

ROBUST D-OPTIMAL DESIGN FOR RESPONSE FUNCTIONS WITH  
A DOWNTURN

A Paper  
Submitted to the Graduate Faculty  
of the  
North Dakota State University  
of Agriculture and Applied Science

By

Jessica Anne Carter

In Partial Fulfillment of the Requirements  
for the Degree of  
MASTER OF SCIENCE

Major Department:  
Statistics  
Master of Applied Statistics

May 2013

Fargo, North Dakota

North Dakota State University  
Graduate School

---

**Title**

ROBUST D-OPTIMAL DESIGN FOR RESPONSE FUNCTIONS WITH A  
DOWNTURN

---

**By**

Jessica Anne Carter

---

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

**MASTER OF SCIENCE**

SUPERVISORY COMMITTEE:

Dr. Seung Won Hyun

---

Chair

Dr. Megan Orr

---

Dr. Elizabeth Birmingham

---

Mr. Curt Doetkott

---

Approved:

5/22/2013

---

Date

Dr. Ronda Magel

---

Department Chair

## ABSTRACT

Carter, Jessica, M.S., Department of Statistics, College of Science and Mathematics, North Dakota State University, May 2013, Robust D-optimal Design for Response Functions with a Downturn. Major Professor: Dr. Seung Won Hyun.

Researchers studying dose-response relationships must allocate limited resources to design points in order to maximize the information gained from the study. D-optimal design is a well-described design that works efficiently to study model parameters. In order to find the D-optimal design, the model that describes the dose-response relationship has to be known. In cases where dose-response relationships show a downturn at high doses, scientists sometimes ignore the downturn to study only the increasing part of the response curve. Here we have two model choices; one describes the overall dose-response relationship, and the other describes only the increasing part of the response curve. The D-optimal designs for these two models will be different and the D-optimal design for one model may not work efficiently for the other model. This research studies robust D-optimal design, a design that works efficiently for both models.

# TABLE OF CONTENTS

ABSTRACT .....	iii
TABLE OF CONTENTS.....	iv
LIST OF TABLES .....	v
LIST OF FIGURES .....	vi
1. INTRODUCTION.....	1
2. BACKGROUND.....	4
2.1. Optimal Design.....	4
2.2. D-optimal Design.....	4
2.3. V-Algorithm.....	5
3. MODELS .....	7
3.1. The General Model Form .....	7
3.2. Model Fitting.....	7
3.3. Fisher Information Matrix .....	8
4. D-OPTIMAL DESIGN.....	10
5. ROBUST D-OPTIMAL DESIGN.....	12
6. EFFICIENCY .....	15
7. CONCLUSIONS.....	18
WORKS CITED .....	20
APPENDIX A. SAS CODE TO FIT THE COMPLETE MODEL.....	21
APPENDIX B. SAS CODE TO FIT THE REDUCED MODEL.....	22
APPENDIX C. R-CODE FOR D-OPTIMAL DESIGN FOR THE COMPLETE MODEL .....	23
APPENDIX D. R-CODE FOR D-OPTIMAL DESIGN FOR THE REDUCED MODEL .....	26
APPENDIX E. R-CODE FOR ROBUST D-OPTIMAL DESIGN .....	29
APPENDIX F. R-CODE FOR RELATIVE EFFICIENCIES .....	33

## LIST OF TABLES

Table 1. Relative design efficiencies and the additional percentage of samples (in parentheses) needed to obtain the same accuracy as the D-optimal design.....	17
-----------------------------------------------------------------------------------------------------------------------------------------------------------------	----

## LIST OF FIGURES

Figure 1. The dose-response data, plotted with the full and reduced models.....	8
Figure 2. Comparison of the two locally D-optimal designs and the robust D-optimal design.....	14

# 1. INTRODUCTION

The design of an experiment is one of the most important factors affecting the outcome of a study. A well designed experiment can help researchers maximize the information gained from a study and ensure that valid and objective conclusions can be made from the results, all while minimizing costs. A poorly designed study, on the other hand, may result in data that are not valid or useful, and thus wasted money and effort (Montgomery, 2009). Since no two studies are exactly alike, there is not one experimental design that will work well for all studies. Thus the study of experimental design has become an important area in the field of statistics research.

Every discipline has particular types of studies that are typical of the research done in that field, and consequently specialized areas of experimental design research have been established that deal with them. One area in which experimental design is often applied is that of biopharmaceutical and toxicology research, where many research studies involve investigating a dose-response relationship. In particular, the dose-ranging study is of key importance. The purpose of a dose-ranging study is to determine the dose of a drug or chemical for further study in large-scale trials (Bornkamp et al., 2007). An important goal for pharmaceutical companies involved in dose-ranging studies is to lower their costs while gathering useful information; this requires well-designed experiments.

