near independence of \( Z(\cdot) \) (i.e. \( R \) drops off quickly) where minimax designs are asymptotically G-optimal and maximin designs are asymptotically D-optimal.
Figure 2.4: (a) Minimax and (b) maximin designs for $n = 6$ and $p = 2$ with Euclidean distance.
CHAPTER 2. A KRIGING APPROACH

2.2.4 Low Discrepancy Designs

Low-discrepancy sequences have been developed for use in numerical integration. These sequences are constructed so that they fill out the design space as much as possible. Hence the empirical measure of these sequences is close to the uniform measure in a Kolmogorov-Smirnov sense.

Definition 5 The star-discrepancy $D^*_n$ of $x_1, \ldots, x_n \in S$ is defined by

$$D^*_n = \sup_{E \in \mathcal{J}} \left| \frac{\sum_{i=1}^{n} 1(x_i \in E)}{n} - V(E) \right|$$

where

$$\mathcal{J} = \{ E : E = [0, t_1) \times \cdots \times [0, t_p) \}$$

and $V(E) = \text{volume of } E = t_1 t_2 \cdots t_n$.

Star-discrepancy is one of the criteria used by numerical analysts to determine the usefulness of sequences for numerical integration. See Niederreiter [45] for an exhaustive review of the literature.

Van de Corput [72] proposed a sequence for $p = 1$ which is defined in terms of a "radical-inverse function." Let $g \geq 2$ be an integer. Then every integer $n \geq 0$ has an expansion

$$n = \sum_{i=0}^{l} a_i g^i \text{ with } a_i \in (0, 1, \ldots, g - 1), \quad 0 \leq i \leq l. \quad (2.39)$$

Definition 6 The radical-inverse function in base $g$ is defined as

$$\phi_g(n) = \sum_{i=1}^{l} a_i g^{-i-1}. \quad (2.40)$$

Definition 7 The Van der Corput sequence is defined as

$$\{ \phi_2(0), \phi_2(1), \ldots, \phi_2(n), \ldots \}. \quad (2.41)$$

Hammersley [28] extended the Van der Corput sequence into $p$-dimensions.
CHAPTER 2. A KRIGING APPROACH

Definition 8 The Hammersley sequence is defined by

\[ (\phi_{g_1}(i), \phi_{g_2}(i), \ldots, \phi_{g_p}(i)), i = 1, 2, \ldots \]  \hspace{2cm} (2.42)

where the g_i's are all relatively prime.

Halton [27] proved that the Hammersley sequences have

\[ D_n^* < O \left( n^{-1} (\log n)^{p-1} \right). \]  \hspace{2cm} (2.43)

Alternatively, Roth [54] proved that there exist an E such that for any sequence

\[ D_n^* > O \left( n^{-1} (\log n)^{(p-1)/2} \right). \]  \hspace{2cm} (2.44)

Beck [5] has further refined this bound.

Faure [18] proposed a sequence that has smaller discrepancy than the Hammersley sequence. While the Hammersley sequence in p-dimensions is based on the first p prime numbers, the Faure sequence uses the same prime number (the smallest r ≥ p) on each axes.

Sacks et al [57] suggest that low-discrepancy sequences should be investigated for use as designs in computer experiments. Owen et al [48] discuss the Hammersley and Faure designs. They prefer the Faure design due to its small sample properties as well as its asymptotic superiority over the Hammersley design. In their work, they have found that a response usually does not depend heavily on all of the input variables. More commonly, there are many response variables with each response variable dependent on a different subset of the input variables. Therefore, robust designs should also be good designs when they are projected onto subsets of the input variable axes. Low-discrepancy designs have the added benefit of having good, nearly uniform, projections. In particular, the Faure design has good projection properties when n = r^l and projected onto any set of l axes.

For example, if l ≥ 2 then the projection of the Faure design onto the plane defined by any pair of inputs will yield r^{l-2} design points in each of the r^2 partitioned squares. There are similar equidistributional properties for three or more axes. Owen et al [48] also found that a permutation scheme alleviated some projection
problems on $(l + 1)$ axes while still retaining the equidistributional properties on lower order projections.

Figure 2.5 is the projection of a $n = 25, p = 5$ Faure design onto the plane defined by variables $x_1$ and $x_2$. Notice that the projection of this design onto any input variable has $n$ different values in equally spaced sub-intervals. Further, there is one and only one point in each of the 25 partitioned squares. This property is true regardless of the pair of input variables selected.

### 2.3 Correlation Functions

As discussed above, the selection of $R(\cdot)$ plays a crucial role in constructing designs and in the predictive process. Consider the example of Section 2.1 where $n = 3, p = 1, D = \{.3,.5,.8\}, y_d = \{.7,.3,.5\}, R(d) = \exp\{\theta d^2\}$ and $\theta = 20$. Figure 2.6(a) shows the effect on prediction for $\theta = 2$. Now $\hat{\beta} = 1.3$ and the surface elasticity is very low. The predictions outside of the design are actually higher than the observed surface since the convex nature of the observed response indicate that the design range contains a local minimum for the total process. Eventually, the extrapolations would return to the value of $\hat{\beta}$. Additionally, the 95% pointwise confident intervals are much narrower within the range of the design than in Figure 2.1. However, sufficiently outside of the design range, the confidence intervals are of similar width. Figure 2.6(b) displays the prediction when $\theta = 100$. Here $\hat{\beta} = .5$ and the surface elasticity is very high. The prediction line is typically $.5$ with smooth curves pulling the surface through the design points. The 95% pointwise confident intervals are larger than before.

This section presents some restrictions on $R(\cdot)$ and six families of correlation functions. Examples of realization of these families will be shown to explain the effect on prediction by varying the parameter of these families. Further, three criteria for estimating the parameters of a family along with techniques for implementing these methods will be discussed in Section 2.4.
Figure 2.5: Marginal Projection of a $n = 25$ and $p = 5$ Faure design.
Figure 2.6: The effects of $\theta$ on prediction.
A restriction on \( R(\cdot) \) is that \( \forall x_1, x_2 \in S \)
\[
R(x_1, x_2) = R(x_1 - x_2)
\] 
(2.45)
or that \( Z(\cdot) \) is stationary. Any nonstationary behavior in the mean function of \( Y(\cdot) \) can be modeled by the linear term in equation (2.1). A further restriction is that the correlation function is only a function of distance.
\[
R(x_1, x_2) = R(|x_1 - x_2|).
\] 
(2.46)
Additionally, for any \( x_1, \cdots, x_n \) and \( \lambda_1, \cdots, \lambda_n \)
\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j R(x_i, x_j) \geq 0
\] 
(2.47)
or that the correlation function is positive definite. This is to ensure that any covariance matrix formed by \( R(\cdot) \) is positive definite and that
\[
\text{Var}(\lambda'Z_D) \geq 0.
\] 
(2.48)
In higher dimensions \( (p \geq 2) \) a product correlation function,
\[
R(x_1, x_2) = \prod_{j=1}^{p} R_j(|x_{1j} - x_{2j}|)
\] 
(2.49)
is used for mathematical convenience. That is, \( R(\cdot) \) is a product of univariate correlation functions and hence, only univariate correlation functions are of interest. The product correlation function has been used for prediction in spatial settings [78, 15, 57, 56, 75]. Currin et al [15] show that the product correlation function is equivalent to information transmitting along paths with all but one variable fixed. Hence, in Figure 2.7
\[
R(x_1 - x_3) = R_1(x_{11} - x_{31}) R_2(x_{12} - x_{32})
\]
\[
= R_1(x_{11} - x_{21}) R_2(x_{22} - x_{32})
\]
\[
= R(x_1 - x_2) R(x_2 - x_3).
\] 
(2.50)
Figure 2.7: The product correlation rule.
CHAPTER 2. A KRIGING APPROACH

2.3.2 Correlation Families

Linear

The linear correlation family is parameterized by \( \rho \in [0, 1] \) and is given by

\[
R(d) = 1 - (1 - \rho) | d |
\]  

(2.51)

for \( d \in [0, 1] \). \( \rho \) is the correlation between the two endpoints, \( \text{corr}(Y(0), Y(1)) = \rho \). A prediction model in one dimension for this family provides linear interpolation between design points. Further, extrapolations are constant lines with value from the closest design point. Therefore, in one dimension the value of \( \rho \) plays no role in the prediction. However, the variance of the BLUP is dependent on \( \rho \) with larger values of \( \rho \) leading to smaller prediction errors. In two dimensions, the predictions are bi-linear. That is, for any rectangular region formed by the design points, the predictions will be linear in each of the input variables. Processes generated with the linear correlation function are not mean square differentiable. Mitchell et al [40] have found necessary and sufficient conditions so that the derivative process has a linear correlation function. This is the cubic correlation family.

Cubic

The cubic correlation family is parameterized by \( \rho \in [0, 1] \) and \( \gamma \in [0, 1] \) and is given for \( d \in [0, 1] \) by

\[
R(d) = 1 - \frac{3(1 - \rho)}{2 + \gamma} d^2 + \frac{(1 - \rho)(1 - \gamma)}{2 + \gamma} d^3
\]  

(2.52)

where \( \rho \) and \( \gamma \) are restricted by

\[
\rho \geq \frac{5\gamma^2 + 8\gamma - 1}{\gamma^2 + 4\gamma + 7}
\]  

(2.53)

to ensure that the function is positive definite [40]. \( \rho \) again is the correlation between endpoints, \( \text{corr}(Y(0), Y(1)) = \rho \) and \( \gamma \) is the correlation of the derivative process, \( \text{corr}(Y'(0), Y'(1)) = \gamma \). The cubic correlation function implies that the derivative process has a linear correlation process with parameter \( \gamma \).
A prediction model in one dimension for this family is a cubic spline interpolator. In two dimensions, the predictions are bi-cubic. That is, in each of the rectangles formed by projecting the design points onto the marginals, the predictor is a cubic polynomial in each of the input variables. In other words, the predictions are piecewise cubic in each variable. Processes generated with the cubic correlation function are once mean square differentiable and therefore are smoother than processes generated with the linear correlation function. Figure 2.8 shows several realizations of processes with the cubic correlation function and parameter pairs (.15,.03),(.45,.20),(.70,.50),(.95,.9). Notice that the realizations are quite smooth and almost linear for parameter pair (.95,.9).

**Exponential**

The exponential correlation family is parameterized by $\theta \in (0, \infty)$ and is given by

$$R(d) = \exp\{-\theta | d |\}$$

for $d \in [0, 1]$. Processes with the exponential correlation function are Ornstein-Uhlenbeck processes [49]. Like the linear correlation function, the exponential correlation function is not mean square differentiable. In fact, as $\theta$ becomes small, the exponential correlation function becomes very similar to the linear correlation function with $\rho = \exp -\theta$.

Figure 2.9 presents several realizations of one dimensional processes with the exponential correlation function and $\theta = 0.5, 2.0, 5.0, 20$. Figure 2.9a is for $\theta = 0.5$ and is similar to results for the linear correlation function with $\rho = .61$. These realizations have very small global trends but much local variation. Figure 2.9d is for $\theta = 20$, and is very jumpy. Mitchell et al [40] also found necessary and sufficient conditions on the correlation function so that the derivative process has an exponential correlation function. These are the smoothed exponential correlation functions.
Figure 2.8: Realizations for the cubic correlation function \((\rho, \gamma) = (a) (.15, .03), (b) (.45, .20), (c) (.70, .50),\) and \((d) (.95, .90).\)
Figure 2.9: Realizations for the exponential correlation function with \( \theta = \) (a) 0.5, (b) 2.0 (c) 5.0, and (d) 20.0.
CHAPTER 2. A KRIGING APPROACH

Smoothed Exponential

The smoothed exponential correlation family is parameterized by $\rho \in [0, 1]$ and $\gamma \in [0, 1]$ and is given for $d \in [0, 1]$ by

$$R(d) = 1 + \frac{1 - \gamma |d| + |d| \ln \gamma}{1 - \gamma + \ln \gamma} (\rho - 1)$$

(2.55)

where $\rho$ and $\gamma$ are restricted by

$$\rho \geq 2 \frac{1 - \gamma}{\ln \gamma} - 1$$

(2.56)

to ensure that the function is positive definite [40]. Again, $\rho$ is the correlation between endpoints, $\text{corr}(Y(0), Y(1)) = \rho$ and $\gamma$ is the correlation of the derivative process, $\text{corr}(Y'(0), Y'(1)) = \gamma$. The smoothed exponential correlation function implies that the derivative process has an exponential correlation process with parameter $\gamma$. Like the cubic correlation function, the smoothed exponential correlation function is only once mean square differentiable.

Gaussian

Sacks et al. generalized the exponential correlation function by using

$$R(d) = \exp^{-\theta |d|^q}$$

(2.57)

where $0 < q \leq 2$ and $\theta \in (0, \infty)$. $q = 1$ is the exponential correlation function. As $q$ increases, this correlation function produces smoother realizations. However, as long as $q < 2$, these processes are not mean square differentiable.

The Gaussian correlation function is the case $q = 2$ and the associated processes are infinitely mean square differentiable. In fact, the process is analytic. In the Bayesian interpretation, this correlation function puts all of the prior mass on analytic functions [15]. This correlation function is appropriate if the simulator output is known to be analytic. Figure 2.10 displays several realizations for various $\theta$ for the Gaussian correlation function. These realizations are very smooth, even when $\theta = 50$. 
Figure 2.10: Realizations for the Gaussian correlation function with $\theta =$ (a) 0.5, (b) 2.0, (c) 10.0, and (d) 50.0.
CHAPTER 2. A KRIGING APPROACH

Bessel

All of the correlation functions described above are either zero, once or infinitely times mean square differentiable. Stein [68] recommends a more flexible family of correlation function [36, 77]. The Bessel correlation function is parameterized by \( \theta \in (0, \infty) \) and \( \nu \in (-1, \infty) \) and is given by

\[
R(d) = \frac{(\theta | d |)^\nu}{\Gamma(\nu)2^{\nu-1}} K_\nu(\theta | d |)
\]  

(2.58)

where \( K_\nu(\cdot) \) is a modified Bessel function of order \( \nu \). The associated process will be \( m \) times differentiable if and only if \( \nu > m \). Hence, the amount of differentiability can be controlled by \( \nu \) while \( \theta \) controls the range of the correlations. This correlation family is more flexible than the other correlation families described above due to the control of the differentiability of the predictive surface. The amount of differentiability should be known a priori since estimating the order of differentiability by a finite design is unrealistic.

Figure 2.11 displays several realizations of processes with the Bessel correlation function with \( \nu = 2.5 \) and various values of \( \theta \). For small values of \( \theta \), the realizations are very smooth and flat while the realizations are erratic for large values of \( \theta \).

The correlation functions described above have been applied in computer experiments. The linear and cubic correlation functions yield predictions that are linear and cubic splines. The exponential predictions are non-differentiable while the Gaussian predictions are infinitely differentiable. The Bessel correlation function is the most flexible since the degree of differentiability and the smoothness of the predictions can be controlled. In general, enough prior information to fix the parameters of a particular correlation family and \( \sigma^2 \) will not be available. A pure Bayesian approach would place a prior distribution on the parameters of a family and use the posterior distribution of the parameter in the estimation process. This procedure will not be discussed here. Alternatively, an empirical Bayesian approach which uses the data to estimate the parameters of a correlation family and \( \sigma^2 \) is used. Two cross-validation criteria and maximum likelihood procedures will be presented and discussed in the next section.
Figure 2.11: Realizations for the Bessel correlation function with $\nu = 2.5$ and $\theta = (a) 2.0$, (b) 4.0, (c) 10.0, and (d) 25.0.
2.4 Correlation Function Estimation

The previous sections of this chapter presented the Kriging model, design criteria and families of correlation functions. The families of correlations are all parameterized by one or two parameters which control the range of correlation and the smoothness of the corresponding processes. This model assumes that $\sigma^2$, the family and parameters of $R(\cdot)$ are known. In general, these values are not completely known a priori. The appropriate correlation family might be known from the simulator’s designer’s experience regarding the smoothness of the function. Also, ranges for $\sigma^2$ and the parameters of $R(\cdot)$ might be known if a similar computer experiment has been performed. A pure Bayesian approach is to quantify this knowledge into a prior distribution on $\sigma^2$ and $R(\cdot)$. How to distribute a non-informative prior across the different correlation families and within each family is unclear. Further, the calculation of the posterior distribution is generally intractable.

An alternative and more objective method of estimating these parameters is an empirical Bayes approach which finds the parameters which are most consistent with the observed data. This section presents three criteria for estimating $\beta$, $\sigma^2$ and the parameters of a fixed correlation family when the underlying distribution of $Z(\cdot)$ is Gaussian. For each of these criteria, the best parameter set from each correlation family can be evaluated to find the overall “best” $\sigma^2$ and $R(\cdot)$. These criteria are the maximum likelihood, maximum cross-validated predictive likelihood and minimum cross-validated squared bias.

