THE UNIVERSITY OF CALGARY

Pseudo-Bayesian Response Surface Analysis

by

Gerard Joseph LaBute

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF MATHEMATICS AND STATISTICS

CALGARY, ALBERTA

AUGUST, 1998

©Gerard Joseph LaBute 1998

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-34971-3

Canada

Abstract

One of the purposes of response surface analysis is to construct a model of the relationship between the input variables and the response variables based on the results of one or more experiments in order to determine the contribution of the input variables to the responses as well as determine their optimal settings. The purpose of this thesis is to use Bayesian principles in order to better achieve the aforementioned goals using the same experimental results.

Acknowledgements

I would like to thank the following people for their help during the development of this thesis:

Dr. Rita Aggarwala

Dr. Michael Lamoureux

Dr. W.K. Nicholson

Dr. Vienna Statsna

my thesis committee members, Dr. Mike Oballa and Dr. David Scollnik

A special thanks to my supervisor, Dr. Peter Ehlers

A special, special thanks to my family, my wife, Susan, our children, Sara and Michael, with their patience and generosity while I was living with our computer.

Contents

Approval Page ii
Abstract iii
Acknowledgements iv
Contents v
List of Figures vii

1 Preliminaries **1**
1.1 Bayesian Perspective 3
1.2 Previous Methodologies 3
1.3 Motivation for New Methodology 6

2 Determination of Variable Importance **8**
2.1 Link Between Bayesian and Least Squares Estimates of B 9
2.2 Elicitation of Prior Anticipation 11
2.2.1 Generation of $\ddot{Y}$ 12
2.2.2 Introduction to Pseudo-Weights 13
2.2.3 $\ddot{Y}$ Sorting Methodology 14
2.3 Search For The Optimal PW Vector 17
2.3.1 Exceptions 37
2.4 Conclusion 38

3 Optimal Operating Conditions **39**
3.1 First Order Model 39
3.1.1 Theoretical Underpinnings 40
3.1.2 Expectations of Bayesian Posterior Vectors 40
3.1.3 Recursive Convergence 41
3.1.4 Construction of Confidence Intervals 44
3.1.5 Methodology 45
3.2 Higher Order Model 58
3.2.1 Methodology 59
3.3 Conclusion 66

4 Extraordinary Circumstances **67**
4.1 Multiple Responses 67
4.1.1 Development of Collapsing Function 67
4.2 Linear Constraint In First Order Model 74
4.3 Variable Importance in Linear Regression 76

5 Conclusion and Further Work **79**

Bibliography 81

Appendix 1 83
Appendix 2 84
Appendix 3 85
Appendix 4 86
Appendix 5 87

List of Figures

2.1 Example 2.3: Interaction of x_1 and x_2 28
2.2 Example 2.7: Interaction of x_1 and x_4 33
2.3 Example 2.7: Interaction of x_2 and x_4 34
2.4 Example 2.7: Interaction of x_2 and x_3 35

3.1 Representation of Confidence Intervals Before and After Convergence 47
3.2 Example 3.1: Angles of Generated Betas With Results 52
3.3 Rotated Beta Vectors In A Higher Dimension 55

4.1 Example 4.2: Interaction of x_1 and x_2 73

Chapter 1

Preliminaries

In the development of new products in the manufacturing sector, research may need to be conducted to determine how to best build the product in order to meet the demanding quality standards of the marketplace. The product may be a new microchip, or a new type of plastic. Whatever the product, at the research and development stage, experiments may be conducted to ascertain the manufacturing conditions necessary to produce a quality product. This process may involve the construction of mathematical equations, or models, to show the relationship between the manufacturing conditions, which are the input variables into the model, and a product attribute, the response variable. If this model is built solely on the evidence provided by an experiment, it is known as an empirical model.

Response surface methodology (RSM) is a statistical technique combining linear regression with experimental design which is used when the relationship between the input variables and the response variable is unclear due to a lack of previous knowledge. The general procedure is to choose an experimental design matrix and conduct the experiment. The experimenter then constructs what is called a first order model. In this model, all of the input variables are raised to only the first

power and the model is strictly additive. If there are k input variables, the model is

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_k x_k$$

If n experiments were conducted, and there are k input variables, the n equations can be written collectively as a matrix equation of the form $Y = XB$. The design matrix X will have dimensions n by k and the response vector, Y, will be of length n. The vector of coefficients, B, of length k, is usually solved for by the method of least squares, in which $B = (X'X)^{-1}(X'Y)$. In general, this will be known as the beta vector, regardless of the order of the model. However, in the first order model presented above, it is known as the path of steepest ascent, if one is searching for a maximum response, or the path of steepest descent, if a minimum response is desired. It has the property of generally pointing the experimenter in the direction of the optimal response, optimal meaning either maximum or minimum.

If the design matrix contains two or more rows that are identical, these are known as replicates and a test for lack of fit can be conducted on the model. If the model is not adequate, this procedure is repeated using a second order design matrix and the construction of a second order model which now includes quadratic terms and interaction terms of the input variables. Though it is possible that a third order model would have to be constructed, it is rare for models to go beyond second order. The objective is to have as simple a model as possible.

In addition to the above goals, the first order model may be used as a screening stage to find the set of variables which contribute most to the variability of the response. If this is the case, the second order model is used to find the settings of the input variables which produce the optimal response.

1.1 Bayesian Perspective

In classical statistics, the beta vector, B, is viewed as a fixed quantity, determined solely by the design matrix and the experimental responses. From a Bayesian viewpoint, however, it is regarded as a random variable, which has a prior distribution which is combined with the experimental results to produce a posterior distribution.

Under the non-Bayesian regime the sampling distribution of B is assumed normal with mean $(X'X)^{-1}(X'Y)$ and variance $(X'X)^{-1}\sigma^2$ where σ^2 is the common variance of the residuals, ε_i.

In a Bayesian context, both the prior and posterior distributions of B are assumed normal with prior mean θ and variance $R^{-1}\sigma^2$ and posterior mean $(X'X+R)^{-1}(X'Y+R\theta)$ and variance $(X'X+R)^{-1}\sigma^2$ in which R is the prior precision matrix of B. If the experimenter is building a first order model to find the path of steepest ascent, the overall purpose of the Bayesian approach is to provide greater precision in the search. If it is a second order model which is being built, the Bayesian objective is to find the settings of the input variables which provide responses greater than those using classical statistical methods. Details will be provided in subsequent chapters as to how these goals will be achieved using techniques developed by this author.

1.2 Previous Methodologies

Bayesian methods have received extensive treatment in the literature, particularly in the area of the design of experiments. The methods center around the use of utility functions and Bayesian alphabetic optimality. The reader is referred to Bernardo [1], Chaloner [4], Draper and Hunter [7], El-Krunz and Studden [9], Lindley [13],

Pukelsheim [17], Verdinelli [18], and Verdinelli and Kadane [19]. It has also been used in the analysis of experiments involving several responses by Box and Draper [2] and experiments involving blocking by Guttman and Menzefricke [11]. To a lesser extent, Bayesian methods have been used to derive the prior hyperparameters θ and R. To illustrate the difficulties involved in deriving these hyperparameters for use in RSM, we present four different approaches found in the literature.

The first is from Kadane et al [12]. In their procedure the expert is asked to provide, for each level of the design matrix, the 1/2, 3/4, 7/8 and 15/16 percentiles of the response from a list of computer generated candidates. All four percentiles had to be provided in order for the corresponding design matrix level to be used in subsequent development of the R matrix. The algorithm is a multi-step procedure in which the R matrix is constructed one dimension at a time, i.e. 1x1, 2x2, 3x3, etc. In the example provided in which road pavement was being developed, the experts were able to choose the 7/8 and 15/16 percentiles for only four of the nine levels of the design matrix. This author attempted to construct the R matrix via this method, using S-Plus under both UNIX and Windows 95 platforms. The UNIX operating system was the faster of the two. Nevertheless, under UNIX, the method took roughly five hours to build a 4x4 matrix and over 24 hours for a 36x36 version, regardless of the type of design matrix being used. This was due to the number of steps required to build the R matrix. The resulting $(X'X + R)$ matrix was unsatisfactory as well; the magnitude of the numbers in the R matrix was so great as to overwhelm any information contained in $X'X$. It was the disappointing experience with this particular algorithm coupled with time constraints which spurred this author to develop an algorithm to produce the R matrix.

The second example is Garthwaite and Dickey [10]. Dickey was one of the co-authors of the Kadane et al paper above, and so this methodology has a passing

resemblance to the construction of the R matrix in that the procedure is sequential. Their elicitation process involves giving the expert the settings for a subset of input variables. The expert is then asked for the levels of the remaining input variables so as to minimize his/her uncertainty in predicting the response for this particular combination of the input variables. The drawback to this method is that the design matrix cannot be one in which the levels of one variable depend on those of the others, such as a fractional factorial design. Once the levels for one row of the design matrix are chosen, the expert is then asked for the 1/4, 1/2 and 3/4 percentiles for the corresponding value of y. Since RSM does use fractional factorial designs, as well as other imbalanced designs in which there are linear dependencies among the input variables, this method is not useful.

The third example is Oman [15]. His method has a prior assumption that certain relationships exist among the input variables, such as in a mixture design matrix in which the proportions of the input variables sum to one, or a fractional factorial design in which one variable is the product of two or more other variables. The restrictive nature of the methodology in this paper is not suitable for the wide variety of response surface designs that exist.

The last example is Toman [16]. This is the only article that mentions RSM specifically. His method is aimed strictly at two- and three-level factorial designs, which excludes other first order designs such as the simplex. He mentions using the method of Kadane et al to derive the R matrix. He assumes that for a given level of the design matrix, the experimenter can provide an estimate of y from which the prior beta vector, θ, can be derived. As with the previous example, his methodology is not suitable for design matrices with linearly dependent input variables.

1.3 Motivation for New Methodology

In developing new techniques to be used in RSM, this author was motivated by the following questions: 1) what is the order of importance of the input variables? 2) how are they correlated with the response variable? 3) what are the optimal conditions of the input variables in order to determine the response desired after variable importance has been determined? The method should be useable regardless of the type of design matrix. It should not assume that there is any prior knowledge available, which may be the case in the development of a new product which has no manufacturing history to call upon. An assumption tied to this is the prior belief that the experimental results are equally likely for all levels of the design matrix.

Before proceeding, a hurdle must be cleared. Chaloner and Verdinelli [6] raise the point that the R matrix is known prior to the experiment. Clearly, in an initial response surface experiment involving a new product, this is not possible. However, the methods to be presented allow the following argument to be posed.

The R matrix is based on observations in the real world. The experiment serves as a model of this real world. If it did not, one would be able to question, at the very least, the methodology used in the experiment, and possibly the entire rationale of the experiment. It is with this understanding that the R matrix can be constructed using the observed experimental data; it is assumed that this data is an accurate reflection of data that would be observed under the same conditions in the real world.

The methods to be introduced are an empirical Bayes approach of sorts since we will construct the R matrix based on the evidence provided by the experiment rather than by prior opinion. This author introduces the term *prior anticipation* to replace the term "prior opinion". The methodology is Bayesian in the sense that the anticipation of how the experiment should have resulted, based on questions 1) and

2) above, determines the nature of the R matrix. It is not Bayesian in that the process is completely deterministic, since the number of answers to the above questions for a given experiment are finite, since the number of input variables is finite.

The general philosophy behind the methods to be introduced is that the experimenter should be allowed as much leeway as is needed in order to determine the optimal operating conditions and the responses under those conditions. This includes the choice of design matrix. The ultimate goal is that, through the introduction of methods based on Bayesian principles, a measure of bias can be introduced that will improve upon the results produced by the standard procedures of least squares and ridge analysis regardless of the type of design matrix chosen by the experimenter. Though the experimenter's goal may be to maximize or minimize the response, or achieve a target value, throughout it is assumed that the goal is to maximize the response.

Chapter 2 addresses the first two questions posed above, chapter 3 the third. Chapter 4 deals with the issues of multiple responses and constrained operating conditions in the first order model.

Chapter 2

Determination of Variable Importance

This chapter begins the development of new tools for analysing the experiment after its completion. If the experiment was performed using a balanced design, analysis can be done using the procedures found in the literature. The two standard references are Box and Draper [3] and Myers and Montgomery [14]. This thesis will introduce methods that can be used on unbalanced designs while producing results consistent with balanced designs. This is to be achieved under a Bayesian framework, in which prior anticipations on the outcome of the experiment are combined with the actual experimental data in order to produce posterior results on which further analysis can be conducted.

In order to do this analysis, a link to the standard methodology is needed. This link is the path of steepest ascent, which was introduced in Chapter 1. This vector is important for two reasons: 1) the magnitude of the coefficients generally provides a measure of the importance of the input variables, and 2) the sign of the coefficient generally indicates the correlation with the response variable, both of which we are concerned with in this chapter.

What we want to show is that a prior anticipation which accurately reflects the actual experimental results will produce a posterior beta vector closer to the path of

steepest ascent than one which does not. We will show mathematically how this closeness is measured by introducing a unique measure of biasedness, and a procedure for finding the least biased beta vector.

2.1 Link Between Bayesian and Least Squares Estimates of B

In least squares linear regression,

$$\hat{\hat{Y}} = X\hat{\hat{B}} = X(X'X)^{-1}(X'Y) \tag{2.1}$$

which is dependent only on the design matrix and the vector of observations. However, in Bayesian linear regression, the posterior

$$\hat{Y} = X\hat{B} = X(X'X + R)^{-1}(X'Y + R\theta) \tag{2.2}$$

in which the values of the $\hat{Y}$ vector depend as well on R and θ, which in turn depend on the prior anticipation. Since the least squares vector is dependent only on the design matrix and the vector of observations, what is needed is a vector to add to the right hand side of equation 2.2 to produce the least squares vector. As will be shown, this vector will provide a measure of the accuracy of a prior anticipation.

If we add $X(X'X)^{-1}R(\hat{B} - \theta)$ [1] to the right hand side of equation 2.2:

$$\begin{aligned}
\hat{Y} &= X\hat{B} + X(X'X)^{-1}R(\hat{B} - \theta) \\
&= X((I + (X'X)^{-1}R)\hat{B} - (X'X)^{-1}R\theta) \\
&= X(((X'X)^{-1}(X'X) + (X'X)^{-1}R)\hat{B} - (X'X)^{-1}R\theta) \\
&= X((X'X)^{-1}(X'X + R)\hat{B} - (X'X)^{-1}R\theta) \\
&= X((X'X)^{-1}(X'X + R)(X'X + R)^{-1}(X'Y + R\theta) - (X'X)^{-1}R\theta) \\
&= X((X'X)^{-1}(X'Y + R\theta) - (X'X)^{-1}R\theta)
\end{aligned}$$

[1] The derivation is provided in Appendix 1

$$= X(X'X)^{-1}(X'Y)$$
$$= X\hat{\hat{B}}$$
$$= \hat{\hat{Y}}$$

Thus, $X(X'X)^{-1}R(\hat{B}-\theta)$, serves as a vector bias, which bridges the gap between the biased Bayesian model response $\hat{Y}$ and the least squares model value, $\hat{\hat{Y}}$. The least squares model serves as an anchor by which one can measure, by means of the vector bias, how far removed the biased model value is, and, by connection, the prior anticipation ultimately responsible for the vector bias.

Under what conditions does the vector bias vanish? Consider the following implications:

Bias = **0**

$X(X'X)^{-1}R(\hat{B}-\theta) = \mathbf{0}$

$\hat{B} = \theta$, since $X(X'X)^{-1}R \neq \mathbf{0}$

$\hat{B} = (X'X+R)^{-1}(X'Y+R\theta)$

$(X'X+R)\hat{B} = X'Y+R\theta$

$X'X\hat{B} = X'Y$

$(X'X)^{-1}X'X\hat{B} = (X'X)^{-1}X'Y$

$\hat{B} = (X'X)^{-1}X'Y$, the path of steepest ascent. The vector bias vanishes only if there is no adjustment between the prior and posterior mean of B. This can only happen if the prior anticipation accurately predicts the result of the experiment. Thus, the goal is to minimize the bias.

Since $X'(X'X)^{-1}R(\hat{B}-\theta)$ is a vector of length n and R is symmetric, $[X'(X'X)^{-1}R(\hat{B}-\theta)]'X(X'X)^{-1}R(\hat{B}-\theta) = (\hat{B}-\theta)'R(X'X)^{-1}R(\hat{B}-\theta))$, is a scalar. We will refer to this as the scalar bias, or simply the bias. As with its vector counterpart, it provides a measure of the distance between the biased Bayesian beta vector and the least squares beta vector.

There are two hyperparameters to solve for: θ and R, both of which are found by quantifying prior anticipations on the outcomes of the experiment. Since the goal is to minimize the bias, we wish to find the prior anticipation which is most closely aligned with the path of steepest ascent.

2.2 Elicitation of Prior Anticipation

There is a wealth of literature on the topic of eliciting subjects' opinions. Each of the fields of mathematics, logic and psychology has extensive libraries. Our goal is to find a psychologically simple method to elicit mathematically logical opinions which are used to construct θ and R. There is a common link among the methods found in the mathematical literature as illustrated by the example in Chapter 1: the subject is required to provide statistics such as percentiles or 95% confidence intervals. There is a fundamental argument that can be used against any method that requires the subject to be mathematically logical: the fact that most people are not. Chaloner [5] admits as much, stating that people make mistakes, such as choosing an upper quantile smaller than a lower quantile, or are inconsistent in choosing a mean and variance for a particular normal distribution, then specifying quantiles which the person thinks belong to the same distribution, but which, in fact, belong to another.

This author's solution is to form a prior anticipation, which is dependent solely on the results of the experiment, based on the following questions: 1) Is the correlation between the input variable and response variable positive or negative? 2) What is the order of importance from most to least of the input variables in their contribution to the variability of the response variable?

Recall that we must solve for the hyperparameters θ, the prior mean beta vector, and R, the prior precision matrix. If we have a matrix of k beta vectors $[B_1 \dots B_k]$,

in which $B_i, i = 1 \ldots k$ is a vector of length p, a mean vector and a covariance matrix can be produced. To produce this matrix by the method of least squares, we need a matrix, $\ddot{Y} = [Y_1 \ldots Y_k]$, and a design matrix X of dimensions n by p. The values in $\ddot{Y}$ are generated according to the two questions posed above. Once the matrix of beta vectors is generated, the mean of the vectors can be computed to produce θ, as well as the variances/covariances to produce R.

2.2.1 Generation of $\ddot{Y}$

Two assumptions will be made. The first assumption is that the y values are normally distributed, each with a particular mean. The second assumption is that the variance is equal for each of the y values. The objective is to produce $\ddot{Y}$ in which each row is randomly drawn from a normal distribution centered about the corresponding y value from the observed Y vector. It is known that 6σ covers almost 100% of the values in a normal distribution. If we have two normal distributions with means μ_1 and μ_2 , we may minimize the overlap between the distributions by each having common standard deviation $|\mu_1 - \mu_2|/6$. With this in mind, $\ddot{Y}$ can be generated as follows:

1) Sort the observed Y vector from smallest to largest. This is to facilitate computer implementation.

2) Find the minimum positive difference between adjacent values. Make the common standard deviation σ equal to 1/6 of this value.

3) For each row of $\ddot{Y}$, generate random normal values with mean equal to the corresponding y value in that row of the observed Y vector and standard deviation σ.

A note about replicates in the design matrix: if the smallest difference found in step 2 is between values corresponding to equal design levels, it may prove useful to

collapse response values by averaging those with equal design levels.

Example 2.1 Given the observed Y vector $\begin{bmatrix} 1.2 & 3.7 & 3.3 & 2.3 \end{bmatrix}'$, the sorted vector is $\begin{bmatrix} 1.2 & 2.3 & 3.3 & 3.7 \end{bmatrix}'$. The vector of differences between adjacent values is $\begin{bmatrix} 1.1 & 1.0 & 0.4 \end{bmatrix}'$. The smallest positive difference is 0.4 Thus we assign a common variance of $0.4/6 = 0.067$. For illustration purposes only, if we generate five values per observation, $\ddot{Y}$ could possibly be

$$\begin{bmatrix} 1.1951 & 1.1590 & 1.2360 & 1.1691 & 1.2635 \\ 2.3177 & 2.2024 & 2.2958 & 2.2202 & 2.3216 \\ 3.4246 & 3.2167 & 3.2534 & 3.3469 & 3.2296 \\ 3.7112 & 3.6204 & 3.6695 & 3.7278 & 3.6451 \end{bmatrix}$$

This generates $\ddot{Y}$ whose values correspond to the observed Y vector. Clearly, each time $\ddot{Y}$ is generated, the values will be different. If it is important to the experimenter to produce bias values which are robust to the variation in $\ddot{Y}$, it may be wise to generate a large number of simulated values for each observed y value. The next step is to sort the rows of $\ddot{Y}$ according to the prior anticipation.

2.2.2 Introduction to Pseudo-Weights

In order to measure the contribution of the input variables to the response, this author developed a tool called pseudo-weights (pw). The algebraic basis of the weights is the sequence of prime number reciprocals $\{1/2, 1/3, 1/5, 1/7, 1/11, \ldots\}$. The reason for choosing this sequence will be explained later.

Each pseudo-weight carries two pieces of information that directly relate to the prior anticipation: 1) in general, given two input variables, x_1 and x_2, if x_1 is anticipated to contribute more than x_2 to the variability of the response, then their corresponding pseudo-weights, pw_1 and pw_2 satisfy the inequality $|pw_1| > |pw_2|$; 2) the sign of a pseudo-weight indicates the anticipated sign of the correlation of the input variable with the response.

We next examine how the pseudo-weights are chosen and used to determine input variable importance and correlation with the response based on the evidence from the experiment. As will be shown, the answers depend on the nature of the design matrix.

2.2.3 $\ddot{Y}$ Sorting Methodology

Consider a 2^2 factorial design with input variables x_1 and x_2 and response vector Y:

$$\begin{bmatrix} x_1 & x_2 & Y \\ 1 & 2 & y_1 \\ 1 & 3 & y_2 \\ 2 & 2 & y_3 \\ 2 & 3 & y_4 \end{bmatrix}$$

Suppose now that both x_1 and x_2 are positively correlated with the response variable Y. We would expect $y_4 \geq y_1$, as well as $y_4 \geq y_3$ and $y_2 \geq y_1$. If x_1 is more important than x_2 in the contribution to the variability of Y, we would expect $y_3 \geq y_2$. Under this scenario, $y_4 \geq y_3 \geq y_2 \geq y_1$.

If this is expected to be the case, we should be able to provide some type of measure by which we can rank the anticipated responses. This measure is the dot product of the rows of either the design matrix or a subset of it with a carefully designed pw vector. The resulting vector of rational numbers, which we refer to as the combined pw, can then be ranked from smallest to largest to indicate the anticipated ranking of the response. Returning to the case at hand, if we assign x_1 a pseudo-weight (pw) of 1/2 and x_2 a pw of 1/3 to indicate that x_1 is considered the more important variable, we can assign each y_i a rank by computing the dot product of each factorial level with the vector $\begin{bmatrix} 1/2 & 1/3 \end{bmatrix}$. The result is:

x_1	x_2	Combined pw	rank
1	2	$7/6 = 1/2 + 2/3$	1
1	3	$3/2 = 1/2 + 3/3$	2
2	2	$5/3 = 2/2 + 2/3$	3
2	3	$2 = 2/2 + 3/3$	4

in which the rankings are from smallest to largest. Similarly, if x_1 and x_2 are both negatively correlated with Y, with x_1 considered more important than x_2, then we could assign the vector $\begin{bmatrix} x_1 & x_2 \end{bmatrix}'$ the pw vector $\begin{bmatrix} -1/2 & -1/3 \end{bmatrix}$ with these results:

x_1	x_2	Combined pw	rank
1	2	-7/6 = −1/2 − 2/3	4
1	3	-3/2 = −1/2 − 3/3	3
2	2	-5/3 = −2/2 − 2/3	2
2	3	−2 = −2/2 − 3/3	1

which is, of course, the opposite of the results under positive correlations.