One way that companies can lower costs is to conduct a single study with multiple objectives. Researchers conducting a dose-ranging study are often interested in many attributes of the dose-response relationship, and a study may have multiple objectives: estimating the model parameters; estimating one or more specific doses, like the minimum effective dose or the ED<sub>p</sub> (100p% of the maximum treatment effect); identifying clinical relevance; and others (Padmanabhan and Dragalin, 2010). Given limited resources, the

desire to speed up the development of pharmaceutical products, and the ethical concerns involved in studying human subjects, studies that are designed to simultaneously address multiple objectives are more beneficial (Brannath et al., 2007, Todd and Stallard, 2005).

From a statistical perspective, the implications of having multiple study objectives relate to how the study should be designed. In a simple, one-objective study, a researcher can generally easily find an experimental design that maximizes the amount of information obtained towards the specific study objective. The design that fills this role is called the optimal design. In the field of optimal design, there are several types of optimality criteria, each of which is preferred for particular study objectives. When the goal of the study is to estimate model parameters, researchers most often use the D-optimality criterion (Padmanabhan and Dragalin, 2010), and that is the focus of this paper.

The locally D-optimal design depends directly on the model parameters to be estimated; they are needed for input into the algorithm to find the optimal design. The term “locally” implies that we assume the true values of the parameters are known and focus on finding the optimal design for those values. When a study has multiple objectives, such as fitting two models to a set of data, the D-optimal designs for the two models will be different because each model has different parameters. This presents a problem to the researcher who is interested in studying two models: which design should be used? The D-optimal design for one model is likely not optimal for the other model, and may be very inefficient. Atkinson et al. (2007) present a method for creating a composite optimality criterion that takes into account all of the attributes of interest.

In this paper, we apply this robust D-optimality criterion with the goal of finding a single design that works efficiently for estimating the parameters of two models. We will consider the case in which a dose-response relationship includes a downturn at higher dose levels, and a researcher is interested in 1) studying the entire dose-response curve, and 2)

studying only the increasing portion of the curve. The motivating data are presented in Welshons et al. (2003). The authors of that paper studied low through high doses of endocrine disrupting chemicals with estrogenic activity (EEDCs), which are chemicals that act like hormones, and the effect on cell proliferation of the MCF-7 human breast cancer cell. They showed that the relationship between the EEDCs and cancer cell growth has a non-monotonic, inverted-U shape with a downturn at the higher doses. Since we are interested in both the whole curve and just the increasing portion of the curve in this research, we need to fit two models to the data. We use the model developed by Hyun (2013) to describe the complete dataset and, for simplicity, we fit the same model minus the quadratic term to the reduced dataset that includes the doses from the minimum up to the dose that is presumed to elicit the maximum response. Since the two models have different shapes, we may expect that the optimal design for one model may not work efficiently for the other model. Given this expectation, we seek to find the robust D-optimal design that will work efficiently for estimating the parameters of both models.

In chapter 2 we present some background information about optimal designs. Chapter 3 describes the models of interest, how the parameters are estimated, and the information matrices, which are needed for finding the optimal designs. In chapter 4 we find the D-optimal designs for the two models separately and in chapter 5 we find the robust D-optimal design. In chapter 6 we compare the efficiencies of the three optimal designs. Finally, in Chapter 7 we present our conclusions.

## 2. BACKGROUND

For a researcher interested in studying the relationship between dose and response, a study objective is often to estimate the model parameters for a model that describes the relationship. In order to do this with the most precision, which is necessary for ensuring the validity of any predictions made from the model, we must carefully select the levels of treatment and how to distribute the available samples over those treatments. We can use optimal design to aid in those selections. In this chapter, we will introduce some concepts important to the study of optimal design and show how optimal designs are found.

### 2.1. Optimal Design

Optimal design is a field of statistics that deals with designing experiments in the most efficient way. Optimal designs tell researchers what levels of treatment to use, and how to distribute the samples across those treatment levels in order to most efficiently meet some study objective. There are several different optimal design criteria; the choice of which one to use depends on the objectives of the study. In general, the optimal design is the design that minimizes the optimality criterion (Atkinson et al., 2007).

### 2.2. D-optimal Design

The D-optimality criterion is a widely used criterion for obtaining optimal designs. D-optimal designs work well for estimating model parameters (Atkinson et al., 2007). The D-optimal design minimizes the volume of the joint confidence ellipsoid for estimating model parameters, thereby minimizing the variance of parameter estimation (Padmanabhan and Dragalin, 2010, Atkinson et al., 2007). We find this design by minimizing the determinant of the inverse of the information matrix  $|M(\xi; \theta)^{-1}|$ , or conversely, maximizing the determinant of the information matrix. Here  $M(\xi; \theta)$  is the Fisher information matrix under design  $\xi$ , and  $\theta$  denotes the vector of parameters for the

given model. The design  $\xi = \{(x_i, w_i), i = 1, 2, \dots, k\}$ , where  $x_i$  is the  $i^{\text{th}}$  dose level, and  $w_i$  is the weight of the  $i^{\text{th}}$  design point. The weight  $w_i = \frac{n_i}{n}$ , where  $n_i$  is the number of samples allocated to the  $i^{\text{th}}$  dose level and  $n$  is the total number of samples, such that  $\sum_{i=1}^k n_i = n$ . We want to find the D-optimal design  $\xi^D$  that minimizes  $|M(\xi; \theta)^{-1}|$ .