2.4.1 Maximum Likelihood

Consider the case where the distribution of $Z(\cdot)$ is Gaussian. Then the distribution for the response at the $n$ design points $Y_D$ is multinormal and the likelihood is given by

$$
\text{lik}(\beta, \sigma^2, R \mid Y_D) = (2\pi)^{-n/2} \sigma^{-n} |R_D|^{-1/2} \\
\times \exp \left\{ -\frac{1}{2\sigma^2} (Y_D - F\beta)' R_D^{-1} (Y_D - F\beta) \right\} \quad (2.59)
$$
CHAPTER 2. A KRIGING APPROACH

where $R_D$ is the design correlation matrix. The log likelihood is

$$l_{ml}(\beta, \sigma^2, R_D \mid Y_D) = -\frac{1}{2n} \ln (2\pi) - \frac{1}{2n} \ln (\sigma^2) - \frac{1}{2} \ln (|R_D|)$$

$$-\frac{1}{2\sigma^2} (Y_D - F\beta)' R_D^{-1} (Y_D - F\beta).$$

(2.60)

Hence

$$\frac{\partial l_{ml}(\beta, \sigma^2, R \mid Y_D)}{\partial \beta} = -\frac{1}{\sigma^2} (F' R_D^{-1} Y_D - F' R_D^{-1} F\beta)$$

(2.61)

which when set to zero yields the maximum likelihood estimate of $\beta$ which is the same as the generalized least squares estimate,

$$\hat{\beta}_{ml} = [F' R_D^{-1} F]^{-1} F' R_D^{-1} Y_D.$$  

(2.62)

Similarly,

$$\frac{\partial l_{ml}(\beta, \sigma^2, R_D \mid Y_D)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (Y_D - F\beta)' R_D^{-1} (Y_D - F\beta)$$

(2.63)

which when set to zero yields the maximum likelihood estimate of $\sigma^2$

$$\hat{\sigma}_{ml}^2 = \frac{1}{n} (Y_D - F\beta)' R_D^{-1} (Y_D - F\beta).$$

(2.64)

Therefore, if $R_D$ is known, the maximum likelihood estimates of $\beta$ and $\sigma^2$ are easily calculated. However, if $R(\cdot)$ is parameterized by $\theta = (\theta_1, \cdots, \theta_p)$,

$$\frac{\partial l_{ml}(\beta, \sigma^2, R_D \mid Y_D)}{\partial \theta_i} = -\frac{1}{2} \frac{\partial |R_D|}{\partial \theta_i} - \frac{1}{2\sigma^2} (Y_D - F\beta)' \frac{\partial R_D^{-1}}{\partial \theta_i} (Y_D - F\beta)$$

$$= -\frac{1}{2} \text{tr} \left\{ R_D^{-1} \frac{\partial R_D}{\partial \theta_i} \right\}$$

$$+\frac{1}{2\sigma^2} (Y_D - F\beta)' R_D^{-1} \frac{\partial R_D}{\partial \theta_i} R_D^{-1} (Y_D - F\beta)$$

(2.65)

does not generally yield an analytic solution for $\theta$ when set to zero for $i = 1, \cdots, p$.

An alternative method of estimating $\theta$ is to use a nonlinear optimization routine using equation (2.60) as the function to be optimized. For a given value of $\theta$, estimates of $\beta$ and $\sigma^2$ are calculated using equations (2.62) and (2.64), respectively. Next, equation (2.65) is used in calculating the partial derivatives of the objective function. See Mardia and Marshall [35] for an overview of the maximum likelihood procedure. Chapter 3 discusses the use of NPSOL [25] in calculating the maximum likelihood estimates for $\beta$, $\sigma^2$ and $\theta$ for various correlation families.
2.4.2 Cross-Validated Maximum Predictive Likelihood

Currin et al [15] have proposed two alternative criteria for estimating $\beta$, $\sigma^2$ and $R(\cdot)$ based upon cross-validation [70]. These criteria minimize the cross-validated squared bias, which will be discussed in the next section, and maximize the expected predictive (or posterior) likelihood. The predictive distribution is the distribution of the unobserved points given the observed design points and is similar to the posterior distribution in the Bayesian approach. See Geisser and Eddy [20] or Besag [6] for a more general discussion on the predictive approach.

Consider predicting $Y_1 = Y(x_1)$ by using the remaining $n - 1$ design responses $y_{(1)} = (y(x_2), \ldots, y(x_n))$ and let

$$\gamma = R_D^{-1}Y_D$$

$$= \begin{bmatrix} r^{11} & r^{12} \\ r^{21} & R_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_{(1)} \end{bmatrix}$$

$$= \begin{bmatrix} r^{11}y_1 + r^{12}y_{(1)} \\ r^{21}y_1 + R_{22}y_{(1)} \end{bmatrix}$$

(2.66)

and

$$\Psi = R_D^{-1}F_D$$

$$= \begin{bmatrix} r^{11} & r^{12} \\ r^{21} & R_{22} \end{bmatrix} \begin{bmatrix} f_1' \\ F_{(1)} \end{bmatrix}$$

$$= \begin{bmatrix} r^{11}f_1' + r^{12}F_{(1)} \\ r^{21}f_1' + R_{22}F_{(1)} \end{bmatrix}$$

(2.67)

where $r^{ij}$ is the $i,j$ partition of $R_D^{-1}$. Hence, the first component of $\gamma$ is

$$\gamma_1 = r^{11}y_1 + r^{12}y_{(1)}$$

$$= r^{11} \left( y_1 - r_{12}R_{22}^{-1}y_{(1)} \right)$$

(2.68)

since $r^{12} = -r^{11}r_{12}R_{22}^{-1}$ from the standard inversion by partition formula. The first row of $\Psi$ is

$$\psi_1' = r^{11} \left( f_1' - r_{12}R_{22}^{-1}F_{(1)} \right)$$

(2.69)
where $R_{ij}$ is the $i, j$ partition of $R_D$. Then the mean of the predictive distribution of $Y_1$ for fixed $\beta$, $\sigma^2$ and $R(\cdot)$ is

\begin{align*}
\hat{Y}_1 &= f_1' \beta + r_{12} R_{22}^{-1} (y_{(1)} - F_{(1)} \beta) \\
&= r_{12} R_{22}^{-1} y_{(1)} + \left( f_1' - r_{12} R_{22}^{-1} F_{(1)} \right) \beta \\
&= y_{1} - \frac{\gamma_1}{r_{11}} + \frac{\psi_1' \beta}{r_{11}} \\
&= y_{1} - \frac{1}{r_{11}} (\gamma_1 - \psi_1' \beta) \tag{2.70}
\end{align*}

and the variance is $\sigma^2_t = \sigma^2/r_{11}$.

If the distribution of $Z(\cdot)$ is Gaussian, then the log predictive (or posterior) likelihood of $Y_1$ is

\begin{align*}
\ln \left[ P_1 \left( y_{1} \mid \beta, \sigma^2, R(\cdot) \right) \right] &= -\frac{1}{2} \left[ \ln \left( 2\pi \right) + \ln \left(\sigma^2_t \right) + \frac{(y_{1} - \hat{Y}_1)^2}{\sigma^2_t} \right] \\
&= -\frac{1}{2} \left[ \ln \left( 2\pi \sigma^2 \right) - \ln \left( r_{11} \right) + \frac{(\gamma_1 - \psi_1' \beta)^2}{\sigma^2 r_{11}} \right]. \tag{2.71}
\end{align*}

By symmetry, the log predictive likelihood of $Y_i$ given the remaining $(n-1)$ design points is

\begin{align*}
\ln \left[ P_i \left( y_{i} \mid \beta, \sigma^2, R(\cdot) \right) \right] &= -\frac{1}{2} \left[ \ln \left( 2\pi \sigma^2 \right) + \ln \left( \delta_i \right) + \frac{\delta_i (\gamma_i - \psi_i' \beta)^2}{\sigma^2} \right] \tag{2.72}
\end{align*}

where $\delta' = (\delta_1, \ldots, \delta_n)$ is the inverse of the diagonal of $R_D^{-1}$. Averaging over the $n$ design points yields the average cross-validated predictive log likelihood

\begin{align*}
l_{cv} \left( \beta, \sigma^2, R(\cdot) \right) &= -\frac{1}{2} \left[ \ln \left( 2\pi \sigma^2 \right) + \frac{1}{n} \sum_{i=1}^{n} \ln (\delta_i) + \frac{1}{n \sigma^2} \sum_{i=1}^{n} \delta_i (\gamma_i - \psi_i' \beta)^2 \right]. \tag{2.73}
\end{align*}

If $R(\cdot)$ is fixed, then $l_{cv}$ can be minimized with respect to $\beta$ and $\sigma^2$ by setting

\begin{align*}
\hat{\beta}_{cv}(R) &= \left[ \sum_{i=1}^{n} \delta_i \psi_i \psi_i' \right]^{-1} \left( \sum_{i=1}^{n} \delta_i \gamma_i \psi_i \right) \tag{2.74}
\end{align*}

and

\begin{align*}
\hat{\sigma}^2_{cv}(R) &= \frac{1}{n} \sum_{i=1}^{n} \delta_i \left[ \gamma_i - \psi_i' \hat{\beta}_{cv}(R) \right]^2. \tag{2.75}
\end{align*}
Hence the average predictive log-likelihood for fixed $R(\cdot)$ is

$$l_{cv}(R) = \frac{-1}{n} \left[ \ln (2\pi) + 1 + \frac{1}{n} \sum_{i=1}^{n} \ln (\gamma_i) + \ln \left( \hat{\sigma}^2_{cv}(R) \right) \right].$$ (2.76)

This quantity can then be maximized with respect to $R(\cdot)$ within a given correlation family by using a non-linear program.

### 2.4.3 Cross-validated Squared Bias

An alternative criterion, that does not require any distributional assumptions is the cross-validated squared bias.

$$l_b [\beta, R(\cdot)] = \frac{1}{n} \sum_{i=1}^{n} [y(x_i) - Y(x_i)]^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \delta_i^2 [\gamma_i - \psi_i' \beta]^2.$$ (2.77)

For fixed $R(\cdot)$, the minimizing $\beta$ of $l_b[\beta, R(\cdot)]$ is

$$\hat{\beta}_b = \left[ \sum_{i=1}^{n} \delta_i^2 \psi_i \psi_i' \right]^{-1} \left( \sum_{i=1}^{n} \delta_i^2 \gamma_i \psi_i \right)$$ (2.78)

and hence the minimum cross-validated squared bias is

$$l_b [R(\cdot)] = \frac{1}{n} \sum_{i=1}^{n} \delta_i^2 \left( \gamma_i - \psi_i' \hat{\beta}_b \right)^2.$$ (2.79)

This criterion is not dependent on $\sigma^2$ and Currin et al [15] recommend using equation (2.75) with $\hat{\sigma}_{cv}$ replaced by $\hat{\beta}_b$.

The use of $\gamma$, $\delta$ and $\Psi$ reduces the computational burden of evaluating these cross-validated criteria for fixed $R(\cdot)$. Within a correlation family, $R$ is parameterized by a vector (or matrix) of parameter values $\Theta$. All three of these criteria, $l_{mi}(R)$, $l_{cv}(R)$ and $l_b(R)$ can be thought of as a function to be optimized with respect to $\Theta$. In principle, this optimization can be performed via any non-linear optimization routine. All three of these criteria can be differentiated with respect to $\Theta$ to increase the efficiency of the optimization routine. Chapter 3 discusses the use and performance of NPSOL [25] in calculating the maximum likelihood estimates of $\beta$, $\sigma^2$ and $\Theta$ with selected correlation families.
Chapter 3

Maximum Likelihood Estimation

Chapter 2 presented three criteria for estimating the $\beta$, $\sigma^2$ and the parameters $\theta$ within a fixed a correlation family. All three of these criteria require the solution of a non-linear function with respect to $\theta$. This chapter focuses on the maximum likelihood estimation procedure and the small sample behavior of the maximum likelihood estimates (MLE). Since the computational cost in each evaluation of equation (2.60) is on order of $n^3$, care is needed in its maximization to ensure minimal equation evaluations. Further, the behavior of the MLE is of interest since they directly effect the predictive ability of this methodology.

This chapter describes the use of NPSOL in directly solving the maximum likelihood problem and reviews some computation-saving alternatives to maximum likelihood estimation. Some of the asymptotic work on maximum likelihood estimators is summarized and some work on small sample behavior is reviewed. Further, the results of a small sample simulation study are presented. This study investigates the bias, variance and correlation of the MLE for $\beta$, $\sigma^2$ and $\theta$ for the Gaussian correlation function in $p = 2, \ldots, 5$ dimensions. Additionally, the results indicate the minimal sample sizes needed for normality of the distribution of these estimates.
3.1 Estimation Methods

3.1.1 The use of NPSOL

NPSOL [25] is a package of Fortran subroutines that minimizes a smooth function \( L(\theta) \), at least twice-continuously differentiable, subject to linear and nonlinear constraints on \( \theta \). NPSOL uses a sequential quadratic programming algorithm [24] in which the search direction is the solution of a quadratic programming (QP) subproblem. The Hessian of each QP subproblem is a positive definite quasi-Newton approximation to the Hessian of the Lagrangian function. The partial derivatives of \( L(\theta) \) with respect to \( \theta \) are required. The nonlinear constraints are useful for estimating the parameters of the cubic and smoothed exponential correlation families.

On the top level of NPSOL, a series of \( \{\theta_i\} \) are found that converge to \( \theta^* \), a first-order Kuhn-Tucker point of nonlinear programming [51]. This series tries to minimize an augmented Lagrangian merit function [21],

\[
\mathcal{L}(\theta, \lambda, s) = L(\theta) - \sum_{j}^{m} \lambda_j [c_j(\theta) - s_j] + \frac{1}{2} \sum_{j}^{m} \rho_j [c_j(\theta) - s_j]^2
\]

where \( \lambda \) are estimates of the Lagrangian multipliers, \( c(\theta) \) are the \( m \) nonlinear constraints and \( s \) are non-negative slack variables. At each iteration,

\[
\theta_i = \theta_{i-1} + \alpha_i \nu_i
\]

where \( \alpha_i \) is a non-negative step length that produces a sufficient decrease in the merit function, and \( \nu_i \) is the search direction and is itself the solution of a quadratic subprogram (QP).

At the \( i \)th iteration, \( \nu_i \) is the solution of the QP subproblem

\[
\min_{\nu} \ g'\nu + \frac{1}{2} \nu' H \nu
\]

subject to the constraints

\[
l \leq \begin{pmatrix} \nu \\ A_l \nu \\ A_n \nu \end{pmatrix} \leq u
\]
where $g = \nabla L(\theta)$, $H$ is the positive-definite quasi-Newton approximation to the Hessian of the Lagrangian function and $A_n$ is the Jacobian of $c_n$. NPSOL solves this QP problem by using LSSOL [22] which is a two-phase QP method. These phases are the feasibility phase which finds an initial feasibility point and the optimality phase which minimizes equation (3.3) within the feasibility region. In the SQP method, the predicted active set from the QP subproblem is correct in a neighborhood of $\theta^*$. NPSOL exploits this feature which in practice leads to optimality of the subproblem in one iteration. See [23] for details on the Hessian approximation.

The major obstacle in maximizing the likelihood is the inversion of $R_D$ at each evaluation of equation (2.60) which requires on order of $n^3$ computational steps. This can be computationally prohibitive if $n$ is too large. Additionally, if components of $\hat{\theta}_{mi}$ are close to zero, then $R_D$, depending upon the design $D$, can be close to being singular and difficult to invert. This is particularly true for designs which do not have good projection properties. The use of generalized inverses can help alleviate this problem during NPSOL optimizations but can also lead to false maximizations. Some alternative methods for approximating the MLE are reviewed below.

NPSOL requires initial estimates of the parameters $\theta$ and offers the option to select NPSOL parameters such as the optimality tolerance level and maximum number of iterations. As noted above, if the initial parameter estimates are too small, the inversion of $R_D$ has numerical instability problems. Additionally, large initial parameter estimates lead to a flat likelihood and only local maximization. This can be alleviated by decreasing the optimality tolerance level. In general, initial estimates of $\theta$ between two and four produce good and stable MLE.

NPSOL returns an integer variable named INFORM which indicates any problems encountered during the optimization. Any non-zero value indicates that the given solution might be incorrect. In these cases different initial estimates usually lead to better results.
CHAPTER 3. MAXIMUM LIKELIHOOD ESTIMATION

3.1.2 Alternative Estimation Procedures

As noted above, direct maximization of the log-likelihood can be computationally prohibitive for large values of \( n \) due to the cost of inverting \( R_D \). As a reference, NPSOL took an average of 2 minutes on a DEC-3100 to compute the MLE for \( n = 50, p = 3 \) and \( R(d) = \exp\{-2d^2\} \). Hence in higher dimensions and with larger designs this method could require hours of time on a workstation. Some alternatives have been proposed to approximate the likelihood function when \( n \) is large. Mardia and Marshall [35] suggest partitioning \( D \) into \( s \) subsets \( D_1, \ldots, D_s \) and performing the maximum likelihood procedure on each subset. The final estimates are the medians of the subset estimates. Schagan [60] has suggested subset sizes equal to two. This approach has drawbacks if the subsamples are too small due to the erratic behavior of small sample estimates as presented below.

Vecchia [73] has proposed more objective approximations to the likelihood by using the identity

\[
P(Y_D) = \prod_{i=1}^{n} P[Y(x_i) \mid Y(x_1), \ldots, Y(x_{i-1})]
\]

(3.5)

for the Gaussian distributional assumption and in particular when the observations have been ordered with respect to one of the input variables. Vecchia proposes approximating \( P(Y_D) \) by

\[
P_m(Y_D) = \prod_{i=1}^{n} P[Y(x_i) \mid Y(x_{im})]
\]

(3.6)

where \( x_{im} \) is a subset of \( D \) of size \( m \) or less. Formally, the optimal choice for the subsets are the points that have the maximum multiple correlation coefficient with \( Y(x_i) \) but this is dependent on the unknown parameters and is very computationally expensive. Vecchia suggests three methods of selecting the subsets: 1) the \( m \) points \( x_j \) with \( j < i \) closest to \( x_i \) in some distance measure such as Euclidean, 2) the \( m \) points without regard to order closest to \( x_i \) and 3) all points with some fixed distance of \( x_i \). The last method works well with lattice designs but is erratic with irregularly spaced designs. The second method does not necessarily converge to \( P(Y_D) \) as \( m \to n \). The first method is advocated by Vecchia since it converges to
\( P(Y_0) \) and is generally close for small values of \( m (\sim 10) \). This method is roughly three times faster than directly computing the maximum likelihood when \( n = 100 \) and \( m = 10 \).

3.2 Asymptotics of the Maximum Likelihood Estimates

Procedures for calculating the maximum likelihood estimators of \( \beta, \sigma^2 \), and \( \theta \) within a correlation family have been discussed above. The behavior of these estimators, particularly the bias, variance and correlation as well as the limiting distribution is of interest. Mardia and Marshall [35] have shown general conditions under the spatial model where these estimators are consistent and asymptotically normally distributed. However, their works requires the design space \( S \) to increase as \( n \) increases so that no two points are too close to each other. In the present setting, the design space is fixed and hence ergodic results do not apply since within the fixed design space there is only a finite amount of information concerning the random function. Hence as \( n \) increases the amount of new information decreases.