There are eight possible pw vectors: $\begin{bmatrix} 1/2 & 1/3 \end{bmatrix}$, $\begin{bmatrix} 1/3 & 1/2 \end{bmatrix}$, $\begin{bmatrix} 1/2 & -1/3 \end{bmatrix}$, $\begin{bmatrix} 1/3 & -1/2 \end{bmatrix}$, $\begin{bmatrix} -1/2 & 1/3 \end{bmatrix}$, $\begin{bmatrix} -1/3 & 1/2 \end{bmatrix}$, $\begin{bmatrix} -1/2 & -1/3 \end{bmatrix}$ and $\begin{bmatrix} -1/3 & -1/2 \end{bmatrix}$ with the x_1 pw listed first. In general, for n input variables, there are $n!2^n$ pw vectors. As evidenced by this chart, the number of vectors quickly mushrooms.

n	# pw vectors
2	8
3	48
4	384
5	3840

Clearly, a systematic approach to finding the optimal pw is needed.

One feature of the pseudo-weight rankings is that they can be used to arrange the rows of the design matrix to show the input variable importance and correlation with the response. Using the matrix

x_1	x_2	dot product
1	2	7/6
1	3	3/2
2	2	5/3
2	3	2

if the pw vector is $\begin{bmatrix} 1/2 & 1/3 \end{bmatrix}$ the rankings of the matrix dot product with the pw vector has the matrix already in the form in which the order of the dot product range from smallest to largest. However, if the pw vector is $\begin{bmatrix} -1/2 & 1/3 \end{bmatrix}$, the

order of the matrix changes to:

x_1	x_2	dot product
2	2	-1/3
2	3	0
1	2	1/6
1	3	1/2

so that the dot product results are arranged from smallest to largest.

One problem to avoid is equal rankings for rows of the design matrix which are not replicates. This is the reasoning behind choosing the sequence of prime number reciprocals as the basis. Consider the case of a 3^3 factorial design with levels (1, 3, 5), (10, 20, 30) and (2, 4, 6) for x_1, x_2, x_3 respectively and the pw vector being $\begin{bmatrix} 1/2 & 1/3 & 1/4 \end{bmatrix}$. These two factorial levels have the same dot product:

x_1	x_2	x_3	dot product
1	10	6	$1/2 + 10/3 + 6/4 = 16/3$
3	10	2	$3/2 + 10/3 + 2/4 = 16/3$

There are six such pairings. However, if we use $\begin{bmatrix} 1/2 & 1/3 & 1/5 \end{bmatrix}$:

x_1	x_2	x_3	dot product
1	10	6	$1/2 + 10/3 + 6/5 = 151/30$
3	10	2	$3/2 + 10/3 + 2/4 = 157/30$

and the rankings become distinct.

Once the pseudo-weight rankings have been computed, they can be used to sort the rows of $\ddot{Y}$ accordingly.

Example 2.1 continued Suppose the rankings of the pw is [2 4 3 1]. The $\ddot{Y}$ matrix

$$\begin{bmatrix} 1.1951 & 1.1590 & 1.2360 & 1.1691 & 1.2635 \\ 2.3177 & 2.2024 & 2.2958 & 2.2202 & 2.3216 \\ 3.4246 & 3.2167 & 3.2534 & 3.3469 & 3.2296 \\ 3.7112 & 3.6204 & 3.6695 & 3.7278 & 3.6451 \end{bmatrix}$$

becomes

$$\begin{bmatrix} 3.7112 & 3.6204 & 3.6695 & 3.7278 & 3.6451 \\ 1.1951 & 1.1590 & 1.2360 & 1.1691 & 1.2635 \\ 3.4246 & 3.2167 & 3.2534 & 3.3469 & 3.2296 \\ 2.3177 & 2.2024 & 2.2958 & 2.2202 & 2.3216 \end{bmatrix}$$

From this matrix of five Y vectors, we can produce a matrix of corresponding beta vectors by the method of least squares. This can be used to produce a prior beta mean vector, θ and a prior beta covariance matrix from which we can solve for R, which can then be used with the design matrix and the vector of observed values to solve for the posterior beta mean vector $(X'X + R)^{-1}(X'Y + R\theta)$ and covariance matrix $(X'X + R)^{-1}$. Thus, if the design matrix has n input variables, we have $n!2^n$ pseudo-weight vectors, as well as $n!2^n$ matchings of θ, R and posterior beta mean vector, all of which are components of the bias. Our goal is to find which pw vector which will minimize the bias.

2.3 Search For The Optimal PW Vector

The strategy to be used depends on the nature of the design matrix. Factors to be taken into consideration are: 1) whether or not the design is factorial; 2) whether or not the design is saturated, as is the case with simplex or 2^{7-4} designs; 3) whether or not the design is balanced; 4) whether or not the levels of one variable depend on the levels of at least one other variable, as is the case with fractional factorial designs; 5) the number of levels in each variable. It should be noted that the results of the analysis are sometimes in agreement with those derived by computation of the main effects of the variables, in particular, for balanced factorial designs. Since computing main effects for these designs is simpler than the methods presented here, this author advises using the simpler method. The value of the methodology introduced herein for these designs is mostly theoretical.

To motivate the strategy, we introduce the concept of *variable order independence.* There are three types: Type I, full independence, Type II, partial indepedence, and Type III, full dependence. These types convey the degree of dependence of the order of a variable in the design matrix in relation to its correlation with the response vector on at least one other variable. To illustrate the difference between the three types, we present some examples.

Example 2.2 This is a Type I design.

$$\begin{bmatrix} x_1 & x_2 \\ 1 & 5 \\ 1 & 12 \\ 2 & 5 \\ 2 & 12 \end{bmatrix}$$

The original matrix is already of the form in which x_1 is the more important variable, and both are positively correlated with the response variable. If x_2 were negatively correlated with the response variable with the same variable importance, the design matrix could be rearranged as:

$$\begin{bmatrix} x_1 & x_2 \\ 1 & 12 \\ 1 & 5 \\ 2 & 12 \\ 2 & 5 \end{bmatrix}$$

Type I independence means that the order of less important variables in the design matrix can be changed if its correlation with the response variable changes. Thus, all full factorial designs are Type I.

Example 2.3

x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
95	5.5	10.77	105	5.5	10.54	115	5.5	10.69
95	5.5	10.77	105	5.5	10.54	115	5.5	10.69
95	6.0	10.77	105	6.0	10.54	115	6.0	10.69
95	6.5	10.77	105	6.5	10.54	115	6.5	10.69
95	5.5	17.66	105	5.5	17.40	115	5.5	16.45
95	6.0	17.66	105	6.0	17.40	115	6.0	16.45
95	6.0	17.66	105	6.0	17.40	115	6.0	16.45
95	6.5	17.66	105	6.5	17.40	115	6.5	16.45
95	5.5	24.30	105	5.5	23.98	115	5.5	22.56
95	6.0	24.30	105	6.0	23.98	115	6.0	22.56
95	6.5	24.30	105	6.5	23.98	115	6.5	22.56
95	6.5	24.30	105	6.5	23.98	115	6.5	22.56

This is a Type II design.The design matrix contains three input variables: x_1 with levels 95, 105 and 115, x_2 with levels 5.5, 6.0 and 6.5, and x_3 with 9 levels that can be grouped into 3 groups: group I with levels 10.54, 10.77 and 10.69, group II with levels 16.45, 17.4 and 17.66, and group III with levels 22.56, 23.98 and 24.3. There are nine replicates which are balanced for x_1: three replicates for each x_1 setting per x_3 group. However, they are not balanced for x_2: 5.5 is replicated with group I, 6.0 with group II, and 6.5 with group III.

x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
95	5.5	10.77	105	5.5	10.54	115	5.5	10.69
95	5.5	10.77	105	5.5	10.54	115	5.5	10.69
95	5.5	17.66	105	5.5	17.40	115	5.5	16.45
95	5.5	24.30	105	5.5	23.98	115	5.5	22.56
95	6.0	10.77	105	6.0	10.54	115	6.0	10.69
95	6.0	17.66	105	6.0	17.40	115	6.0	16.45
95	6.0	17.66	105	6.0	17.40	115	6.0	16.45
95	6.0	24.30	105	6.0	23.98	115	6.0	22.56
95	6.5	10.77	105	6.5	10.54	115	6.5	10.69
95	6.5	17.66	105	6.5	17.40	115	6.5	17.40
95	6.5	24.30	105	6.5	23.98	115	6.5	22.56
95	6.5	24.30	105	6.5	23.98	115	6.5	22.56

The above table shows the ordering with all variables positively correlated with the response variable and $x_1 \succ x_2 \succ x_3$. If this order is maintained, but the correlation of either x_2 or x_3 or both is changed, their order within the design

matrix can be accommodated. Below is the design matrix with x_2 negatively correlated and the other two positively correlated with the response.

x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
95	6.5	10.77	105	6.5	10.54	115	6.5	10.69
95	6.5	17.66	105	6.5	17.40	115	6.5	16.45
95	6.5	24.30	105	6.5	23.98	115	6.5	22.56
95	6.5	24.30	105	6.5	23.98	115	6.5	22.56
95	6.0	10.77	105	6.0	10.54	115	6.0	10.69
95	6.0	17.66	105	6.0	17.40	115	6.0	16.45
95	6.0	17.66	105	6.0	17.40	115	6.0	16.45
95	6.0	24.30	105	6.0	23.98	115	6.0	22.56
95	5.5	10.77	105	5.5	10.54	115	5.5	10.69
95	5.5	10.77	105	5.5	10.54	115	5.5	10.69
95	5.5	17.66	105	5.5	17.66	115	5.5	16.45
95	5.5	24.30	105	5.5	23.98	115	5.5	22.56

However, if x_3 is considered the most important variable, it does not matter if x_1 is more important than x_2 or how x_1 is correlated with the response variable; the order of the variables within the design matrix cannot be changed. This is because the levels of x_1 are tied to those of x_3. The following table illustrates this, with x_2 and x_3 positively correlated with the response variable.

x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3
105	5.5	10.54	115	5.5	16.45	115	5.5	22.56
105	5.5	10.54	115	6.0	16.45	115	6.0	22.56
105	6.0	10.54	115	6.0	16.45	115	6.5	22.56
105	6.5	10.54	115	6.5	16.45	115	6.5	22.56
115	5.5	10.69	105	5.5	17.40	105	5.5	23.98
115	5.5	10.69	105	6.0	17.40	105	6.0	23.98
115	6.0	10.69	105	6.0	17.40	105	6.5	23.98
115	6.5	10.69	105	6.5	17.40	105	6.5	23.98
95	5.5	10.77	95	5.5	17.66	95	5.5	24.30
95	5.5	10.77	95	6.0	17.66	95	6.0	24.30
95	6.0	10.77	95	6.0	17.66	95	6.5	24.30
95	6.5	10.77	95	6.5	17.66	95	6.5	24.30

Type II designs convey the information that the design matrix is changeable or not depending on the order of the variable importance. Irregular fractional designs are Type II.

Example 2.4 This is a Type III design.

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

This is a 2^{7-4} fractional factorial design. Any three variables form a 2^3 design which is of Type I. However, any variable added to the 2^3 design does not add any information, since the levels of a fourth variable are determined by the other three. The nature of a Type III design is that of an augmented Type I design. All the information is contained within the various Type I designs inside the Type III design. All fractional factorial designs are of Type III.

We will now present strategies to solve for the optimal pw vector.

Full 2-Level Factorial

To motivate the selection process, consider the following 2^2 design:

$$\begin{bmatrix} x_1 & x_2 \\ 20 & 5 \\ 20 & 6 \\ 30 & 5 \\ 30 & 6 \end{bmatrix}$$

Let pw_i correspond to $x_i, i = 1, 2$. Without loss of generality, suppose both x_1 and x_2 are positively correlated with the response variable. The proper choice of pw depends on which of the two factors is more important. If x_1 is more important, we have $20pw_1 + 6pw_2 < 30pw_1 + 5pw_2$ or $pw_2 < 10pw_1$. If we chose $pw_1 = 1/2$, then we are allowed any other fraction from the basis of prime number reciprocals such as $1/3$. On the other hand, if x_2 is more important, we have $pw_2 > 10pw_1$. If $pw_2 =$

1/2, the largest possible choice from the basis for pw_1 is 1/23. Identical results follow for the other correlation combinations.

The reader will note that the choice of pw depends on the range of the individual variables. In the above example, the range of x_1 is 30 - 20 = 10, that of x_2 is 6 - 5 = 1. The relative weight of x_2 to x_1 is 1/10. If x_2 is considered more important and pw_2 is chosen as 1/2, pw_1 could be chosen by first computing $1/2 * 1/10 = 1/20$, then choosing any pw from the basis which is less, such as 1/23.

This strategy can be extended to the 2^k factorial design, $k > 2$. Consider:

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ 10 & 1 & 7 \\ 10 & 1 & 21 \\ 10 & 2 & 7 \\ 10 & 2 & 21 \\ 30 & 1 & 7 \\ 30 & 1 & 21 \\ 30 & 2 & 7 \\ 30 & 2 & 21 \end{bmatrix}$$

The reader will note that the ranges of x_1, x_2 and x_3 are respectively, 20, 1 and 14. Assuming that the factors are positively correlated with the response variable and that $x_2 \succ x_3 \succ x_1$, the design matrix can be rearranged to reflect the relative importance of the factors:

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ 10 & 1 & 7 \\ 30 & 1 & 7 \\ 10 & 1 & 21 \\ 30 & 1 & 21 \\ 10 & 2 & 7 \\ 30 & 2 & 7 \\ 10 & 2 & 21 \\ 30 & 2 & 21 \end{bmatrix}$$

Performing the same algebraic computations with rows 2 and 3 (or equally with rows 6 and 7) and with rows 4 and 5, we derive:

$$20pw_1 < 14pw_3$$

$$20pw_1 + 14pw_3 \quad < \quad pw_2$$

If $pw_2 = 1/2$, clearly $pw_3 < 1/2 * 1/14$. If it is chosen to be, say, $1/101$, pw_1 must satisfy two conditions: $pw_1 < 1/20 * 14/101$ and $pw_1 < 1/20 * (1/2 - 14/101)$, i.e. $pw_1 < 1/144$. Any fraction from the prime number reciprocal basis less than $1/144$, say, $1/1001$, will satisfy both conditions.

Note that the importance of the variables was chosen to be inversely proportional to the range of the levels, which guarantees that the pw basis of $\begin{bmatrix} 1/2 & 1/101 & 1/1001 \end{bmatrix}$ would permute the design matrix according to the variable importance chosen. This motivates the general strategy for a 2^k factorial. If we have factors x_1 through x_k with corresponding level ranges r_1 through r_k where $r_i = (\mu_i + \sigma_i) - (\mu_i - \sigma_i)$, and $r_i < r_{i+1}$, and corresponding pw, pw_1 through pw_k, where $pw_i > pw_{i+1}, i = 1 \ldots (k-1)$, the pw must satisfy the following conditions:

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & -r_{k-1} & r_k \\ 0 & 0 & \cdots & -r_{k-2} & r_{k-1} & r_k \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -r_1 & r_2 & \cdots & r_{k-2} & r_{k-1} & r_k \end{bmatrix} \begin{bmatrix} pw_1 \\ pw_2 \\ \vdots \\ pw_k \end{bmatrix} < \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.3}$$

Example 2.5

$$\begin{bmatrix} x_1 & x_2 & Y \\ 1 & 5 & 0.4 \\ 1 & 12 & 0.21 \\ 2 & 5 & 0.61 \\ 2 & 12 & 0.89 \end{bmatrix}$$

This example is a 2^2 factorial design before coding the variables x_1 and x_2. It will illustrate, in full detail, how the bias is computed for a given pseudo-weight vector.

The range of x_1 is 1, that of x_2 is 7. The basis of the pw is chosen to be $1/2$ and $1/17$ since both $-1/2 + 5/17 < 0$ and $-5/2 + 1/17 < 0$, satisfying equation 2.3. The minimum difference in the y values is 0.4 - 0.21 = 0.19. Thus σ is computed as $0.19/6 = 0.03167$. Under the various pw permutations, using 1000 simulated values of Y, the biases are presented:

pw permutations	bias	pw permutations	bias
(-1/2, -1/17)	0.1582	(-1/17, -1/2)	0.0441
(-1/2, 1/17)	0.5028	(-1/17, 1/2)	0.5306
(1/2, -1/17)	0.0473	(1/17, -1/2)	0.0726
(1/2, 1/17)	0.0220	(1/17, 1/2)	0.1895

To illustrate how the biases were computed, we will use five simulated values of Y and the the pw vectors (1/2, 1/17) and (-1/17, 1/2). First,

$$\ddot{Y} = \begin{bmatrix} 0.2282292 & 0.2196398 & 0.1825924 & 0.1556550 & 0.2244890 \\ 0.3584038 & 0.3991967 & 0.4181021 & 0.3969132 & 0.3892801 \\ 0.6209335 & 0.6069105 & 0.6164507 & 0.6048003 & 0.6390413 \\ 0.8327723 & 0.9472272 & 0.8817289 & 0.9114386 & 0.8883125 \end{bmatrix}$$

We begin with the pw vector (1/2, 1/17). To determine how the rows of $\ddot{Y}$ should be sorted, we observe the rankings of the combined pw:

$$\begin{bmatrix} 1 & 5 \\ 1 & 12 \\ 2 & 5 \\ 2 & 12 \end{bmatrix} \begin{bmatrix} 1/2 \\ 1/17 \end{bmatrix} = \begin{bmatrix} 0.79 \\ 1.21 \\ 1.29 \\ 1.71 \end{bmatrix}$$

The ranking of the combined pw vector is (1, 2, 3, 4). Since the rows of $\ddot{Y}$ are already arranged in this order, there is no need to rearrange the rows. The next step is to produce the beta vector using the method of least squares using X and $\ddot{Y}$. The matrix is:

$$\ddot{B} = \begin{bmatrix} 0.302247547 & 0.28880883 & 0.27798663 & 0.27871392 & 0.30533790 \\ 0.009246454 & 0.01645144 & 0.01602328 & 0.01564523 & 0.01206074 \end{bmatrix}$$

Computing the mean vector, θ, and covariance matrix $\ddot{\Sigma}$ of $\ddot{B}$, we have

$$\theta = \begin{bmatrix} 0.29061897 \\ 0.01388543 \end{bmatrix} \ddot{\Sigma} = \begin{bmatrix} 164.11342 & -33.35092 \\ -33.35092 & 9.775336 \end{bmatrix} \text{x}10^{-6}$$

To solve for R, we need the mse from the least squares model. In this example, we have the mse = 0.1949323. Solving for $R = \text{mse} \cdot \ddot{\Sigma}^{-1}$, we have

$$R = \begin{bmatrix} 3873.184 & 13214.30 \\ 13214.302 & 65025.02 \end{bmatrix}$$

Solving for $\hat{B} = (X'X + R)^{-1}(X'Y + R\theta)$, we have

$$\hat{B} = \begin{bmatrix} 0.29083084 \\ 0.01382308 \end{bmatrix}$$

Solving for the bias, $(\hat{B} - \theta)'R(X'X)^{-1}R(\hat{B} - \theta)$, the result is 0.01966979. The reader should note that the bias using five simulated value of Y does not differ much from that using 1000 simulated values.

If (-1/17, 1/2) is the chosen pw vector, the combined pw vector is

$$\begin{bmatrix} 1 & 5 \\ 1 & 12 \\ 2 & 5 \\ 2 & 12 \end{bmatrix} \begin{bmatrix} -1/17 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 2.441176 \\ 5.941176 \\ 2.382353 \\ 5.882353 \end{bmatrix}$$

which is in the order (2, 4, 1, 3). We rearrange $\ddot{Y}$ to reflect this with the result

$$\ddot{Y} = \begin{bmatrix} 0.3584038 & 0.3991967 & 0.4181021 & 0.3969132 & 0.3892801 \\ 0.8327723 & 0.9472272 & 0.8817289 & 0.9114386 & 0.8883125 \\ 0.2282292 & 0.2196398 & 0.1825924 & 0.1556550 & 0.2244890 \\ 0.6209335 & 0.6069105 & 0.6164507 & 0.6048003 & 0.6390413 \end{bmatrix}$$

This changes θ, R and $\hat{B}$ with

$$\theta = \begin{bmatrix} -0.1120674 \\ 0.0793107 \end{bmatrix} R = \begin{bmatrix} 14114.46 & 72760.9 \\ 72760.90 & 414975.9 \end{bmatrix} \hat{B} = \begin{bmatrix} -0.11119630 \\ 0.07915114 \end{bmatrix}$$

with a resulting bias of 0.5608.

Since (1/2, 1/17) has the minimum bias, x_1 is deemed the more important variable with both variables being positively correlated with Y. Note that there is no permutation that would sort the vector $\begin{bmatrix} 0.21 & 0.4 & 0.61 & 0.89 \end{bmatrix}'$ into the vector of observed values $\begin{bmatrix} 0.4 & 0.21 & 0.61 & 0.89 \end{bmatrix}'$ since that would require $pw_1 + 12pw_2 < pw_1 + 5pw_2$ or $pw_2 < 0$ and $2pw_1 + 5pw_2 < 2pw_1 + 12pw_2$ or $pw_2 > 0$, a contradiction.

Multi-Level Full Factorial

Full factorial designs with three or more levels per factor employ the same strategy as the 2-level factorial designs, with one modification: the minimum and maximum ranges of the levels in a factor must be taken into account.

Example 2.6 Let x_1, x_2 and x_3 have respective levels $(1,3,5)$, $(10,20,30)$ and $(7,14,21)$ from a 3^3 factorial experiment. This table indicates the respective ranges:

	min	max
x_1	2	4
x_2	10	20
x_3	7	14

If $x_1 \succ x_2 \succ x_3$ is the prior anticipation, then the pw must be chosen to satisty the following conditions:

$$10pw_2 > 14pw_3$$

$$2pw_1 > 20pw_2 + 14pw_3$$

so that the design matrix can be arranged according to the above importance ranking regardless of the correlation of x_1, x_2 and x_3 with the response. The basis $\begin{bmatrix} 1/2 & 1/101 & 1/1001 \end{bmatrix}$ is satisfactory. If $x_3 \succ x_1 \succ x_2$, then the pseudo-weight inequalities change to:

$$2pw_1 > 20pw_2$$

$$7pw_3 > 4pw_1 + 20pw_2$$

In this case, $\begin{bmatrix} 1/11 & 1/1001 & 1/2 \end{bmatrix}$ satisfies the conditions.

The reader will note from the above example that the same selection of pw in various permutations cannot necessarily be used. Each permutation must be examined individually.