### 2.3. V-Algorithm

Fedorov and Hackl (1997) discuss an iterative method for finding optimal designs called the V-algorithm. For a continuous response  $y_{ij}$ , suppose  $y_{ij} = \mu_i(x_i, \theta) + \varepsilon_i$ ,  $\varepsilon_i \sim N(0, \sigma^2)$ . Based on design  $\xi$ , the Fisher information matrix for  $\theta$  is expressed by  $M(\xi; \theta) = \sum_{i=1}^k w_i f(x_i)^T f(x_i)$ , where  $f(x_i)$  is the vector of first-order partial derivatives of the mean function  $\mu_i(x_i, \theta)$  with respect to  $\theta$ . The V-algorithm works by minimizing the maximum of the standardized variance,  $d_n$ , shown in (1), where  $n$  refers to the  $n^{\text{th}}$  iteration.

$$d_n = f(x)^T M(\xi_n; \theta)^{-1} f(x) \quad (1)$$

The standardized variance is evaluated for each  $x$  in the design space and the  $x^*$  that minimizes  $\max(d_n)$  is chosen to be included in the next iteration of the design,  $\xi_{n+1}$ .

$M(\xi_n; \theta)$  is then recalculated by  $M(\xi_{n+1}; \theta) = (1 - \alpha_{n+1})M(\xi_n; \theta) + \alpha_{n+1}f(x^*)^T f(x^*)$ , where  $\alpha_{n+1} = \frac{1}{n+1}$ . When  $d_n - p$  is close to zero (eg.  $p = 0.005$ ), the corresponding design  $\xi_n$

becomes the D-optimal design, and the iteration stops. Here  $p$  is the number of parameters to be estimated.

The V-algorithm requires a starting design, which is generally chosen to be uniform. The starting design is based on the range of the data, so that the endpoints of the data range are included in the design, along with evenly-spaced points in the middle, if necessary. The number of design points is determined based on the number of parameters to be estimated. At least  $p$  design points are needed for estimating  $p$  parameters, and Caratheodory's Theorem gives an upper bound for the number of points in the design when finding a D-optimal design. According to this theorem, the

number of design points in the initial design should be no more than  $\left(\frac{p(p+1)}{2} + 1\right)$ , where  $p$  is the number of parameters to be estimated in the model.

### 3. MODELS

In this section, we will introduce the models under consideration for this paper, and determine the Fisher information matrix that we will use to find the D-optimal and robust D-optimal designs.

#### 3.1. The General Model Form

In toxicology studies, the probit model is often used to describe dose-response relationships. When there is a downturn in the response at high dose levels, including a second-order term for dose in the model can account for this (Ting, 2006). In this paper, a probit model with three parameters is used to describe a dose-response relationship with both increasing and decreasing portions (2). Here  $\Phi$  is the cumulative distribution function of the standard normal. It was decided prior to fitting the models that, for simplicity, a probit model with two parameters (3) would be used to describe just the increasing portion of the dose-response curve, up to the dose presumed to elicit the maximum response.

Suppose a continuous response  $y_{ij}$  follows  $y_{ij} = \mu_p(x_i, \theta) + \varepsilon_{ij}$ , where  $x_i$  is the  $i^{\text{th}}$  log dose,  $i=1,2,\dots,k$ ,  $j=1,2,\dots,n_i$ ,  $p=1,2$ ,  $\varepsilon_{ij} \sim N(0, \sigma^2)$ , and  $\sigma^2$  is unknown. Define  $\theta_1 = (\theta_1, \theta_2, \theta_3)$  and  $\theta_2 = (\theta_1, \theta_2)$ . Then it follows that:

$$\mu_1(x_i, \theta_1) = \Phi(-(\theta_1 + \theta_2 x_i + \theta_3 x_i^2)); \quad (2)$$

$$\mu_2(x_i, \theta_2) = \Phi(-(\theta_1 + \theta_2 x_i)). \quad (3)$$

#### 3.2. Model Fitting

To see how these two models fit Welshons' data, we used the non-linear procedure in SAS 9.2 to fit both the three-parameter model to the complete design space (Appendix A) and the two-parameter model to the reduced design space, which considers only the increasing portion of the response curve where the log dose ranges from -14 to -6 (Appendix B). The estimated three-parameter model is shown in (4) and the estimated two-parameter model is shown in (5).

$$\mu_1(x_i, \theta_1) = \phi(-(4.6359 + 1.2327x_i + 0.072x_i^2)) \quad (4)$$

$$\mu_2(x_i, \theta_2) = \phi(-(-2.0381 - 0.1926x_i)) \quad (5)$$

In both cases, the p-value for the model was highly significant ( $<.0001$ ), however the complete model followed the curves of the data much better than the reduced model (Figure 1). In other words, the two models have completely different shapes. Thus we expect that the optimal design for each model may work poorly for the other model.