Stein [66] shows that if the correlation function \( R(\cdot) \) is known that \( \hat{\sigma}^2_{\text{ml}} \) converges almost surely no matter where the design points are placed. Unfortunately, these results do not apply when the parameters of \( R(\cdot) \) are also estimated. Stein does conjecture on asymptotic normality for the modified maximum likelihood estimators and shows that this estimator is unbiased. Hence, over many realizations of the random function the estimators are on average unbiased, but for only one realization the estimators can be biased even for large \( n \). Stein [65] has shown conditions, for a fixed design space, where function estimates are asymptotically efficient even if the covariance function is misspecified. Generally, the conditions are that the covariance function is estimated well for small distances.
3.3 Small Sample Behavior

A general theory of the asymptotic distribution for the maximum likelihood esti-
mators would be useful for large \( n \). However, for small \( n \) the usefulness would
depend on the strength of the asymptotics. The practical situation in computer
experiments is with relatively small \( n \) and hence simulation studies appear to be
the only approach for determining the behavior of the MLE. Mardia and Marshall
[35] performed a simulation study in two dimensions with \( n = 36 \) and 100. Their
results suggest that the asymptotic normality appears valid for \( n = 100 \) when the
design space increases with \( n \). They found that for smaller \( n \) that \( \hat{\sigma}_{ml}^2 \) is underesti-
mated while \( \hat{\beta}_{ml} \) is overestimated with an expected high correlation between them.
\( \hat{\beta}_{ml} \) appeared unbiased and normally distributed even for \( n = 36 \). Questions still
remaining are whether these results hold true for higher dimensions, a fixed design
spaces and different correlation functions. These are some of the goals of the fol-
lowing study.

3.3.1 Introduction

This simulation study focuses on the behavior of the maximum likelihood estima-
tors of \( \beta, \sigma^2 \) and \( \theta \) as \( n \) increases in two through five dimensions. In particular,
interest is in the model where \( k = 1 \) (i.e. \( \beta = \mu \) a constant) and when the corre-
lation function is Gaussian with \( \theta_i = 2 \) for \( i = 1, \ldots, p \). Some additional simulations
are performed in three dimensions where \( \theta' = (4, 4, 4) \) and \((1, 2, 4) \) and also for the
Bessel correlation function with \( \theta' = (2, 2, 2) \) and \( \nu' = (2.5, 2.5, 2.5) \). Since the de-
sign space is fixed, ergodic theory does not directly apply. Hence as \( n \) increases,
the maximum likelihood estimators do not necessarily converge to the true values.
Even over many realizations of the process, the estimators are not assured of con-
verging to the true values for small \( n \). The purpose of this study is to find values of
\( n \), for various dimensions and correlation functions, that have corresponding max-
imum likelihood estimators that are accurate and precise.
3.3.2 Methods

A sequential design is needed to accurately assess the effects of increasing \( n \) on the maximum likelihood estimators. The Faure design is the only design discussed in Section 2.2 that is sequential and is therefore utilized in this study. As mentioned above, interest is in the model where the linear component is a simple constant and without loss of generality \( \mu = \beta = 0 \) and \( \sigma^2 = 1 \). Further, for most of the simulation runs, a Gaussian correlation function is used with \( \theta_i = 2 \) for all \( i \). The value of \( \theta \) essentially effects the range of the observed random function. As \( \theta \) increases, the correlations drop off faster and hence the random function has more variation. This is equivalent to using a larger design space with smaller values of \( \theta \).

The simulation study involves a number of steps. For each dimension \( p \), the corresponding Faure design with a large value of \( n \) was created. Here, the maximum \( n \) is dependent on the dimension. For example, the maximum \( n \) is equal to 25 in two dimensions but is 80 in five dimensions. Next, the corresponding covariance matrix was computed and a Cholesky decomposition performed. The transpose of the Cholesky matrix was then post-multiplied by \( B \) vectors of independent standard-normal pseudo-random variables. This yielded \( B \) realizations of the random function at the design points. The number of realizations generated was different for the different dimensions but either 30 or 70.

For each of the \( B \) realizations, the NPSOL routine described above was employed to find the MLE of \( \mu, \sigma^2 \) and for each of the \( \theta_i \). Caution was used to confirm that the estimates calculated were the true MLE and not just saddle points. If INFORM was non-zero, other initial estimates were tried until satisfactory results were obtained. These \( B \) set of estimates were then pooled together to approximate the true distribution of the MLE. The bias, variance, skewness and correlation between estimators were then evaluated.

Let \( \sigma_n^2(\omega) \) be the between realization variability for \( \omega = \mu_{ml}, \sigma_{mi}, \text{ and } \theta_{ml} \) as a function of \( n \). That is,

\[
\sigma_n^2(\omega) = \frac{1}{B} \sum_{b=1}^{B} (\omega_b - \bar{\omega})
\]

(3.7)
where $\bar{\omega} = 1/B \sum_{b=1}^{B} \omega_b$ and $\omega_b$ is the MLE of $\omega$ for the $b^{th}$ realization and by using only $n$ design points. To assess the reduction in $\sigma_n^2(\cdot)$ as $n$ increases a regression was performed on the model

$$\sigma_n^2(\cdot) = \frac{\kappa}{n^2}$$

(3.8)

to find the rate constant $q$. Explicitly, a linear regression was performed on the log model

$$\ln \left[ \sigma_n^2(\cdot) \right] = \ln [\kappa] + q \ln [n]$$

(3.9)

to find $\hat{q}$. $R^2$ fit statistics are provided below to indicate the reliability of $\hat{q}$. The estimates of $q$ are only valid for the range of $n$ investigated.

### 3.3.3 Results

#### Two Dimensions

Values of $n = 5, 6, \ldots, 25$ were investigated in two dimensions and $B = 70$. The maximum value for $n$ is restricted by the stability of the covariance matrix. The number of parameters estimated is four in two dimensions and hence these estimates will be highly variable for very small $n$. $\mu_{ml}$ is very well behaved even for small $n$. There was only one outlier, when $n = 6$, out of all the $70 \times 21$ estimates. In general the estimates are unbiased, symmetrically distributed and uncorrelated with the other parameter estimates. Although within a realization there are occasional jumps in the estimates as $n$ increases, generally the estimates of $\mu$ are very stable and do not change much past $n = 8$. In fact, $\sigma_n^2(\mu_{ml})$ only decreases at a rate approximately proportional to $n^{-2}$ with $R^2 = .50$. Figure 3.1a shows $\mu_{ml}$ as $n$ increases for the seventy realizations.

Alternatively, $\sigma_{ml}^2$ is not well behaved even for values of $n$ as high as seventeen. In general, the estimates are biased low while the distribution is skewed to the right with the amount of skewness and bias reducing as $n$ increases. Hence, while the average estimate of $\sigma^2$ is close to 1 for different values of $n$, most of the estimates are less than 1 with some large outliers pulling the average up. See Figure 3.2a for the estimates of $\sigma^2$ for increasing $n$ in two dimensions. $\sigma_n^2(\sigma_{ml}^2)$ has an associated $\hat{q}$ of .8 with $R^2 = .14$. Further, $\sigma_{ml}^2$ has high positive-correlation with all components.
Figure 3.1: Sample paths of $\mu_{ml}$ for $p = 2 - 5$ as $n$ increases.
of $\theta$. These correlations are around -.55 for $n = 25$ while slightly lower for smaller values of $n$.

The distribution for each $\theta_i$ is very unstable for small values of $n$. As expected, for $n < 10$ the distribution of $\theta$ is very non-normal with some very large outliers and high correlation (.6 - .9) between $\theta_1$ and $\theta_2$. For $n > 15$, the distribution is close to symmetric although still slightly biased high and the correlations do not appear significant. $\sigma_n^2(\theta_{ml})$ reduces at a very fast rate proportional to $n^{-3.8}$ with $R^2 = .92$. Figure 3.3 displays nine paths of $\theta$ for $n = 5, \ldots, 25$. These can have very wild paths but they all converge closely to the correct values of (2,2) for $n > 15$.

Three Dimensions

In three dimensions, $n = 5, 10, \ldots, 50$ are investigated with $B = 30$. Again $\mu_{ml}$ looks normally distributed for all values of $n$ and as Figure 3.1b shows the estimates are almost constant after $n = 30$. $\sigma_n^2(\mu_{ml})$ reduces at a rate almost proportional to $n^{-3}$ with $R^2 = .85$.

The distribution for $\sigma_{ml}^2$ is very non-normal for $n = 5$ as is expected since there are five parameters to estimate. For $n \geq 10$ the distribution is better behaved although still skewed to the right and slightly biased low until $n \geq 30$. $\sigma_n^2(\sigma_{ml}^2)$ seems to reduce proportionally to $n^{-5}$, the same as in two dimensions, with $R^2 = .42$. The correlations between $\sigma_{ml}^2$ and $\theta_{ml}$ are in the -.45 to -.65 range.

Small sample sizes force $\theta_{ml}$ to set some components to zero to reduce the dimensionality of the problem. For $n = 5$ over thirty of the ninety estimates of components of $\theta$ are zero while by the time $n = 20$, none of these estimates are zero. Overall the estimates are biased high and $\sigma_n^2(\theta_{ml})$ reduces proportionally to $n^{-3}$ for $n \leq 35$ with $R^2 = .98$. For larger $n$ the reduction is proportional to $n^{-2}$. In general, for larger $n$, the correlation between the components of $\theta$ are positively correlated in the .2 - .4 range. Only a few of these are statistically significant at the 95% level due to the small replication level. See below for changes in these estimators when the correlation function changes.
Figure 3.2: Sample paths of $\sigma_m^2$ for $p = 2 \rightarrow 5$ as $n$ increases.
Figure 3.3: Sample paths of $\theta_{ml}$ for $p = 2$ and $n = 5, \ldots, 25$. 
CHAPTER 3. MAXIMUM LIKELIHOOD ESTIMATION

Four Dimensions

$B$ was increased to 70 for the four dimensional study and the $n = 10, 15, \ldots, 50$. Again $\mu_{ml}$ is well behaved and the estimates do not generally change much for $n \geq 30$. $\sigma_n^2(\mu_{ml})$ appears to reduce proportionally to $n^{-5}$ with $R^2 = .93$ while $\mu_{ml}$ is not significantly correlated with the other parameter estimates.

As Figure 3.2c displays, the distribution of $\sigma_n^2(\mu_{ml})$ is highly skewed for small $n$ with the median of the distribution lower than the true value of one. As the skewness reduces with increasing $n$, the distribution becomes biased low. $\sigma_n^2(\sigma_n^2)_{ml}$ is reducing but at a rate of $n^{-1.4}$ with $R^2 = .96$. The $\sigma_n^2, \theta_{ml}$ correlations are in the range -.35 to -.5 and are generally smaller for smaller $n$.

$\theta_{ml}$ behaves similarly in four dimensions as in three dimensions. When $n = 10, 80$ of the 280 parameter estimates are zero indicating the same tendency to reduce the dimensionality of the problem. This tendency is not alleviated until $n \geq 30$. For all sample sizes, there are very large estimates for components of $\theta$ which causes the estimated mean of the distribution to be biased high. Further, $\sigma_n^2(\theta_{ml})$ reduces proportionally to $n^{-3.3}$ with $R^2 = .97$. The correlations between components are between .0 and .2 for $n = 50$.

Five Dimensions

For the five dimensional study, $B = 30$ and $n = 10, 15, \ldots, 80$. Once again the distribution of $\mu_{ml}$ is well behaved as displayed in Figure 3.1d. The estimates are unbiased for all values of $n$ with no skewness. The estimates do not change much for $n \geq 40$ with $\sigma_n^2(\mu_{ml})$ decreasing by $n^{-5}$ with $R^2 = .95$.

Figure 3.2 shows the estimates of $\sigma_n^2(\mu_{ml})$ as a function of $n$ over the thirty realizations. The estimates are skewed to the right, particularly for small sample sizes. The skewness makes the mean of the distribution larger than one while the median is less than one. The estimates exhibit stability for $n \geq 45$ as indicated by $\sigma_n^2(\sigma_n^2)_{ml}$ which does not decrease significantly for $n \geq 55$. For small $n$, $\sigma_n^2(\sigma_n^2)_{ml}$ reduces proportionally to $n^{-q}$ for $q$ between .5 and .8. The correlations between $\sigma_n^2_{ml}$ and $\theta_{ml}$ range from -.35 to -.65 for $n \geq 40$. The correlations appear to increase with $n$. 
As with the other dimensions, $\theta_{ml}$ arbitrarily sets some components to zero to reduce the dimensionality of the problem when $n$ is small, in this case $n \leq 40$. Further, the distribution is highly skewed until $n \approx 50$. $\sigma_n^2(\theta_{ml})$ is decreasing proportionally to $n^{-2}$ for large $n \geq 50$ with $R^2 = .90$. The ten within component correlations are, except for one, all positive with range -.25 to .4.

Ten Dimensions

To assess the maximum likelihood estimators in higher dimensions, a study was performed in 10 dimensions with $B = 20$ and $n = 15(5)20(10)100$. For $n = 30$, $\mu_{ml}$ is well estimated with no reduction in the variance for larger $n$. The variance of $\sigma^2_{ml}$ is still reducing for $n = 100$ and the distribution is slightly skewed. The most critical problem is with components of $\theta_{ml}$ having zero estimates. For $n = 25$, five out of the ten components on average of $\theta$ are estimated to be zero. Even for $n = 100$ one or two of the components are typically estimated to be zero. This severely limits the size of the dimensionality that the spatial methodology can model.

Other Results

Three addition studies were performed for $p = 3$ and $B = 30$ with: 1) $\theta' = (4,4,4)$, 2) $\theta' = (.1,2,4)$ and 3) the Bessel correlation function with $\theta' = (2,2,2)$ and $\nu' = (2.5,2.5,2.5)$. These are to test if the above results can be expected to hold for various correlation functions. These studies suggest that in all cases $\mu_{ml}$ is unbiased and symmetrically distributed. As $n$ increases, the reduction in $\sigma_n^2(\mu_{ml})$ is large for small $n$ but the effect decreases for large $n$. In fact, for 1) $n \geq 25$ in the $\theta' = (4,4,4)$ case, 2) $n \geq 30$ in the $\theta' = (.1,2,4)$ case, 3) $n \geq 35$ in the $\theta' = (2,2,2)$ case, and for 4) $n \geq 20$ in the Bessel case $\sigma_n^2(\mu_{ml})$ does not decrease significantly (See Table 3.1). This is due to the restricted design space indicating that additional observations do not help in estimating $\mu$.

The results are similar for $\sigma^2_{ml}$ when $\theta$ increases from 2 to 4. The estimates are biased low for all $n$ with the bias actually increasing with larger sample sizes due to the reduction of the distribution skewness. $\sigma_n^2(\sigma^2_{ml})$ rate reduction seems
to be stronger than $n^{-\frac{3}{2}}$ and slightly smaller (see Table 3.2) while the $\sigma_{ml}^2$, $\theta$ correlations are slightly smaller. The estimates of $\theta$ are still biased high at close to the same standardized rate as with $\theta = 2$. The coefficient of variation for $\theta_{ml}$, $\text{CV}(\theta_{ml}) = \sigma_{n}^2(\theta_{ml})/\theta$, looks larger for the increased $\theta$.

When the Bessel correlation function is used with $\theta' = (2, 2, 2)$ the estimates are very similar although not identical. The $\sigma_{n}^2(\cdot)$ for the Bessel correlation function is larger for both $\mu_{ml}$ and $\sigma_{ml}^2$ as shown in Tables 3.1 and 3.2 respectively.

When the parameters of the Gaussian correlation are different the results indicate that $\sigma_{ml}^2$ is severely biased low even for large $n$, similarly to the case $\theta = (4, 4, 4)$. $\sigma_{n}^2(\sigma_{ml}^2)$ is smaller than $\theta = 2$ but larger than the $\theta = 4$ case (see Table 3.2). The bias of $\theta_{ml}$ is a little higher than the other simulations described above while $\sigma_{n}^2(\theta_{ml})$ reduces at a comparable rate. The between component correlations are all positive with a range of .25 to .4.

3.3.4 Summary

The results reveal that that the maximum likelihood estimator of $\mu$ is unbiased for all values of $n$ with $\sigma_n^2(\mu_{ml})$ reducing, for moderate $n$, at a rate around $O(n^{-q})$ where $q$ is approximately $.4 - .6$. As the sample size increases, $\sigma_n^2(\mu_{ml})$ appears to
### Table 3.2: Between realization standard error of $\sigma_{ml}^2$, $\sigma_n(\sigma_{ml}^2)$, for $p = 3$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Gaussian $(2,2,2)$</th>
<th>Gaussian $(4,4,4)$</th>
<th>Gaussian $(.1,2,4)$</th>
<th>Bessel $(2,2,2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>.58</td>
<td>.89</td>
<td>.87</td>
<td>.56</td>
</tr>
<tr>
<td>15</td>
<td>.53</td>
<td>.59</td>
<td>.75</td>
<td>.81</td>
</tr>
<tr>
<td>20</td>
<td>.80</td>
<td>.53</td>
<td>.52</td>
<td>.71</td>
</tr>
<tr>
<td>25</td>
<td>.62</td>
<td>.39</td>
<td>.57</td>
<td>.49</td>
</tr>
<tr>
<td>30</td>
<td>.45</td>
<td>.34</td>
<td>.48</td>
<td>.60</td>
</tr>
<tr>
<td>35</td>
<td>.48</td>
<td>.32</td>
<td>.41</td>
<td>.80</td>
</tr>
<tr>
<td>40</td>
<td>.43</td>
<td>.34</td>
<td>.40</td>
<td>.53</td>
</tr>
<tr>
<td>45</td>
<td>.43</td>
<td>.31</td>
<td>.38</td>
<td>.47</td>
</tr>
<tr>
<td>50</td>
<td>.43</td>
<td>.26</td>
<td>.35</td>
<td>.46</td>
</tr>
</tbody>
</table>

### Table 3.3: Between-realization coefficient-of-variation of $\theta_{ml}$, $\sigma_n(\theta_{ml})/\theta$, for $p = 3$ and the Gaussian correlation function.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Gaussian $(2,2,2)$</th>
<th>Gaussian $(4,4,4)$</th>
<th>Gaussian $(.1,2,4)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>381</td>
<td>203</td>
<td>911</td>
</tr>
<tr>
<td>15</td>
<td>91</td>
<td>227</td>
<td>245</td>
</tr>
<tr>
<td>20</td>
<td>58</td>
<td>234</td>
<td>384</td>
</tr>
<tr>
<td>25</td>
<td>48</td>
<td>114</td>
<td>162</td>
</tr>
<tr>
<td>30</td>
<td>38</td>
<td>54</td>
<td>116</td>
</tr>
<tr>
<td>35</td>
<td>23</td>
<td>36</td>
<td>72</td>
</tr>
<tr>
<td>40</td>
<td>21</td>
<td>29</td>
<td>71</td>
</tr>
<tr>
<td>45</td>
<td>18</td>
<td>25</td>
<td>53</td>
</tr>
<tr>
<td>50</td>
<td>16</td>
<td>22</td>
<td>43</td>
</tr>
</tbody>
</table>
approach a non-zero asymptote maybe due to the compactness of the design space. There is reason to believe that if the Gaussian correlation function is used and if the function is completely known in some neighborhood, then the MLE would be consistent due to the analyticity of the random function generated with a Gaussian correlation function. However, the results observed in this study indicate that this could not only occur very quickly. $\mu_{ml}$ is not significantly correlated with the other parameters. $\sigma^2_{ml}$ appears to be biased low for all values of $n$, but the average is masked by the high positive skewedness for $n < 20$. Further, $\sigma^2_n(\sigma^2_{ml})$ reduces at the rate around $O(n^{-5})$. This rate is stronger for small $n$, and as Figure 3.2 and Table 3.2 exhibit, this reduction looks to also not converge to 0. As expected, $\sigma^2$ has high positive correlation with all components of $\theta$.