In general, if we have factors x_1 through x_k with corresponding minimum level ranges $r_{1.min}$ through $r_{k.min}$ and maximum ranges $r_{1.max}$ through $r_{k.max}$, and corresponding pw, pw_1 through pw_k, where $pw_i > pw_{i+1}, i = 1 \ldots (k-1)$, the pw must be chosen to satisfy the following conditions:

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & -r_{k-1.min} & r_{k.max} \\ 0 & 0 & \cdots & -r_{k-2.min} & r_{k-1.max} & r_{k.max} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -r_{1.min} & r_{2.max} & \cdots & r_{k-2.max} & r_{k-1.max} & r_{k.max} \end{bmatrix} \begin{bmatrix} pw_1 \\ pw_2 \\ \vdots \\ pw_k \end{bmatrix} < \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.4)$$

Imbalanced Designs

Example 2.3 continued The full data set is provided in Appendix 2 using y_1 as the response. The prior anticipation is that all three input variables are positively correlated with the response variable y_1. There were 1000 random generations per observation with $6\sigma = 0.02$ since 0.02 is the smallest positive difference from the observed y_1 vector. The ranges of the input variables are:

	x_1	x_2	x_3
minimum	10	0.5	0.08
maximum	20	1.0	13.76

In order to satisfy the inequalities in equation 2.4 using the above ranges, the pw were chosen as shown in the following table with 1 indicating the variable deemed the most important:

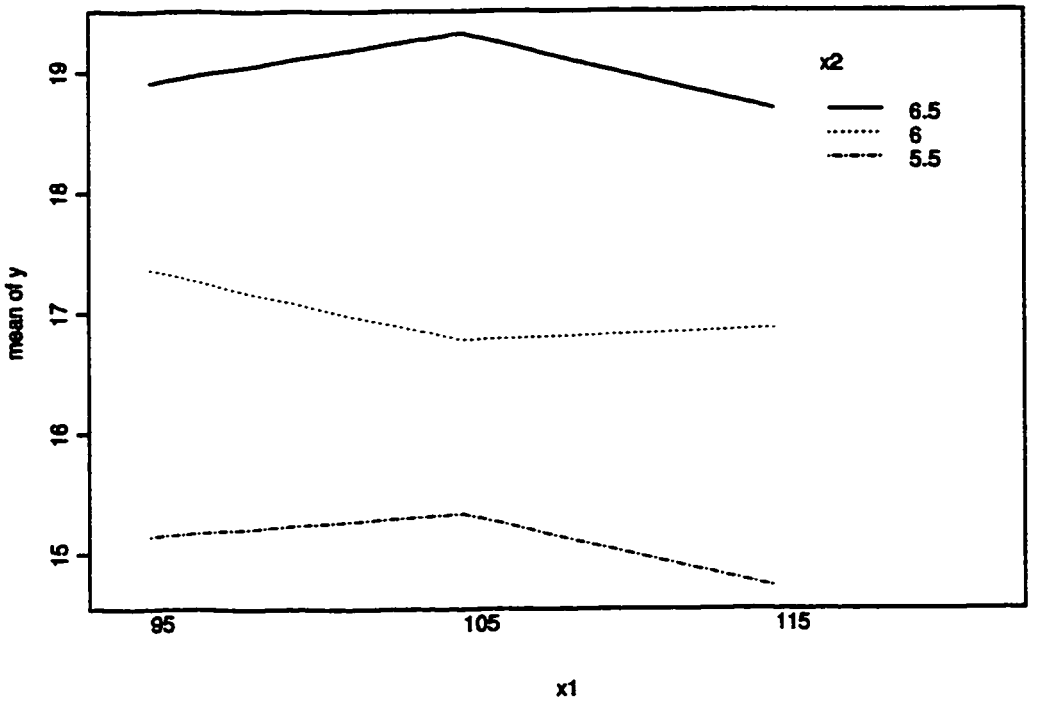
x_1	x_2	x_3	x_1 pw	x_2 pw	x_3 pw
1	2	3	1/2	1/3	1/101
1	3	2	1/2	1/367	1/29
2	1	3	1/83	1/2	1/10001
3	1	2	1/100001	1/2	1/59
2	3	1	1/503	1/10001	1/2
3	2	1	1/10001	1/29	1/2

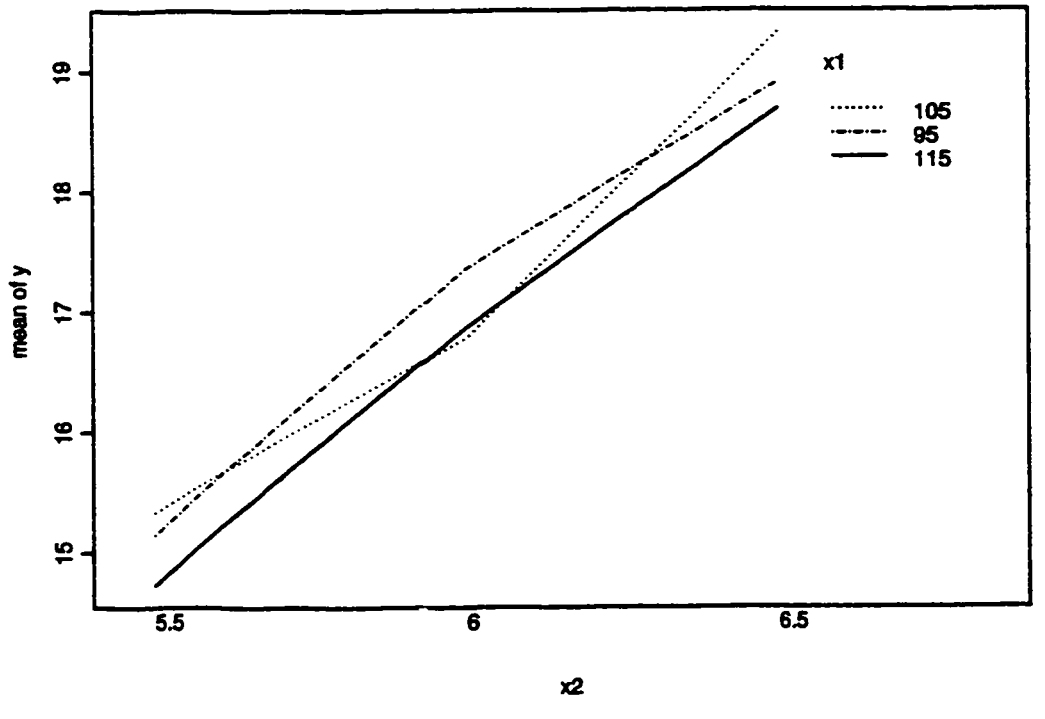
The bias was computed three times with a fresh set of 1000 simulated values generated each time. The results, rounded to three decimal places were:

x_1	x_2	x_3	bias 1	bias 2	bias 3
1	2	3	2099.785	2099.799	2099.781
1	3	2	2018.137	2018.143	2018.168
2	1	3	1314.241	1314.251	1314.258
3	1	2	1290.281	1290.274	1290.271
2	3	1	8.466	8.467	8.468
3	2	1	8.468	8.465	8.467

Clearly, x_3 is the most important variable. The biases for the last two rows are the same since we have a Type II design. Comparing x_1 with x_2, the average amount of bias for x_1 when it is ranked first is 2058.9688, compared to 1302.2627 for

Figure 2.1: Example 2.3: Interaction of x_1 and x_2





x_2, indicating that x_2 is the more important of the two. Two tools can confirm this conclusion: interaction plots and marginal bias computations. Figure 2.1 shows the interaction of x_1 and x_2 with y. As shown, the variation in y is greater for x_2 than for x_1, indicating that x_2 is the more important input variable.

The marginal bias computations for x_1 and x_2 involve subsetting the design matrix and the results vector for fixed values of x_3 . Since x_3 can be partitioned into three distinct groups, marginal biases can be derived for each as previously. To illustrate, the partial design matrix and response vector corresponding to the Group I levels of x_3 is:

row	x_1	x_2	y	row	x_1	x_2	y	row	x_1	x_2	y
1	95	5.5	9.89	13	105	5.5	9.48	25	115	5.5	8.99
2	95	5.5	9.44	14	105	5.5	9.11	26	115	5.5	10.59
3	95	6.0	7.74	15	105	6.0	9.48	27	115	6.0	8.02
4	95	6.5	8.33	16	105	6.5	10.06	28	115	6.5	9.74

Computing the bias for all three groups yields the following results:

	x_1	x_2	Group I bias	Group II bias	Group III bias	Average Bias
rank	1	2	0.0588	1.1358	21.2591	7.4846
	2	1	2.9355	0.5312	5.7972	3.0880

This design had nine replicates. If the values for the replicates are averaged, the above bias chart reads as follows:

	x_1	x_2	Group I bias	Group II bias	Group III bias	Average Bias
rank	1	2	0.3926	0.4702	18.5326	6.4651
	2	1	3.2824	0.8233	2.5538	2.2199

The average bias for both confirms that x_2 is the more important variable.

Recall that the prior anticipation was that both variables were positively correlated with the response variable. To confirm this, prior anticipations with at least one negative correlation should be examined. Of the six possible combinations, the pw vector $\begin{bmatrix} -1/2 & -1/3 \end{bmatrix}$ had the smallest bias average of 3.5825 using 1000 replicates of the response vector, which is larger than the smallest average bias of

2.2199 from the above table. This confirms that both variables are positively correlated with the response variable.

The reader should note that if interaction charts prove inconclusive, bias computations can be used to determine input variable importance. The following example illustrates this point.

Example 2.7

x_1	x_2	x_3	x_4	Y
−1	−1	−1	−1	5.3483
−1	−1	−1	1	4.5101
−1	−1	1	−1	3.9182
−1	−1	1	1	5.2637
−1	1	−1	−1	6.1347
−1	1	−1	1	3.5249
−1	1	1	−1	6.8008
−1	1	1	1	0.9871
1	−1	−1	1	5.3539
1	−1	1	−1	5.0506
1	1	−1	−1	5.0725
1	1	1	1	4.5834

This is an irregular fractional factorial design which is almost Resolution V. The initial decision was to test three variables at a time in order to minimize the number of pw needed to determine the proper variable weights and correlations. The pw basis of $\left[\begin{array}{ccc} 1/2 & 1/3 & 1/7 \end{array}\right]$ was chosen to satisfy equation 2.4. Beginning with x_1, x_2 and x_3 , and 1000 replications of Y, testing under a logical sequence of correlations and weights produced the following:

x_1 weight	x_2 weight	x_3 weight	bias
1/2	1/3	1/7	36.30498
-1/2	1/3	1/7	28.85875
-1/2	-1/3	1/7	22.95018
-1/2	-1/3	-1/7	16.62441
-1/2	-1/7	-1/3	16.3097
-1/3	-1/7	-1/2	13.1459
-1/7	-1/3	-1/2	11.62014

Based on the above calculations, the conclusion was that $x_3 \succ x_2 \succ x_1$, with all three variables negatively correlated with the response variable. The next variable combination choice was x_2, x_3 and x_4. However, the pw vectors $\begin{bmatrix} -1/7 & -1/3 & 1/2 \end{bmatrix}$ and $\begin{bmatrix} -1/7 & -1/3 & -1/2 \end{bmatrix}$, in which the weights correspond to the variables in order of x_2, x_3 and x_4, produced the same bias of 8.6368. As with the previous example, this is due to this being a Type II design. It was then decided to pair x_4 with the other variables to find its proper weight and correlation.

x_3 weight	x_4 weight	bias
-1/3	1/2	27.42
-1/3	-1/2	0.80
-1/2	-1/3	4.00

x_2 weight	x_4 weight	bias
-1/3	1/2	27.91
-1/3	-1/2	1.25
-1/2	-1/3	5.17

x_1 weight	x_4 weight	bias
-1/3	1/2	42.09
-1/3	-1/2	10.91
-1/2	-1/3	19.80

In the above computations of the bias, the response was averaged across equal design levels. After averaging, the correlation of the input variables with the response was established first, followed by the variable importance. The pw basis of $\begin{bmatrix} 1/2 & 1/3 \end{bmatrix}$ was chosen to satisfy equation 2.3. Based on the above evidence, x_4 was deemed the most important variable being negatively correlated with the response variable. The reader may have noticed the increase in the bias as x_4 was paired with less important variables. The relative magnitude of the bias in these pairings serves as a tool in determining variable importance.

An examination of the interaction charts of some variable pairings, particularly those with strong interactions, demonstrates how bias computation can be helpful in

situations in which the charts are inconclusive. The interaction plots on the following pages are drawn from this data set.

Type III designs

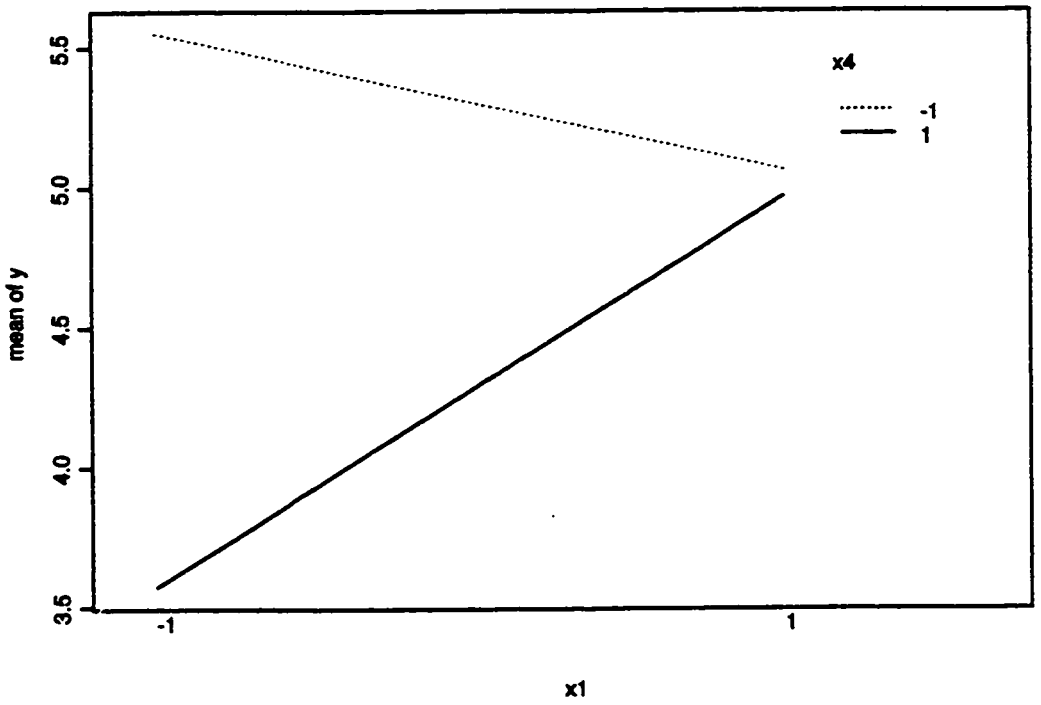
There are three main types of designs of Type III: fractional factorial, Plackett-Burman, and mixed fraction. The essential point to keep in mind is that all Type III designs contain various Type I designs. The basic strategy is to perform analysis on Type I subsets of the original design matrix.

Example 2.8

x_1	x_2	x_3	x_4	Y
−1	−1	−1	−1	8.0050
−1	−1	1	1	9.5715
−1	1	−1	1	8.5010
−1	1	1	−1	9.1714
0	−1	−1	1	4.6379
0	−1	1	−1	3.9728
0	1	−1	−1	2.5343
0	1	1	1	6.1502
1	−1	−1	−1	7.6290
1	−1	1	1	3.7933
1	1	−1	1	3.3766
1	1	1	−1	6.5514

This is a mixed fractional design. It is almost Resolution V. Any three variables form a Type I design. The initial strategy follows that of the previous example. The results are as follows:

Figure 2.2: Example 2.7: Interaction of x_1 and x_4



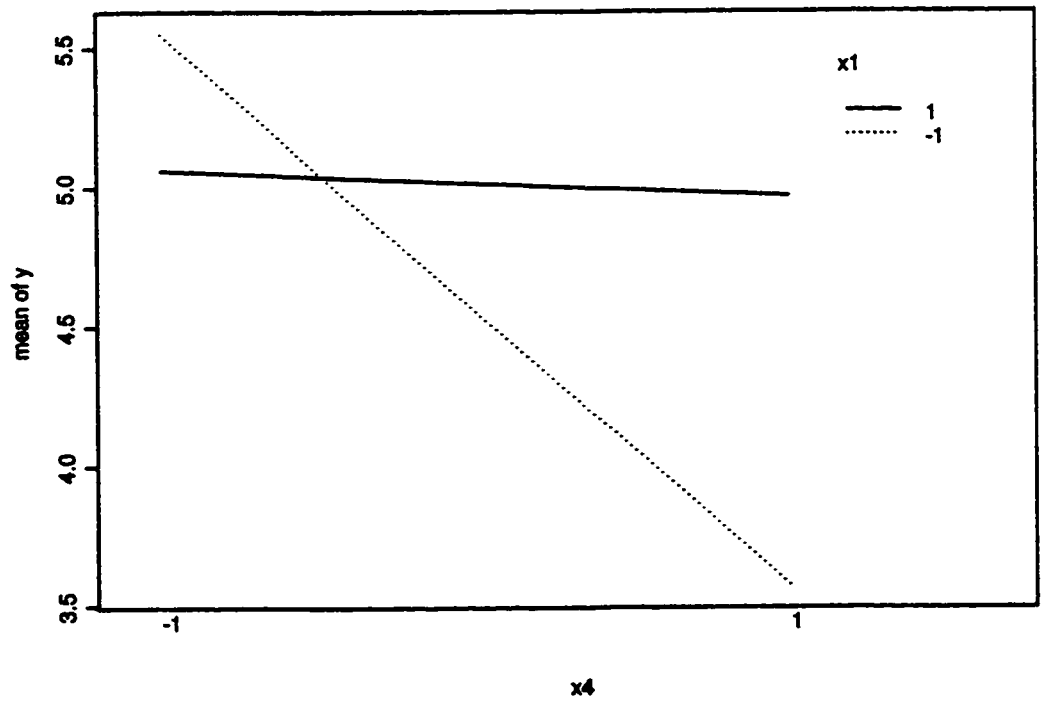
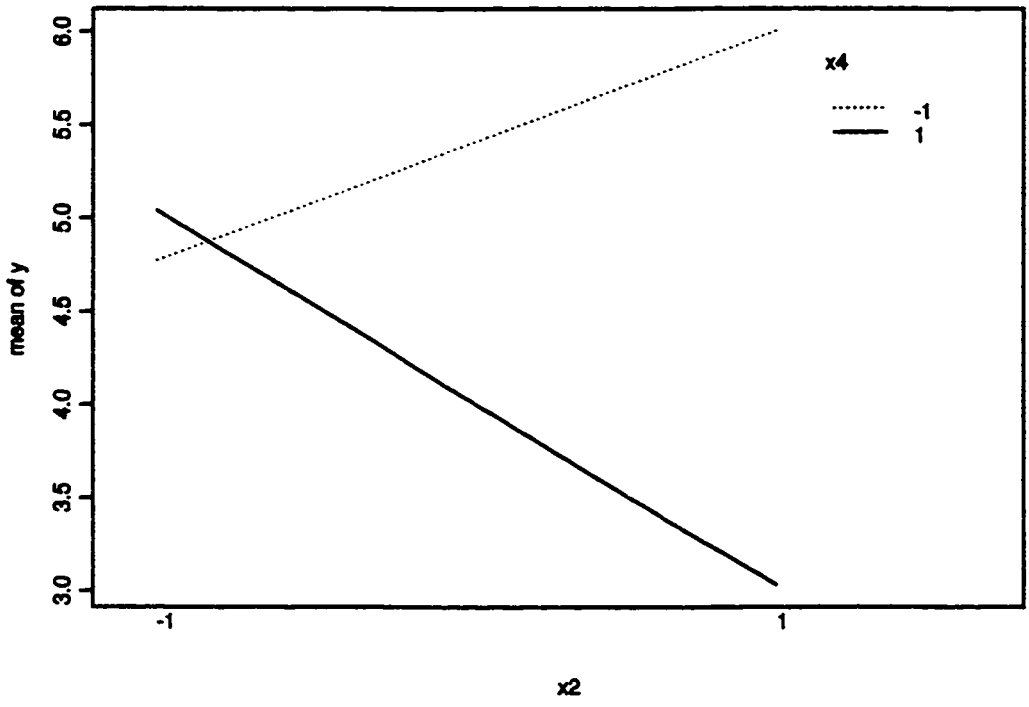


Figure 2.3: Example 2.7: Interaction of x_2 and x_4



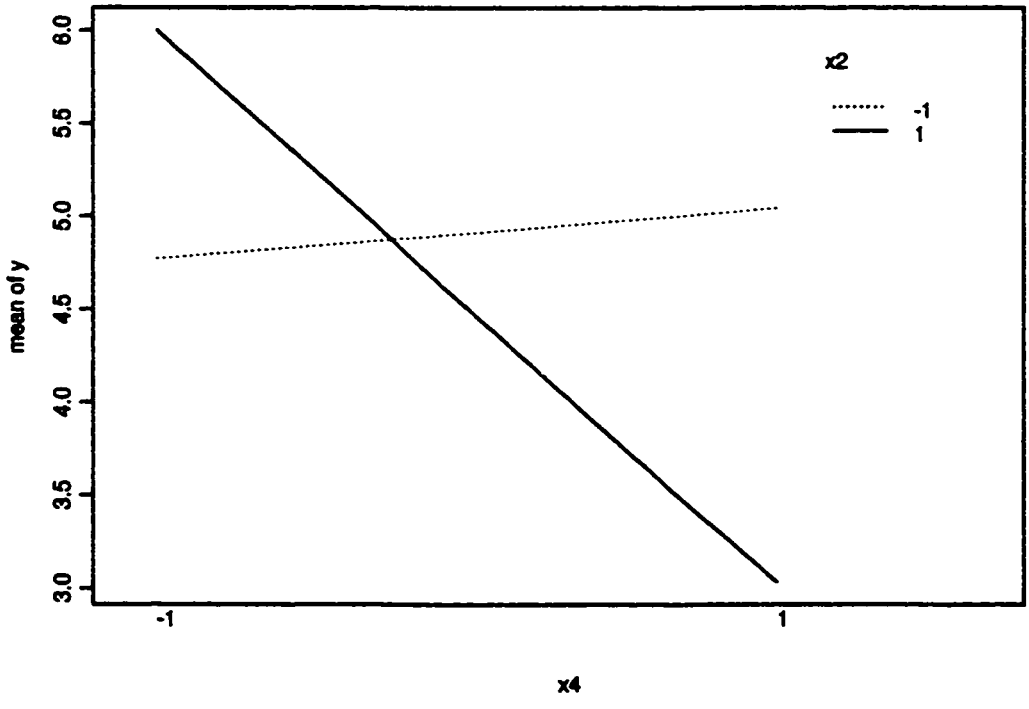
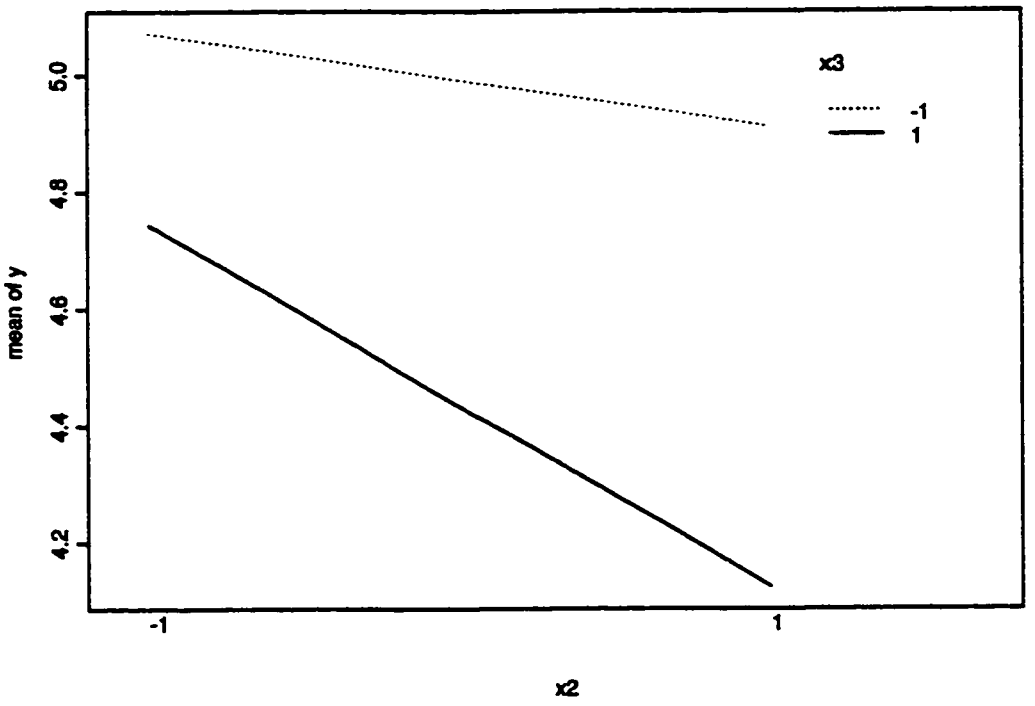
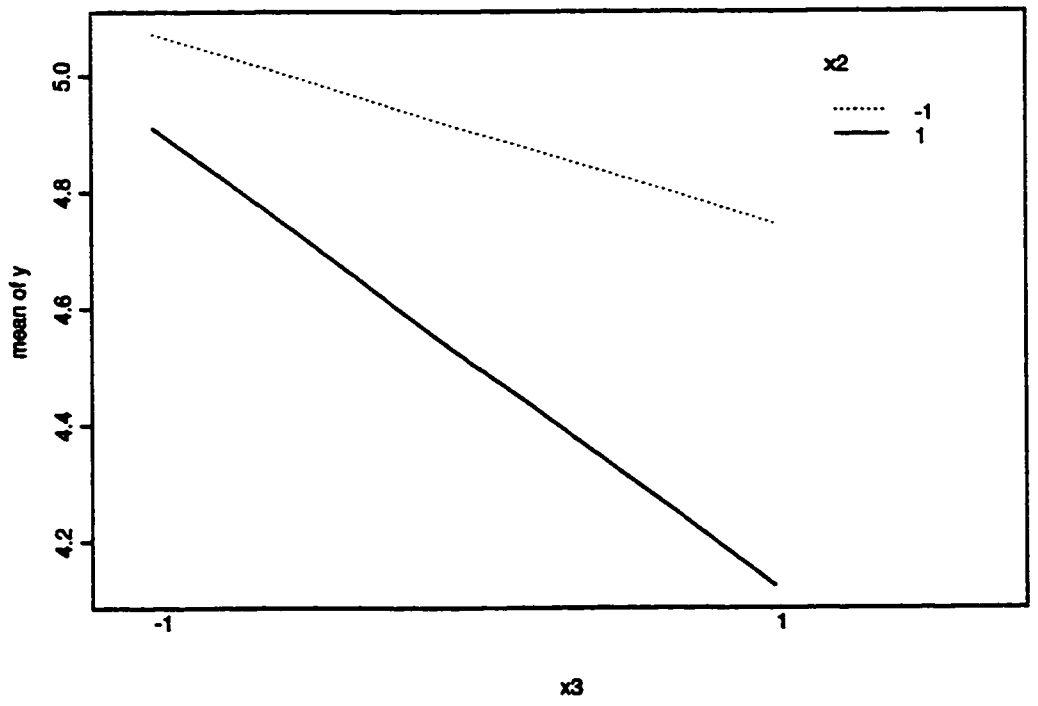


Figure 2.4: Example 2.7: Interaction of x_2 and x_3





x_1 weight	x_2 weight	x_3 weight	x_4 weight	bias
1/2	1/3	1/7	-	148.5473
-1/2	1/3	1/7	-	27.12483
-1/2	-1/3	1/7	-	19.55255
-1/2	-1/3	-1/7	-	29.40098
-1/2	-1/7	1/3	-	**14.87727**
-1/3	-1/7	1/2	-	38.77794
-1/3	-	1/7	1/2	63.25925
-1/3	-	1/7	-1/2	47.15386
-1/2	-	1/7	-1/3	18.33796
-1/2	-	1/3	-1/7	**14.45265**
-	-1/3	1/2	-1/7	45.6551
-	-1/7	1/2	-1/3	45.32922
-	-1/2	-	-1/3	55.29393
-	-1/3	-	-1/2	**53.77354**

As previously, the correlation with the response was established first. The bias values for the last two rows were computed simply to determine which of x_2 and x_4 is more important in its contribution to the variability of the response. Based on the above evidence, we conclude $x_1 \succ x_3 \succ x_4 \succ x_2$, with x_3 being the only variable positively correlated with the response variable.