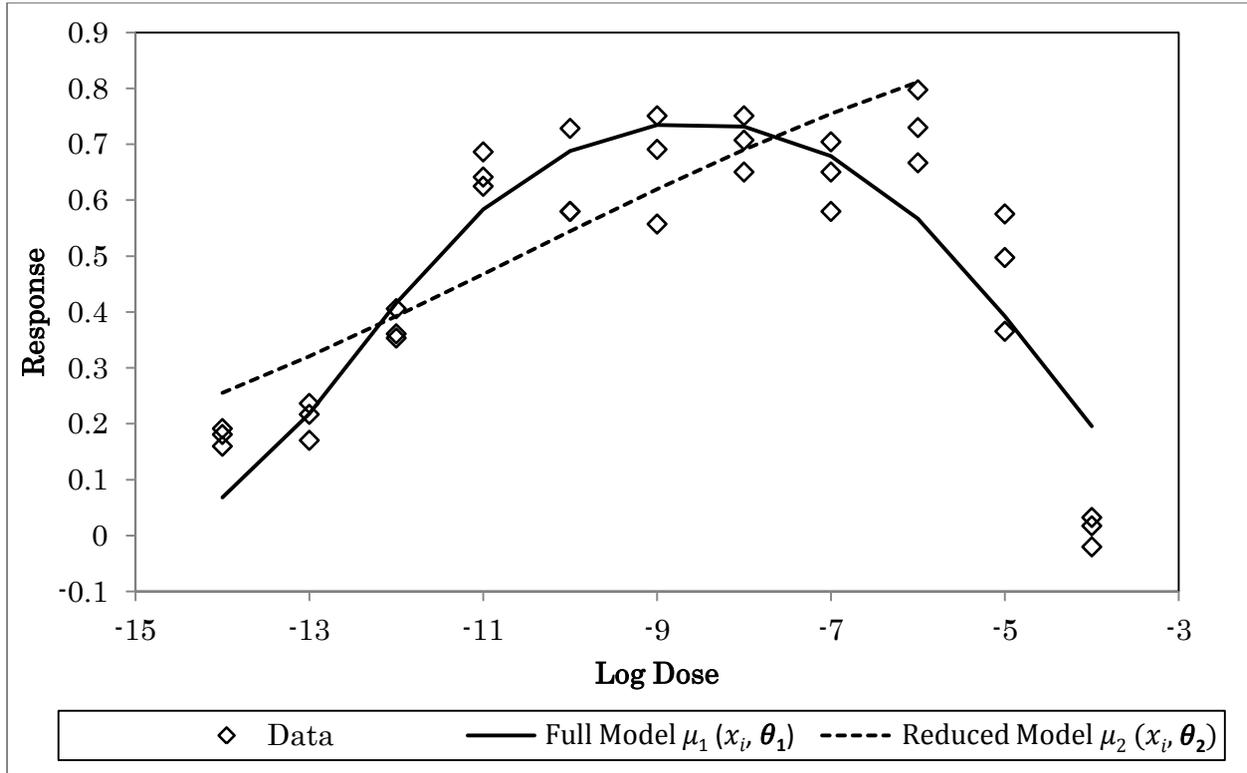


Figure 1. The dose-response data, plotted with the full and reduced models.

### 3.3. Fisher Information Matrix

One of the necessary components needed for finding the optimal designs is the Fisher information matrix. As mentioned in Chapter 2, the information matrix is used to calculate the standardized variance, (1), which is the key for finding the D-optimal design. Here we will construct the information matrices, denoted as  $M_p(\xi; \theta)$ , using the formula in (6).

$$M_p(\xi; \theta) = \frac{n}{\sigma^2 \sqrt{2\pi}} \sum_{i=1}^k w_i f(x_i)^T f(x_i) \quad (6)$$

Here,  $f(x_i)$  is a vector of the first-order derivatives of the mean function  $\mu_p(x_i, \theta)$ . For the mean response (2) that is,

$$\begin{aligned} f(x_i) &= \left( \frac{\partial \mu_1(x_i, \boldsymbol{\theta}_1)}{\partial \theta_1}, \frac{\partial \mu_1(x_i, \boldsymbol{\theta}_1)}{\partial \theta_2}, \frac{\partial \mu_1(x_i, \boldsymbol{\theta}_1)}{\partial \theta_3} \right)^T \\ &= \left( e^{-\frac{1}{2}(\theta_1 + \theta_2 x_i + \theta_2 x_i^2)^2}, x e^{-\frac{1}{2}(\theta_1 + \theta_2 x_i + \theta_2 x_i^2)^2}, x^2 e^{-\frac{1}{2}(\theta_1 + \theta_2 x_i + \theta_2 x_i^2)^2} \right). \end{aligned}$$

It follows that the Fisher information matrix for the mean response (2) is as shown below:

$$M_1(\xi; \boldsymbol{\theta}_1) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^k w_i e^{-(\theta_1 + \theta_2 x_i + \theta_2 x_i^2)^2} \begin{bmatrix} 1 & x & x^2 \\ x & x^2 & x^3 \\ x^2 & x^3 & x^4 \end{bmatrix}.$$