$\theta_{ml}$ is biased high for most values of $n$ in this study but with the bias reducing with increasing $n$. The distribution of $\theta_{ml}$ is bimodal for small values of $n$ with one mode at zero and the other mode larger than $\theta$. Hence, for small values of $n$ the MLE of $\theta$ tend to set one of the components of $\theta$ equal to zero to reduce the dimensionality. This causes the surface to appear noisy in the remaining variables and therefore the estimates of the other components are too large. The distribution of $\theta_{ml}$ is unbiased and symmetric and appears normal when $n \approx 10 \times p$. This rule of thumb is only good for $p = 1, \ldots, 5$ and the for the correlation functions investigated. The components of $\theta$ are slightly positively correlated for large $n$.

The above conclusions suggest that computer experiment designs with relatively small sample sizes may yield badly distributed MLE. While the distribution of $\mu_{ml}$ is stable, the distribution of $\sigma^2_{ml}$ is highly skewed. A suggested minimum sample size rule for good estimates of $\sigma^2$ is $n_{\text{min}} = 4 \times p$. On the other hand, too many samples do not significantly improve the MLE of $\mu$ and $\sigma^2$. Another suggested rule for the maximum sample size is $n_{\text{max}} = 10 \times p$ for efficiency of estimation.

More critical is the distribution of $\theta_{ml}$. This is very non-normal for very small $n$ having a tendency of estimating components zero while over estimating the other components. If the sample size is small compared to the dimensionality of the design space, the maximum likelihood method will arbitrarily estimate no relationship
between some of the inputs and the response while overestimating the other relationships. A minimum sample size to guard against this is \( n_{\text{min}} = (5, 10, 15, 25, 45) \) for \( p = 1, \ldots, 5 \), respectively. The variance of \( \theta \) reduces very quickly for small \( n \) and did not converge for any of the simulations studied. Hence, there is inconclusive evidence whether \( \theta_{\text{ml}} \) is a consistent estimate if the design space is saturated.

As a practical note, predictions are more precise as the sample size increases since the approximating model is correct at more points. However, the inversion of \( V_D \) becomes more computationally intense. Further, the actual effect on prediction of changes in the parameter estimates is unclear. Therefore, the sample sizes suggested above are only rules for good maximum likelihood estimation and not necessarily good for overall prediction.
Chapter 4

A Design Comparison Study

This chapter presents the results of a simulation study investigating the robustness properties of the maximin ($D_M$), integrated mean square error ($D_I$) and Faure ($D_F$) computer experimental designs. In particular, 9-point designs in two dimensions and 25-point designs in four dimensions are evaluated. While mean absolute error and maximum error criteria were investigated, the main criterion of interest is the integrated mean square error (IMSE) of prediction.

The Kriging model, equation (2.1), approximates the simulator function by the sample path of a random function. For the purposes of evaluating the performance of different designs, this model is assumed correct with the hypothesized simulator functions being actual realizations of Gaussian random functions. In particular, the performance of the designs with respect to IMSE is evaluated for different underlying correlation functions. Equation (2.34) can be used to quantify the expected value of this criterion when the correlation function is known. However, in practice the correlation function is at best known up to a family parameterized by $\theta$. This study takes into consideration the added difficulty in using MLE for $\beta, \sigma^2$ and $\theta$ for known correlation families.

As discussed above, $D_I$ is dependent upon $R(\cdot)$ while $D_M$ and $D_F$ are not. The two dimensional $D_I$ is that recommended in Sacks et al [56], while the four dimensional $D_I$ was constructed for $R(d) = \exp\{-\theta \sum_{j=1}^{4} d_j^2\}$ with $\theta = 2$. The linear component of the model is assumed to be a constant $\mu$ and without loss of
CHAPTER 4. A DESIGN COMPARISON STUDY

generality $\mu = 0$ and $\sigma^2 = 1$ for all simulations.

4.1 Methods

For $p = 2, 4$, $D_{MA}$, $D_F$, and $D_I$ designs were created. $D_I$ is generally optimal with respect to the expected IMSE for only one selected correlation function. $D_I$ is not guaranteed to be IMSE optimal for surface generated with other correlation functions. Further, an evaluation grid $(X_E)$ of size $N$ was created by using a Halton-Hammersley sequence [28]. These four sets of points were combined to form

$$X' = (D'_{MA}, D'_F, D'_I, X'_E)$$

the realization set of points where the simulated surfaces will be observed. For various correlation functions as described below, the covariance matrix $V_X$ is computed and a Cholesky decomposition is performed to find $C_X$ where $V_X = C'_X C_X$.

The Cholesky matrix is then used in generating the 100 and 120 realizations of the random function for the given correlation function for two and four dimensions respectively. For each realization, the vector of observations is computed by $Y = C'_X \xi$ where $\xi$ is a vector of independent standard normal random variables generated by a pseudo-random number generator. Hence $Y$ can be decomposed into the realizations for each of the three designs and the evaluation grid by

$$Y' = (Y'_{MA}, Y'_F, Y'_I, Y'_E).$$

For notational purposes, let $D$ denote any of the three designs and $Y_D$ the observed realizations at those points.

For each simulated realization and for each design $D$, MLE for $\mu, \sigma^2$ and $\theta$ were estimated. See below for problems encountered in using the MLE. $Y_E$ was predicted by using equation (2.7) with the MLE. Hence, each design produces estimates of $Y_E$ and the actual errors of prediction on the evaluation grid are computed yielding $\epsilon_{MA}, \epsilon_F$ and $\epsilon_I$. These errors are used in approximating the performance criteria
for each design and surface realization by

\[
ASE = \frac{1}{N} \sum_{i=1}^{N} \epsilon^2 \tag{4.3}
\]

\[
AAE = \frac{1}{N} \sum_{i=1}^{N} |\epsilon| \tag{4.4}
\]

\[
ME = \max\{|\epsilon|\} \tag{4.5}
\]

where ASE is the average squared error, AAE is the average absolute error and ME is the maximum error. Both ASE and AAE are averages over the evaluation grid and are hoped to be close to their associated integrated quantities. ME is the maximum error over the evaluation grid and will always underestimate the true maximum error over the design space. These criteria are pooled over the \(B\) realizations for comparison across the designs for a given correlation function. For example, \(\overline{ASE} = 1/B \sum_{b=1}^{B} ASE_b\). The method allows for comparison of the prediction performance of these three types of designs over identical surface realizations.

### 4.2 Results

#### 4.2.1 Two dimensions

Figure 4.1 displays the nine-point \(D_M\), \(D_I\), and \(D_F\) designs in two dimensions. \(D_M\) (0) has eight out of nine points on the edge of the design space with the remaining point in the center. Incidentally, this design is the same as the optimal maximum entropy design for correlation functions with rapidly dropping correlations. Both \(D_F\) (+) and \(D_I\) (*) designs have all their points in the interior of the design space. \(D_F\) has close to uniform marginal projections onto either of the axes. \(D_I\)'s marginal projections show nine unique points but with clumps on the extremes and fewer points in the middle. This design is optimal for \(\theta = 1\) when there is a quadratic mean term. Without the quadratic term, the design looks similar to \(D_M\) but smaller shrunk towards the center \(S\) with the middle column of points shifted. A 300 point Halton-Hammersley sequence was used for the evaluation grid.
Figure 4.1: \( n = 9 \) and \( p = 2 \) designs: \( "0" = D_M, "+" = D_F, \) and \( "*" = D_I. \)
CHAPTER 4. A DESIGN COMPARISON STUDY

For mathematical convenience, a product correlation rule is adopted and hence only univariate correlation functions need to be investigated. The exponential, Gaussian and Bessel correlation families were used to generate the random function realizations. The exponential family is given by \( R_e(d) = \exp\{-\theta d\} \) and produces non-differentiable responses. The Gaussian correlation function is \( R_g(d) = \exp\{\theta d^2\} \) and produces infinitely mean-square differentiable responses. The Bessel correlation function is \( R_b(d) = \sqrt{2/9\pi} \theta |d|^{2.5} K(\theta |d|) \) where \( K(\cdot) \) is a modified Bessel function of order 2.5. Realizations from this correlation function are twice mean-square differentiable. In all three of the correlation functions, \( \theta \) controls the range of correlations with small values of \( \theta \) indicating high levels of correlations while large values of \( \theta \) yield lower levels of correlations.

Table 4.1 shows the ASE over 100 simulations for the three designs and for the Gaussian correlation function with various \( \theta \) pairs. The expected IMSE, when the true values of \( \theta \) are known, is also given. As expected, \( D_I \) out performs the other designs with respect to IMSE with \( D_F \) consistently better than \( D_M \). In general, the use of the MLE increases \( \overline{ASE} \) over the expected IMSE when \( \theta \) is known. The relative increase, \( ASE/IMSE \) ranges from 3.2 to 137.5 with the relative increase greater for small values of the expected IMSE. Therefore, designs with small expected IMSE are affected the most by using the MLE of the parameters. The increases in expected IMSE by using \( D_M \) instead of \( D_I \), \( IMSE_M/IMSE_I \) ranges from 3.8 to 12.6 while the actual increases \( \overline{ASE}_M/\overline{ASE}_I \) are only 1.6 to 2.0. For \( D_F \) these increases are 1.6 to 3.8 for \( IMSE_F/IMSE_I \) and 1.1 to 1.7 for \( \overline{ASE}_F/\overline{ASE}_I \).

While \( D_I \) is shown to be the best design for smooth surfaces, the expected relative efficiency drastically inflates the true efficiency over the other designs.

When the correlation family is the exponential, \( D_F \) and \( D_I \) perform almost identically in both expected IMSE and \( \overline{ASE} \) as indicated in Table 4.2. Both of these designs are better than \( D_M \). Again \( \overline{ASE} \) is larger than the expected IMSE with ratios from 1.0 to 2.4. This indicates that when the surface is not differentiable, \( D_F \) performs similarly to the optimal design for differentiable surfaces.

The results when using the Bessel correlation function are not much different from the Gaussian results. Specifically, \( D_M \) performs poorly. \( D_I \) is generally better
CHAPTER 4. A DESIGN COMPARISON STUDY

<table>
<thead>
<tr>
<th>$\exp{-\theta \times d^2}$</th>
<th>$D_M$</th>
<th>$D_F$</th>
<th>$D_I$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASE</td>
<td>EIMSE</td>
<td>ASE</td>
</tr>
<tr>
<td>(.5,.5)</td>
<td>0.043</td>
<td>0.0004</td>
<td>0.018</td>
</tr>
<tr>
<td>(.5,2)</td>
<td>0.23</td>
<td>0.010</td>
<td>0.13</td>
</tr>
<tr>
<td>(.5,10)</td>
<td>0.77</td>
<td>0.24</td>
<td>0.52</td>
</tr>
<tr>
<td>(2,2)</td>
<td>0.44</td>
<td>0.019</td>
<td>0.28</td>
</tr>
<tr>
<td>(2,10)</td>
<td>0.83</td>
<td>0.24</td>
<td>0.66</td>
</tr>
<tr>
<td>(10,10)</td>
<td>1.85</td>
<td>0.42</td>
<td>1.85</td>
</tr>
</tbody>
</table>

Table 4.1: ASE and expected IMSE for $p = 2$ and $n = 9$ designs with the Gaussian correlation family.

<table>
<thead>
<tr>
<th>$\exp{-\theta \times d}$</th>
<th>$D_M$</th>
<th>$D_F$</th>
<th>$D_I$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASE</td>
<td>EIMSE</td>
<td>ASE</td>
</tr>
<tr>
<td>(.5,.5)</td>
<td>.31</td>
<td>.16</td>
<td>.25</td>
</tr>
<tr>
<td>(.5,2)</td>
<td>.65</td>
<td>.37</td>
<td>.53</td>
</tr>
<tr>
<td>(.5,5)</td>
<td>.98</td>
<td>.68</td>
<td>.68</td>
</tr>
<tr>
<td>(.5,20)</td>
<td>1.09</td>
<td>1.10</td>
<td>.93</td>
</tr>
<tr>
<td>(2,2)</td>
<td>.88</td>
<td>.53</td>
<td>.78</td>
</tr>
<tr>
<td>(2,5)</td>
<td>.98</td>
<td>.76</td>
<td>.89</td>
</tr>
<tr>
<td>(2,20)</td>
<td>1.08</td>
<td>1.06</td>
<td>1.06</td>
</tr>
<tr>
<td>(5,5)</td>
<td>1.07</td>
<td>.89</td>
<td>1.02</td>
</tr>
<tr>
<td>(5,20)</td>
<td>1.07</td>
<td>1.06</td>
<td>1.10</td>
</tr>
<tr>
<td>(20,20)</td>
<td>1.10</td>
<td>1.09</td>
<td>1.11</td>
</tr>
</tbody>
</table>

Table 4.2: ASE and expected IMSE for $p = 2$ and $n = 9$ designs with the exponential correlation family.
CHAPTER 4. A DESIGN COMPARISON STUDY

<table>
<thead>
<tr>
<th>Bessel</th>
<th>(D_M)</th>
<th>(D_F)</th>
<th>(D_I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta)</td>
<td>ASE</td>
<td>IMSE</td>
<td>ASE</td>
</tr>
<tr>
<td>(0,2)</td>
<td>.017</td>
<td>.0026</td>
<td>.071e-4</td>
</tr>
<tr>
<td>(0,4)</td>
<td>.102</td>
<td>.032</td>
<td>.152e-3</td>
</tr>
<tr>
<td>(0,10)</td>
<td>.38</td>
<td>.33</td>
<td>.015</td>
</tr>
<tr>
<td>(2,2)</td>
<td>.011</td>
<td>.005</td>
<td>.003</td>
</tr>
<tr>
<td>(2,4)</td>
<td>.045</td>
<td>.035</td>
<td>.017</td>
</tr>
<tr>
<td>(4,4)</td>
<td>.093</td>
<td>.064</td>
<td>.053</td>
</tr>
<tr>
<td>(2,10)</td>
<td>.32</td>
<td>.33</td>
<td>.085</td>
</tr>
<tr>
<td>(4,10)</td>
<td>.38</td>
<td>.35</td>
<td>.19</td>
</tr>
<tr>
<td>(2,25)</td>
<td>.35</td>
<td>.83</td>
<td>.12</td>
</tr>
</tbody>
</table>

Table 4.3: ASE and expected IMSE for \(p = 2\) and \(n = 9\) designs with the Bessel correlation family.

when both components of \(\theta\) are positive. However, when the first component is set to zero, \(D_F\) is significantly better than the other designs. Specifically, \(D_M\) is poor since it only has three unique projection points. \(D_I\) while having good projections has double ASE and expected IMSE of \(D_F\). Interestingly, the increase in ASE by using the MLE is not as pronounced for the Bessel correlation family. The ratios ASE/IMSE, are from 1.0 to 1.6. These results indicate that if one of the input variables has no relationship with the response, \(D_F\) is significantly better than \(D_I\). However, \(D_I\) is still slightly better than \(D_F\) while \(D_M\) is a consistently poor.

4.2.2 Four dimensions

\(D_M\) was not investigated in four dimensions due to its poor performance in two dimensions. Additionally, finding \(D_M\) is a unique geometric problem for each value of \(n\) and \(p\) and generally difficult to solve, particularly when \(p > 2\). \(n = 25\) since \(D_F\) is best when \(n\) is an integer power of a prime number and \(25 = 5^2\). Figure 4.2 shows all pairwise marginal projections onto planes of \(D_F\) (+) and optimal \(D_I\) (*) design for the Gaussian correlation function with \(\theta' = (2,2,2,2)\). The marginal projections of \(D_F\) have 25 distinct points in each of the axes. The range of \(D_F\) in each variable is .02 to .98. \(D_I\) marginal projections group into nine clumps that
CHAPTER 4. A DESIGN COMPARISON STUDY

look similar to a central composite design. The individual input variable ranges are consistently from .11 to .89. Hence the convex hull of $D_I$ covers at most 40% of the design space compared to 85% for $D_F$.

Only five different correlation functions were investigated in four dimensions. Four involve the Gaussian correlation family and one uses the Bessel correlation family. Three of the functions have one of the components of $\theta$ smaller than the rest. The choice of which component to change is arbitrary and hence four subsets of simulations were performed with each set having a different component of $\theta$ different. 120 response surfaces were generated for each unique correlation function. A Halton-Hammersley evaluation grid of size $N = 200$ was utilized for evaluating the predictions. The size of the evaluation grid is restricted due to the computational expense and numerical instability of performing the Cholesky decomposition on the $(2n+N) \times (2n+N)$ covariance matrix of the two designs and the evaluation grid.

Some problems were encountered in calculating the MLE, particularly for $D_I$ with one component of $\theta$ small. This is due to the high correlations within the marginal projection clumps. This adds a selection bias to the study. The relationship between the validity of the MLE and the prediction performance is unclear. If the difficulty in estimation indicates poor prediction performance then the results given are biased in favor of $D_I$.