If the bias values are computed pairwise with the proper correlations, a table of the values will corroborate the above conclusions. Drawing a fresh set of 1000 replicated Y values:

	x_1	x_3	x_4	x_2
x_1	-	16.67996	22.88221	24.27083
x_3	40.30613	-	43.54208	44.11251
x_4	49.69935	51.0251	-	53.71403
x_2	51.73672	53.11702	55.29556	-

The variable deemed the more important runs from top to bottom. For example, comparing the bias values for x_1 and x_3, if x_1 is considered the more important variable, the bias value is 16.67996. On the other hand, if x_3 is considered more important, the bias value is 40.30613. By comparing bias values pairwise, one can determine the importance order of the variables.

2.3.1 Exceptions

The experimenter may be faced with the possibility that the design matrix fits into none of the three types introduced earlier. This situation may arise if the experimenter faces cost or time constraints and chooses a second order design such as central composite.

If this is the case, variable importance must be determined one variable at a time. The pw chosen can be either 1 or -1 to determinine the correlation with the response variable. As before, the amount of bias determines the importance of the variable.

Example 2.9

$$\begin{bmatrix} x_1 & x_2 & x_3 & Y \\ -1 & -1 & -1 & 52.2 \\ 1 & 1 & -1 & 68.1 \\ 1 & -1 & 1 & 57.9 \\ -1 & 1 & 1 & 55.3 \\ -\sqrt{2} & 0 & 0 & 59.8 \\ \sqrt{2} & 0 & 0 & 58.5 \\ 0 & -\sqrt{2} & 0 & 67.3 \\ 0 & \sqrt{2} & 0 & 70.2 \\ 0 & 0 & -\sqrt{2} & 55.8 \\ 0 & 0 & \sqrt{2} & 47.4 \\ 0 & 0 & 0 & 60.3 \end{bmatrix}$$

The experimenter chose a small central composite design. No two variables form a Type I design. In order to determine bias one variable at a time, the sorted Y values must be averaged across equal level values, so that for each input variable level, there is only one average response value. This procedure produced the following results:

bias	x_1	x_2	x_3
-1	1197.283	1207.447	668.551
1	740.413	703.977	688.551

Based on the above results, the experimenter concluded that $x_3 \succ x_2 \succ x_1$ with x_3 being negatively correlated with the response variable and the other two

positively correlated.

Variable importance can also sometimes be deduced from the magnitude of the bias when the variable is wrongly correlated with the response variable. In the above example, the bias for x_2 is less than that of x_1 when both are properly correlated with the response. The opposite is true when both are wrong correlated. This principle will resurface in Chapter 4 under the subject of variable importance in linear regression.

2.4 Conclusion

As shown, the methodology presented to find variable importance is consistent regardless of the type of design matrix used. If the matrix is balanced, the results are consistent with the computation of the main effects. If the design is not balanced, this method provides an accurate measure of variable importance. If the experimenter is concerned with being able to conduct meaningful analysis on an imbalanced design matrix, the methods presented here allow the experimenter considerable leeway in choosing levels for the input variables.

Chapter 3

Optimal Operating Conditions

Once the input variable importance and correlation with the response variable has been established, the goal becomes to find the optimal operating conditions under which the maximum response can be found. The approach depends on whether one is building a first order model or one of a higher degree.

3.1 First Order Model

The first order model algorithm is a multi-step procedure involving the path of steepest ascent and the least biased beta vector with their respective confidence regions in which we wish to find the true path of steepest ascent. The general objective is to have the confidence region of one vector contained within that of the other. At the same level of confidence, the confidence region of the biased vector is the smaller of the two, due to the presence of the prior precision matrix R in the posterior covariance matrix $(X'X + R)^{-1}$. As well, the difference between each individual coefficient of the biased vector and that of the least squares beta vector must be less than the standard error of the least squares coefficient. This will be referred to as the standard error criterion. Given a least squares coefficient, b_j, we have $se(b_j) = \sqrt{\hat{\sigma}^2 C_{jj}}$ where $\hat{\sigma}^2$ is the mean square error of the least squares equation and C_{jj} is the j^{th} diagonal element of $(X'X)^{-1}$. We present first the theory

necessary for the methodology, then the procedure itself.

3.1.1 Theoretical Underpinnings

As with any measure involving variation, the path of steepest ascent has a $1-\alpha$ confidence region for the true path of steepest ascent. The least biased beta vector also has a $1-\alpha$ confidence region which may or may not coincide with that of the path of steepest ascent. Since our goal is to find the optimal operating conditions, we want to show that a biased vector can be made to converge towards the path of steepest ascent, and its corresponding covariance matrix converge to a zero matrix, so that its confidence region can be enclosed inside that of the path of steepest ascent. Before we proceed with the theory of the convergence we examine some properties of Bayesian posterior vectors.

3.1.2 Expectations of Bayesian Posterior Vectors

Since the path of steepest ascent is an unbiased estimator of B, the true path of steepest ascent, we would like to show that a Bayesian posterior beta vector is also an unbiased estimator of B. It is assumed that $\mathrm{E}(\varepsilon) = 0$. If
$Y = XB + X(X'X + R)^{-1}R(B-\theta) + \varepsilon$:

$$\begin{aligned} E(\hat{B}) &= E((X'X+R)^{-1}(X'Y+R\theta)) \\ &= E((X'X+R)^{-1}(X'(XB+X(X'X)^{-1}R(B-\theta)+\varepsilon)+R\theta)) \\ &= E((X'X+R)^{-1}(X'XB+RB+X'\varepsilon) \\ &= (X'X+R)^{-1}(X'XB+RB) \\ &= (X'X+R)^{-1}(X'X+R)B \\ &= B \end{aligned}$$

Thus, $E(\hat{Y}) = XB$

However, $Var(\hat{Y})$ is dependent on the bias. Recall under the normality assumptions that

$$\begin{aligned} Var(\hat{B}) &= (X'X + R)^{-1}\sigma^2 \\ Var(\hat{Y}) &= Var(X\hat{B}) \\ &= XVar(\hat{B})X' \\ &= X(X'X + R)^{-1}X'\sigma^2 \end{aligned}$$

Since $(X'X + R)^{-1}$ is used in the derivation of the bias (see Appendix 1), the variance of both statistics depends on the quality of the prior opinion on the outcome of the experiment. This will prove important later in the computation of standard errors.

3.1.3 Recursive Convergence

Recall from equation 2.2 that the posterior sampling distribution of B is normal with mean $(X'X + R)^{-1}(X'Y + R\theta)$ and variance $(X'X + R)^{-1}\sigma^2$ in which σ^2 is the common variance of ε_i, θ is the prior mean of B and $R = \ddot{\Sigma}^{-1}\sigma^2$, in which $\ddot{\Sigma}$ is the prior covariance matrix of B.

We show first that a Bayesian posterior beta vector converges to the path of steepest ascent and its corresponding covariance matrix to a zero matrix by using the fact that the posterior distribution of a parameter can be used as the prior for the next posterior distribution of the same parameter. In the proofs to follow, X and Y are assumed to remain constant. While it is not realistic to assume that Y would remain the same for a given X design matrix each time the experiment is conducted, it does serve the purpose of showing that the path of steepest ascent, derived by the method of least squares, is the asymptotic first order beta vector, regardless of the prior anticipation initially chosen.

We denote as b_1 and Σ_1 the initial prior beta vector and covariance matrix, respectively, and r_i, b_i and Σ_i, the subsequent parameters at the ith iteration, in which

$$r_i = \sigma^2\Sigma_i^{-1} \tag{3.1}$$

$$b_{i+1} = (X'X + r_i)^{-1}(X'Y + r_i b_i) \tag{3.2}$$

$$\Sigma_{i+1} = \sigma^2(X'X + r_i)^{-1} \tag{3.3}$$

We wish to show via proof by induction that at the nth iteration,

$$\Sigma_n = \sigma^2((n-1)X'X + \sigma^2\Sigma_1^{-1})^{-1} \tag{3.4}$$

and

$$b_n = ((n-1)X'X + \sigma^2\Sigma_1^{-1})^{-1}((n-1)X'Y + \sigma^2\Sigma_1^{-1}b_1) \tag{3.5}$$

We begin with the proof for Σ_n.

The case of $k = 1$ is trivially true. We assume that it is true for $k = n$ and prove for $k = n + 1$ that

$$\Sigma_{n+1} = \sigma^2(nX'X + \sigma^2\Sigma_1^{-1})^{-1}$$

Using equations 3.3 and 3.1, we derive

$$\begin{aligned}
\Sigma_{n+1} &= \sigma^2(X'X + r_n)^{-1} \\
&= \sigma^2(X'X + \sigma^2\Sigma_n^{-1})^{-1} \\
&= \sigma^2(X'X + \sigma^2(\sigma^{-2}((n-1)X'X + \sigma^2\Sigma_1^{-1})))^{-1} \\
&= \sigma^2(nX'X + \sigma^2\Sigma_1^{-1})^{-1}
\end{aligned}$$

completing the proof for Σ_n. Using this as well as equations 3.1 through 3.3 we follow the same inductive steps to show that

$$b_{n+1} = (nX'X + \sigma^2\Sigma_1^{-1})^{-1}(nX'Y + \sigma^2\Sigma_1^{-1}b_1)$$

As previously, we assume the inductive hypothesis to be true for $k = n$.

$$\begin{aligned} b_{n+1} &= (X'X + r_n)^{-1}(X'Y + r_n b_n) \\ &= (X'X + \sigma^2\Sigma_n^{-1})^{-1}(X'Y + \sigma^2\Sigma_n^{-1}b_n) \\ &= (X'X + \sigma^2(\sigma^{-2}((n-1)X'X + \sigma^2\Sigma_1^{-1})))^{-1} \bullet \\ &\quad (X'Y + \sigma^2(\sigma^{-2}((n-1)X'X + \sigma^2\Sigma_1^{-1})) \bullet \\ &\quad ((n-1)X'X + \sigma^2\Sigma_1^{-1})^{-1}((n-1)X'Y + \sigma^2\Sigma_1^{-1}b_1)) \\ &= (nX'X + \sigma^2\Sigma_1^{-1})^{-1}(nX'Y + \sigma^2\Sigma_1^{-1}b_1) \end{aligned}$$

completing the proof for b_n. Clearly,

$$\begin{aligned} \Sigma_n &= \sigma^2((n-1)X'X + \sigma^2\Sigma_1^{-1})^{-1} \\ &= \frac{\sigma^2}{n-1}(X'X + \frac{\sigma^2\Sigma_1^{-1}}{n-1})^{-1} \end{aligned}$$

which tends to a zero matrix as $n \to \infty$. As well,

$$\begin{aligned} b_n &= ((n-1)X'X + \sigma^2\Sigma_1^{-1})^{-1}((n-1)X'Y + \sigma^2\Sigma_1^{-1}b_1) \\ &= (X'X + \frac{\sigma^2\Sigma_1^{-1}}{n-1})^{-1}(X'Y + \frac{\sigma^2\Sigma_1^{-1}b_1}{n-1}) \end{aligned}$$

which converges to the path of steepest ascent as $n \to \infty$.

The interpretation of the preceeding derivations must be made clear. If an experiment is conducted n times with the same experimental results, and the prior distribution is based on the previous $n-1$ experiments, the posterior distribution of the biased vector will converge to that of the path of steepest ascent with certainty. This is due to the fact that $E(Y) = XB$, in which B is the path of steepest ascent, serves as the average of experimental results. Using this information, we can construct confidence intervals satisfying the required inclusion properties.

3.1.4 Construction of Confidence Intervals

Box and Draper [3] state that a 1 - α confidence region for a beta vector with k parameters is the set $X_1 \ldots X_k$, $\sum_{i=1}^{k} X_i^2 = 1$ in which

$$\frac{\sum_{i=1}^{k}[b_i - (\sum_{i=1}^{k} b_i X_i)X_i]^2}{k-1} \leq S_b^2 F_{\alpha,(k-1),\nu_b}$$

where S_b^2 is the estimated common variance of the b_i and ν_b is the error degrees of freedom associated with the regression.

From a geometric viewpoint, the confidence region is a hypercone, with a corresponding angle,

$$\theta = \arcsin \sqrt{\frac{(k-1)S_b^2 F_{\alpha,(k-1),\nu_b}}{\sum_{i=1}^{k} b_i^2}} \tag{3.6}$$

in which θ is the semi-plane angle between the beta vector and the perimeter of the hypercone. The percentage of acceptable beta vectors originating from the origin is given by the fractional solid angle 2ϕ in which

$$P[t_{k-1} > 2\phi] = \sqrt{\frac{\sum_{i=1}^{k} b_i^2}{S_b^2 F_{\alpha,(k-1),\nu_b}} - (k-1)} \tag{3.7}$$

The quantity 2ϕ can be interpolated from tables of the t distribution.

It is intuitively clear that the confidence region of a biased vector can lie within that of the path of steepest ascent if $\theta_{\text{biased}} < \theta_{\text{steepest ascent}}$.

The key to this inclusion is the variable S_b^2. For the path of steepest ascent derived from a balanced design, S_b^2 is constant for all b_i, $i = 1 \ldots k$. Such is not the case for a biased vector due to the inclusion of R in the posterior variance of B.

However, since Σ and, hence, $(X'X + R)^{-1}$ can be made arbitrarily small by recursive convergence towards the zero matrix, S_b^2 can be taken as the average of the diagonal elements of Σ after sufficient convergence, so as to be treated as a constant.

Sufficient convergence has been reached once the confidence interval angle of the biased vector includes the path of steepest ascent. To show that this can always be the case, we present the following argument. Let ϕ be the angle between u, the path of steepest ascent, and v, the biased vector. Let θ be the confidence interval angle of v. We standardize u and v so that the magnitude of each is unity. Then,

$$\begin{aligned} \phi &= \arccos(u \cdot v) \\ \theta &= \arcsin\sqrt{(k-1)S_b^2 F_{\alpha,(k-1),\nu_b}} \end{aligned}$$

If $\theta > \phi$, then

$$\begin{aligned} \arcsin\sqrt{(k-1)S_b^2 F_{\alpha,(k-1),\nu_b}} &> \arccos(u \cdot v) \\ \sqrt{(k-1)S_b^2 F_{\alpha,(k-1),\nu_b}} &> \sin(\arccos(u \cdot v)) \\ (k-1)S_b^2 F_{\alpha,(k-1),\nu_b} &> \sin^2(\arccos(u \cdot v)) \\ (k-1)S_b^2 F_{\alpha,(k-1),\nu_b} + (u \cdot v)^2 &> 1 \end{aligned}$$

Choose α such that $F_{\alpha,(k-1),\nu_b} = S_b^{-2}$. Since $k \geq 2$, the last inequality holds. Thus a $1-\alpha$ confidence interval of a biased vector can be constructed to include the path of steepest ascent. This confidence interval indicates the region of acceptable beta vectors after n experiments, n being the number of iterations converging the biased vector towards the path of steepest ascent using equation 3.5

3.1.5 Methodology

Before proceeding with the methodology, there are two concepts which need to be introduced: the region of interest and the region of operability. The region of interest is the area of the input variables from which the maximum response may be derived. The region of operability is the actual physical arena for the input variables. An example may be an experiment in which an object is baked in an oven. While the model may say a maximum response would be achieved with the

oven temperature set at 1000°, the actual oven temperature may be limited to 600°. Any response surface model always has to take the region of operability into consideration. While the ideal is to have the region of interest contained within the region of operability, the above example illustrates that this may not always be the case.

In this section, we refer to the path of steepest ascent as B_0, the minimum bias vector as B_1. The general strategy at the first order level is to compute the angle between the two vectors as well as their respective confidence region angles at various levels of confidence. Once a level of confidence has been chosen, if the confidence region of the biased vector does not contain the path of steepest ascent, it will be necessary to converge it sufficiently by choosing an arbitrarily large value of n from equation 3.5 to achieve this status. Figure 3.1 illustrates the objective.

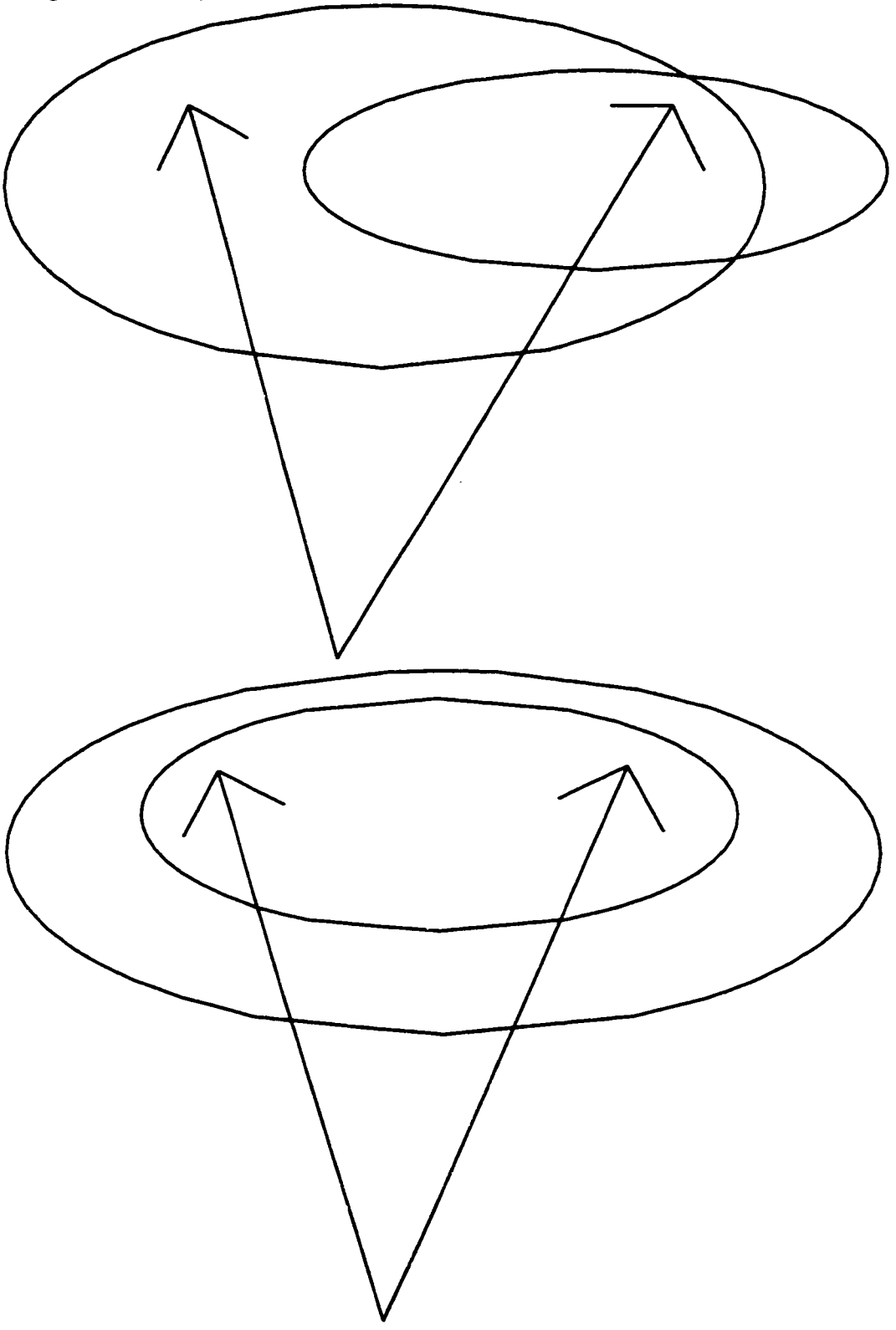
Once the necessary inclusion properties have been achieved, the next step is to find the optimal operating conditions. Two choices are available depending on the region of operability. If the optimal levels from the design matrix serve as boundary points for the region of operability, the exercise is complete. If they do not, the next step is to find that beta vector within the confidence region of the least biased beta vector which yields the maximum response, then use the method of Lagrange multipliers by fixing the input variables to a certain radius to find the optimal variable settings.

Example 3.1

$$X = \begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \quad Y = \begin{bmatrix} 20.21 \\ 21.40 \\ 20.89 \\ 19.62 \end{bmatrix}$$

In this experiment, the levels of X were chosen to be well within the operability limits of -1.4 and 2. By the method of least squares, B_0 is $\begin{bmatrix} -0.275 & -0.020 \end{bmatrix}'$.

Figure 3.1: Representation of Confidence Intervals Before and After Convergence



Using the methodology of Chapter 2, B_1 was derived to be $\begin{bmatrix} -0.6128 & -0.2719 \end{bmatrix}'$, using the least squares b_0 value of 20.53 for both vectors. The angle between the two vectors is 19.8°. The standard error of each coefficient in B_0 is 0.615. Since the difference between the coefficients of the two vectors is less than 0.615, B_1 satisfies the standard error criterion. In order to compute the confidence interval of B_1, we need the posterior covariance matrix, which, using 1000 values of Y, was computed to be

$$\Sigma = \begin{bmatrix} 170.8957 & 1.8235 \\ 1.8235 & 180.9369 \end{bmatrix} \text{x}10^{-5}$$

Confidence interval angles were computed up to a maximum of 180° using equation 3.6, which allows each hypercone boundary to be 90° from the vector in point.