For the mean response (3),  $\theta_3$  equals zero, so there are only two terms in  $f(x_i)$ :

$$\begin{aligned} f(x_i) &= \left( \frac{\partial \mu_2(x_i, \boldsymbol{\theta}_2)}{\partial \theta_1}, \frac{\partial \mu_2(x_i, \boldsymbol{\theta}_2)}{\partial \theta_2} \right)^T \\ &= \left( e^{-\frac{1}{2}(\theta_1 + \theta_2 x_i + \theta_2 x_i^2)^2}, x e^{-\frac{1}{2}(\theta_1 + \theta_2 x_i + \theta_2 x_i^2)^2} \right), \end{aligned}$$

and the Fisher information matrix is as follows:

$$M_2(\xi; \boldsymbol{\theta}_2) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^k w_i e^{-(\theta_1 + \theta_2 x_i)^2} \begin{bmatrix} 1 & x \\ x & x^2 \end{bmatrix}.$$

These information matrices are the key to finding the D-optimal designs, as discussed in the following section.

## 4. D-OPTIMAL DESIGN

In this chapter we will find the locally D-optimal designs for the complete and reduced models. We say that these designs are locally D-optimal because they are the optimal designs based on the given values initially estimated for the model parameters. These designs may not be optimal for the true values of the model parameters, which are unknown. However, we assume that the values initially estimated are the true values of the model parameters.

In this paper we found the optimal designs for the mean response (2) (Appendix C) and mean response (3) (Appendix D) using the V-algorithm in R. To use the V-algorithm, two elements must first be input: the values of the model parameters and an initial design. As discussed earlier, the initial design should have between  $p$  and  $(\frac{p(p+1)}{2} + 1)$  design points, where  $p$  is the number of parameters in the model. For mean response (2), the design should have between three and seven design points, and for mean response (3) the design should have between two and four design points. For simplicity, we use the minimum number of design points and a uniform design based on the range of the data. For mean response (2) we use three design points, and the data range from -14 to -4. Therefore, the initial design has 3 points at -14, -9, and -4, each with one third of the weight. The initial design for mean response (3) has two design points at -14 and -6, each with one half of the weight.

The D-optimal design for the complete model ( $\xi_C^D$ ) has four design points, with one third of the weight near each of the two endpoints of the data range and the rest split equally between two points near the middle of the range, while the D-optimal design for the reduced model ( $\xi_R^D$ ) has two design points with equal weight near the endpoints of the data range. These are the designs that maximize  $|M_1(\xi; \boldsymbol{\theta}_1)|$  and  $|M_2(\xi; \boldsymbol{\theta}_2)|$ , respectively:

$$\xi_C^D = \begin{Bmatrix} -12.73 & -9.21 & -7.91 & -4.39 \\ 0.33 & 0.17 & 0.17 & 0.33 \end{Bmatrix};$$

$$\xi_R^D = \begin{Bmatrix} -14 & -6.84 \\ .50 & .50 \end{Bmatrix}.$$

## 5. ROBUST D-OPTIMAL DESIGN

When two models are of interest to a researcher, the D-optimal design for one model will likely not be optimal for the other model. The purpose of robust D-optimal design is to find the design that minimizes the variance of estimating model parameters for both models at the same time. This is accomplished by modifying the V-algorithm to work with both models, and defining the robust D-optimality criterion.

In chapter 2, we showed that the D-optimality criterion minimizes the determinant of the inverse of the information matrix. Since we are now concerned with two models, we have two D-optimality criteria to minimize. The robust D-optimality criterion  $\phi(\xi)$  combines the two D-optimality criteria into a compound optimality criterion that maximizes the D-efficiency under both models (Atkinson et al., 2007, Chapter 21). Therefore, instead of minimizing the determinant of the inverse of the information matrices, the robust D-optimal design maximizes (7).

$$\phi(\xi) = k \left( \frac{\log |M_1(\xi; \theta_1)|}{p_1} \right) + (1 - k) \left( \frac{\log |M_2(\xi; \theta_2)|}{p_2} \right) \quad (7)$$

Here  $k$  is some constant weight between zero and one for each model,  $M_1(\xi; \theta_1)$  is the information matrix for the complete model,  $p_1$  is the number of parameters to be estimated in the complete model,  $M_2(\xi; \theta_2)$  is the information matrix for the reduced model, and  $p_2$  is the number of parameters to be estimated in the reduced model. The derivative function of (7) is shown in (8). This puts the criterion in the same form as (1).

$$\left( \frac{k}{p_1} \right) (f_1(x)^T M_1(\xi_n; \theta_1)^{-1} f_1(x)) + \left( \frac{1-k}{p_2} \right) (f_2(x)^T M_2(\xi_n; \theta_2)^{-1} f_2(x)) \leq 1 \quad (8)$$

In order to use the V-algorithm to find the robust D-optimal design where two models are of interest, the sensitivity function is replaced with the left side of (8). The parameter  $k$  allows us to specify a weight of interest in each model by inputting a value

between 0 and 1 for  $k$ . This identifies which model is more important, or for  $k=0.5$ , that the models have equal importance.