Table 4.4 gives the number out of 120 simulations that produced valid MLE, and $\overline{\text{ASE}}$ and the expected IMSE if the parameters were known. For $\theta' = (2, 2, 2, 2)$, $D_I$ is roughly twice as efficient as $D_F$. When one of the components of $\theta$ is set to .1 this efficiency ratio is lowered to 1.15. When one of the components of $\theta$ is set to zero, the results are mixed. For two of the components, $D_I$ is better while $D_F$ is better for the other two. On average the two designs appear to be equally efficient. However, all of $D_I$ have maximum likelihood estimation problems. In particular, when the fourth component of $\theta$ is set to zero, only 6 out of the 120 surface realizations yield valid MLE. Similar problems are encountered for the Bessel correlation family when all components of $\theta$ are set to 2 except one which is set to .1. On average, $\overline{\text{ASE}}_F/\overline{\text{ASE}}_I$ is 1.10 for the Bessel case while $\text{IMSE}_F/\text{IMSE}_I$
Figure 4.2: Pairwise marginal projections of $n = 25$ and $p = 2$ designs: 
"+" = $D_F$ and "*" = $D_I$. 
<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$D_F$</th>
<th>$D_I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$B$</td>
<td>$\overline{ASE}$</td>
</tr>
<tr>
<td>$(2,2,2,2)$</td>
<td>112</td>
<td>$(.388)^2$</td>
</tr>
<tr>
<td>$(1,2,2,2)$</td>
<td>119</td>
<td>.039</td>
</tr>
<tr>
<td>$(2,1,2,2)$</td>
<td>115</td>
<td>.048</td>
</tr>
<tr>
<td>$(2,2,1,2)$</td>
<td>114</td>
<td>.042</td>
</tr>
<tr>
<td>$(2,2,2,1)$</td>
<td>120</td>
<td>.043</td>
</tr>
<tr>
<td>Average</td>
<td>117</td>
<td>$(.207)^2$</td>
</tr>
<tr>
<td>$(0,2,2,2)$</td>
<td>82</td>
<td>.022</td>
</tr>
<tr>
<td>$(2,0,2,2)$</td>
<td>103</td>
<td>.022</td>
</tr>
<tr>
<td>$(2,2,0,2)$</td>
<td>102</td>
<td>.025</td>
</tr>
<tr>
<td>$(2,2,2,0)$</td>
<td>90</td>
<td>.027</td>
</tr>
<tr>
<td>Average</td>
<td>94</td>
<td>$(.155)^2$</td>
</tr>
<tr>
<td>$(1,2,3,4)$</td>
<td>120</td>
<td>.188</td>
</tr>
<tr>
<td>$(2,3,4,1)$</td>
<td>117</td>
<td>.181</td>
</tr>
<tr>
<td>$(3,4,1,2)$</td>
<td>120</td>
<td>.177</td>
</tr>
<tr>
<td>$(4,1,2,3)$</td>
<td>119</td>
<td>.189</td>
</tr>
<tr>
<td>Average</td>
<td>119</td>
<td>$(.429)^2$</td>
</tr>
<tr>
<td>Bessel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(.1,2,2,2)$</td>
<td>109</td>
<td>.0055</td>
</tr>
<tr>
<td>$(2,1,2,2)$</td>
<td>105</td>
<td>.0056</td>
</tr>
<tr>
<td>$(2,2,1,2)$</td>
<td>102</td>
<td>.0055</td>
</tr>
<tr>
<td>$(2,2,2,1)$</td>
<td>108</td>
<td>.0056</td>
</tr>
<tr>
<td>Average</td>
<td>106</td>
<td>$(.075)^2$</td>
</tr>
</tbody>
</table>

Table 4.4: $\overline{ASE}$ and expected IMSE for $p = 4$ and $n = 25$ designs.

is .65. When the components of $\theta$ are set arbitrarily to different values from the set $\{1,2,3,4\}$ the results suggest that $D_I$ is, as expected, over 1.5 times as efficient as $D_F$.

The $\overline{ASE}$ is from 1.3 to 1.7 times larger than the expected IMSE. The $AAE$ measure is typically .66 to .75 the size of $\sqrt{\overline{ASE}}$ while the $ME$ is typically 5 to 6 times the size of $AAE$. These results indicate that the 25-point $D_I$ in 4 dimensions performs better than $D_F$ when all of the input variables sufficiently affect the response variable. However, if one of the input variables is not related to the response, $D_I$ might cause difficulty in finding the MLE and will not provide significantly better results that $D_F$. 
4.3 Summary

A simulation study investigating the performance of the $D_M$, $D_F$ and $D_I$ has been presented. The results indicate, that overall $D_M$ is a poor design in terms of prediction and should be avoided. When information is known concerning the type of correlation family and the general magnitude of $\theta$ $D_I$ is the best selection. It is robust to small changes in $\theta$ and generally performs well particularly for smooth surfaces. However, if one expects that some of the input variables have no or little relationship with the response, then $D_I$ may be a poor design. In this situation, $D_F$ performs the best in both prediction performance and in avoiding maximum likelihood estimation problems.

$D_F$ is computationally inexpensive to generate while $D_I$ requires a $n \times p$ dimensional optimization for a given correlation function. Hence, the cost of obtaining $D_I$ must be evaluated against the cost of taking additional simulations using $D_F$. Further, in finding optimal designs, the expected IMSE from equation (2.34) will be smaller than the true error due to the uncertainty in the parameter values. This increase is dependent on the magnitude of the expected IMSE with smaller expected values yielding larger prediction errors.
Chapter 5

The Root Mean Square Error

5.1 Introduction and Background

As with response surface methods, the Kriging model provides an estimate of the prediction error. That is, with each prediction, an associated standard deviation of the prediction is also given. In least squares regression theory, the measure of error or standard error is derived from the measurement error distributional assumptions. These distributional assumptions are not valid in computer experiments since deterministic computer simulators have no random or measurement error. In the Kriging model, the measure of error is the root mean square error (RMSE) which is based upon the distributional assumptions of the random function $Z(\cdot)$, which models the systematic bias of the model. The RMSE is also known as the posterior standard deviation. This chapter investigates the performance of the root mean square error (or the posterior standard deviation) as a measure of prediction error.

The root mean square error (RMSE) for a prediction $\hat{Y}(x_0)$, where $x_0 \in S$, is given by

$$\text{RMSE}[\hat{Y}(x_0)] = \sqrt{\sigma^2 - (f'(x_0), v_{x_0}) \left( \begin{array}{cc} 0 & F'_D \\ F_D & V_D \end{array} \right)^{-1} \left( \begin{array}{c} f(x_0) \\ v_{x_0} \end{array} \right)}.$$  \hspace{1cm} (5.1)

RMSE is dependent upon the design $D$, the linear terms of the model through $F_D$ and $f(x_0)$, the correlation function $R(\cdot)$, and the random function variance $\sigma^2$. 

77
CHAPTER 5. **THE ROOT MEAN SQUARE ERROR**

RMSE will be very small near design points and in fact equals zero for $x \in D$. As a prediction point moves away from all design points, the RMSE increases. Clearly, $\sigma^2$ plays an important role since RMSE is proportional to $\sigma$. Further, changes in $R(\cdot)$ have great effect on RMSE. If $R(\cdot)$ has relatively small correlations, the RMSE will be relatively large while larger correlations correspond to smaller values of RMSE. The effect of changing the linear component of the model, that is $F_D$ and $f(x_0)$, is unclear and will not be discussed here.

The RMSE at a particular point $x_0$ is also the standard deviation of the error distribution for the prediction, $\hat{Y}(x_0)$. Consider many random realizations of a random function $Z(\cdot)$ with fixed parameters (i.e. $\beta$, $\sigma^2$ and $\theta$, the parameters of the correlation function) and that all have the same $Y_D$. Then if the parameters of the random function are known, the realized values of the many realizations at the point $x_0$ would be distributed with mean $\hat{Y}(x_0)$ and standard deviation $\text{RMSE}[\hat{Y}(x_0)]$. If the random function, $Z(\cdot)$, is assumed to be Gaussian, then the standardized error

$$
\epsilon(x_0) = \frac{[Y(x_0) - \hat{Y}(x_0)]}{\text{RMSE}[\hat{Y}(x_0)]}
$$

is distributed as a standard normal. Therefore, RMSE appears to be a useful measure of the the error in prediction when the underlying parameters are known.

In most computer experiments, the underlying parameters of the Kriging model are unknown and must be estimated. Further, when modeling one response, prediction errors are needed for many points within the design space and these points are correlated. That is, within a realization, points that are close to each other have similar prediction errors. Hence, whether the standardized errors, $\epsilon$, will look jointly Gaussian is unclear. Sacks et al [57] provide an example where they evaluate the performance of $\text{RMSE}_{ml}$. Their example concerns the modeling of circuit clock skews with a 32 point design in 6 dimensions. They plot the standardized errors from an evaluation grid of one hundred random points in the design space versus the standard normal quantiles. This plot is known as a Q-Q plot. In their example, this plot closely follows the $45^\circ$ line indicating close agreement to a standard normal distribution. Their results indicate that the $\text{RMSE}_{ml}$ does indeed provide
a good measure of prediction error. However, more investigation is needed as the example below indicates.

Consider the example of a basic experiment given in Figure 5.1. Here, the experiment takes place in one dimension with a two point design \( D = \{0, 1\} \). The observed values for both design points are zero. Hence the BLUP for all points within the design space are also zero. If the true response is positive between the design points, then every prediction point will be estimated too low and all of the corresponding \( \epsilon \)'s will be positive. In this example, the Q-Q plot (Figure 5.1(b)) does not follow the expected 45° line of a standard normal. A Q-Q plot that follows any straight line indicates close agreement to a Gaussian distribution. Further, the slope of the line is inversely related to the average \( \text{RMSE}_{ml} \).

The following simulation study provides some insight into the performance of the \( \text{RMSE}_{ml} \) within the assumptions of the Kriging model. This study simulates responses at random within the framework of the Kriging model. The realizations on a fixed design are then used to estimate the the parameters of the process. These estimates are utilized in predicting the generated response on a random grid along with the \( \text{RMSE}_{ml} \). The standardized errors are evaluated by fitting a line to the Q-Q plot. The coefficients of the line are used in making generalizations concerning the validity of \( \text{RMSE}_{ml} \) as a predictor of error.

### 5.2 Methods

This simulation study follows the methods for the design comparison study presented in Section 4.1. For this study, the nine-point and twenty-five point Faure designs, in two dimensions and four dimensions respectively, were utilized. Additionally, a 300 point in two dimensions and a 200 point in four dimensions Halton-Hammersley sequence [27, 28] was used to evaluate the standardized errors. Specifically, for both two and four dimensions, let

\[
X' = (D', E')
\]  

(5.3)
Figure 5.1: A $n = 2$ example with (a) the realization and (b) the corresponding Q-Q plot.
be the composite of the Faure design and the evaluation grid, and let \( V_X \) be the covariance matrix of \( X \) for the covariance functions specified below. The Cholesky decomposition of \( V_X \) was performed to find \( C_X \) where \( V_X = C_X' C_X \).

The Cholesky matrix was then used to generate approximately one hundred realizations of the random function for each of the covariance functions discussed below. In the results discussed below, only random functions with zero mean \((\mu = \beta = 0)\) and \(\sigma^2 = 1\) were investigated. Each realization is computed by \( Y = C_X' \xi \) where \( \xi \) is a vector of independent standard normal random variables generated by a pseudo-random number generator. \( Y \) can be decomposed into the realization for the design and the evaluation grid by

\[
Y' = (Y_D', Y_E').
\]  

(5.4)

Even though the underlying parameters of the random function are known, \( Y_D \) was used to find the maximum likelihood estimates for \( \mu, \sigma^2 \), and \( \theta \) for each of the realizations. This was done to better reflect the results expected from actual computer experiments. Next \( Y_E \) was predicted by using equation (2.7) and the \( \text{RMSE}_{ml} \) was calculated by using equation (5.1) with the maximum likelihood estimates. The standardized errors were then computed by

\[
\varepsilon_i = \frac{[Y(x_i) - \hat{Y}(x_i)]}{\text{RMSE}_{ml}(\hat{Y}(x_i))}
\]  

(5.5)

for each of the \( N \) points in the evaluation grid.

A linear regression of the form

\[
\varepsilon_i = b_0 + b_1 \zeta_i + e_i
\]  

(5.6)

where \( \zeta_i \) is the \( i/N \) standard normal quantile, was calculated to summarize the performance of the standardized residuals for each of the realizations. The Q-Q plots are anticipated to be close to linear indicating the standardized errors are approximately normally distributed. \( b_0 \) is an estimate of the overall bias in prediction over the evaluation grid while \( b_1 \) is an estimate of the scale of the standardized error distribution. If \( \hat{b}_1 \approx 1 \) and \( \hat{b}_0 \approx 0 \), then the \( \text{RMSE}_{ml} \) is a valid estimate of
the prediction error. However, if $\hat{b}_1 > 1$ then the standardized errors are too large meaning that the RMSE_{ml}s are on average too small. Alternatively, if $\hat{b}_1 < 1$ then the standardized errors are too small indicating that the RMSE_{ml}s are too large on average.

5.3 Results

5.3.1 Nine Point Faure Design in Two Dimensions

The nine-point Faure design was used for the two dimensional analysis and is displayed by the "+" symbol in Figure 4.1. Random realizations of random functions with $\mu = 0$, $\sigma^2 = 1$, and the Bessel, Gaussian, and exponential correlation families for various pairs of parameters, $\theta$, were generated. Maximum likelihood estimates of the process parameters were estimated from the responses generated at the design points. The standardized errors on the 300 point evaluation grid were plotted against the associated quantiles of a standard normal and a simple linear regression line was estimated.

As an example, Figure 5.2 displays the Q-Q plots from four of the 100 realizations generated with the Bessel correlation function, $\theta = (2, 2)$, and $\nu = (2.5, 2.5)$. Table 5.1 contains the maximum likelihood estimates of $\mu$, $\sigma$ and $\theta$ along with the estimated regression coefficients. The first plot shows close agreement with a standard normal. $\hat{b}_1 \approx 1$ while $\hat{b}_0$ is slightly negative indicating that the surface is overestimated. The second and third plots look fairly straight but the slopes are both too large. This indicates that the $\text{RMSE}_{ml}$ is too small. Notice that both of these plots have large intercepts, ($\hat{b}_0$), but of opposite direction. Alternatively, the fourth plot has very small intercept and small slope ($\hat{b}_1 = .51$) suggesting that the $\text{RMSE}_{ml}$ for this realization are overall too large. Over the 100 realizations, the intercepts average close to zero while the slopes average greater than one (See Table 5.3).

As another example, Figure 5.3 displays the Q-Q plots from four realizations of a random function with the Bessel correlation function, $\theta = (2, 4)$, and $\nu = (2.5, 2.5)$. 
Figure 5.2: Q-Q plots from 4 realizations with the Bessel correlation function and $\theta = (2, 2)$. 
CHAPTER 5. THE ROOT MEAN SQUARE ERROR

<table>
<thead>
<tr>
<th></th>
<th>( \hat{\mu} )</th>
<th>( \hat{\sigma} )</th>
<th>( \hat{\theta}_1 )</th>
<th>( \hat{\theta}_2 )</th>
<th>( \hat{b}_0 )</th>
<th>( \hat{b}_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.25</td>
<td>0.45</td>
<td>2.70</td>
<td>3.46</td>
<td>-0.11</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.77</td>
<td>0.17</td>
<td>2.74</td>
<td>7.18</td>
<td>-0.38</td>
<td>1.57</td>
</tr>
<tr>
<td>3</td>
<td>0.65</td>
<td>0.98</td>
<td>1.45</td>
<td>1.55</td>
<td>0.38</td>
<td>1.65</td>
</tr>
<tr>
<td>4</td>
<td>0.45</td>
<td>0.86</td>
<td>2.15</td>
<td>3.11</td>
<td>-0.01</td>
<td>0.51</td>
</tr>
</tbody>
</table>

Table 5.1: Maximum likelihood estimates and Q-Q coefficients from four realizations with the Bessel correlation function and \( \theta = (2, 2) \).

<table>
<thead>
<tr>
<th></th>
<th>( \hat{\mu} )</th>
<th>( \hat{\sigma} )</th>
<th>( \hat{\theta}_1 )</th>
<th>( \hat{\theta}_2 )</th>
<th>( \hat{b}_0 )</th>
<th>( \hat{b}_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.52</td>
<td>0.94</td>
<td>0.74</td>
<td>4.89</td>
<td>0.00</td>
<td>1.66</td>
</tr>
<tr>
<td>2</td>
<td>-0.28</td>
<td>1.16</td>
<td>1.56</td>
<td>4.02</td>
<td>-0.10</td>
<td>0.70</td>
</tr>
<tr>
<td>3</td>
<td>-0.55</td>
<td>1.09</td>
<td>0.47</td>
<td>3.93</td>
<td>-0.26</td>
<td>4.97</td>
</tr>
<tr>
<td>4</td>
<td>-1.22</td>
<td>0.57</td>
<td>3.65</td>
<td>7.79</td>
<td>-0.13</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.2: Maximum likelihood estimates and Q-Q coefficients from four realizations with the Bessel correlation function and \( \theta = (2, 4) \).

The maximum likelihood estimates and estimated coefficients are given in Table 5.2. The first Q-Q plot is typical with zero intercept and slope around 1.6. However Q-Q plots similar to the second plot are common. This plot has slope smaller than one. Similarly, the third plot while being extreme with slope close to five is not uncommon. The fourth plot is not very linear but the fitting line has slope equal to one and does not adversely effect the investigation of \( \hat{b}_1 \). The 100 realizations for this correlation function indicates that the RMSE\(_{mis} \)s are smaller than the \( \theta = (2, 2) \) case with the mean, \( \tilde{b}_1 = 1.7 \), and the median, \( \tilde{b}_1 = 1.3 \).