B_0		B_1	
Level of Confidence	Angle	Level of Confidence	Angle
1%	4°	60%	9.99°
5%	20°	70%	14.26°
10%	41°	80%	22.45°
15%	65°	90%	47.06°
20%	92°	95%	106.93°
25%	135°	95.5%	126.54°
30%	180°	96%	180.00°

In the computations of the confidence intervals for B_1, s_b^2 is taken as $(170.8957 + 180.9169)/2 \text{ x } 10^{-5} = 175.9163 \text{ x } 10^{-5}$. This compares against 0.378225 for B_0. The squared magnitude of B_1 is 0.439829; that of B_0 is 0.076025. Since X has $n = 4$ rows and $k = 2$ columns, the degrees of freedom for the F statistic are $k - 1 = 1$ and $n - (k + 1) = 1$. Using the 70% confidence interval of B_1 as an example, $F_{0.7,1,1} = 3.85184$, with the resulting angle being

$$\theta = 2 \arcsin \sqrt{\frac{(0.001759163)(3.85184)}{0.439829}}$$

Suppose the experimenter chose a 70% level of confidence with corresponding angle 14.26°. Since the angle between the two vectors, 19.8°, is greater than 14.26°, the

first step was to converge B_1 towards B_0 using equation 3.5 so that the confidence region of B_1 would include B_0. This was carried out using $n = 1000$ with the resulting B_1 vector being $\begin{bmatrix} -0.3339 & -0.0644 \end{bmatrix}'$. The resulting angle with B_0 is 6.8°. The new 70% confidence region angle of B_1 is 13.695°. Once an acceptable confidence region is determined, the beta vector lying in the intersection of the two confidence regions yielding the maximum response within the region of operability using the method of Lagrange multipliers must be found. This is done first with B_0 and B_1 to indicate the general direction the experimenter should head in. The side condition for the Lagrange multiplier is that $x_1^2 + x_2^2 = 2$. This is to maintain consistency with the design matrix. The X vector within the region of operability producing the greatest response with B_0 is $\begin{bmatrix} -1.4 & -0.2 \end{bmatrix}$ with a response of 20.92. The corresponding X vector for B_1 is $\begin{bmatrix} -1.38862 & -0.26783 \end{bmatrix}$ with a response of 21.0109. This indicates that the experimenter should look in the direction of B_1 to find the true path of steepest ascent. The next step is to generate a set of n beta vectors, each lying inside the intersection of the confidence intervals of B_0 and B_1. The number n is arbitary; the goal is to converge towards the beta vector and, by extension, its corresponding levels in the design matrix which will produce the maximum response. We will call an arbitrary vector generated by this method B_2. If there are two input variables, the vector B_2 can be generated through the rotation

$$\begin{bmatrix} \cos(\alpha\theta) & -\sin(\alpha\theta) \\ \sin(\alpha\theta) & \cos(\alpha\theta) \end{bmatrix} B_1, \; 0 < |\alpha| < 1 \tag{3.8}$$

Note that this rotation preserves the magnitude of B_1.

Under the first order model scenario with two variables, it is necessary to produce $n = 2$ rotations only. The rationale is as follows.

We wish to find the maximum response under the optimal rotation $\theta \in (-\pi/2, \pi/2)$. Given a beta vector $b = \begin{bmatrix} b_1 & b_2 \end{bmatrix}$, we define the rotated

coefficients as

$$\begin{aligned}\hat{b}_1 &= \cos\theta b_1 - \sin\theta b_2 \\ \hat{b}_2 &= \sin\theta b_1 + \cos\theta b_2\end{aligned}$$

The model equation is

$$\begin{aligned}\hat{y} &= \hat{b}_1 x_1 + \hat{b}_2 x_2 \\ &= (\cos\theta b_1 - \sin\theta b_2)x_1 + (\sin\theta b_1 + \cos\theta b_2)x_2\end{aligned}$$

Computing the first derivative with respect to θ,

$$\frac{d\hat{y}}{d\theta} = (-\sin\theta b_1 - \cos\theta b_2)x_1 + (\cos\theta b_1 - \sin\theta b_2)x_2$$

Setting the derivative equal to zero,

$$\begin{aligned}\cos\theta(b_1x_2 - b_2x_1) &= \sin\theta(b_1x_1 + b_2x_2) \\ (b_1x_2 - b_2x_1) &= \tan\theta(b_1x_1 + b_2x_2) \\ \tan\theta &= \frac{b_1x_2 - b_2x_1}{b_1x_1 + b_2x_2} \\ \theta &= \arctan\frac{b_1x_2 - b_2x_1}{b_1x_1 + b_2x_2}\end{aligned}$$

Since $\tan(-\pi/2) = -\infty$ and $\tan(\pi/2) = \infty$, choosing points at the edge of the region of operability will produce the outermost angles. To ensure that the maximum has been computed, the second derivative must satisfy:

$$-\cos\theta b_1x_1 + \sin\theta b_2x_1 - \sin\theta b_1x_2 - \cos\theta b_2x_2 < 0$$

Thus, if a biased beta vector has total confidence region angle θ, it is necessary to produce only the two rotations $-\theta/2$ and $\theta/2$.

Example 3.1 continued: Following the above strategy, the B_2 vector $\begin{bmatrix} -0.3240 & -0.1033 \end{bmatrix}$ produced the maximum response of 20.957 when multiplied by $\begin{bmatrix} -1 & -1 \end{bmatrix}$ from the design matrix. Using Lagrange multipliers, the optimal X

vector was $\begin{bmatrix} -1.34779 & -0.42958 \end{bmatrix}$ producing a response of 21.011. The angle between B_1 and B_2 is 6.7592°, placing it within the B_1 70% confidence half angle of 6.8475°. The experimenter decided that the newly found X vector would serve as the optimal setting.

Figure 3.2 illustrates the results if 25 beta vectors had been generated within the 70% confidence interval of B_1. The x-axis of the first figure is the angle between B_1 and the generated B_2 vectors. The x-axis of the second figure is the angle of rotation of B_1. The second figure clearly demonstrates the monotonic nature of beta generation in the first order model.

As with any measure that involves sampling, the estimated response has a standard error given by

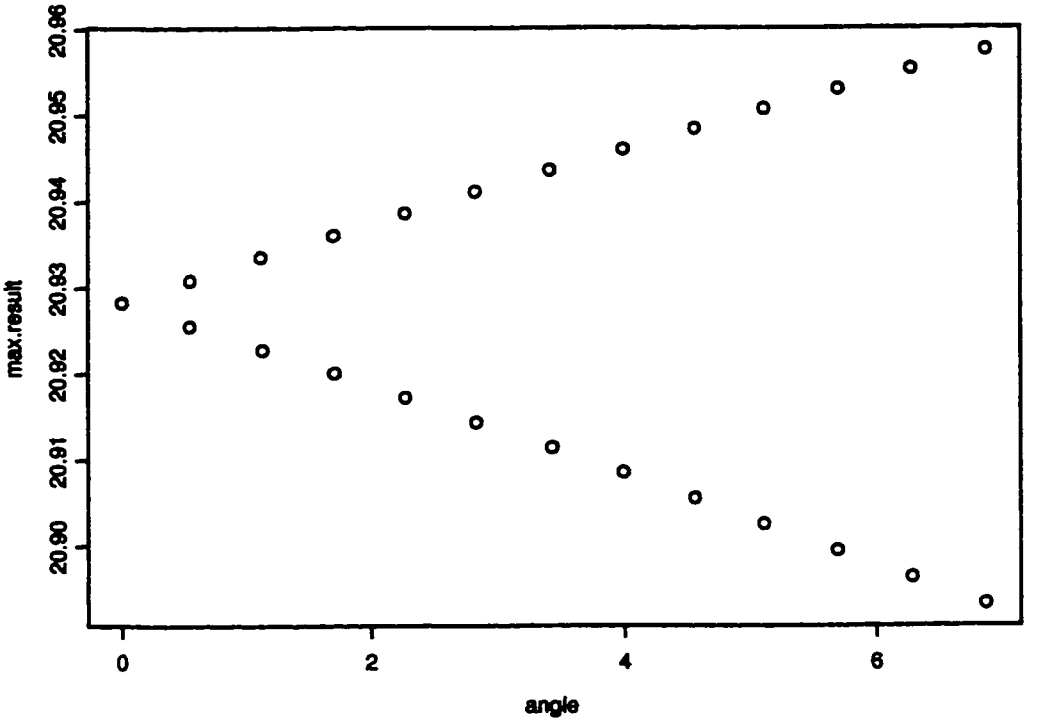
$$s_{\hat{y}(x)} = s\sqrt{x^{(m)\prime}(X'X + R)^{-1}x^{(m)}}$$

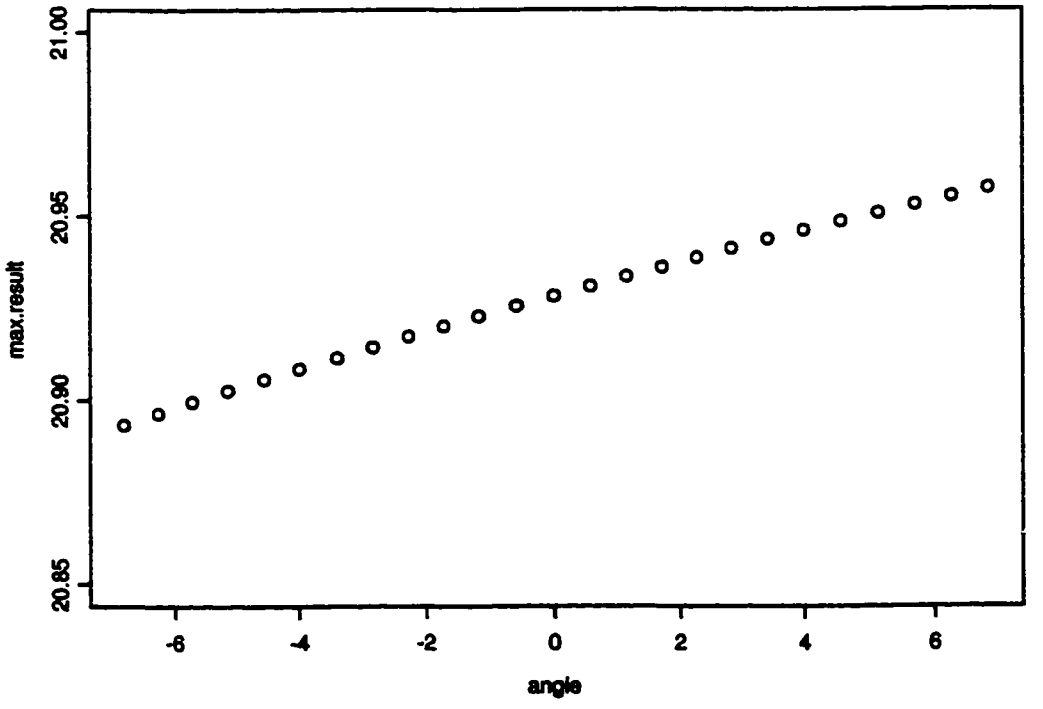
in which s is the square root of the mse and $x^{(m)}$ is the basis of the model equation which, in this case, is $\begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix}'$. This involves augmenting the design matrix by a column of ones. Given

$$\begin{aligned} \text{mse} &= 1.5594 \\ x^{(m)\prime} &= \begin{bmatrix} 1 & -1.34779 & -0.42958 \end{bmatrix}' \\ (X'X + R)^{-1} &= \begin{bmatrix} 73.63983 & 1.05101 & -0.41868 \\ 1.05101 & 78.32335 & -0.92352 \\ -0.41868 & -0.92352 & 84.56158 \end{bmatrix} \text{x}10^{-5} \end{aligned}$$

the estimated error of response is 0.06. It should be noted that $(XX + R)^{-1}$ is derived from the least biased beta vector before convergence towards the path of steepest ascent, so as to serve as a supremum of standard errors under the Bayesian regime. This may be compared with 1.07 for the non-Bayesian error of response using the least squares mse of 1.5129 in which $(X'X + R)^{-1}$ is replaced by $(X'X)^{-1}$.

Figure 3.2: Example 3.1: Angles of Generated Betas With Results





If the number of model parameters, p exceeds 2, the most convenient strategy is to fix $(p-2)$ parameters and rotate the other two through their common plane. We will refer to these beta parameters as b_1 and b_2. The primary consideration in this rotation is that the magnitude of the beta vector must be preserved. There are now two angles that relate to the biased vector, the internal angle ϕ of the rotation on the $b_1 b_2$ plane and the external angle θ which preserves the magnitude of the beta vector. The new b_1 and b_2 coefficients are derived through rotation of ϕ by (3.8), and the methodology continues as for the two-parameter model. The derivation of ϕ is as follows:

We will refer to the two vectors as B_1 and B_2, in which B_2 is B_1 after rotation through the angle θ. The angle θ can be thought of as the external angle of B_1 and B_2. Thus,

$$\begin{aligned}\theta &= \arccos\left[\frac{B_1 \cdot B_2}{\|B_1\|\|B_2\|}\right] \\ &= \arccos\left[\frac{B_1 \cdot B_2}{\|B_1\|^2}\right]\end{aligned}$$

since the magnitude of the two vectors is the same. Now,

$$\begin{aligned}B_1 \cdot B_2 &= \sum_{i=1}^{p} b_{1i} b_{2i} \\ &= \sum_{i=1}^{2} b_{1i} b_{2i} + \sum_{i=3}^{p} b_{1i}^2\end{aligned}$$

It follows that

$$\cos\theta \|B_1\|^2 = \sum_{i=1}^{2} b_{1i} b_{2i} + \sum_{i=3}^{p} b_{1i}^2$$

The rotation through ϕ affects only the parameters b_1 and b_2, so

$$\begin{bmatrix} b_{21} \\ b_{22} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix}$$

$$b_{21} = b_{11}\cos\phi - b_{12}\sin\phi$$

$$b_{22} = b_{11}\sin\phi + b_{12}\cos\phi$$

$$\begin{aligned}
\sum_{i=1}^{2} b_{1i}b_{2i} &= b_{11}[b_{11}\cos\phi - b_{12}\sin\phi] + b_{12}[b_{11}\sin\phi + b_{12}\cos\phi] \\
&= (b_{11}^2 + b_{12}^2)\cos\phi \\
\cos\phi &= \frac{\cos\theta\|B_1\|^2 - \sum_{i=3}^{p} b_{1i}^2}{\sum_{i=1}^{2} b_{1i}^2} \\
\phi &= \arccos\left[\frac{\cos\theta\|B_1\|^2 - \sum_{i=3}^{p} b_{1i}^2}{\sum_{i=1}^{2} b_{1i}^2}\right] \bullet \mathrm{sgn}(\theta)
\end{aligned}$$

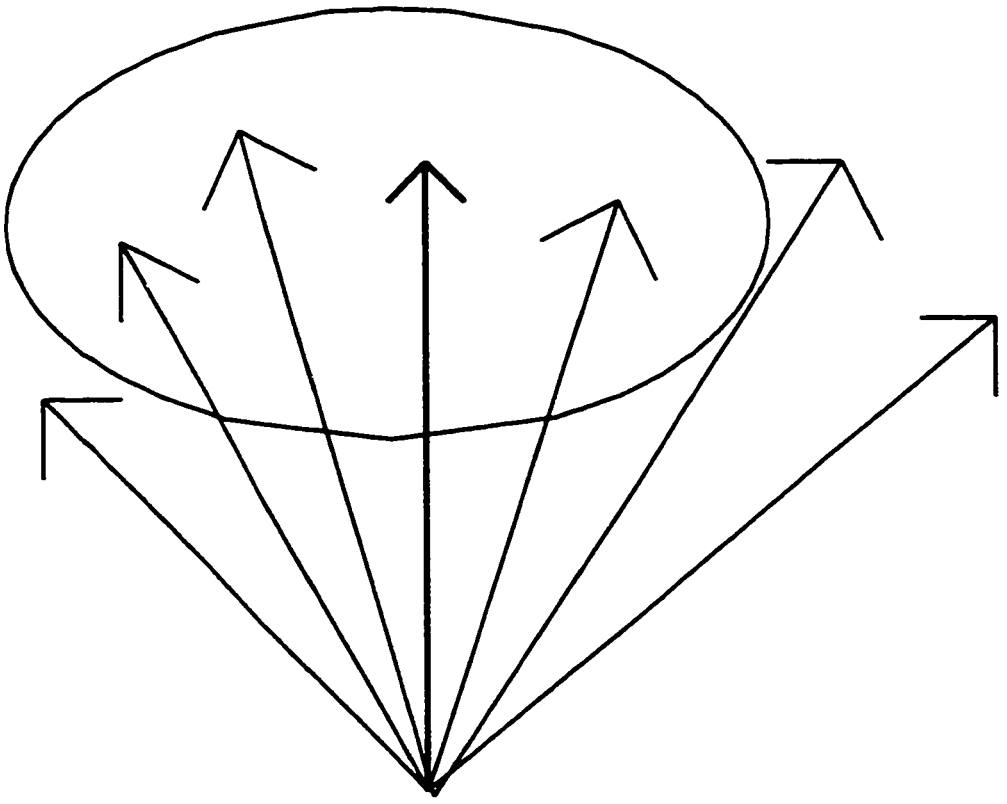
since $\cos(\theta) = \cos(-\theta)$. It should be noted that this method serves as an alternative to the Newton-Raphson method for solving systems of equations involving Lagrange multipliers in which an analytical solution is not possible or is difficult. It has the advantage that no derivatives need be calculated. This will prove particularly useful in higher order models.

The strategy for the rotation changes when there are more than two input variables. For the first rotation, it is still possible to generate only the two beta vectors at either extreme of the confidence interval hypercone as was done previously. However, subsequent beta vectors generated by rotating on a different plane must lie within the angle between the least biased beta vector and the most recently built B_2 vector yielding the currently maximum response under the least squares model. Of these, only those lying within the confidence interval hypercone of the least biased beta vector can be accepted. Figure 3.3 illustrates the general principle. The vector at 90° to the base represents the B_1 vector.

The first variables of choice are those which contribute the most to the variability of the response. However, if the experimenter feels that rotation of other variables will aid convergence towards the optimal beta vector, this should be done.

Example 3.2

Figure 3.3: Rotated Beta Vectors In A Higher Dimension



$$X = \begin{bmatrix} -1 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix} \quad Y = \begin{bmatrix} 2.95 \\ 3.84 \\ 5.12 \\ 2.87 \\ 5.89 \\ 6.41 \\ 4.05 \\ 5.26 \end{bmatrix}$$

Using 1000 simulated values of Y, the least squares beta vector, B_0, and the least biased beta vector, B_1, are:

$$B_0 = \begin{bmatrix} 0.85375 \\ -0.22375 \\ 0.04625 \end{bmatrix} \quad B_1 = \begin{bmatrix} 1.1213306 \\ -0.4988541 \\ 0.1187435 \end{bmatrix}$$

with a common intercept value of 4.54875. The standard error of each B_0 coefficient is 0.432, placing each B_1 coefficient within the standard error of its respective B_0

coefficient. The maximum response using the design matrix under B_0 is 5.6725, that under B_1, 6.2877. Their angle is 9.609°. The confidence intervals angles are:

B_0		B_1	
Level of Confidence	Angle	Level of Confidence	Angle
30%	51.18°	99%	2.63°
40%	63.65°	99.9%	4.86°
50%	77.98°	99.99%	8.74°
60%	96.36°	$(100 - 10^{-3})\%$	15.63°
70%	125.33°	$(100 - 10^{-4})\%$	28.01°
75%	155.70°	$(100 - 10^{-5})\%$	51.00°
80%	180.00°	$(100 - 10^{-7})\%$	180.00°

The experimenter decided upon a 99% confidence interval. The region of operability for the X values was the interval $\left(-1.1,\ \ 1.1 \right)$. Convergence towards B_0 produced the B_1 vector $\left[\begin{array}{ccc} 0.869827 & -0.239988 & 0.051356 \end{array} \right]'$ which has an angle of 0.78125° with B_0 and a confidence interval angle of 1.1998°. However, the use of Lagrange multipliers on both vectors produces levels of X outside the region of operability, for both the first and second coefficients, under the restriction that $x_1^2 + x_2^2 + x_3^2 = 3$, with the first component greater than 1.1 and the second component less than -1.1. Under the circumstances, the interim optimal X vector was chosen as $\left[\begin{array}{ccc} 1.1 & -1.1 & 0.76158 \end{array} \right]$ which we call xx. Using xx, the responses under B_0 and B_1 are, respectively, 5.7692 and 5.8087. This suggests searching for the true path of steepest ascent in the vicinity of B_1. Generating two beta vectors for the first rotation, followed by sets of 25 for subsequent rotations in the neighborhood of B_1, produced the following results:

First Beta	Second Beta	coefficients Rotated	Result With xx	New Beta	Angle with B_1
-	B_1	1,2	5.815902	B_2	0.5995902°
B_1	B_2	1,3	5.815935	B_3	0.5994507°
B_1	B_3	2,3	5.817526	B_4	0.5914114°

The B_4 vector is $\begin{bmatrix} 0.8673 & -0.2484 & 0.0547 \end{bmatrix}'$. Using xx, it produced the greatest response of the four beta vectors of 5.817526. If the experimenter chose settings away from the extremes of -1.1 and 1.1 for the first two components, B_4 still indicates the direction to achieve maximum response.

In the previous example, each rotation produced a new maximum response. This will not always be the case, particularly in higher order models. The experimenter may have multiple beta vectors producing the same maximum response, suggesting a hyperplane in the model. The experimenter may also rotate through fairly important coefficients with no improvement in the maximum response and then suddenly realize an improvement rotating through the plane of a relatively unimportant coefficient coupled with an important one. However, there is a way to determine the size of the angle which must be chosen in order to have an increased response regardless of the order of the model. Let

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + k$$

in which b_1 and b_2 are each the coefficient to be rotated of a quadratic, interaction or single exponent term. The terms x_1 and x_2 are chosen from the levels of the design matrix which contribute to the maximum response under the least squares model. The last term, k is the sum of the remaining terms. Let $\theta \in \left(0, \ \pi/2 \right)$ be the angle of rotation. Let

$$\hat{\hat{y}} = b_0 + (\cos\theta b_1 - \sin\theta b_2)x_1 + (\sin\theta b_1 + \cos\theta b_2)x_2 + k$$

If $\hat{\hat{y}} > \hat{y}$ for given values of x_1 and x_2, then

$$(\cos\theta b_1 - \sin\theta b_2)x_1 + (\sin\theta b_1 + \cos\theta b_2)x_2 > b_1 x_1 + b_2 x_2$$

$$\sin\theta(b_1 x_2 - b_2 x_1) > (1 - \cos\theta)(b_1 x_1 + b_2 x_2)$$

$$\sin\theta > \frac{(1 - \cos\theta)(b_1 x_1 + b_2 x_2)}{(b_1 x_2 - b_2 x_1)}$$

$$2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) > \frac{2\sin^2(\frac{\theta}{2})(b_1x_1+b_2x_2)}{(b_1x_2-b_2x_1)}$$

$$\tan\left(\frac{\theta}{2}\right) < \frac{b_1x_2-b_2x_1}{b_1x_1+b_2x_2}$$

$$\theta < 2\arctan\left(\frac{b_1x_2-b_2x_1}{b_1x_1+b_2x_2}\right)$$

provided $\frac{b_1x_2-b_2x_1}{b_1x_1+b_2x_2} > 0$. If this is otherwise or $\theta \in \left(-\pi/2, \ 0 \right)$, the inequality is reversed. Regardless of the method chosen, the reader can appreciate the time-consuming nature of this method if the number of input variables is large. It is compensated for by the fact that the experimenter can narrow the search for the true path of steepest ascent as much as need be. In the previous example, sets of 25 beta vectors were generated at a time. This is not to suggest that this is an optimal number of vectors to produce; the experimenter can have more or fewer, depending on the degree of accuracy needed to find the true path of steepest ascent.

3.2 Higher Order Model

If a goodness-of-fit test indicates that a higher order model is needed, it is necessary to provide an appropriate posterior least biased beta vector. The information required from the first order phase are the variable importance and their general region of interest. If the first order model was constructed as a screening experiment, then the second order experiment will usually have fewer variables. As well, using the region of interest, the range of the variables in the second order model will be narrower than that of the first order. If both a first order design matrix and a second order design matrix have the same variables at identical levels, the latter matrix will have a higher bias due to the fact that design matrices for higher order models do not fit into any of the three type of design matrices described in Chapter 2. Recall that the bias is zero if the biased vector is equal to the least squares beta vector since there is no adjustment between the prior and

posterior distribution of the biased vector. Thus, a larger bias indicates a larger angle between a biased vector and the least squares beta vector.

3.2.1 Methodology

In a higher order model, the beta vector can be split into two parts: the higher order component and the first order component. The construction of the least biased beta vector first involves building the first order component. We will now define B_0 as the least squares beta vector of known dimension, and B_1 as the least biased beta vector, a modification of the definition on page 46. To indicate the components we define

$$B_0 = \begin{bmatrix} B_{(1)} \\ B_{(2)} \end{bmatrix} \quad B_1 = \begin{bmatrix} B^*_{(1)} \\ B^*_{(2)} \end{bmatrix}$$

in which the subscripts indicate the order of the component.

There are two options for the construction: the experimenter can use the order importance from the first order experiment or the order importance from $B_{(1)}$. The first criterion to use for judging which vector to choose is that of the least squares standard error. This criterion is particularly useful if the order importance for constructing one vector differs from that of the other. If one vector satisfies the criterion and the other does not, the satisfactory vector is chosen. If both vectors satisfy the criterion, or neither does, whichever forms the smaller angle with $B_{(1)}$ should be chosen in order to minimize the level of confidence needed to contain $B_{(1)}$ inside the confidence interval region of $B^*_{(1)}$.