As with the D-optimal design, the inputs to the program to find the robust D-optimal design include the specified parameter values, this time for both models, and a single initial design. Since the robust D-optimal design must work for both models, the initial design has three points which are based on the range of the complete dataset [-14, -4]. This satisfies the requirements for both models, since the reduced model requires at least two design points, and the range of the reduced dataset [-14, -6] is included within the complete dataset. For this paper, we only evaluated the D-optimal design with equal weight of interest for the two models (i.e. weights of interest of 0.5 for both models).

The robust D-optimal design ( $\xi_{Rob}^D$ ) is the design that maximizes both  $|M_1(\xi; \theta_1)|$  and  $|M_2(\xi; \theta_2)|$ :

$$\xi_{Rob}^D = \begin{Bmatrix} -12.9 & -10.5 & -7.5 & -4.4 \\ 0.4 & 0.03 & 0.29 & 0.28 \end{Bmatrix}.$$

The robust D-optimal design is similar to the D-optimal design for the complete model ( $\xi_C^D$ ) in that they both have 4 design points, with approximately two thirds of the weight split between two points near the endpoints of the design space. Unlike  $\xi_C^D$ ,  $\xi_{Rob}^D$  has only a very small portion (3%) of the weight at one of the middle points. Figure 2 shows a comparison of the two locally D-optimal designs and the robust D-optimal design.

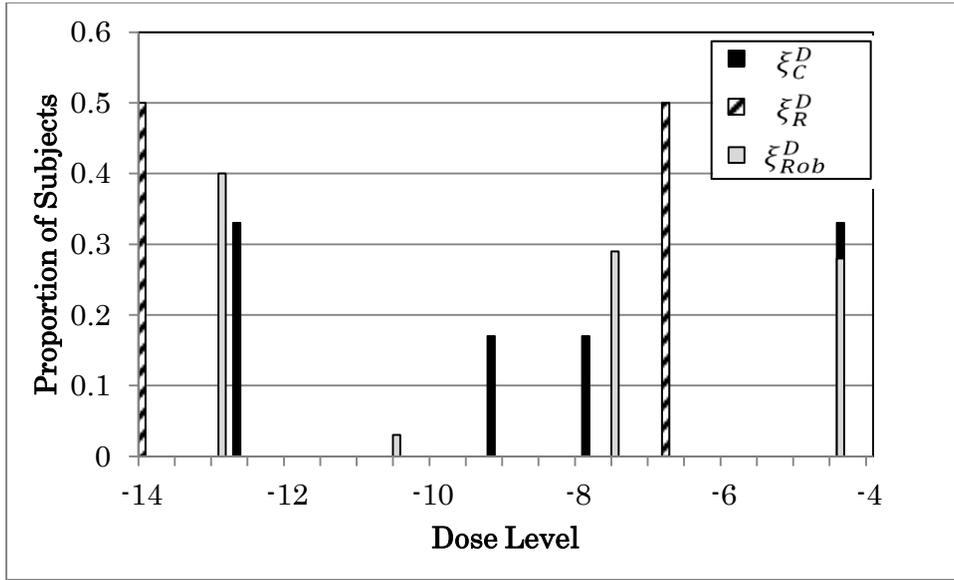


Figure 2. Comparison of the two locally D-optimal designs and the robust D-optimal design.

## 6. EFFICIENCY

In this chapter we will discuss design efficiency and compare the relative efficiencies of the designs found in the previous chapter. Design efficiency tells us how a design performs relative to some criteria. Since we are concerned with D-optimal design in this paper, we will compute the D-efficiency. The formula for D-efficiency is shown in (9), where the determinant of the information matrix for a design  $\xi$  is divided by the determinant of the information matrix for the locally D-optimal design  $\xi^D$ , and the ratio is scaled by the number of parameters ( $p$ ) to be estimated in the model.

$$\text{Eff}^D(\xi) = \left( \frac{|\mathbf{M}(\xi; \theta)|}{|\mathbf{M}(\xi^D; \theta)|} \right)^{1/p} \quad (9)$$

D-efficiencies range from zero to one, with the optimal design having efficiency one and non-optimal designs having efficiencies smaller than one. The D-efficiency of a design  $\xi$  tells us the sample size required for parameter estimation to have the same accuracy under  $\xi$  as it does under the locally D-optimal design  $\xi^D$  (Padmanabhan and Dragalin, 2010). For example, if the D-efficiency of design  $\xi$  is  $q$ , then  $100(1/q - 1)\%$  more samples are needed for design  $\xi$  to provide the same accuracy for parameter estimation as  $\xi^D$ .