The rest of the results for the Bessel correlation function are summarized in Table 5.3. This Table contains the mean, \( \bar{b} \), and the median, \( \tilde{b} \), of \( \hat{b}_0 \) and \( \hat{b}_1 \) over the 100 realizations for the various parameter pairs. Further, Figure 5.4 shows the scatterplots of the corresponding estimated coefficients of the Q-Q plots for the various parameter pairs. These results show that for all \( \theta \) investigated, the intercept term averages near zero. Further, the results indicate that when the components of \( \theta \) are medium (.5 – 5.) the slope of the Q-Q plots are generally too larger than one indicating that the RMSE\(_{mis} \)s are too small. However, when one of the components of \( \theta \) is zero, this tendency is reversed with a slope less than one and hence the
Figure 5.3: Q-Q plots from 4 realizations with the Bessel correlation function and $\theta = (2, 4)$. 
CHAPTER 5. THE ROOT MEAN SQUARE ERROR

<table>
<thead>
<tr>
<th>$\theta_2$</th>
<th>2</th>
<th>4</th>
<th>10</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>$b_0$</td>
<td>$b_1$</td>
<td>$b_0$</td>
<td>$b_1$</td>
</tr>
<tr>
<td>0</td>
<td>.00</td>
<td>.83</td>
<td>.00</td>
<td>.98</td>
</tr>
<tr>
<td></td>
<td>.00</td>
<td>.80</td>
<td>.01</td>
<td>.83</td>
</tr>
<tr>
<td>2</td>
<td>-.01</td>
<td>1.42</td>
<td>.01</td>
<td>1.70</td>
</tr>
<tr>
<td></td>
<td>.00</td>
<td>1.15</td>
<td>-.01</td>
<td>1.27</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>.05</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>.02</td>
<td>1.19</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: Mean and median Q-Q plot coefficients for the Bessel correlation function

RMSE$_{ml}$ are too large. As $\theta$ becomes large, the slope of the Q-Q plots average close to one with a standard deviation between realizations of approximately .4. The median of coefficients for the 100 realizations are also provided since as the scatterplots indicate, the distributions of $\hat{b}_1$ are generally skewed to the right. In fact, $\hat{b}_1$ appears to have a log-normal distribution.

Table 5.4 and Figure 5.5 contain the results from the simulations generated with the exponential correlation function. Overall, these results show that $\hat{b}_0$ averages zero for all $\theta$ investigated. Additionally, $\hat{b}_1 > 1$ in both mean and median over the hundred realizations. This tendency is most pronounced when $\theta_j < 2$ with medians ranging from 1.24 to 1.32, and less severe when $\theta_j > 5$. Generally, the distribution of $\hat{b}_1$ is skewed to the right with some estimates greater than 4 as shown in Figure 5.5. Not only is there a general tendency for the RMSE$_{ml}$ to underestimate the true error but in some cases these estimates can be off by a factor of 4.

Table 5.5 and Figure 5.6 contain the results from the simulations generated with the Gaussian correlation function. Overall, these results show that $\hat{b}_0$ averages zero for all values of $\theta$ except for $\theta = (10, 10)$. Additionally, $\hat{b}_1$ is significantly larger than one in both average and median over the hundred realizations for all values of $\theta$. This is most extreme for $\theta = (10, 10)$ where $\hat{b}_1 > 6.0$. For the Gaussian correlation function, RMSE$_{ml}$ that are too small by a factor of 5 are uncommon
Figure 5.4: Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the Bessel realizations in $p = 2$. 
Figure 5.5: Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the exponential realizations in $p = 2$. 
CHAPTER 5. THE ROOT MEAN SQUARE ERROR

<table>
<thead>
<tr>
<th>$\theta_2$</th>
<th>.5</th>
<th>2</th>
<th>5</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>$b_0$</td>
<td>$b_1$</td>
<td>$b_0$</td>
<td>$b_1$</td>
</tr>
<tr>
<td>.5</td>
<td>.02</td>
<td>1.52</td>
<td>.01</td>
<td>1.50</td>
</tr>
<tr>
<td>2</td>
<td>.02</td>
<td>1.24</td>
<td>.04</td>
<td>1.32</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>.00</td>
<td>1.46</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Mean and median Q-Q plot coefficients for the exponential correlation function

<table>
<thead>
<tr>
<th>$\theta_2$</th>
<th>.5</th>
<th>2</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>$\beta_0$</td>
<td>$\beta_1$</td>
<td>$\beta_0$</td>
</tr>
<tr>
<td>.5</td>
<td>-.02</td>
<td>1.53</td>
<td>-.02</td>
</tr>
<tr>
<td>2</td>
<td>-.02</td>
<td>1.03</td>
<td>-.01</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Mean and median Q-Q plot coefficients for the Gaussian correlation function

but do occur.

5.3.2 Twenty-five point Faure design in Four Dimensions

The study in 4 dimensions used the 25-point Faure design and a 200-point Halton-Hammersley Sequence for the evaluation grid. The marginal projections of the Faure design are displayed in Figure 4.2 by the “+” symbols. The correlation functions investigated were the Bessel with all components of $\nu = 2.5$ and the Gaussian, both with various combinations of $\theta$ as shown in Table 5.6. For the correlation function with unequal components of $\theta$, the components were rotated and the results averaged for Table 5.6.

For all of the correlation functions $b_0$ average to zero in both mean and median.
Figure 5.6: Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the Gaussian realizations in $p = 2$. 
CHAPTER 5. THE ROOT MEAN SQUARE ERROR

<table>
<thead>
<tr>
<th>Family</th>
<th>$\theta$</th>
<th>$B$</th>
<th>$b_0$</th>
<th>$b_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Median</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>(2,2,2,2)</td>
<td>104</td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td>Gaussian</td>
<td>(.1,2,2,2)</td>
<td>382</td>
<td>-.01</td>
<td>-.02</td>
</tr>
<tr>
<td>Gaussian</td>
<td>(0,2,2,2)</td>
<td>346</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>(1,2,3,4)</td>
<td>396</td>
<td>-.01</td>
<td>.00</td>
</tr>
<tr>
<td>Bessel</td>
<td>(.1,2,2,2)</td>
<td>386</td>
<td>-.01</td>
<td>.00</td>
</tr>
</tbody>
</table>

Table 5.6: Mean and median Q-Q plot coefficients for various correlation functions in four dimensions

However, the results for $b_1$ are mixed. For $\theta = (2,2,2,2)$ in the Gaussian case, $\tilde{b}_1 = .32$ and $\tilde{b}_{11} = .28$. In fact, over the 104 realizations for this case, all of the $\tilde{b}_i$'s are less than .9 as shown in Figure 5.7(a). Similarly, when one component is dropped to zero, $\tilde{b}_1 = .27$ and $\tilde{b}_{11} = .27$. Figure 5.7(b) shows that most of the $\tilde{b}_i$ are less than .5 and that in most cases, with these correlation functions, the RMSE$_{mis}$ are at least two times too large. Additionally, for these correlations functions, the RMSE$_{mis}$ are on average too large by a factor of 3 or 4.

For the rest of the correlations presented in Table 5.6 have the averages of $\tilde{b}_1$ larger than one. When one of the components of $\theta$ is set to .1, the $\tilde{b}_1 = 1.83$ and 1.40 for the Bessel and Gaussian families respectively. When $\theta$ is equal to some permutation of (1,2,3,4), $\tilde{b}_1 = 1.45$. For these cases, the RMSE$_{mis}$ are on average too small and in fact 97% of the estimates for the Bessel function were too small.

5.4 Summary

The simulation study presented above, investigates the performance of the RMSE$_{mi}$ as a method of quantifying prediction error. The results were obtained by fitting a simple linear regression line to the Q-Q plot of the standardized errors. Estimates of the regression coefficients were obtained for many realizations of a random function with fixed mean and covariance function. These coefficients are used to evaluate the performance of the RMSE$_{mis}$ in estimating prediction errors. The intercept coefficient’s departure from zero indicate the overall prediction bias
Figure 5.7: Scatterplots of $(\hat{b}_0, \hat{b}_1)$ for the realizations in $p = 4$. 

(a) Gaussian (2,2,2,2)
(b) Gaussian (0,2,2,2)
(c) Gaussian (1,2,3,4)
(d) Bessel (.1,2,2,2)
CHAPTER 5. THE ROOT MEAN SQUARE ERROR

for that generated surface. Departures of the slope coefficient, $b_1$, indicates that
the RMSE$_{m1}$s are biased by a factor inversely related to the departure.

Overall the results suggest that $b_0$ is unbiased. Intuitively, this result is expected
by symmetry. However, the results for $b_1$ are not so clear cut. For most of the
correlation functions investigated, $b_1 > 1$ on average. Consequently, for these
cases, the RMSE$_{m1}$ are too small and their use produces unfound confidence in the
predictions. Alternatively, for some of the correlation functions $b_1 < 1$ on average
and the RMSE$_{m1}$s are too conservative. These latter cases strictly involve the
Gaussian family and are when one component of $\theta$ is zero in two and dimensions and
when all components of $\theta = 2$ in four dimensions. There were too few combinations
of $\theta$ to suggest a generalization of these results.

As noted in Chapter 3, the maximum likelihood estimates of $\sigma^2$ are on average
biased low. Since the RMSE$_{m1}$ is proportional to $\sigma$, this underestimation might lead
to $b_1 > 1$. However, $\sigma$ is also estimated low for the correlations functions associated
with low $\hat{b}_1$. Also, correlations between $\hat{\sigma}_{m1}$ and $\hat{b}_1$ are very low, usually less than
.06. This indicates that the variation in the maximum likelihood estimates between
realizations does not directly effect the overall performance of RMSE$_{m1}$. Further
investigations into the maximum likelihood estimates of $\theta$ found no relationship to
$\hat{b}_1$.

In conclusion, the results of this chapter suggest that the reliability of the root
mean square error as a measure of prediction error is questionable for small sample
sizes. For most of correlation functions investigated, the RMSE$_{m1}$ is too small
leading to narrower confidence bands for the desired confidence level. For some of
the correlations functions, particularly when one of the components of $\theta$ is zero,
the RMSE$_{m1}$ is too large leading to larger, conservative confidence bands. More
research is needed to conclude whether these results have an underlying factor
affecting the magnitude of the RMSE$_{m1}$. 
Chapter 6

Conclusion

A deterministic simulator evaluates a complicated system of equations with no measurement or random error. A computer experiment is appropriate when the evaluations of the simulator are computationally expensive. A design, $D$, is evaluated by the simulator and the results used to build a computationally-inexpensive approximating model. Since the simulator is deterministic, this approximating model’s error is all bias. The Kriging methodology treats the bias term as a realization of a Gaussian random function. This paper has presented the results of several studies designed to investigate some design and estimation issues in the Kriging methodology. This chapter summarizes the results from these studies and proposes areas for future research.

6.1 Summary

Chapter 2 presented and discussed the Kriging model which consists of a linear term and an error term $Z(\cdot)$ which describes the systematic bias of the linear term. $Z(\cdot)$ is controlled by a correlation function $R(\cdot)$. Six families of correlation functions were presented. They are the linear and exponential which yield non-differentiable responses, the cubic and smoothed exponential which yield once differentiable surfaces, the Gaussian which analytic responses, and the Bessel family which controls the order of differentiability with the parameter $\nu$. 
CHAPTER 6. CONCLUSION

The maximum likelihood, cross-validated maximum predictive likelihood, and cross-validated squared bias criteria for estimating the parameters \((\beta, \sigma^2, \theta)\) of the Kriging model were also introduced in Chapter 2. Additionally, 5 design optimality criteria were discussed, and figures showing some of their associated designs were presented. Algorithms have been programmed into PACE for finding the maximum entropy and IMSE designs which are generally only optimal for a specified correlation function. There are no general prescriptions for generating maximin and minimax designs while the Faure design is quick to generate.

Chapter 3 discussed in more depth the MLE of the model parameters. The workings and usefulness of NPSOL, a nonlinear program for solving these parameters, was presented. Since the design space is compact, ergodic theory is not applicable in proving consistency of the maximum likelihood estimators. The results of a simulation study that investigates the small sample properties of the MLE for \(p = 2 - 5\) was presented. The results indicate that \(n \geq 4p\) is sufficient for the distribution of these estimators to be well behaved. Alternatively, \(n \geq 10p\) does not significantly improve the distribution of \(\hat{\mu}_{ml}\) and \(\hat{\sigma}_{ml}^2\). This bound on \(n\) is only important for the MLE of \(\beta\) and \(\sigma^2\). Within the range of \(n\) investigated, larger \(n\) improve the MLE of \(\theta\). Further, larger \(n\) should improve the predictions of the model since the model will be correct in more locations. An alarming result indicates that the MLE of \(\theta\) arbitrarily sets some components to zero when \(n\) is small. This leads to false conclusions regarding the relationship between the input and response variables. \(n_{\text{min}} = (5, 10, 15, 25, 45)\) for \(p = 1, \ldots, 5\) is usually sufficient to guard against this tendency.

Chapter 4 presented the results of a comparative design study. This study compared the performance of the maximin \((D_M)\), Faure \((D_F)\), and optimal IMSE \((D_I)\) designs when the underlying correlation function changes. Specifically, \(n = 9\) in 2 dimensions and \(n = 25\) in 4 dimensions were compared in terms of the average square error (ASE) over an evaluation grid. The results suggest that \(D_I\) is the best when all of the input variables significantly affect the response variable. However, if one or more of the input variables have no effect on the response variable \(D_F\) can be significantly better. Further, \(D_I\) can have problems with calculating the MLE.
$D_M$ fares poorly compared to the other designs.

The model presented in Chapter 2 has an added benefit of providing an estimate of the prediction errors in the root mean square error (RMSE). These are dependent on the underlying covariance function $\sigma^2 R(\cdot)$. If $\sigma^2 R(\cdot)$ is known, the true errors from the model, $Y - \hat{Y}$, when divided by RMSE will be distributed as a standard normal variable. However, in practice $\sigma^2$ and $\theta$ need to be estimated as discussed in Chapters 2 and 3. The effect of using these estimators was investigated in Chapter 5. The results show that $\text{RMSE}_{ml}$ are not reliable with a tendency to under estimate the true prediction error for the designs investigated. This leads to confidence bands that are too narrow, sometimes by a factor of 6 for some $R(\cdot)$. Alternatively, in some situations the error predictions are consistently over estimated leading to larger, conservative confidence bands. This occurred in this study only for the Gaussian correlation family when one component of $\theta$ is zero in $p = 2$ and when all components of $\theta = 2$ for $p = 4$. More research is needed to make more general conclusions concerning the under and over estimation tendencies of $\text{RMSE}_{ml}$.

The results presented in Chapters 3–5 are from simulation studies designed to investigate the performance of the Kriging model. The results do not imply general results but do suggest tendencies and problems when the Kriging model is the correct model. The next section suggest area where new or more research is needed.

### 6.2 Future Work

The MLE results presented in Chapter 3 are mostly for the Gaussian correlation family. Similar studies are needed for the other correlation families, particularly the Bessel. The one result using the Bessel correlation function indicates that $\sigma^2_n(\cdot)$ is larger than for the Gaussian family. Further Table 4.4 suggest that $\theta$ is harder to estimate for the Bessel family than the Gaussian. Also, knowing if one family, such as the Bessel family is flexible enough to be confidently used exclusively in real computer experiments would save analysis time and effort.
CHAPTER 6. CONCLUSION

Chapter 2 presented two other criteria for estimating $\beta, \sigma^2$, and $\theta$. A comparison study of the maximum likelihood, cross-validated maximum predictive likelihood, and cross-validated minimum squared bias estimators to see if one estimator is consistently superior would be interesting. A further study to directly connect the effect of estimating these parameters on prediction is important. The results in Chapter 4 suggest that not knowing the parameters of the model has a large effect on prediction. Knowing how well these parameters need to be estimated for valid prediction would be a major contribution to understanding the practical application of the Kriging methodology.

The results from the design comparison study presented in Chapter 4 are only for two $(n,p)$ combinations. Expanding this study to include other $n$ and $p$ would provide added insight whether the results hold true generally. These results indicate that $D_I$ is the best design out of the three studied when all of the input variables significantly affect the response. However, when one or more of the input variables is not significant, $D_I$ has problems finding the MLE and is not better than $D_F$. Modifying the IMSE design optimality criterion to explicitly include the possibility of insignificant input variables is possible. A discrete prior on $\theta$ could be easily incorporated into the PACE algorithm by forming a set of size $s$, $\Theta = \{\theta_1, \ldots, \theta_s\}$ with associated weights $W = \{w_1, \ldots, w_s\}$. The IMSE criterion (equation 2.34) could be averaged over $\Theta$. A general approach for continuous $\Theta$ does not appear feasible. Also, computer experimental designs have been investigated for their ability for good prediction. Creating designs that are also good for estimation have not investigated.

The results on the estimated root mean square error (RMSE$_{ml}$) from Chapter 5 are discouraging in that RMSE$_{ml}$ is not a reliable estimate of the true error. An expansion of this study is needed to find general statements connecting the magnitude of the bias to the correlation function.

In the manufacturing sciences, deterministic simulators help describe the relationships between product design, and the manufacturing process to the product's
final characteristics. This allows the product to be designed and manufactured efficiently. Equally important are the effects of uncontrollable variation in the manufacturing parameters to the end product. If the product's characteristics are sensitive to slight variations in the manufacturing process, the yield, or percentage of marketable units produced, may decrease. Further, understanding the sensitivities of the product's characteristics can help design more reliable products and increase the overall quality of the product.

Many simulators need to solve differential equations and can provide the gradient of the response at a design point with little or no additional computational cost. However, some simulators require that the gradient be approximated by a difference equation. Then the cost of finding a directional derivative at a point is equal to evaluating an additional point while approximating the total gradient requires $p$ additional runs.

Consider Figure 6.1 for an example in $p = 1$ showing the effects of including gradient information on prediction. The solid lines, $Y$ in Figure 6.1(a) and $Y'$ in Figure 6.1(b), are the true function and its derivative, respectively, while the long dashed lines are a $n = 3$ Kriging predictor $\hat{Y}_3$ and $\hat{Y}'_3$. As expected $\hat{Y}_3$ goes through the design points, $D = \{.2,.5,.8\}$, but $\hat{Y}'_3$ is a poor predictor of $Y'$. The short dashed lines are the $n = 3$ predictors with derivative information $\hat{Y}'_3$ and $\hat{Y}'_3$. Notice that this predictor now matches $Y'$ and $Y$ at $D$ and the interpolations are over all much better. The addition of gradient information substantially improves the fits of both $Y$ and $Y'$. The dotted lines are the $n = 6$ predictors $\hat{Y}_6$ and $\hat{Y}'_6$ and is a fairer comparison if the derivative costs are equal to the response cost. $\hat{Y}_6$ is a little better on the interior of $S$ but $\hat{Y}'_6$ is worse at $x = 0$ than $\hat{Y}'_3$.