Before adding $B^*_{(2)}$ to the biased vector B_1, it may be necessary to converge $B^*_{(1)}$ towards $B_{(1)}$, as per equation 3.5. Once this is done, the higher order coefficients are added to the biased vector in such a way so as to satisfy the standard error criterion. If the angle between $B^*_{(1)}$ and $B_{(1)}$ is maintained, this guarantees that successive components can be added. The reasoning is as follows.

To simplify the equations we define

$$\begin{aligned} K &= \sum_{i=1}^{n-1} B_i b_i \\ J &= \sum_{i=1}^{n-1} B_i^2 \\ M &= \sum_{i=1}^{n-1} b_i^2 \end{aligned}$$

in which B_i is the i$^{\text{th}}$ coefficient of B_0 and b_i is the corresponding coefficient from B_1. If the angle between the two vectors up to $n-1$ is θ, then

$$\cos\theta = \frac{K}{\sqrt{JM}}$$

Since $\cos\theta \in \left(0, \ 1 \right)$, we have $JM \geq K^2$. With B_n known, if b_n is added so as to maintain θ, then,

$$\begin{aligned} \frac{K}{\sqrt{JM}} &= \frac{K + B_n b_n}{\sqrt{J + B_n^2}\sqrt{M + b_n^2}} \\ K^2(J + B_n^2)(M + b_n^2) &= JM(K + B_n b_n)^2 \\ (K^2 J + K^2 B_n^2 - JMB_n^2)b_n^2 - 2KJMB_n b_n + K^2 MB_n^2 &= 0 \end{aligned}$$

Solving for b_n, the requirement for a solution is that the discriminant be non-negative.

$$\begin{aligned} 4(KJMB_n)^2 - 4(K^2 MB_n^2)(K^2 J + K^2 B_n^2 - JMB_n^2) &\geq 0 \\ J^2 M - K^2 J - K^2 B_n^2 + JMB_n^2 &\geq 0 \\ (J + B_n^2)(JM - K^2) &\geq 0 \end{aligned}$$

which is a tautology since both $(J + B_n^2)$ and $(JM - K^2)$ are non-negative. Furthermore, if $|B_i - b_i| < \varepsilon$, $i = 1 \ldots n-1$, for an arbitrarily small ε, then $\cos\theta > \alpha$ for an arbitrarily large α. It is clear that $\varepsilon \to 0 \ \leftrightarrow \alpha \to 1$. Thus if the coefficient b_n is added to construct B_1, and the angle θ is maintained with $\cos\theta > \alpha$, then $|B_n - b_n| < \varepsilon$. If ε is chosen as a fraction of the infimum of the least square

beta vector standard errors, then the added components to B_1 will fall within their respective least squares standard errors. The methodology for solving for b_n is as follows.

For the sake of simplicity, we will relabel B_0 as B consisting of the coefficients $\left[\begin{array}{cccccccc} B_1 & B_2 & \dots & B_{n-1} & B_n & \dots & B_p \end{array}\right]$, b the vector B_1 at the state $\left[\begin{array}{cccccccc} b_1 & b_2 & \dots & b_{n-1} & 0 & \dots & 0 \end{array}\right]$, θ the angle between $\left[\begin{array}{cccc} B_1 & B_2 & \dots & B_{n-1} \end{array}\right]$ and $\left[\begin{array}{cccc} b_1 & b_2 & \dots & b_{n-1} \end{array}\right]$, and b_n the n^{th} coefficient to be added to b. Beginning with the dot product equation, we have:

$$\cos\theta = \frac{\sum_{i=1}^{n-1} B_i b_i + B_n b_n}{\sqrt{\sum_{i=1}^{n} B_i^2}\sqrt{\sum_{i=1}^{n-1} b_i^2 + b_n^2}}$$

$$\cos^2\theta = \frac{\left(\sum_{i=1}^{n-1} B_i b_i + B_n b_n\right)^2}{\sum_{i=1}^{n} B_i^2 \sum_{i=1}^{n-1} b_i^2 + b_n^2 \sum_{i=1}^{n} B_i^2}$$

$$\cos^2\theta \sum_{i=1}^{n} B_i^2 \sum_{i=1}^{n-1} b_i^2 + \cos^2\theta b_n^2 \sum_{i=1}^{n} B_i^2 = \left(\sum_{i=1}^{n-1} B_i b_i\right)^2 + 2B_n b_n \left(\sum_{i=1}^{n-1} B_i b_i\right) + B_n^2 b_n^2$$

We can now solve this quadratic equation in b_n. Whichever of the two solutions is closer to B_n is the coefficient of choice for b_n. Once the least biased beta vector has been constructed, the approach to finding the optimal conditions now parallels that of the first order model in that we rotate the biased vector along the plane of two of its coefficients. For higher order models, the most convenient method for doing this is to fix one beta vector and uniformly generate B_2 vectors in the plane of the chosen coefficients between the fixed vector and the vector of interest. The governing criteria are maintaining the variables within the region of operability and the radius of an X vector in which

$$radius(x_1 \dots x_k) = \sqrt{\sum_{i=1}^{k} x_i^2}$$

In comparing two X vectors, they must be compared in the same model and at either equal radii or equal responses. The goal is to minimize the radius while maximizing the response. If two X vectors produce the same response, but one has a shorter radius, the response of that vector will have a smaller standard error. The least squares model is used to compare X vectors, since this model is the standard by which to compute responses. As well, commercially available software, such as S-Plus with the DOX module, can provide optimal settings under the least squares model.

Once the set of B_2 vectors is generated, the design matrix can be used to find the optimal constrained X vector from the set. Let

$$f(x) = b_0 + b_1 x_1 + b_2 x_2 + c$$

be the least biased model in which b_1 and b_2 are the coefficients on whose plane the B_2 vectors will be generated, with x_1 and x_2 being the generalized X variables. They may be quadratic, interactive or single power terms. b_0 is the intercept term and c is the sum of the remaining terms. We define x_1^o and x_2^o as the specific values generated from the design matrix row which produces the maximum response with B_0 from all the rows in X. We label this row x^o. As well, $f_\theta(x)$ and $f_\phi(x)$ is $f(x)$ rotated through the angles θ and ϕ respectively, $\cos\theta > \cos\phi$. We have

$$\begin{aligned} f_\theta(x) &= b_0 + (\cos\theta b_1 - \sin\theta b_2)x_1 + (\sin\theta b_1 + \cos\theta b_2)x_2 + c \\ f_\phi(x) &= b_0 + (\cos\phi b_1 - \sin\phi b_2)x_1 + (\sin\phi b_1 + \cos\phi b_2)x_2 + c \end{aligned}$$

Suppose $f_\theta(x^o) > f_\phi(x^o)$. Then

$$\begin{aligned} & (\cos\theta b_1 - \sin\theta b_2)x_1^o + (\sin\theta b_1 + \cos\theta b_2)x_2^o \\ > \; & (\cos\phi b_1 - \sin\phi b_2)x_1^o + (\sin\phi b_1 + \cos\phi b_2)x_2^o \end{aligned}$$

Let x_1^o and x_2^o be subject to the constraint $(x_1^o)^2 + (x_2^o)^2 = k$. We will call the constrained optimum under f_θ and f_ϕ, x_θ^c and x_ϕ^c respectively. Under f_θ we have

$$x_{\theta.1}^c = \frac{\sqrt{k}(\cos\theta b_1 - \sin\theta b_2)}{\sqrt{b_1^2 + b_2^2}}$$
$$x_{\theta.2}^c = \frac{\sqrt{k}(\sin\theta b_1 + \cos\theta b_2)}{\sqrt{b_1^2 + b_2^2}}$$

with similar results for f_ϕ. Now

$$\begin{aligned} f(x_\theta^c) &= b_0 + b_1\left(\frac{\sqrt{k}(\cos\theta b_1 - \sin\theta b_2)}{\sqrt{b_1^2 + b_2^2}}\right) + b_2\left(\frac{\sqrt{k}(\sin\theta b_1 + \cos\theta b_2)}{\sqrt{b_1^2 + b_2^2}}\right) + c \\ &= b_0 + \frac{\sqrt{k}}{\sqrt{b_1^2 + b_2^2}}\cos\theta(b_1^2 + b_2^2) + c \end{aligned}$$

with the result that $f(x_\theta^c) > f(x_\phi^c)$ since $\cos\theta > \cos\phi$. Thus, the design matrix can be used to find the optimal constrained X vector from a set of B_2 vectors.

Example 3.3 The design matrix and response are in Appendix 3. This is Example 6.3 from Myers & Montgomery [14] in which a second order model is fit. Using S-Plus with the DOX module, a maximum response of 59.0027 was achieved with the X vector $\begin{bmatrix} 0.1378 & -1.4142 & -1.1976 & -0.2950 \end{bmatrix}$ which has a radius of 1.881544 used in conjunction with the least squares model. Our goal is to see if this can be improved on, by either achieving the same response using an X vector with a smaller radius, or achieving a greater response with an X vector at a radius of 1.881544. Since S-Plus uses the Newton-Raphson method to compute constrained optimum, the former goal appears to be the more attainable.

Using the 2^4 subset of the design matrix, the path of steepest ascent is $\begin{bmatrix} 5.11875 & -9.50625 & -1.21875 & -1.61875 \end{bmatrix}$. The variable importance from most important to least important is x_2, x_1, x_4, x_3 with x_1 positively correlated with the response variable and the others negatively correlated. Using this information, $B_{(1)}^*$

was constructed, using 1000 simulations of the Y vector, to be $\begin{bmatrix} 7.9747 & -14.4638 & -1.6541 & -3.1367 \end{bmatrix}$. Its angle with $B_{(1)}$ is 13.17°. However, $B_{(1)}$ is $\begin{bmatrix} 4.9435 & -8.7152 & 1.3019 & -1.5071 \end{bmatrix}$ suggesting that the variable importance remains the same, but the correlation of x_3 with the response is positive instead of negative. This information was used to construct another $B^*_{(1)}$, using the same 1000 simulations of the Y vector. The result was $\begin{bmatrix} 7.9747 & -14.4638 & 1.6541 & -3.1367 \end{bmatrix}$ Its angle with $B_{(1)}$ is 3.009°. Since its angle is the smaller of the two, it was chosen as the first order component of the least biased beta vector. However, the standard error of $B_{(1)}$ is 9.92194, placing the second coefficient of $B^*_{(1)}$ beyond the acceptable standard error limit. As well, the 99% confidence interval angle is 0.0971°. It was necessary to converge $B^*_{(1)}$ towards $B_{(1)}$ using equation 3.5 so as to satisfy both the standard error and confidence interval criteria. Choosing the 99% confidence level, $B^*_{(1)}$ after convergence was $\begin{bmatrix} 5.0179 & -8.8565 & 1.3158 & -1.5565 \end{bmatrix}$. Its angle with $B_{(1)}$ is 0.1467° while its confidence interval angle is 0.1507°. The remaining coefficients were then added to maintain the angle of 0.1467°. From this vector, the five most important beta coefficients, in order of magnitude are b_2, b_{23}, b_{44}, b_1 and b_{33}.

To begin, the maximum response under B_0 and B_1 with the design matrix are 52.49809 and 52.69549 respectively. We use 52.69549 as the interim design matrix maximum. We refer to subsequent rotated biased beta vectors as B_2. In order to find the optimal settings, each time B_2 was rotated, its maximum under the design matrix was computed and compared to the previous interim design matrix maximum. If it was greater, the optimal X vector was found for the model using the biased beta vector. This X vector was then used in the least squares model to find the response which would be compared to 59.0027 (see page 63) while computing its radius. It was decided to rotate on the $b_2 b_{23}$ plane since they are the two most

important beta coefficients. The following table shows the results.

First Beta	Second Beta	New Beta	Design Matrix Maximum	Least Squares Maximum
x_1	x_2	x_3	x_4	Radius
B_0	B_1	B_2	52.50025	59.0027
0.1380	$-\sqrt{2}$	-1.1974	-0.2950	1.881422
B_1	B_2	B_3	52.50068	59.0027
0.1380	$-\sqrt{2}$	-1.1973	-0.2950	1.881363
B_1	B_3	B_4	52.50167	59.0027
0.1381	$-\sqrt{2}$	-1.1971	-0.2949	1.881264
B_1	B_4	B_5	52.5035	59.0027
0.1383	$-\sqrt{2}$	-1.1968	-0.2949	1.88105
B_1	B_5	B_6	52.50732	59.0027
0.1388	$-\sqrt{2}$	-1.1961	-0.2947	1.880638
B_1	B_6	B_7	52.51496	59.00268
0.1396	$-\sqrt{2}$	-1.1948	-0.2944	1.879814
B_1	B_7	B_8	52.53009	59.00261
0.1413	$-\sqrt{2}$	-1.1922	-0.2937	1.878155

The reader will recall that the maximum response using the vector of observations with the least squares model was 59.0027 with a radius of 1.881544. The reader will have noticed that x_2 remained at $-\sqrt{2}$ throughout the rotation, which was the lower limit of the variables in the design matrix. The reader will note there are five X vectors all with radius less than 1.881544 achieving the response of 59.0027. At a radius of 1.8812, placing it in tbe middle of the radii of the X vectors generated above, the X vector $\begin{bmatrix} 0.1380 & -1.4140 & -1.1973 & -0.2950 \end{bmatrix}$ which was generated by the least squares model could produce a response of only 58.99885 compared to 59.00261 produced by $\begin{bmatrix} 0.1413 & -\sqrt{2} & -1.1922 & -0.2937 \end{bmatrix}$ at a radius of 1.878155. Of these two X vectors, the former was chosen as the optimal settings.

Computation of the standard error involves the $(X'X + R)^{-1}$ matrix from the second order model using the variable importance used in choosing the first order component of the biased vector, as well as the MSE of the least squares model since

the X vectors were compared on this model. Under this arrangement, the standard error of the optimal settings vector chosen above is 0.0031 under the Bayesian model and 12.49308 under the least squares model, where once again, $(X'X + R)^{-1}$ was replaced by $(X'X)^{-1}$.

3.3 Conclusion

The methods presented in this chapter should prove useful in situations in which a high degree of precision is required regardless of the design matrix used. It does have the drawback of being computationally intensive. Appendix 5 contains the S-Plus code used in computations. The code is easily translated into C/C++ to provide access to a wide variety of users.

Chapter 4

Extraordinary Circumstances

This chapter is concerned with the employment of the techniques introduced in Chapters 2 and 3 in three situations that may occasionally occur. The first is the case of multiple responses in RSM, the second situation concerns first-order models in RSM subject to a linear constraint, and the third is the determination of variable importance in linear regression.

4.1 Multiple Responses

It is not unusual in RSM for a study to have multiple response variables, all of which are measured simultaneously for each level of the design matrix. An example would be a chemical experiment in which the goal is to maximize the production of the desired product while minimizing the amount of a chemical by-product. We wish to develop a function which would take as the argument a multivariate response matrix and produce a single composite response vector upon which the analysis would be conducted, as in the previous chapters.

4.1.1 Development of Collapsing Function

In the development of the aforementioned function, two factors must be taken into consideration: the magnitude of the responses and the weighting of the individual

responses in the final response vector. An example of the former consideration is the case in which the range of the first response may be from 100 to 200, while that of the second may be from 0.5 to 1.0. We need to scale the responses so that the resulting scales of the two are similar. An example of the latter consideration may be the situation in which there are three responses, and the experimenter decides that the importance of the responses is in the ratio 2:1:1.

Scaling the results of a single response does not affect the bias, whether the response is multiplied by a scalar or raised to an exponent. However, the bias is affected when the biases from different responses with different optimization criteria are combined, such as when one response must be maximized while another must be minimized. We wish to develop a function that produces a response vector whose bias accurately reflects the best compromise of the individual responses while taking their relative importance into consideration.

Derringer and Suich [8] developed a desirability function for multiple responses that takes into consideration whether the experimenter wishes to maximize, minimize or target the individual responses. The experimenter chooses a lower limit A, an upper limit C, and a target value B. If the goal is to achieve the target B, the desirability measure d, with weights s and t, is defined as

$$d = \begin{cases} \left(\frac{\hat{y}-A}{B-A}\right)^s & , \quad A \leq \hat{y} \leq B \\ \left(\frac{\hat{y}-C}{B-C}\right)^t & , \quad B \leq \hat{y} \leq C \end{cases}$$

and zero otherwise. If the goal is to maximize the response, then

$$d = \begin{cases} \left(\frac{\hat{y}-A}{B-A}\right)^s & , \quad A \leq \hat{y} \leq B \\ 1 & , \quad \hat{y} > B \\ 0 & , \quad \hat{y} < A \end{cases}$$

If the goal is to minimize the response, then

$$d = \begin{cases} \left(\frac{\hat{y}-C}{B-C}\right)^s & , \quad B \leq \hat{y} \leq C \\ 1 & , \quad \hat{y} < B \\ 0 & , \quad \hat{y} > C \end{cases}$$

The weights s and t are chosen to indicate how important it is for the response to meet the target, with larger weights indicating greater importance. We will develop a similar approach that accommodates the methodology introduced in Chapters 2 and 3. The goal is to build a composite response that is to be maximized. This will be accomplished by transforming each response variable so that each transformed response variable is to be maximized. The composite response can be constructed by a linear combination of the transformed response variables.

In order to have different responses on the same scale while minimizing the subjectiveness of the method of Derringer and Suich, A and C will be, respectively, the minimum and maximum values of the response from the experiment. The following shows the appropriate transformation for each objective.

Objective	Transformation
Maximize	$\frac{C-y}{C-A}$
Minimize	$\frac{y-A}{C-A}$
Target B	$\frac{\lvert B-y\rvert}{C-A}$

If the experimenter has n transformed variables, $y_1 \ldots y_n$, and weights $w_1 \ldots w_n$, with $\sum_{i=1}^{n} w_i = 1$, then the composite response y_c is defined as

$$y_c = exp\left(-\sum_{i=1}^{n} w_i y_i\right) \tag{4.1}$$

The goal becomes to maximize y_c.

Example 4.1 This is example 6.6 from Myers and Montgomery [14]. The data is provided in Appendix 4. The goal is to maximize y_1 and have y_2 achieve a goal of 57.5. For y_1, $A = 80$ and $B = 97$. For y_2, $A = 55$ and $B = 60$. The following table shows the responses achieved under the Derringer-Suich function for permutations of the weights 0.1, 1.0 and 10.0. For y_2, s and t were chosen to be equal.

y_1 weight	y_2 weight	y_1	y_2
0.1	0.1	95.18	57.50
0.1	1.0	95.18	57.50
0.1	10.0	94.08	57.50
1.0	0.1	95.18	57.50
1.0	1.0	95.18	57.50
1.0	10.0	95.18	57.50
10.0	0.1	96.93	59.77
10.0	1.0	95.18	57.50
10.0	10.0	95.18	57.50

The second order models for y_1 and y_2 are:

$$\begin{aligned} y_1 &= 81.09 + 1.0284x_1 + 4.03x_2 + 6.2037x_3 \\ &- 1.8336x_1^2 + 2.9382x_2^2 - 5.1915x_3^2 \\ &+ 2.125x_1x_2 + 11.375x_1x_3 - 3.875x_2x_3 \\ y_2 &= 59.85 + 3.583x_1 + 0.2546x_2 + 2.2298x_3 \\ &+ 0.83479x_1^2 + 0.07484x_2^2 + 0.05716x_3^2 \\ &- 0.3875x_1x_2 - 0.375x_1x_3 + 0.3125x_2x_3 \end{aligned}$$

Under the seven weights producing the responses of 95.18 and 57.5 the settings from the second order models producing these responses simultaneously are $x_1 = -0.5486, x_2 = 1.682$ and $x_3 = -0.5955$.

Direct comparison with the procedure developed by this author is not possible due to the difference in the meaning of the weights. To illustrate the difference, we use the new method giving each of the two response variables equal weight. Under this scenario, $B_{(1)} = \begin{bmatrix} -0.0598 & 0.0378 & 0.0241 \end{bmatrix}$. The pseudo-weight vector was chosen to be $\begin{bmatrix} -1/2 & 1/11 & 1/97 \end{bmatrix}$ to reflect the range of the variables in the design matrix. Generating 1000 simulations of y_c, $B^*_{(1)} = \begin{bmatrix} -0.1484 & 0.0377 & 0.0282 \end{bmatrix}$. However the standard error of $B_{(1)}$ is 0.0006, placing the first and third coefficients of $B^*_{(1)}$ beyond the standard error range. After convergence, $B^*_{(1)} = \begin{bmatrix} -0.0603 & 0.0377 & 0.0241 \end{bmatrix}$. Its angle with $B_{(1)}$ is 0.2534° and has an 80%

confidence interval angle of 0.499°. Under both B_0 and B_1, the optimal settings were found to be $x_1 = -0.3268, x_2 = 1.682$ and $x_3 = -0.4177$, with $y_1 = 95.2776$ and $y_2 = 58.4748$. However, further rotations did not change the conditions found above, even if the standard errors of the coefficients were greater than the least squares standard errors. The following table shows B_0 with the tenth generated beta vector $B_{2.10}$ along with the upper and lower standard error limits of B_0.

B_0	Lower Limit	Upper Limit	$B_{2.10}$
-0.0598	-0.0605	-0.0592	-0.0549
0.0378	0.0372	0.0384	0.0394
0.0241	0.0235	0.0247	0.0263
-0.0660	-0.0664	-0.0656	-0.0649
0.0231	0.0227	0.0235	0.0263
-0.0788	-0.0792	-0.0783	-0.0784
0.0208	0.0198	0.0218	0.0162
0.0438	0.0427	0.0448	0.0427
-0.0449	-0.0460	-0.0439	-0.0533

If another experiment were conducted with the same design matrix and $B_{2.10}$ were the least squares beta vector, apportioning each response equal weight under the new method, the optimal operating conditions would remain the same. While it may be due to the robustness of the method, a more likely reason is the constraint in solving for simultaneous optimal solutions. The problem of finding settings which maximize one response while minimizing another is illustrated in the next example.

Example 4.2 The full data set is in Appendix 2. The reader may recognize the X variables as the Type II design from Example 2.3. The purpose of this experiment is to maximize y_1 and minimize y_2. The transformed response vector is $y.mix$. The prior expectation was that all three input variables were positively correlated with the new response variable. The following table shows the biases under various pseudo-weights.

x_1 pw	x_2 pw	x_3 pw	bias
1/2	1/3	1/101	117.771
1/2	1/101	1/5	103.8826
1/1001	1/10001	1/2	8.299978
1/10001	1/1001	1/2	8.299978
1/1001	1/10001	-1/2	382.9142
1/10001	1/1001	-1/2	382.9142

The results are that x_3 is the most important variable; its correlation with the combined response variable is positive. Since the design matrix is Type II, further analysis is required with x_1 and x_2. The interaction charts for these two variables at the x_3 Group III levels are shown in figure 4.1. The reader can appreciate that these charts are not open to easy interpretation. There is clearly strong interaction. The x_1 setting of 95 appears to have the least amount of variability. Its interaction with the x_2 setting of 5.5 produced the largest $y.mix$ response for the x_3 Group III level.