D-efficiencies for all three optimal designs,  $\xi_C^D$ ,  $\xi_R^D$ , and  $\xi_{Rob}^D$ , for both the complete and reduced models were computed in R (Appendix F) and are shown in Table 1. Using design  $\xi_C^D$  for the reduced model resulted in a D-efficiency of 0.83. If we were to use design  $\xi_C^D$  for the reduced model, we would need 21 percent more samples to obtain the same accuracy as  $\xi_R^D$ . When calculating the D-efficiency of using design  $\xi_R^D$  for the complete model, a singular matrix resulted, because there were fewer design points than the number of parameters to be estimated. Therefore, to calculate the D-efficiency of using design  $\xi_R^D$  for the complete model we used an alternative design that was close to  $\xi_R^D$ , (with efficiency 0.99) but which had three design points instead of two ( $\xi_{RA}^D$ ):

$$\xi_{RA}^D \begin{Bmatrix} -14 & -6.84 & -6.20 \\ .50 & .25 & .25 \end{Bmatrix}.$$

Using the alternative design  $\xi_{RA}^D$  to estimate the D-efficiency for the complete model resulted in an efficiency of 0.24, meaning that 317 percent more samples would be needed to obtain the same accuracy of estimation as design  $\xi_C^D$ . While the non-optimal design  $\xi_R^D$  had a relatively low efficiency for the complete model, the non-optimal design  $\xi_C^D$  performed relatively well for the reduced model. The difference here is the loss of information that occurs when using the reduced range found in the design  $\xi_R^D$  for estimating the complete model.

For both models, the robust D-optimal design,  $\xi_{Rob}^D$ , outperformed the non-optimal design. In the case of the complete model, the D-efficiency of the robust design was 0.97, so only three percent more samples would be needed to match the efficiency of estimating parameters under  $\xi_C^D$ . For the reduced model, the D-efficiency of the robust design was 0.89. To use the robust design for the reduced model, 13 percent more samples would be needed to obtain the same accuracy of parameter estimation as  $\xi_R^D$ .

Just looking at the D-efficiencies, it is difficult to tell which model performed best, since both  $\xi_C^D$  and  $\xi_{Rob}^D$  performed relatively well for the reduced model. Therefore, I calculated the average of the two D-efficiencies for each design. The robust D-optimal design  $\xi_{Rob}^D$  had an average D-efficiency of 0.93, while  $\xi_C^D$  averaged 0.91 (Table 1). This confirms that the robust D-optimal design is the most efficient design.

Table 1. Relative design efficiencies and the additional percentage of samples (in parentheses) needed to obtain the same accuracy as the D-optimal design.

Design	Model				Average
	Complete (2)		Reduced (3)		
$\xi_C^D$	1	(0%)	0.83	(21%)	0.91
$\xi_R^D$	0.24	(317%)	1	(0%)	0.62
$\xi_{Rob}^D$	0.97	(3%)	0.89	(13%)	0.93

## 7. CONCLUSIONS

D-optimal design is a very useful tool for establishing an experimental design for research in which the main objective is to estimate model parameters. Here we have examined D-optimal designs for two models fit to a dose-response dataset in which the response has a downturn at higher dose levels. We found the D-optimal designs for a complete model that was fit to the entire dataset, and a reduced model that was fit to just the increasing portion of the dataset. We also found the robust D-optimal design that takes into account both models.

We used a three-parameter model for the complete design space, and for simplicity, we used a two-parameter model for the reduced design space. Although the reduced model did not fit the data very well, it worked well for the purposes of demonstrating the effectiveness of the robust D-optimal criterion. The D-optimal design for the reduced model performed very poorly for the complete model. The D-optimal design for the complete model performed relatively well for the reduced model. Despite the complete and reduced models showing very different shapes and the D-efficiency of the reduced design being very poor for the complete model, by combining the two optimality criteria into one compound criterion we were able to find a robust D-optimal design that was efficient for both models. By comparing the D-efficiencies, we showed that the robust D-optimal design was more efficient overall than either of the other designs.

Several related ideas are of interest for future study. In this paper, we considered a case where two models are of equal interest to the researcher. In the future, we would like to examine how the robust design criterion performs when we consider different weights of interest in the two models. We have shown that the robust D-optimality criterion is efficient in the case of the two models presented here; however, there are several other candidate models to describe the dose-response relationship that shows a downturn at high

doses. We would like to study whether a robust D-optimal design that works well under all candidate models can be found. Finally, since optimal design depends on the unknown values of the model parameters, we would like to study a robust optimal design that is less sensitive to these unknown values.