Still for $p = 1$, but more generally, consider approximating a function $g(t) \in C^4[a,b]$ by a cubic spline on $[a,b]$. Let $D$ consist of $n$ equally spaced points $t_i = a + (i-1)(b-a)/(n-1)$. Additionally, let $g'(a)$ and $g'(b)$ be known. Then

$$\max_{[a,b]} | g - c_n | \leq \frac{5}{384} \left( \frac{b-a}{n-1} \right)^4 \max_{[a,b]} | g^{(4)} | \tag{6.1}$$
Figure 6.1: (a) An example of a response (\(Y\)) and three predictors (\(\hat{Y}_3, \hat{Y}_5, \hat{Y}_6\)).

(b) An example of a derivative (\(Y'\)) and three predictors (\(\hat{Y}'_3, \hat{Y}'_5, \hat{Y}'_6\)).
where \( c_n \) is a complete cubic spline interpolator [16]. Alternatively, if \( g'(D) \) is known

\[
\max_{[a,b]} | g - h_n | \leq \frac{1}{384} \left( \frac{b - a}{n - 1} \right)^4 \max_{[a,b]} | g^{(4)} |
\]  

(6.2)

where \( h_n \) is a cubic Hermite interpolator. If \( D \) consists of only \( n/2 \) points with derivatives

\[
\max_{[a,b]} | g - h_{n/2} | \leq \frac{16}{384} \left( \frac{b - a}{n - 1} \right)^4 \max_{[a,b]} | g^{(4)} |
\]

(6.3)

If no derivative information is available, the maximum error of a cubic spline is \( O(n^{-3}) \). Using derivative information from the interior of \( S \) reduces the maximum error bound by a factor of 5. If \( n \) must be reduced due to the added cost of obtaining this information, the error bound increases by a factor of 3.

These results are not easily extended for \( p > 1 \). An interesting study could compare the performance of the Kriging model for \( p > 1 \) in terms of both response and gradient predictions. Several designs could be compared. For example, let \( D_{n,0} \) be a \( n \) points design with only response data, \( D_{n,1} \) the same \( n \) design points each with one directional derivative, and let \( D_{n,p} \) contain the response and the total gradient information for the same \( n \) design points. A fourth design, \( D_{np,0} \) with \( np \) points and no derivative information could also be compared if derivative costs are important. The comparison could be on a series of fixed functions or on generated realizations of random functions.

The Kriging methodology easily extends to model gradients. To see this for \( p = 1 \), let \( E[Y(\cdot)] = \mu \) and \( d = t_2 - t_1 \), then

\[
\text{Cov} [Y(t_1), Y'(t_2)] = E[Y(t_1)Y'(t_2)] - E[Y(t_1)] E[Y'(t_2)].
\]

(6.4)

Now due to the stationarity of \( Y(\cdot) \), \( E[Y'(\cdot)] = 0 \) and

\[
\text{Cov} [Y(t_1), Y'(t_2)] = E[Y(t_1)Y'(t_2)]
\]

\[
= E \left[ Y(t_1) \lim_{\delta \to 0} \frac{Y(t_2 + \delta) - Y(t_2)}{\delta} \right]
\]

\[
= E \left[ \lim_{\delta \to 0} \frac{Y(t_1)Y(x_2 + \delta) - Y(t_1)Y(x_2)}{\delta} \right]
\]

\[
= \sigma^2 \lim_{\delta \to 0} \frac{R(d + \delta) - R(d)}{\delta}
\]

\[
= \sigma^2 R'(d)
\]

(6.5)
for differentiable $R(\cdot)$. Similarly,
\[
\text{Cov} \left[ Y''(t_1), Y(t_2) \right] = -\sigma^2 R'(d)
\]  
(6.6)
and
\[
\text{Cov} \left[ Y''(t_1), Y''(t_2) \right] = -\sigma^2 R''(d)
\]  
(6.7)
For more general $p$ and for higher derivatives [43] let
\[
Y^{(a_1, \ldots, a_p)}(t) = \frac{\partial^a}{\partial t_1^{(a_1)} \ldots \partial t_p^{(a_p)}} Y(t)
\]  
(6.8)
where $a = \sum_{j=1}^{p} a_j$ and $t_j$ is the $j^{ith}$ component of $t$. Then $E[Y^{(a_1, \ldots, a_p)}] = 0$ and
\[
\text{Cov} \left[ Y^{(a_1, \ldots, a_p)}(t_1), Y^{(b_1, \ldots, b_p)}(t_2) \right] = (-1)^a \sigma^2 \prod_{j=1}^{p} R_j^{(a_j+b_j)}(t_{2j} - t_{1j})
\]  
(6.9)
for $R(d) = \prod_{j=1}^{p} R_j(d_j)$.
Further, for directional derivatives, let $Y'_v(t)$ be the directional derivative of $Y(t)$ in the direction $\nu = (\nu_1, \ldots, \nu_p)'$, $\sum_{j=1}^{p} \nu_j^2 = 1$,
\[
Y'_v(t) = \sum_{j=1}^{p} \frac{\partial Y(t)}{\partial t_j} \nu_j = \langle \nabla Y(t), \nu \rangle .
\]  
(6.10)
Then $E[Y'_v(t)] = 0$ and for $\delta = t - \xi$,
\[
\text{Cov} \left[ Y(\delta), Y'_v(t) \right] = E \left[ Y(\delta) Y'_v(t) \right] \\
= \sum_{j=1}^{p} E \left[ Y(\delta) \frac{\partial Y(t)}{\partial t_j} \nu_j \right] \\
= \sum_{j=1}^{p} \text{Cov} \left[ Y(\delta), \frac{\partial Y(t)}{\partial t_j} \right] \nu_j \\
= \sigma^2 \sum_{j=1}^{p} \frac{\partial \hat{R}(d)}{\partial d_j} \nu_j \\
= \sigma^2 \langle \hat{R}(d), \nu \rangle
\]  
(6.11)
where $\hat{R}(d) = [\partial R(d)/\partial d_1, \ldots, \partial R(d)/\partial d_p]'$. Similarly,
\[
\text{Cov} \left[ Y'_v(\delta), Y(t) \right] = -\sigma^2 \langle \hat{R}(d), \nu \rangle
\]  
(6.12)
and

\[ \text{Cov} \left[ Y'_{\nu}(x), Y'_{\nu}(t) \right] = -\sigma^2 \nu' \tilde{R}(d) \nu \]  \hspace{1cm} (6.13)

where

\[ \left( \tilde{R}(d) \right)_{kl} = \frac{\partial^2 R(d)}{\partial d_k \partial d_l} \]  \hspace{1cm} (6.14)

is the matrix of 2\textsuperscript{nd} partial derivatives evaluated at \( d \).

The Kriging methodology is modified to model gradient information by letting

\[ y^*_D = \left[ y(x_1), \ldots, y(x_n), y'_{\nu_1}(x_1), y'_{\nu_2}(x_1), \ldots, y'_{\nu_m}(x_n) \right]' \]  \hspace{1cm} (6.15)

where \( \nu_{ii} \) is the direction of the \( l \)th directional derivative at \( x_i \). Also let

\[ \mu^* = (\mu_1', \Omega')' \]  \hspace{1cm} (6.16)

and \( V^* \) be the combined covariance matrix for the design responses and derivatives with the entries as prescribed above. Then

\[ \hat{Y}(x_0) = \mu + v^*_{x_0} V^{-1} \left( y^*_D - \mu^* \right) \]  \hspace{1cm} (6.17)

and

\[ \hat{Y}'(x_0) = v^*_{x_0, \nu} V^{-1} \left( y^*_D - \mu^* \right) \]  \hspace{1cm} (6.18)

where \( v^*_{x_0} = \text{Cov}[Y(x_0), Y^*_D] \), and \( v^*_{x_0, \nu} = \text{Cov}[Y'(x_0), Y^*_D] \).

The Kriging model for the total gradient has already been programmed into one version of PACE. One problem with using the total gradient information is the rapid increase in the covariance matrix. For each additional design point, \( V^* \) increases by \( p + 1 \) rows and columns. Fortunately, these new rows and columns generally have lower correlations than the corresponding rows and columns for an equal number of response. The inversion of \( V^* \) is more computationally stable than for an equally sized \( V_D \). More research is needed to provide general guidelines for using gradient information efficiently.
Appendix A

The Best Linear Unbiased Predictor

The Kriging approach models the response $\forall x \in S = [0, 1]^p$ as

$$Y(x) = \sum_{j=1}^{k} \beta_j f_j(x) + Z(x)$$

(A.1)

where the $f_j$'s are known fixed functions, the $\beta_j$'s are unknown coefficients, and $Z(x)$ is a stationary Gaussian random function with $E[Z(x)] = 0$ and

$$\text{Cov}[Z(x_i), Z(x_j)] = \sigma^2 R(x_j - x_i).$$

(A.2)

Let $Y_D = \{Y(x_1), \ldots, Y(x_n)\}$ be the responses for the given design $D$, and consider a linear predictor of a new point, $x_0 \in S$,

$$\hat{Y}(x_0) = \lambda'(x_0) Y_D.$$ 

(A.3)

The best linear unbiased predictor (BLUP) [26] of $Y(x_0)$ finds the $\lambda'(x_0)$ that minimizes the mean square error

$$\text{MSE}[\hat{Y}(x_0)] = E[\lambda' Y_D - Y(x_0)]^2$$

(A.4)

subject to the unbiased condition $E[\hat{Y}(x_0)] = E[Y(x_0)]$. 
Letting
\[
\begin{align*}
\lambda &= \lambda(x_0) \\
Z_D' &= (Z(x_1), \ldots, Z(x_n)) \\
f'(x_0) &= (f_1(x_0), \ldots, f_k(x_0)) \\
(F_D)_{ij} &= f_j(x_i) \\
(V_D)_{ij} &= \text{Cov}[Z(x_i), Z(x_j)] \\
v'_{x_0} &= (\text{Cov}[Z(x_0), Z(x_1)], \ldots, \text{Cov}[Z(x_0), Z(x_n)])
\end{align*}
\]
and substituting equation (A.1) for \(Y(\cdot)\),
\[
\text{MSE}[\hat{Y}(x_0)] = E[\lambda'(F_D\beta + Z_D) - f'(x_0)\beta - Z(x_0)]^2
\]
\[
= E\{[\lambda'F_D\beta - f'(x_0)\beta] + [\lambda'Z_D - Z(x_0)]\}^2
\]
\[
= [\lambda'F_D\beta - f'(x_0)\beta]^2 + 2[\lambda'F_D\beta - f'(x_0)\beta]E[\lambda'Z_D - Z(x_0)]
\]
\[
\quad + E[\lambda'Z_D - Z(x_0)]^2
\]
\[
\text{(A.5)}
\]
Since \(E[Z(\cdot)] = 0\), the second component of equation (A.5) vanishes and
\[
\text{MSE}[\hat{Y}(x_0)] = [\lambda'F_D\beta - f'(x_0)\beta]^2 + \lambda'\text{E}[Z_DZ_D']\lambda - 2\lambda'\text{E}[Z_DZ(x_0)]
\]
\[
+ E[Z(x_0)]^2
\]
\[
= [\lambda'F_D\beta - f'(x_0)\beta]^2 + \lambda'V_D\lambda - 2\lambda'v_{x_0} + \sigma^2.
\]
\[
\text{(A.6)}
\]
The unbiased condition implies that the first component of equation (A.6) is zero and that \(\lambda'F_D = f'(x_0)\).

To minimize the MSE subject to the unbiased condition by use of Lagrange multipliers, \(\gamma\), let
\[
G(\lambda) = \lambda'V_D\lambda - 2\lambda'v_{x_0} + \sigma^2 + 2(\lambda'F_D - f'(x_0))\gamma.
\]
\[
\text{(A.7)}
\]
Differentiating with respect to \(\lambda\) and setting equal to zero yields
\[
F_D\gamma + V_D\lambda = v_{x_0}.
\]
\[
\text{(A.8)}
\]
Including the unbiased condition and writing in matrix form,
\[
\begin{bmatrix}
0 & F_D' \\
F_D & V_D
\end{bmatrix}
\begin{bmatrix}
\gamma \\
\lambda
\end{bmatrix}
= \begin{bmatrix}
f(x_0) \\
v_{x_0}
\end{bmatrix}
\]
\[
\text{(A.9)}
\]
or

\[
\begin{pmatrix}
\gamma \\
\lambda
\end{pmatrix} = \begin{bmatrix}
0 & F'_D \\
F_D & V_D
\end{bmatrix}^{-1} \begin{pmatrix}
f(x_0) \\
v'_{x_0}
\end{pmatrix}.
\] (A.10)

Hence the BLUP for \(Y(x_0)\) given \(Y_D = y_D\) is

\[
\lambda'_{yD} = \begin{pmatrix}
\gamma' \\
\lambda'
\end{pmatrix} \begin{pmatrix}
0 \\
y_D
\end{pmatrix}
\]

\[
= \begin{pmatrix}
f'(x_0) \\
v'_{x_0}
\end{pmatrix} \begin{bmatrix}
0 & F'_D \\
F_D & V_D
\end{bmatrix}^{-1} \begin{pmatrix}
0 \\
y_D
\end{pmatrix}
\]

\[
= \begin{pmatrix}
f'(x_0) \\
v'_{x_0}
\end{pmatrix} \begin{bmatrix}
-A & AF'_D V_D^{-1} \\
V_D^{-1} F_D A & V_D^{-1} - V_D^{-1} F_D A F'_D V_D
\end{bmatrix} \begin{pmatrix}
0 \\
y_D
\end{pmatrix}. \quad \text{(A.11)}
\]

from inverting by partitions and letting \(A = [F'_D V_D^{-1} F_D]^{-1}\).

Hence,

\[
\hat{Y}(x_0) = \lambda'_{yD}
\]

\[
= \begin{pmatrix}
f'(x_0) \\
v'_{x_0}
\end{pmatrix} \left( \begin{array}{c}
AF'_D V_D^{-1} y_D \\
V_D^{-1} y_D - V_D^{-1} F_D A F'_D V_D y_D
\end{array} \right)
\]

\[
= \begin{pmatrix}
f'(x_0) \\
v'_{x_0}
\end{pmatrix} \left( \begin{array}{c}
\hat{\beta} \\
V_D^{-1} y_D - V_D^{-1} F_D \hat{\beta}
\end{array} \right)
\]

\[
= f'(x_0) \hat{\beta} + v'_{x_0} V_D^{-1} \left( y_D - F_D \hat{\beta} \right). \quad \text{(A.12)}
\]

where

\[
\hat{\beta} = AF'_D V_D^{-1} y_D
\]

\[
= [F'_D V_D^{-1} F_D]^{-1} F'_D V_D^{-1} y_D \quad \text{(A.13)}
\]

is the generalized least squares solution to \(\beta\). Hence,

\[
\lambda'(x_0) = f'(x_0) B + v'_{x_0} V_D^{-1} - v'_{x_0} V_D^{-1} F_D B
\]

\[
\text{where } B = AF'_D V_D^{-1}. \quad \text{Note that } B V_D B' = A. \quad \text{(A.14)}
\]
The mean square error of $\hat{Y}(x_0)$ is

\[
\text{MSE}[\hat{Y}(x_0)] = \lambda' V_D \lambda - 2 \lambda' v_{x_0} + \sigma^2
\]

\[
= \left[ f'(x_0)B + v_{x_0}' V_D^{-1} - v_{x_0}' F_D' F_D B \right] V_D
\times \left[ B' f'(x_0) + V_D^{-1} v_{x_0} - B' F_D' V_D^{-1} v_{x_0} \right]
- 2 \left[ f'(x_0)B + v_{x_0}' V_D^{-1} - v_{x_0}' V_D^{-1} F_D B \right] v_{x_0} + \sigma^2
\]

\[
= f'(x_0)A f(x_0) + v_{x_0}' V_D^{-1} v_{x_0} + v_{x_0}' V_D^{-1} F_D B v_{x_0}
+ 2 f'(x_0)B v_{x_0} - 2 f'(x_0) A F_D' V_D^{-1} v_{x_0} - 2 v_{x_0}' V_D^{-1} F_D B v_{x_0}
- 2 f'(x_0)B v_{x_0} - 2 v_{x_0}' V_D^{-1} v_{x_0} + 2 v_{x_0}' V_D^{-1} F_D B v_{x_0} + \sigma^2
\]

\[
= \sigma^2 + f'(x_0)A f(x_0) - 2 f'(x_0) A F_D' V_D^{-1} v_{x_0}
- v_{x_0}' V_D^{-1} v_{x_0} + v_{x_0}' V_D^{-1} F_D B v_{x_0}
\]

\[
= \sigma^2 - \begin{pmatrix} f'(x_0) & v_{x_0}' \end{pmatrix} \begin{pmatrix} -A & A F_D' V_D^{-1} \\ V_D^{-1} F_D A & V_D^{-1} - V_D^{-1} F_D B \end{pmatrix} \begin{pmatrix} f(x_0) \\ v_{x_0} \end{pmatrix}
\]

and finally

\[
\text{MSE}[\hat{Y}(x_0)] = \sigma^2 - \begin{pmatrix} f'(x_0) & v_{x_0}' \end{pmatrix} \begin{pmatrix} 0 & F_D' \\ F_D & V_D \end{pmatrix}^{-1} \begin{pmatrix} f(x_0) \\ v_{x_0} \end{pmatrix}. \quad (A.15)
\]
Appendix B

PACE User’s Guide

The Program for the Analysis of Computer Experiments (PACE) is an interactive program that aids in building approximating Kriging models to deterministic simulators. PACE can create optimum computer experimental designs, estimate parameters of the model, compute predictions, and summarize the model by providing the main and two-way interactions effects. The purpose of this guide is to inform users of the structure and idiosyncrasies of PACE. The guide has five sections: the overview, a section describing the complete instructions to using PACE, a similar section for PACE-II which uses gradient information, a section that summarizes PACE’s subroutines, and a section on proposed changes for future versions of PACE.

B.1 Overview

PACE performs five major tasks:

1. Computes optimal integrated mean square error (IMSE) and entropy designs.

2. Evaluates specified model parameters by three estimation criteria.

3. Computes the optimum correlation parameters for these three criteria.

4. Predicts the response and RMSE$_m$ for user specified input files or for profiles within $S$. 

107
5. Computes main and two-way interaction effects.