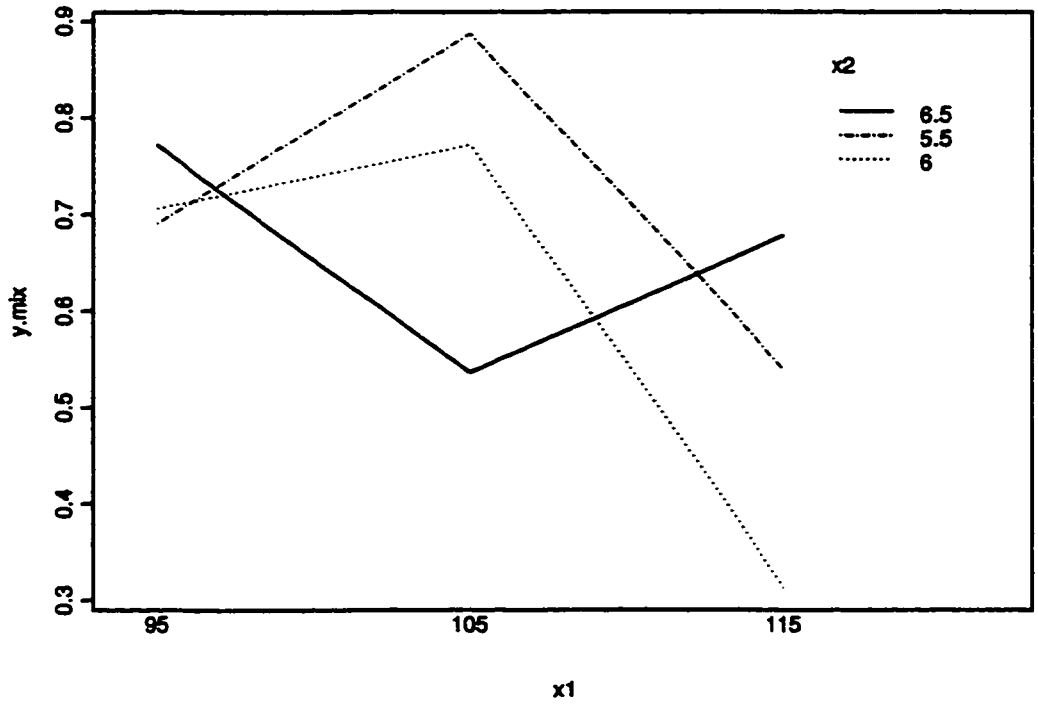
The nine settings of variable x_3 can be grouped into three groups: Group I, under 12, Group II, 12 to 20, and Group III, over 20. As there are nine replicates in the experiment, biases were computed before and after averaging the replicates. The top two pw were $pw_1 = \begin{bmatrix} 1/101 & 1/2 \end{bmatrix}'$ and $pw_2 = \begin{bmatrix} -1/2 & -1/3 \end{bmatrix}'$. The results for the three groups, as well as the group average are as follows:

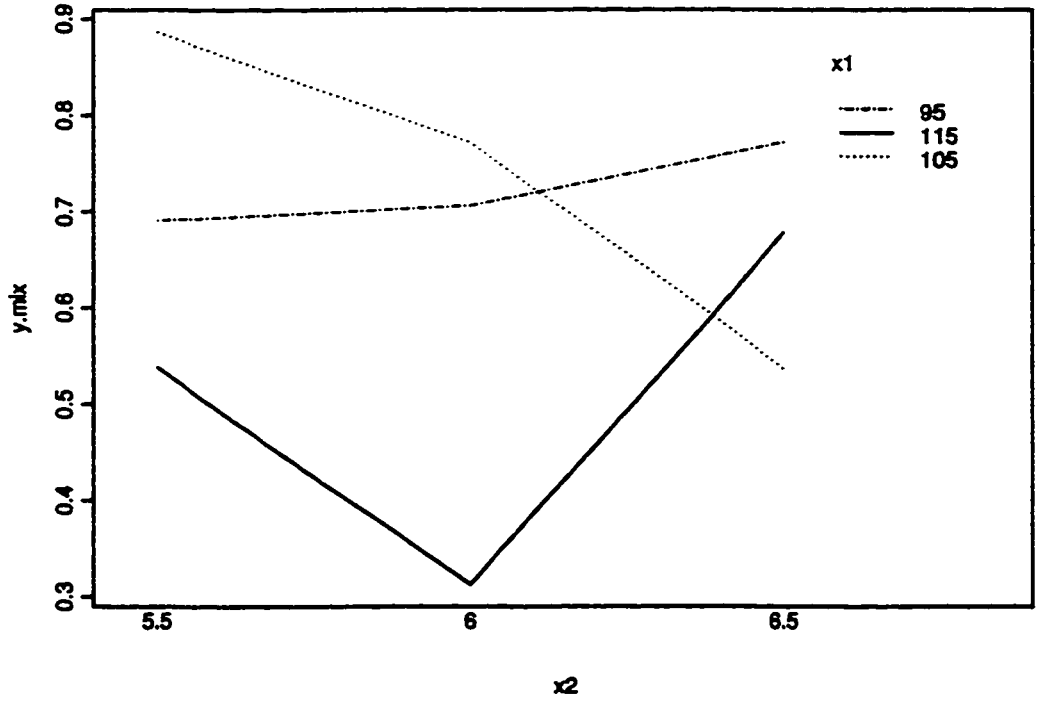
	Before Averaging Duplicates		After Averaging Duplicates	
	pw_1	pw_2	pw_1	pw_2
Group I	0.0098	0.0079	0.0387	0.0024
Group II	0.0289	0.4178	0.0524	0.1587
Group III	0.0870	11.9329	5.2339	10.3865
Group Average	0.0419	4.1195	1.775	3.5158

Based on the above analysis, the conclusion was that the three input variables were positively correlated with $x_3 \succ x_2 \succ x_1$.

The next step is to find the optimal operating conditions for the combined response vector. A second order model was chosen. The $(100 - 10^{-14})\%$ confidence

Figure 4.1: Example 4.2: Interaction of x_1 and x_2





interval angle for the least biased beta vector was less than 1° but the $(100 - 10^{-15})\%$ was 180°. This is due to the computer being restricted to 17 decimal places. The experimenter decided to focus attention on vectors within a certain radius of the least biased beta vector rather than a level of confidence to see if there would be any change in the optimal settings for the input variables at different radii. Using the methods of Chapter 3, optimal setting were sought for vectors within 3° and 6° of the least biased beta vector. In both cases the optimal settings were found to be 95 for x_1, 5.5 for x_2 and 24.3 for x_3, producing individual responses of 24.3 for y_1 and 0.2058 for y_2 under the individual least squares models. By way of comparision, the following table shows the optimal operating conditions and response for the individual models.

Response Variable	x_1	x_2	x_3	Optimal Response
y_1	115	6.5	24.3	25.01846
y_2	95	5.5	10.54	0.0739

4.2 Linear Constraint In First Order Model

If the path of steepest ascent in the first order model is subject to a linear constraint of the form

$$c_0 + c_1x_1 + c_2x_2 + \ldots + c_kx_k = 0$$

then it is intuitively clear that any biased beta vector should be subject to the same constraints. We are concerned with, in particular, the least biased beta vector.

There are two parameters of interest in a constrained linear vector: the point at which the beta vector is redirected, and the new direction. Using the linear model

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + \ldots + b_kx_k$$

the redirected beta vector starts at the point $[p_0 b_1, p_0 b_2, \ldots, p_0 b_k]'$ in which

$$p_0 = \frac{-c_0}{\sum_{i=1}^{k} b_i c_i} \tag{4.2}$$

and the direction is now $b_j - dc_j$ for $j = 1, 2, \ldots, k$ in which

$$d = \frac{\sum_{i=1}^{k} b_i c_i}{\sum_{i=1}^{k} c_i^2} \tag{4.3}$$

and the analysis continues as presented in earlier chapters.

Example 4.3

x_1	x_2	x_3	y
-1	-1	-1	255.5
-1	-1	1	387.5
-1	1	-1	421.0
-1	1	1	190.0
1	-1	-1	378.5
1	-1	1	209.0
1	1	-1	315.5
1	1	1	264.5

The region of operability is the interval $\left(-1.3, \quad 1.2 \right)$. We wish to build a first order model subject to the constraint that $x_1 + x_2 \geq -2$. Under this constraint, $c_0 = -2, c_1 = -1, c_2 = -1$ and $c_3 = 0$. The first order least squares model is

$$\hat{y} = 302.6875 - 10.8125x_1 - 4.9375x_2 - 39.9375x_3$$

Using equations 4.2 and 4.3 the beta vector redirects at the point $\left[-1.3732 \quad -0.6271 \quad -5.0721 \right]$ and heads in the direction $\left[-8.1875 \quad -2.3125 \quad -39.9375 \right]$.

The least biased beta vector before redirection and after convergence towards the least squares beta vector using 1000 simulated values of y is $\left[-11.04587 \quad -5.09425 \quad -40.45131 \right]$. Using this vector in place of the least squares beta vector, the vector redirects at $\left[-1.3686 \quad -0.6312 \quad -5.0119 \right]$ and

heads in the direction $\begin{bmatrix} -13.7359 & -2.4043 & -40.45131 \end{bmatrix}$. It should be noted that the angle between the two vectors increases from 0.1733° to 7.1588° after redirection. This, however, does not affect the confidence region angles of the vectors since the covariance matrices are not affected by redirected vectors.

At the point of redirection the least squares equation becomes

$$\hat{y} = 523.1985 - 8.1875x_1 - 2.3125x_2 - 39.9375x_3$$

and the least biased equation becomes

$$\hat{y} = 523.7583 - 13.7359x_1 - 2.4043x_2 - 40.45131x_3$$

The least squares constrained optimum within the region of operability is $x_1 = -1.1015, x_2 = 0.3111$ and $x_3 = -1.3$ producing a response of 584.8552. The corresponding settings for the biased model is $x_1 = -1.1274, x_2 = -0.1973$ and $x_3 = -1.3$ with a response of 592.3052. The standard error of the path of steepest ascent is 34.6289 placing the biased coefficients well within the standard error limit. Thus, the biased vector serves as a better choice of the two vectors for the true path of steepest ascent.

4.3 Variable Importance in Linear Regression

In the construction of a least squares model in linear regression, the variable selection process uses either the all-possible regression method or the stepwise method, depending on how many variables are involved. We present here an alternative to these methods using one-at-a-time pseudo-weights.

The reader will note that the solution for the least squares beta vector is the same for both linear regression and experimental design. The only difference is that the X matrix in linear regression does not fall into any of the three types of design matrices introduced in Chapter 2. However, this does not prohibit the experimenter

to determine variable importance. It simply must be performed one variable at a time, using a pw of +1 if the prior expectation is that the variable is positively correlated with the response, or -1 otherwise, and solving for the bias as previously. The theory is that more influential variables will have a smaller bias than those less so, provided the correlation is correctly determined. Similarly, if the correlation is wrongly opined, the bias for the more important variables may be greater than for those less important. Both sets of correlation may be needed in order to determine variable importance. The following example illustrates this need.

Example 4.4

x_1	x_2	x_3	x_4	y
78	48.2	2.75	29.5	57.5
69	45.5	2.15	26.3	52.8
77	46.3	4.41	32.2	61.3
88	49.0	5.52	36.5	67.0
67	43.0	3.21	27.2	53.5
80	48.0	4.32	27.7	62.7
74	48.0	2.31	28.3	56.2
94	53.0	4.30	30.3	68.5
102	58.0	3.71	28.7	69.2

This data is Example 12.9 in Walpole and Myers [20]. The least squares model is

$$\hat{y} = 7.1475 + 0.1000x_1 + 0.7264x_2 + 3.0758x_3 + 0.0300x_4$$

Using forward stepwise regression, x_1 and x_3 were found to be the most important variables. However, using the backward elimination procedure, x_2 and x_3 were considered the most important. The respective R^2 values for each of the two models are 0.9882 and 0.9905, which does not indicate a superiority of one pair of variables over another.

Using the one-at-a-time pseudo-weight procedure, the following biases were produced using 1000 simulated values of the response:

pw	x_1	x_2	x_3	x_4
+1	0.0004	1.3549	0.9808	0.3129
-1	20.52317	9.1548	74.8742	6.4335

It is clear that all the variables are positively correlated with the response, with x_1 the most important due to the low bias value of 0.0004. One might be inclined to choose x_4 as the next most important variable since it has the second lowest bias value for positive correlation. However, inspection of the negative correlation biases reveals that x_3 is the next most important due to the magnitude of its bias compared to that of x_4. Based on this analysis, x_1 and x_3 are chosen as the most important variables in a subset regression.

Chapter 5

Conclusion and Further Work

Pseudo-Bayesian response surface analysis was developed to allow experimenters to analyze their results under a wide variety of experimental conditions. No prior manufacturing history is required and it can be conducted on any type of design matrix. Its two primary goals are to determine input variable importance and their settings to achieve the optimal response.

The former goal is achieved by introducing a unique type of bias. This bias is derived from the prior beta vector θ and prior precision matrix R, both of which are determined by the prior expectation of how the input variables are ordered and correlated with the response. This information is carried in a new type of vector called a pseudo-weight. The goal then narrows to finding the pseudo-weight which has the least amount of bias associated with it.

The latter goal is achieved by rotating the biased vector within the region of operability to seek improved responses with the design matrix. Each time this is done, the constrained optimum under the biased vector is determined and then used in the least squares model to note any improvement. This method serves as an alternative to the Newton-Raphson method which is the common method used by computer programs. At present, any improvement provided by the new methods presented in this thesis over established methods are marginal at best.

Future work could involve improvements to the process to achieve better operating conditions under which to find optimal responses. Different types of design matrices have different regions of operability and interest. An area of further research would examine whether pseudo-Bayesian RSM achieves better results under certain types of matrices than with others.The last chapter showed how to use pseudo-weights to determine variable importance in simple linear regression; perhaps this method could be applied to other regression model types.

Bayesian response surface analysis is still in its infancy. It is this author's hope that it will contribute much in helping engineers, scientists and statisticians analyse experimental data in the years to come.

Bibliography

[1] Bernardo, J.M. (1979), "Expected Information As Expected Utility," *The Annals Of Statistics*, 7, 686-690.

[2] Box, G.E.P., and Draper, N.R. (1965), "The Bayesian Estimation Of Common Parameters From Several Responses," *Biometrika*, 52, 355-365.

[3] Box, G.E.P., and Draper, N.R. (1987), *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, New York.

[4] Chaloner, Kathryn (1984), "Optimal Bayesian Experimental Design For Linear Models," *The Annals Of Statistics*, 12, 283-300.

[5] Chaloner, Kathryn (1996), "Elicitation Of Prior Distributions," *Bayesian Biostatistics* edited by D.A. Berry and D.K. Stangl, Marcel Dekker, Inc., New York, 141-156.

[6] Chaloner, K., and Verdinelli, I. (1995) "Bayesian Experimental Design: A Review," *Statistical Science*, 10, 237- 304.

[7] Draper, N.R., and Hunter, W.G. (1967), "The Use Of Prior Distributions In The Design of Experiments For Parameter Estimation In Non-Linear Situations," *Biometrika*, 54, 147-153.

[8] Derringer, G., and Suich, R. (1980), "Simultaneous Optimization Of Several Response Variables," *Journal Of Quality Technology*, 12, 214-219.

[9] El-Krunz, S.M., and Studden, W.J. (1991), "Bayesian Optimal Designs For Linear Regression Models," *The Annals Of Statistics*, 19, 2183-2208.

[10] Garthwaite, H., and Dickey, J.M. (1988), "Quantifying Expert Opinion In Linear Regression Problems," *Journal Of The Royal Statistical Society, Series B*, 50, 462-474.

[11] Guttman, I., and Menzefricke, U. (1994), "Prediction Based On Response Surface Data Obtained With Random Blocking," *Test*, 3, 87-99.

[12] Kadane, J.B., Dickey, J.M., Winkler, R.L., Smith, D.S., and Peters, S.C. (1980), "Interactive Elicitation Of Opinion For A Normal Linear Model," *Journal Of The American Statistical Association*, 75, 845-854.

[13] Lindley, D.V. (1956), "On A Measure Of The Information Provided By An Experiment," *The Annals Of Statistics*, 27, 986- 1005.

[14] Myers, R.H., and Montgomery, D.C. (1995), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York.

[15] Oman, S.D. (1985), "Specifying A Prior Distribution In Structured Regression Problems," *Journal Of The American Statistical Association*, 80, 190-195.

[16] Toman, Blaza (1994), "Bayes Optimal Designs For Two- and Three-Level Factorial Experiments," *Journal of the American Statistical Association*, 89, 937-946.

[17] Pukelsheim, Friedrich (1980), "On Linear Regression Designs Which maximize Information," *Journal Of Statistical Planning And Inference*, 4, 339-364.

[18] Verdinelli, Isabella (1996), "Bayesian Design For The Normal Linear Model With Unknown Error Variance," *Technical Report 647*, Carnegie Mellon University, Pittsburgh, PA, U.S.A., and University of Rome, Italy.

[19] Verdinelli, I., and Kadane, J.B. (1992), "Bayesian Designs For Maximizing Information and Outcome," *Journal Of The American Statistical Association*, 87, 510-515.

[20] Walpole, R.E., and Myers, R.H. (1993), *Probability and Statistics For Engineers and Scientists*, fifth edition, Prentice Hall, Englewood Cliffs, N.J.

Appendix 1

If $B = (X'X)^{-1}(X'Y)$,

then $X'XB = X'Y$

and $Y = XB$, since X is a non-zero matrix with linearly independent columns.

Similarly, the following deductions:

$$\begin{aligned} B &= (X'X + R)^{-1}(X'Y + R\theta) \\ (X'X + R)B &= X'Y + R\theta \\ X'XB + R(B - \theta) &= X'Y \\ X'XB + X'X(X'X)^{-1}R(B - \theta) &= X'Y \\ Y &= XB + X(X'X)^{-1}R(B - \theta) \end{aligned}$$

Appendix 2

Data for Examples 2.3 and 4.2

x_1	x_2	x_3	y_1	y_2	$y.mix$
95	5.5	10.77	9.89	0.07	0.4147
95	5.5	10.77	9.44	0.07	0.4044
95	6.0	10.77	7.74	0.25	0.2652
95	6.5	10.77	8.33	0.07	0.3802
95	5.5	17.66	17.32	0.16	0.5328
95	6.0	17.66	17.96	0.16	0.5521
95	6.0	17.66	18.44	0.17	0.5569
95	6.5	17.66	18.18	0.16	0.5589
95	5.5	24.30	23.93	0.22	0.6905
95	6.0	24.30	25.30	0.25	0.7058
95	6.5	24.30	24.09	0.12	0.8357
95	6.5	24.30	25.04	0.24	0.7084
105	5.5	10.54	9.48	0.16	0.3442
105	5.5	10.54	9.11	0.18	0.3251
105	6.0	10.54	9.48	0.16	0.3442
105	6.5	10.54	10.06	0.21	0.3246
105	5.5	17.40	18.90	0.26	0.4851
105	6.0	17.40	16.91	0.25	0.4421
105	6.0	17.40	16.77	0.25	0.4387
105	6.5	17.40	17.07	0.45	0.3101
105	5.5	23.98	23.84	0.08	0.8863
105	6.0	23.98	23.96	0.16	0.7714
105	6.5	23.98	24.48	0.36	0.5520
105	6.5	23.98	25.68	0.43	0.5197
115	5.5	10.69	8.99	0.37	0.2286
115	5.5	10.69	10.59	0.29	0.2891
115	6.0	10.69	8.02	0.31	0.2415
115	6.5	10.69	9.74	0.62	0.1513
115	5.5	16.45	17.57	0.25	0.4587
115	6.0	16.45	18.28	0.23	0.4949
115	6.0	16.45	18.42	0.24	0.4898
115	6.5	16.45	18.39	0.39	0.3723
115	5.5	22.56	21.75	0.29	0.5384
115	6.0	22.56	22.75	0.62	0.3124
115	6.5	22.56	23.30	0.28	0.5978
115	6.5	22.56	23.33	0.15	0.7585

Appendix 3

Data for Example 3.3

$$
X = \begin{bmatrix}
-1 & -1 & -1 & -1 \\
-1 & -1 & -1 & 1 \\
-1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 \\
-1 & 1 & -1 & -1 \\
-1 & 1 & -1 & 1 \\
-1 & 1 & 1 & -1 \\
-1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 \\
1 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 \\
1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
1 & 1 & -1 & 1 \\
1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 \\
-1.41 & 0 & 0 & 0 \\
1.41 & 0 & 0 & 0 \\
0 & -1.41 & 0 & 0 \\
0 & 1.41 & 0 & 0 \\
0 & 0 & -1.41 & 0 \\
0 & 0 & 1.41 & 0 \\
0 & 0 & 0 & -1.41 \\
0 & 0 & 0 & 1.41 \\
0 & 0 & 0 & 0
\end{bmatrix}
\quad
Y = \begin{bmatrix}
58.2 \\
23.4 \\
21.9 \\
21.8 \\
14.3 \\
6.3 \\
4.5 \\
21.8 \\
46.7 \\
53.2 \\
23.7 \\
40.3 \\
7.5 \\
13.3 \\
49.3 \\
20.1 \\
31.1 \\
43.1 \\
49.9 \\
34.2 \\
17.5 \\
49.7 \\
31.1 \\
28.1 \\
32.8
\end{bmatrix}
$$

Appendix 4

Data for Example 4.1

x_1	x_2	x_3	y_1	y_2
−1.000	−1.000	−1.000	74	53.2
1.000	−1.000	−1.000	51	62.9
−1.000	1.000	−1.000	88	53.4
1.000	1.000	−1.000	70	62.6
−1.000	−1.000	1.000	71	57.3
1.000	−1.000	1.000	90	67.9
−1.000	1.000	1.000	66	59.8
1.000	1.000	1.000	97	67.8
−1.682	0.000	0.000	76	59.1
1.682	0.000	0.000	79	65.9
0.000	−1.682	0.000	85	60.0
0.000	1.682	0.000	97	60.7
0.000	0.000	−1.682	55	57.4
0.000	0.000	1.682	81	63.2
0.000	0.000	0.000	81	59.2
0.000	0.000	0.000	75	60.4
0.000	0.000	0.000	76	59.1
0.000	0.000	0.000	83	60.6
0.000	0.000	0.000	80	60.8
0.000	0.000	0.000	91	58.9

Appendix 5

S-Plus code used in the text

add.component

```
function(x, beta, beta1, alpha)
{
# x is the design matrix
# beta is the least squares beta vector
# beta1 is the least biased beta vector
# alpha is the dot product angle between the two vectors
# the purpose of this function is to add components
# to the beta1 vector until it is the same length
# as beta while maintaining the dot product angle
# between them
# this uses the function beta.sum
     alpha <- alpha * (pi/180)
     beta.new <- beta1
     k <- nrow(beta.new)
     n <- nrow(beta)
     yhat <- x %*% beta
     for(i in (k + 1):n) {
          blob1 <- sum(beta[1:(i - 1)] * beta.new)
          blob2 <- (cos(alpha))^2 * beta.sum(beta[1:i])
          blob3 <- blob2 - (beta[i])^2
          blob4 <- blob2 * beta.sum(beta.new) - (blob1)^2
          blob5 <- blob1 * beta[i]
          b1 <- (blob5 - sqrt(blob5^2 - blob3 * blob4))/blob3
          b2 <- (blob5 + sqrt(blob5^2 - blob3 * blob4))/blob3
          test1 <- rbind(beta.new, b1)
          test2 <- rbind(beta.new, b2)
          y1 <- x[, c(1:i)] %*% test1
          y2 <- x[, c(1:i)] %*% test2
          d1 <- sum((yhat - y1)^2)
          d2 <- sum((yhat - y2)^2)
          if(d1 < d2)
               beta.new <- test1
          else beta.new <- test2
     }
     beta.new
}
```

---

best.beta

```
function(x, beta, beta.array, n)
{
# x is the fully expanded design matrix
# beta is the least biased beta vector
# beta.array is an array of beta vector
# generated by either beta.generate or
# beta.generate2
```

```
# b0 is the least squares intercept term
# n is the number of beta vectors in beta.array
# this uses the function dot.product.angle
     max.result <- matrix(0, nrow = n)
     for(i in 1:n)
          max.result[i] <- max(x %*% beta.array[,  , i])
     no1 <- seq(along = max.result)[max.result == max(max.result)]
     if(length(no1) == 1)
          beta.array[,  , no1]
     else {
          no2 <- length(no1)
          angle <- matrix(0, nrow = no2)
          for(i in 1:no2)
               angle[i] <- dot.product.angle(beta,
                                beta.array[,  , no1[i]])
          k <- seq(along = angle)[angle == min(angle)]
          beta.array[,  , no1[k]]
     }
}
```