The first task has two options. The first is to compute the IMSE or entropy criteria for the specified design and correlation function (see Sections 2.2.1 and 2.2.2). This can aid in comparing specified designs under these criteria. The second task is to optimize the design criterion selected with respect to \( D \). Both of these criteria are dependent on the correlation function. The entropy design is for the diffuse prior on the mean function [see equation (2.31)].

The second task is to evaluate estimates of \( \theta \) by three criteria: the cross-validated predictive likelihood, the cross-validated squared-bias, and the likelihood. Each of these criteria provide estimates of \( \mu \) and \( \sigma^2 \). This task is helpful in comparing parameter estimates by the three criteria.

The third task is to find the parameter estimates that optimize a selected criterion. The cross-validated predictive likelihood and the likelihood criteria are maximized while the cross-validated squared bias criterion is minimized. These optimizations are calculated by NPSOL. The derivatives of these criteria are programmed into PACE to reduce the computations. The restrictions on \( \theta \) for the cubic and smoothed exponential correlation families so that

\[
\text{Var}(\lambda'Z_D) > 0
\]  

are easily handled by NPSOL.

The fourth task in PACE is the prediction of the response and error (RMSE). These are computed two ways. The first is on user specified input files while the second is on 50-point profiles through \( S \). These predictions are written onto user specified output files.

The fifth task is to summarize the model by computing the first order main effects. Further, user specified second order interactions are also computed on a grid. These can be written out for use in a graphics package for displaying the average effects of each input variable alone or in pairs. Additionally, a coefficient of variation which measures the amount of additivity in the model is provided.

The current version of PACE assumes that the mean function is a constant. Some of the code can easily be extended to handle more general mean functions.
Further, PACE assumes that the design is scaled to -1 and 1 for each input variable. The next section describes in detail the running of PACE.

B.2 Running PACE

B.2.1 Initialization

PACE begins by prompting for the current problem’s size parameters. These are the number of design points \( n \), the number of input variables \( p \), and the number of desired covariate regressors. The last parameter must be 1 in the current version. Next the names of the input files containing the scaled design and the responses are requested. These file names must be \( \leq 15 \) characters. Further, PACE reads these files with a free format and the input data must be separated by commas, blanks and/or carriage returns. Also, the input files cannot contain any extra columns of data.

After the initialization phase, PACE provides the prompt

Enter command (0 for menu):

which is waiting for input from the following menu:

-1 - QUIT

0 - MENU

1 - COMPUTE OPTIMAL DESIGN CRITERIA

2 - EVALUATE THE CORRELATION FUNCTION

3 - COMPUTE OPTIMUM CORRELATION PARAMETERS

4 - PREDICTION

5 - COMPUTE MAIN EFFECTS
B.2.2 Compute Optimal Designs

Selecting 1 from the menu starts the optimal design driver. The optimality criteria computed in PACE require the correlation function. This routine prompts for the correlation family as follows:

1 - Exponential
2 - Smoothed Exponential
3 - Cubic
4 - Gaussian
5 - Linear
6 - Bessel

Next the desired design optimality criterion is requested. These are (1) for the the IMSE and (2) for the entropy criteria. At this point PACE asks if the correlation parameters will be entered from a file or manually. If the parameters are in a file, then PACE will prompt for the file name. Otherwise, PACE prompts for correlation parameters sequentially.

PACE then questions whether just the selected criterion should be evaluated for the given design or whether it should attempt to find the design that optimizes the specified criterion. If only the criterion is calculated, PACE immediately returns to the main prompt. Otherwise, NPSOL is invoked to solve the optimization problem. NPSOL requires an input file containing optimization parameters such as the maximum number of major iterations, the optimality tolerance, and the major print level (see [25] for details). The following is an example of the file:

```
BEGIN Options for NPSOL 4.02.
Major Iterations Limit 90
Major Print Level 10
Optimality Tolerance 1.0E-7
Difference Interval .001
```
Central Difference Interval .001
Function Precision 8.0E-10
Hessian No * Ready for the next run.
END

The '*' comments out the rest of the current line.
The output that follows is from NPSOL and contains three sections:

1 the settings for NPSOL's internal parameters

2 the intermediate results from each major iteration

3 the final solution.

For further information on NPSOL's output see [25]. And lastly, the name of an output file is requested for saving the optimum design. PACE also prints the average squared movement of the design. PACE uses as the initial design, the current design. The current design is changed to the optimal design found.

B.2.3 Evaluate a Correlation Function

Selection 2 on the main menu is used to evaluate the correlation parameters by three criteria: the cross-validated predictive likelihood, the cross-validated squared bias, and the likelihood (see Section 2.4). The correlation family and parameter are entered as in the above option. The output is a table containing the criteria value, $\hat{\mu}$, and $\hat{\sigma}$ for each of the three criteria. This table can then be written to a file if so desired.

B.2.4 Compute Optimum Correlation Parameters

Selection 3 from the main menu is for finding the optimum correlation parameters for one of the three optimality criteria through NPSOL. As with the other options, the correlation family must be specified. Next, PACE prompts for the desired correlation criterion as follows:
APPENDIX B. PACE USER'S GUIDE

1 = Predictive Deficiency

2 = Squared Bias

3 = Maximum Likelihood

NPSOL requires initial estimates of the correlation parameters. These are entered in the usual way either by an input file or manually. An initial estimate of $\mu$ is also needed. The solution is fairly robust to this initial value but an intelligent choice reduces the computations. Additionally, NPSOL requires an input file containing optimization parameters such as the maximum number of major iterations, the optimality tolerance, and the major print level (see [25] and above for details).

The output that follows is from NPSOL and contains three sections:

1 the settings for NPSOL's internal parameters

2 the intermediate results from each major iteration

3 the final solution.

For further information on NPSOL's output see [25]. And lastly, PACE prints out the same table as described above with the three criteria values for the solution found. The optimum parameters can then be saved to a specified file. This file will contain INFORM, $\hat{\theta}$, optimum criterion value, $\hat{\mu}_{ml}$, and $\hat{\sigma}_{ml}$.

B.2.5 Prediction

Selecting 4 from the main menu starts the prediction driver. The correlation family and parameters are entered in the usual way. After their entry, PACE questions whether the predictions should be on an user supplied input file. The user is prompted for an output file for the predictions. This file contains the response and RMSE predictions in two columns. Alternatively, the predictions can be performed on profiles. These consist of 50 equally spaced points on a line segment between two user specified points in $S$. This method also prompts for an output file which contains the predictions as well as the input variable points. The prediction driver
loops on the profile routine so that many profile predictions can be computed on one routine call.

B.2.6 Main Effects

The last selection, 5, on the main menu is for computing the main effects. This begins as the others requesting the correlation family and coefficients. After inverting the covariance matrix, this option requests the name of an output file where it puts places $p$ blocks of 51 predictions of the main effects, one block for each input variable. The main effect for input variable $j$ is the average effect if the other input variable effects are integrated out. The routine also provides the percentage of the variation of the design points that are explained by the one-way main effects.

The main effects driver also prompts for pairs of input variable subscript for which two-way interaction effects are evaluated on an 30 by 30 equally spaced grid. These are also written to a unique user supplied output file. The data is written to the file with the first subscript as the inner counter (by column). The two-way interaction effects have the one-way effects included.

One trick implemented into PACE is to use a negative integer for the correlation family to indicate that the same correlation function is desired for the current task as was used previously. This can save considerable time if $n$ is large.

B.3 PACE-II

PACE-II is organized as PACE but uses total gradient information as well. This section describes the differences between PACE-II and PACE.

B.3.1 Initialization

PACE-II needs to read in the gradient information as well as the response. Currently, PACE-II only deals with complete gradient information and assumes the gradient information is from differences. As with PACE, PACE-II begins by prompting
for \( n, p \), and the number of mean functions (currently can only be 1). Then PACE-II requires the name of the design input file and the response input file. The design input file is as in PACE-II but also contains the \( p \) shifts of the design for the gradient information. The last line of the design input file should contain the assumed standard deviations of the input variables. These are used in computing the transmitted variability. The response input file contains \( p + 1 \) blocks of information where the first block is the response information while each subsequent block is the responses for the shifted design. This data is internally converted into directional derivatives.

B.3.2 Prediction

PACE-II computes the optimal \( \theta \) and also evaluates the correlation function for the three optimality criteria. However, these routines do not incorporate any gradient information and are identical to PACE's routines. Further, PACE-II does not contain any design optimization nor mean effects features. The prediction changes are described below.

The prediction routine works identically as PACE but the output files are slightly different. The first columns contains the response predictions while the second columns contains the transmitted standard deviation predictions. The third column contains the estimated RMSE predictions for the response while the fourth columns is the RMSE for the transmitted variability. Currently these last two columns do not appear to be working properly. If the predictions are on profiles, the output file contains additional columns with the input variable values for the predictions.

B.4 Fortran Subroutines

PACE is controlled by a main driver which reads and processes the inputed data and controls the main menu. The following list describes, in alphabetical, order PACE's subroutines and functions. Additionally, each subroutine called by and
APPENDIX B. PACE USER'S GUIDE

called from these subroutines are listed.

BESSK Computes the modified Bessel function for different orders. Called by COR-
FUC, DCORF1, and DTCOR. BESSK needs functions BESSK0, BESSK1,
BESSI0, BESSI1, and GAMMA either directly or indirectly. All these sub-
routines are contained in the same file as BESSK and are from numerical
recipes [52]. GAMMA is also called by CORFUC.

BLP Computes the best linear unbiased predictor (BLUP). This routine can handle
general linear functions. Called by PREDCT. Calls FUNCTX, DISTX, and
COVX.

CONFN This routine is used by NPSOL to calculate the nonlinear constraints on \( \theta \)
and their first derivatives. This is used in the parameter estimation routines
for the cubic and smoothed exponential correlation functions. This routine is
external to OPTIM and is called only by NPSOL. It does not call any routines.

COR2IN This function is the driver for calculating integrals of products of the correla-
tion functions for the IMSE design calculations. Romberg numerical intergration
(QROMB/TRAPZ/POLINT [52]) is used for the smoothed exponential, cubic, Gaussian and Bessel correlation functions. These routines are in the
same file. Called by IMSE and calls CORFUC, QROMB2, TRAPZ2, and
POLINT.

COR3IN This function is to the driver for calculating integrals of the product of
a correlation function and the derivative of a correlation function. This
is calculated using Romberg numerical integration routines QROMB3 and
TRAPZ3 which are in the same file. Called by IMSE and calls CORFUC, and DCORF1.

CORFUC This function computes the correlations for various correlation families. COR-
FUC is called by COR2IN, COR3IN, CORINT, COVX, IMSE, AND MAN-
EFF and calls BESSK and GAMMA.
APPENDIX B. PACE USER'S GUIDE

CORINT This function is to the driver for calculating integrals of the correlation functions. For the Gaussian and Bessel functions Romberg numerical integration via QROMB1 and TRAPZ1 is used. These routines are in the same file along with POLINT. Called by IMSE AND MANEFF and calls CORFUC.

COVX This routine computes the correlation matrix \( V_D \) for the various correlation families. Called by BLP, ENTROP, EVAL, IMSE, MANEFF, OBJFN1, OBJFN2, OBJFN3, and PREDCT. Calls CORFUC, THTCHK.

CRANK This subroutine computes the variables needed for prediction of both the response and the error (RMSE). Specifically, these variables are \( \hat{\beta} \), \( \hat{\sigma}^2 \),

\[
g = V_D \left( y_D - F_D \hat{\beta} \right)
\]

and

\[
W = \begin{pmatrix} 0 & F'_D \\ F_D & V_D \end{pmatrix}^{-1}
\]

CRANK can handle general linear functions. CRANK is called from MANEFF and PREDCT and calls INVERT.

DCORF1 This function computes correlations between \( Y(t) \) and \( Y'(s) \) for all of the correlation families. DCORF1 is called by COR3IN, ENTROP and IMSE. It only called BESSK.

DESIM This routine is the driver for computing the IMSE and entropy design criteria. It will either calculate the selected criterion for the design entered or will use NPSOL to calculate the optimal design for a given correlation function. DESIM is called by MAIN and calls DISTX, DOPT, ENTROP, IMSE, OUTPT, and THETA.

DISTX This subroutine returns either a \( n(n+1)/2 \times p \) matrix in symmetric storage mode or a \( n \) length vector containing the absolute distance between design points or between a prediction points and \( D \), respectively. DISTX is called from BLP, DESIM, ENTROP, IMSE and MAIN. It does not call any other routines.
APPENDIX B. PACE USER’S GUIDE

DOPT This routine is used to call NPSOL to find optimum design points for the IMSE and maximum entropy criteria. DOPT is only called by DESIM and calls NPSOL routines. It passes ENTROP, IMSE and CONFN2 as externals to NPSOL. CONFN2 is a dummy subroutine contained in the same file.

DTCOR This function computes the derivative of correlation function with respect to \( \theta_j \). This function helps NPSOL compute the derivatives of the estimation criteria. DTCOR is called by OBJFN1, OBJFN2, OBJFN3 and calls only BESSK.

ENTROP This subroutine computes the entropy criterion, and its derivatives with respect to \( D \) if desired, for a given design. ENTROP is called by DESIM and NPSOL, and is passed by DOPT to NPSOL. It calls COVX, DISTX, and INVERT.

EVAL This routine is the driver for estimating the models parameters via the three estimation criteria discussed in Section 2.4. EVAL is called by MAIN and OPTIM. It calls COVX, THETA and EVALCOR while computes the criteria.

EVALCOR This subroutine returns \( \hat{\mu}, \hat{\sigma} \) and performance statistics for the three predictive criteria discussed in Section 2.4. EVALCOR is called by EVAL and calls INVERT.

FILNAM This subroutine opens files for reading. FILNAM is called by DOPT, MAIN, OPTIM, PREDCT and THETA and does not call any other routines.

FUNCTX This subroutines is set up to compute the \( f_i(\cdot) \) for the linear component of the model. Currently, it only returns 1’s as PACE only handles the constant term. FUNCTX is called by BLP and MAIN while not calling any other routines.

IMSE This function returns the IMSE for the given \( D \). IMSE is called by DESIM, NPSOL, and is passed by DOPT to NPSOL. It calls DISTX, CORFUC, CORINT, COVX, and INVERT.
INVERT  This routine uses the singular-value decomposition to compute the inverse of matrices. It also returns the eigenvalues. INVERT is called by CRANK, ENTROP, EVALCOR, IMSE, OBJFN1, OBJFN2, and OBJFN3. It calls SVDCMP which performs the singular-value decomposition. If the singular-value decomposition yields eigenvalues that are too small, then INVERT will return a generalized inverse. An error message is returned notifying the user.

MANEFF  This routine is the driver for computing all of the one-way main effects and also selected two-way interaction effects as well as giving measures of the model's additivity. This routine needs to be modified if more complicated fixed linear functions are implemented. MANEFF is called by MAIN and calls COVX, CRANK, OUTPT, and THETA.

OBJFN1  This routine computes the cross-validated predictive likelihood criteria and its derivatives. OBJFN1 is called from NPSOL and passed to NPSOL by OPTIM. It calls COVX, DTCOR, and INVERT.

OBJFN2  This routine computes the cross-validated squared-bias criteria and its derivatives. OBJFN2 is called from NPSOL and passed to NPSOL by OPTIM. It calls COVX, DTCOR, and INVERT.

OBJFN3  This routine computes the likelihood criteria and its derivatives. OBJFN3 is called from NPSOL and passed to NPSOL by OPTIM. It calls COVX, DTCOR, and INVERT.

OPTIM  This routine is the driver for estimating the parameters of the Kriging model through any of the 3 criteria discussed in Section 2.4. This routine asks for initial values of the parameters and then uses NPSOL to search for the optimum values. OPTIM is only called by MAIN and calls EVAL, FILNAM, THETA, and NPSOL routines NPSOL and NFILE. Further it passes OBJFN1, OBJFN2, OBJFN3, and CONFN to NPSOL.

OUTPT  This subroutine opens files for writing. OUTPT is called by DESIM, EVAL, MANEFF, and PREDCT while it does not call any other routines.
APPENDIX B. PACE USER'S GUIDE

PREDCT This is the prediction driver. It prompts for θ, a file containing any new input data to be predicted. This new data is then written to a file. This routine also allows profiles to be made. PREDCT is called by MAIN only and calls BLP, COVX, CRANK, FILNAM, OUTPT, THETA, AND XVEC.

SVDCMP This is singular-value decomposition routines fro numerical recipes [52]. It is called by INVERT and calls no other routines.

THETA This subroutine prompts the user and reads θ. THETA is called by the sub-drivers: DESIM, EVAL, MANEFF, OPTIM, and PREDCT. It only calls FILNAM.

THTCHK This subroutine checks that the θ is valid for the specified correlation family. THTCHK is called by COVX and does not call any other routines.

XVEC This routine prompts for two coordinate values and then computes 50 equally spaced points along the vector between the two values. These points are then predicted by PREDCT. XVEC is called by PREDCT and does not call any other routines.

The internal size of PACE is controlled by the include file ‘parameter.inc’. This defines the maximum number of design points, input variables and mean functions (currently equal to 1). If larger experiments need to be analyzed then PACE has to be re-compiled with ‘parameter.inc’ modified appropriately. As a note, PACE assumes that the design space is \( S = [-1, 1]^p \) although the model parameters are for \( S = [0, 1]^p \). This should not cause too much confusion in the operation of PACE but can cause difficulties in programming. Future versions will have this problem corrected.

B.5 Future Improvements

Future versions of PACE will be more flexible and contain more options. A routine will be included to automatically scale and rescale the input variables so that no
manual data manipulation will be necessary. This will allow users to input their data and receive output in its natural scale. However, the internal calculations would still be for the unit cubic input space. Improvements in the design area will be to add optional generation of the Faure designs. Additionally, random initial designs will be added so that the optimizations for the IMSE and entropy designs can be found without initial designs. Also, this addition will allow users to optimize for many starting points and have a better chance of finding the true optimum designs.

PACE will also be generalized to include more flexible mean functions such as linear or quadratic terms. Further, more general functions could be defined by programming a subroutine such as FUNCTX. This will allow users to customize the Kriging model to their particular problem. An additional generalization is to combine the attributes of PACE and PACE-II into one program that can handle responses, total gradients, and/or directional derivatives. This will allow for the investigation of models with partial gradient information when available. These improvements will make PACE more general and easier to use for both the scientist in analyzing computer experiments and the research in investigating the behavior of the Kriging methodology.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