---

best.x

```
function(x, y, beta)
{
# x is the design matrix
# y is the vector of observations
# beta is a beta vector from which we want to
# find the optimal x settings under the least
# squares model
# it uses the following functions:
# beta.prior
# beta.sum
# order.make
     module(dox)
     k <- nrow(beta)
     n <- ncol(x)
     test.x <- order.make(x[, 1], x[, 2], x[, 3])
     x.one <- cbind(rep(1, nrow(x)), test.x)
     full.beta <- beta.prior(x.one, y)
     beta0 <- full.beta[c(2:(k + 1))]
     b0 <- full.beta[1]
     test.y <- test.x %*% beta0 + b0
     test.df <- cbind(x, test.y)
     rsm.result <- rsm.lm(test.df)
     test <- optim.rsm.lm(rsm.result)
     xx <- matrix(test$optim[c(1:4)], nrow = 1)
     radius <- sqrt(beta.sum(t(xx)))
     xx <- order.make(xx[, 1], xx[, 2], xx[, 3])
     result <- xx %*% beta0 + b0
     list(x = xx[c(1:n)], y = result, radius = radius)
}
```

beta.converge

```
function(beta0, beta1, x, y, cv, mse, alpha)
{
# beta0 is the least squares beta vector
# beta1 is a biased vector
# x is the design matrix
# y is the vector of responses
# cv is the posterior covariance matrix
# mse is the least squares mean square error
# alpha is the level of confidence
# the purpose of the function is to converge the
# beta1 vector towards the beta0 vector so that
# the ci angle of beta1 is greater than the dot product
# angle between beta0 and beta1
# it also produces the corresponding covariance of the new beta vector
# this uses the following functions:
# beta.n
# ci.angle
# cv.n
# dot.product.angle
     x <- as.matrix(x)
     y <- as.matrix(y)
     dot.angle <- dot.product.angle(beta0, beta1)
     beta1.angle <- ci.angle(x, beta1, cv, alpha)
     n <- flag <- 0
     while(flag < 1) {
          if(beta1.angle > 2*dot.angle)
               flag <- 1
          else {
               n <- n + 1
               beta1 <- beta.n(x, y, beta1, cv, mse, 1000*n)
               cv1 <- cv.n(x, cv, mse, 1000*n)
               beta1.angle <- ci.angle(x, beta1, cv1, alpha)
               dot.angle <- dot.product.angle(beta0, beta1)
          }
     }
     return(beta1, cv1)
}
```

beta.generate

```
function(beta, theta, b1, b2, n)
{
# beta is the beta vector
# theta is the total confidence interval angle in degrees
# b1 and b2 are the two parameters of interest
# with b1 < b2
# to rotate while the other parameters remain fixed
# n is the number of beta vectors to generate
# that lie within theta/2 either clockwise or counterclockwise
```

```
# it uses the following functions:
# beta.sum
# dot.product.angle
     theta <- (theta * pi)/180
     k <- nrow(beta)
     beta.rotate <- matrix(beta[c(b1, b2)])
     if(k > 2)
          beta.fix <- matrix(beta[ - c(b1, b2)])
     beta.array <- array(beta, dim = c(k, 1, n))
     angle <- seq( - theta/2, theta/2, length = n)
     for(i in 1:n) {
          if(k == 2)
               u <- angle[i]
          else {
               theta.r <- angle[i]
               blob <- (cos(theta.r) * beta.sum(beta) -
                         beta.sum(beta.fix))/beta.sum(beta.rotate)
               if(blob < 0 | blob > 1)
                    u <- 0
               else u <- acos(blob) * sign(theta.r)
          }
          if(b1 * b2 > 0)
               u.matrix <- matrix(c(cos(u), sin(u),
                           - sin(u), cos(u)), nrow = 2)
          else u.matrix <- matrix(c(cos(u),  - sin(u),
                            sin(u), cos(u)), nrow = 2)
          result <- u.matrix %*% beta.rotate
          beta.array[c(b1, b2),  , i] <- round(result, 4)
     }
     beta.array
}
```

---

beta.generate2

```
function(beta.f, beta.r, b1, b2, n)
{
# beta.f is the fixed beta vector
# beta.r is the rotated beta vector
# b1 and b2 are the two parameters of interest
# with b1 < b2
# to rotate while the other parameters remain fixed
# n is the number of beta vectors to generate
# it uses the following functions:
# beta.sum
# dot.product.angle
     theta <- (dot.product.angle(beta.f, beta.r) * pi)/180
     k <- nrow(beta.f)
     beta.rotate <- matrix(beta.r[c(b1, b2)])
     if(k > 2)
          beta.fix <- matrix(beta.r[ - c(b1, b2)])
     beta.array <- array(beta.r, dim = c(k, 1, n))
```

```
     angle <- seq(theta/n, theta, theta/n)
     for(i in 1:n) {
          if(k == 2)
               u <- angle[i]
          else {
          theta.r <- angle[i]
          blob <- (cos(theta.r) * beta.sum(beta.r) -
                    beta.sum(beta.fix)/beta.sum(beta.rotate)
          if(blob < 0 | blob > 1)
               u <- 0
          else u <- acos(blob)
     }
     if(b1 * b2 > 0)
          u.matrix <- matrix(c(cos(u), sin(u),
                         - sin(u), cos(u)), nrow = 2)
     else u.matrix <- matrix(c(cos(u),  - sin(u),
                         sin(u), cos(u)), nrow = 2)
     result <- u.matrix %*% beta.rotate
     beta.array[c(b1, b2),  , i] <- round(result, 4)
     }
     beta.array
}
```

---

beta.n

```
function(x, y, beta, cv, mse, n)
{
# x is the design matrix
# y is the observed y vector
# beta is the prior beta vector
# cv is the prior covariance matrix
# mse is the mean squared error
# n is the number of iterations to perform
     solve(n * t(x) %*% x + mse * solve(cv)) %*% (n * t(x) %*% y
       + mse * solve(cv) %*% beta)
}
```

---

beta.prior

```
function(x, y)
{
# x is the design matrix
# y is the vector of y values
     solve(t(x) %*% x) %*% t(x) %*% y
}
```

---

beta.solve

```
function(x, y, r, b)
{
```

```
# x is the design matrix
# y is the observed y vector
# r is the prior beta covariance matrix
# b is the prior beta vector
     solve(t(x) %*% x + r) %*% (t(x) %*% y + r %*% b)
}
```

---

beta.sum

```
function(beta)
     t(beta) %*% beta
```

---

ci.angle

```
function(x, beta, cv, alpha)
{
# x is the design matrix
# beta is the beta vector
# cv is the beta covariance matrix
# alpha is the level of confidence
# this function uses the following functions:
# sum.diag
# beta.sum
     k <- ncol(x)
     n <- nrow(x)
     sb <- sum.diag(cv)
     beta2 <- beta.sum(beta)
     f.value <- qf(alpha, k - 1, n - k - 1)
     chunk <- ((k - 1) * sb * f.value)/beta2
     if(chunk > 1)
          180
     else (360/pi) * asin(sqrt(chunk))
}
```

---

collapse

```
function(x)
{
# x is a data frame in which the last column is the response
# it returns a data frame in which the rows are averaged
# for rows of equal input variables
     x <- as.matrix(x)
     n <- nrow(x)
     k <- ncol(x) - 1
     collapsed.df <- average.df <- NULL
     flag <- 0
     for(i in 2:n) {
          tf <- all(x[(i - 1), c(1:k)] == x[i, c(1:k)])
          if(tf == T & i < n) {
               average.df <- rbind(average.df, x[(i - 1),  ])
```

```
            flag <- 1
        }
        if(tf == T & i == n) {
            average.df <- rbind(average.df, x[(i - 1),  ])
            average.df <- rbind(average.df, x[i,  ])
            average.df <- apply(average.df, 2, mean)
            collapsed.df <- rbind(collapsed.df, average.df)
        }
        if(tf == F & flag == 0 & i < n)
            collapsed.df <- rbind(collapsed.df, x[(i - 1),  ])
        if(tf == F & flag == 0 & i == n) {
            collapsed.df <- rbind(collapsed.df, x[(i - 1),  ])
            collapsed.df <- rbind(collapsed.df, x[i,  ])
        }
        if(tf == F & flag == 1 & i < n) {
            average.df <- rbind(average.df, x[(i - 1),  ])
            average.df <- apply(average.df, 2, mean)
            collapsed.df <- rbind(collapsed.df, average.df)
            flag <- 0
            average.df <- NULL
        }
        if(tf == F & flag == 1 & i == n) {
            average.df <- rbind(average.df, x[(i - 1),  ])
            average.df <- apply(average.df, 2, mean)
            collapsed.df <- rbind(collapsed.df, average.df)
            collapsed.df <- rbind(collapsed.df, x[i,  ])
        }
    }
    data.frame(collapsed.df)
}
```

---

constrained

```
function(x, y, x.c)
{
# x is the design matrix
# y is the vector of observations
# x.c is a (p+1) vector in which the first p terms
# are the coefficients of the constrained equation
# involving x and the last term is the limit of the
# x equation. For example, if 25x1 + 25x2 = 300, is the
# constrained equation, and there are 3 input variables,
# x1, x2 and x3, the x.c vector is (25, 25, 0, 300)
# this solves for the constrained vector and the point
# of intersection in the least squares scenario.
# it uses the following functions:
# beta.prior
# beta.sum
    n <- nrow(x)
    p <- ncol(x)
    x.one <- cbind(rep(1, n), x)
```

```
    beta <- beta.prior(x.one, y)[c(2:(p + 1))]
    z <- sum(x.c[1:p] * beta)
    d <- z/(beta.sum(x.c[1:p]))
    p0 <- x.c[(p + 1)]/z
    x0 <- p0 * beta
    beta.c <- beta - d * x.c[1:p]
    return(x = x0, beta = beta.c)
}
```

---

constrained.bias

```
function(beta, x.c)
{
# beta is a biased vector
# x.c is a (p+1) vector in which the first p terms
# are the coefficients of the constrained equation
# involving x and the last term is the limit of the
# x equation. For example, if 25x1 + 25x2 = 300, is the
# constrained equation, and there are 3 input variables,
# x1, x2 and x3, the x.c vector is (25, 25, 0, 300)
# this solves for the constrained vector and the point
# of intersection in the biased scenario.
# it uses the following functions:
# beta.sum
    p <- nrow(beta)
    z <- sum(x.c[1:p] * beta)
    d <- z/(beta.sum(x.c[1:p]))
    p0 <- x.c[(p + 1)]/z
    x0 <- p0 * beta
    beta.c <- beta - d * x.c[1:p]
    return(x = x0, beta = beta.c)
}
```

---

cov.solve

```
function(x, r, mse)
{
# x is the design matrix
# r is the prior precision matrix
# mse is the mean squared error
    mse * solve(t(x) %*% x + r)
}
```

---

cv.n

```
function(x, cv, mse, n)
{
# x is the design matrix
# cv is the prior covariance matrix
# mse is the mean squared error
```

```
# n is the number of iterations to perform
     mse * solve(n * t(x) %*% x + mse * solve(cv))
}
```

---

dot.product

```
function(x,y)
     t(x) %*% y
```

---

dot.product.angle

```
function(x, y)
{
# this uses the function dot.product
     (180/pi) * acos((dot.product(x, y)/(sqrt(dot.product(x, x) *
       dot.product(y, y)))))
}
```

---

e.solve

```
function(x, y, yrandom, d)
{
# x is the design matrix
# y is the observed y vector
# yrandom is the random y matrix
# d is the direction vector
# the purpose of this function is to solve
# for posterior values
# this function uses the following functions:
# beta.prior
# beta.solve
# cov.solve
# error.solve
# r.solve
     n <- ncol(yrandom)
     x <- as.matrix(x)
     y <- as.matrix(y)
     result <- r.solve(x, yrandom, d)
     cov.p <- result$cv
     beta.p <- result$b
     one <- rep(1, nrow(x))
     x.one <- cbind(one, x)
     b0 <- beta.prior(x.one, y)[1]
     y.hat <- x %*% beta.p + b0
     mse <- t(y - y.hat) %*% (y - y.hat)/(nrow(x) - ncol(x) - 1)
     r <- diag(mse, nrow = nrow(cov.p)) %*% solve(cov.p)
     beta <- beta.solve(x, y, r, beta.p)
     covariance <- cov.solve(x, r, mse)
     error <- error.solve(x, r, beta, beta.p)
     list(beta = beta, covariance = covariance,
       mse = mse, error = error)
}
```

e.solve2

```
function(x, y, yrandom, d)
{
# this is the e.solve program to use for higher order models
# x is the design matrix
# y is the observed y vector
# yrandom is the random y matrix
# d is the direction vector
# this function uses the following functions:
# beta.prior
# beta.solve
# cov.solve
# error.solve
# order.make
# r.solve2
# the programmer must ensure that order.make is in the correct form
# the programmer must also ensure that r.solve2 has x in the
# proper form
     n <- ncol(yrandom)
     x <- as.matrix(x)
     y <- as.matrix(y)
     x <- order.make(x[, 1], x[, 2], x[, 3], x[, 4])
     result <- r.solve2(x, yrandom, d)
     cov.p <- result$cv
     beta.p <- result$b
     one <- rep(1, nrow(x))
     x.one <- cbind(one, x)
     b0 <- beta.prior(x.one, y)[1]
     y.hat <- x %*% beta.p + b0
     mse <- t(y - y.hat) %*% (y - y.hat)/(nrow(x) - ncol(x) - 1)
     r <- diag(mse, nrow = nrow(cov.p)) %*% solve(cov.p)
     beta <- beta.solve(x, y, r, beta.p)
     covariance <- cov.solve(x, r, mse)
     error <- error.solve(x, r, beta, beta.p)
     list(beta = beta, covariance = covariance,
       mse = mse, error = error)
}
```

error.solve

```
function(x, r, b, bp)
{
# x is the design matrix
# r is the prior beta covariance matrix
# b is the posterior beta vector
# bp is the prior beta vector
     t(b - bp) %*% r %*% solve(t(x) %*% x) %*% r %*% (b - bp)
}
```

mse.solve

```
function(x, y)
{
# x is the design matrix
# y is the response vector
# this uses the function beta.prior
# this function solves for the least squares mse
     n <- nrow(x)
     k <- ncol(x)
     x <- as.matrix(x)
     y <- as.matrix(y)
     x.one <- cbind(rep(1, n), x)
     beta <- beta.prior(x.one, y)
     y.hat <- x.one %*% beta
     sum((y - y.hat)^2)/(n - k - 1)
}
```

---

mse.solve2

```
function(x, y, beta)
{
# x is the design matrix
# y is the response vector
# this function solves for the mse for the
# given beta vector
# it uses the function beta.prior
     n <- nrow(x)
     k <- ncol(x)
     x <- as.matrix(x)
     y <- as.matrix(y)
     x.one <- cbind(rep(1, n), x)
     b0 <- beta.prior(x.one, y)[1]
     y.hat <- x %*% beta + b0
     sum((y - y.hat)^2)/(n - k - 1)
}
```

---

mse.solve3

```
function(x, y)
{
# x is the design matrix
# y is the vector of responses
# this is the function to use to solve for the mse
# when the x matrix has replicates
     n <- nrow(x)
     k <- ncol(x)
     sum.df <- NULL
     flag <- sum <- df <- df.total <- 0
     for(i in 2:n) {
          tf <- all(x[(i - 1), c(1:k)] == x[i, c(1:k)])
          if(tf == T & i < n) {
               sum.df <- rbind(sum.df, y[(i - 1),  ])
```

```
            df <- df + 1
            flag <- 1
        }
        if(tf == T & i == n) {
            sum.df <- rbind(sum.df, y[(i - 1), ])
            sum.df <- rbind(sum.df, y[i, ])
            df <- df + 1
            sum <- sum + apply(sum.df, 2, var) * df
            df.total <- df.total + df
        }
        if(tf == F & flag == 1) {
            sum.df <- rbind(sum.df, y[(i - 1), ])
            sum <- sum + apply(sum.df, 2, var) * df
            df.total <- df.total + df
            flag <- df <- 0
            sum.df <- NULL
        }
    }
    sum/df.total
}
```

---

order.make

```
function(x, y, z)
{
# each letter is a column from a matrix
# the programmer must ensure that the program has the appropriate
# number of letters in the argument to match the matrix
    x <- as.matrix(x)
    y <- as.matrix(y)
    z <- as.matrix(z)
    w <- matrix(0, nrow = nrow(x), ncol = 9)
    w[, 1] <- x
    w[, 2] <- y
    w[, 3] <- z
    w[, 4] <- x^2
    w[, 5] <- y^2
    w[, 6] <- z^2
    w[, 7] <- x * y
    w[, 8] <- x * z
    w[, 9] <- y * z
    w
}
```

---

r.solve

```
function(x, yrandom, d)
{
# x is the design matrix
# yrandom is the random y matrix
# d is the direction vector
```

```
# this function uses the following functions:
# beta.prior
# sort.x
     random.y <- sort.x(x, yrandom, d)
     bp <- t(beta.prior(x, random.y))
     cv <- var(bp)
     b <- apply(bp, 2, mean)
     return(cv, b)
}
```

---

r.solve2

```
function(x, yrandom, d)
{
# x is the design matrix
# yrandom is the random y matrix
# d is the direction vector
# this function uses the following functions:
# beta.prior
# sort.x
     random.y <- sort.x(x[, c(1:4)], yrandom, d)
     bp <- t(beta.prior(x, random.y))
     cv <- var(bp)
     b <- apply(bp, 2, mean)
     return(cv, b)
}
```

---

r.solve3

```
function(x, yrandom, d)
{
# x is the design matrix
# yrandom is the random y matrix
# d is the direction vector
# this function uses the following functions:
# beta.prior
# sort.x
     k <- ncol(x)
     random.y <- sort.x(x[, c(2:k)], yrandom, d)
     bp <- t(beta.prior(x, random.y))
     cv <- var(bp)
     b <- apply(bp, 2, mean)
     return(cv, b)
}
```

---

se.compute

```
function(xp, x, y, yrandom, d)
{
# xp is the x vector for which the standard error
```

```
# is needed
# x is the design matrix
# y is the observed y vector
# yrandom is the random y matrix
# d is the direction vector
# this function uses the following functions:
# beta.prior
# beta.solve
# cov.solve
# error.solve
# r.solve3
     x <- as.matrix(x)
     y <- as.matrix(y)
     one <- rep(1, nrow(x))
     x.one <- cbind(one, x)
     result <- r.solve3(x.one, yrandom, d)
     cov.p <- result$cv
     beta.p <- result$b
     y.hat <- x.one %*% beta.p
     mse <- t(y - y.hat) %*% (y - y.hat)/(nrow(x) - ncol(x) - 1)
     r <- diag(mse, nrow = nrow(cov.p)) %*% solve(cov.p)
     xxr <- solve(t(x.one) %*% x.one + r)
     covariance <- diag(mse, nrow = nrow(cov.p)) %*% xxr
     se <- sqrt(t(xp) %*% covariance %*% xp)
     list(se = se, mse = mse, xxr = xxr)
}
```

---

sort.x

```
function(x, randomy, d)
{
# x is the design matrix
# randomy is the random y matrix
# d is the direction vector
     no <- rank(x %*% d)
     randomy[c(no),  ]
}
```

---

steep.ascent.ci

```
function(x, y, alpha)
{
# x is the design matrix
# y is the observed y vector
# alpha is the level of confidence
# this function uses the following functions:
# beta.prior
# ci.angle
# mse.solve
     x <- as.matrix(x)
     y <- as.matrix(y)
```

```
     beta <- beta.prior(x, y)
     mse <- mse.solve(x, y)
     cv <- diag(mse, nrow = ncol(x)) %*% solve(t(x) %*% x)
     ci.angle(x, beta, cv, alpha)
}
```

---

sum.diag

```
function(x)
{
     n <- nrow(x)
     sum <- 0
     for(i in 1:n)
          sum <- sum + x[i, i]
     sum/n
}
```

---

x.find

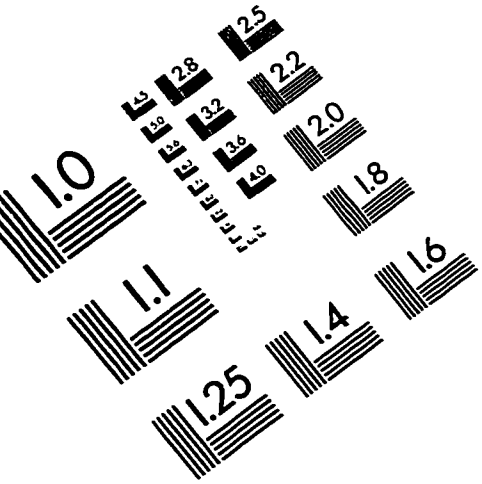
```
function(beta)
{
# the programmer should ensure that
# beta has the proper number of
# components to match the function as
# well as the proper sum of squares
     x1 <- sqrt(3/(1 + beta[2]^2/beta[1]^2 +
        beta[3]^2/beta[1]^2)) * sign(beta[1])
     x2 <- beta[2]/beta[1] * x1
     x3 <- beta[3]/beta[1] * x1
     matrix(c(x1, x2, x3))
}
```

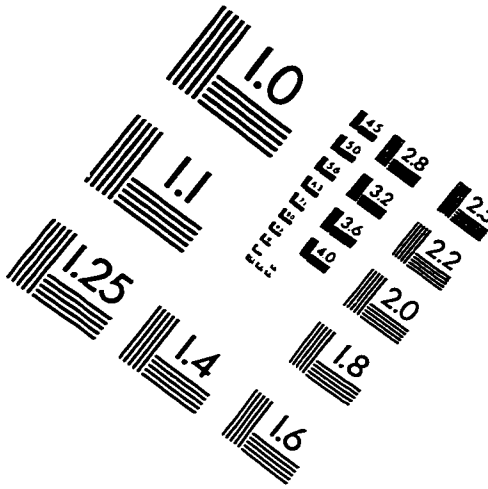
---

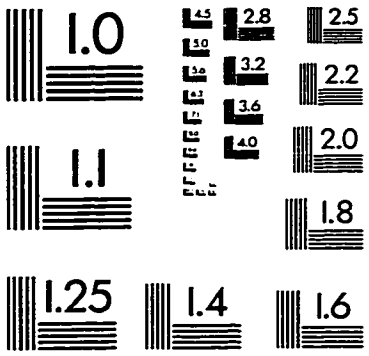
y.generate

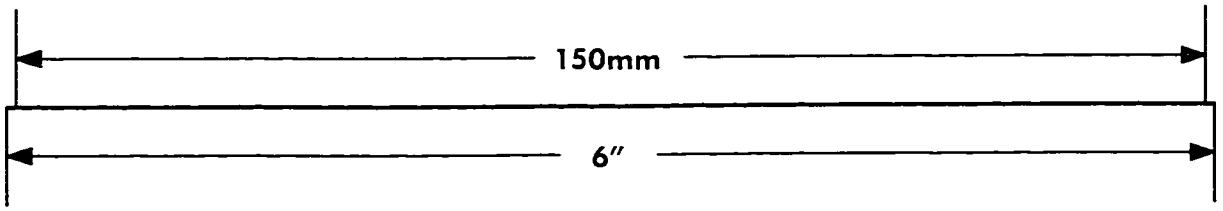
```
function(y, n)
{
# y is the observed or prior y matrix
# n is the number of iterations to perform
     k <- nrow(y)
     y <- sort(y)
     y.sort <- matrix(0, nrow = (k - 1))
     for(i in 1:(k - 1))
          y.sort[i] <- y[i + 1] - y[i]
     y.sort <- y.sort[y.sort > 0]
     sigma <- min(y.sort)/6
     random.y <- matrix(rep(0, k * n), nrow = k)
     for(i in 1:k)
          random.y[i,  ] <- rnorm(n, y[i], sigma)
     random.y
}
```

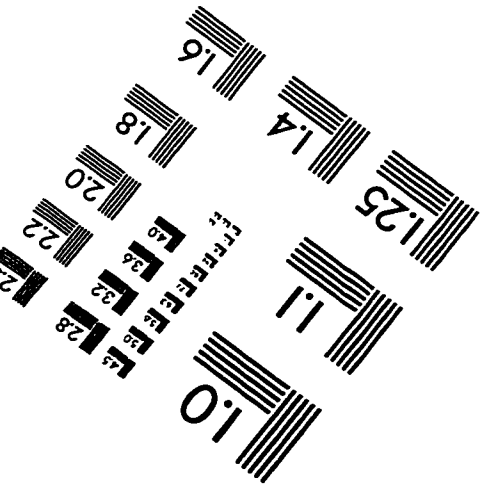
IMAGE EVALUATION
TEST TARGET (QA-3)

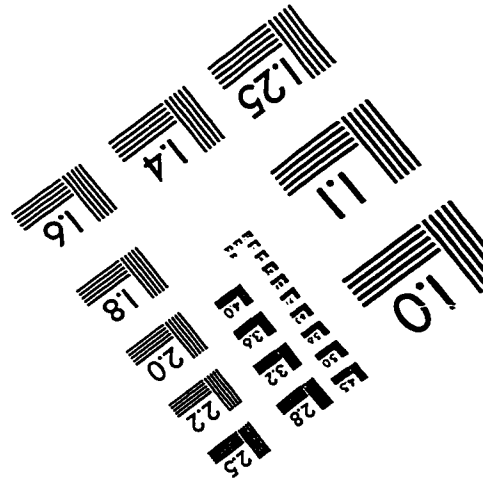












APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc.. All Rights Reserved