## WORKS CITED

- Atkinson, A. C., Donev, A. N., and Tobias, R. D. (2007), *Optimum Experimental Designs, with SAS*, Oxford: Oxford University Press.
- Bornkamp, B., Bretz, F., Dmitrienko, A., Enas, G., Gaydos, B., Hsu, C.-H., König, F., Krams, M., Liu, Q., Neuenschwander, B., Parke, T., Pinheiro, J., Roy, A., Sax, R. and Shen, F. (2007), “Innovative approaches for designing and analyzing adaptive dose-ranging trials,” *Journal of Biopharmaceutical Statistics* 17, 965–1032.
- Brannath, W., Koenig, F. and Bauer P. (2007), “Multiplicity and flexibility in clinical trials,” *Pharmaceutical Statistics* 6, 205–216.
- Fedorov, V. V., and Hackl, P. (1997), *Model-Oriented Design of Experiments* (Vol. 25, Lecture Notes in Statistics), New York: Springer
- Hyun S. W. (2013), “Optimal designs for a probit model with a quadratic term,” *Statistics in Biopharmaceutical Research* 5, 18-26.
- Montgomery, D.C. (2009), *Design and Analysis of Experiments* (7<sup>th</sup> Edition), Wiley & Sons, Inc.
- Padmanabhan, S. K. and Dragalin, V. (2010), “Adaptive  $D_c$ -optimal designs for dose finding based on a continuous efficacy endpoint,” *Biometrical Journal* 52, 836–852.
- Ting, N. (2006), *Dose Finding in Drug Development*, New York: Springer.
- Todd, S. and Stallard, N. (2005) “A new clinical trial design combining phases II and III: sequential designs with treatment selection and a change of endpoint,” *Drug Information Journal* 39:109–118.
- Welshons, W.V., Thayer, K.A., Judy, B.M., Taylor, J.A., Curran, E.M., Saal, F.S. (2003), “Large effects from small exposures. I. Mechanisms for endocrine-disrupting chemicals with estrogenic activity,” *Environmental Health Perspectives* 111, 994–1006.

## APPENDIX A. SAS CODE TO FIT THE COMPLETE MODEL

```
data one;
input x y @@;
x1=log10(x);
cards;
1E-14 .19157199 1E-14 .18106978 1E-14 .16006537
1E-13 .17056757 1E-13 .21707734 1E-13 .23658144
1E-12 .40611707 1E-12 .36110761 1E-12 .35360603
1E-11 .64166656 1E-11 .62516309 1E-11 .68667602
1E-10 .72868484 1E-10 .58015363 1E-10 .58015363
1E-9 .69117696 1E-9 .55764891 1E-9 .75118957
1E-8 .70768043 1E-8 .65066845 1E-8 .75118957
1E-7 .58015363 1E-7 .7046798 1E-7 .65066845
1E-6 .79769935 1E-6 .66717192 1E-6 .73018516
1E-5 .4976363 1E-5 .36560855 1E-5 .57565269
1E-4 -.0199725 1E-4 .03253857 1E-4 .01753542
;;;
proc nlin data=one;
  parameters alpha0=-10 to 0 by 1 alpha1=-2 to 2 by 1 alpha2=-.5 to .5 by .1;
  model y=probnorm(-(alpha0+alpha1*x1+alpha2*(x1**2)));
  output out=three_2 p=predict;
run;

symbol1 color=black interpol=none value=dot;
symbol2 color=black interpol=join value=none;

proc gplot data=three_2;
plot y*x1=1 predict*x1=2/overlay;
run;
```

## APPENDIX B. SAS CODE TO FIT THE REDUCED MODEL

```
data two;
input x y @@;
x1=log10(x);
cards;
1E-14 .19157199 1E-14 .18106978 1E-14 .16006537
1E-13 .17056757 1E-13 .21707734 1E-13 .23658144
1E-12 .40611707 1E-12 .36110761 1E-12 .35360603
1E-11 .64166656 1E-11 .62516309 1E-11 .68667602
1E-10 .72868484 1E-10 .58015363 1E-10 .58015363
1E-9 .69117696 1E-9 .55764891 1E-9 .75118957
1E-8 .70768043 1E-8 .65066845 1E-8 .75118957
1E-7 .58015363 1E-7 .7046798 1E-7 .65066845
1E-6 .79769935 1E-6 .66717192 1E-6 .73018516
;;;
proc nlin data=two;
  parameters alpha0=-10 to 0 by 1 alpha1=-2 to 2 by 1 ;
  model y=probnorm(-(alpha0+alpha1*x1));
  output out=two_1 p=predict2;
run;

symbol1 color=black interpol=none value=dot;
symbol2 color=black interpol=join value=none;

proc gplot data=two_1;
plot y*x1=1 predict2*x1=2/overlay;
run;
```

## APPENDIX C. R-CODE FOR D-OPTIMAL DESIGN FOR THE COMPLETE MODEL

```

#Number of parameters
k=3
#Value of parameters
alpha=4.6359
beta=1.2327
gamma=0.0720

#Initial design
x0=c(-14,-9,-4)
n0=length(x0)
w=rep(1/n0,n0)
D=rbind(x0,w)
#Initial Information matrix
A<-rep(0,n0)
B<-rep(0,n0)
C<-rep(0,n0)
E<-rep(0,n0)
F<-rep(0,n0)
for (i in 1:n0)
{
A[i]=w[i]*exp(-1*(alpha+(beta*x0[i])+(gamma*x0[i]^2))^2)
B[i]=x0[i]*A[i]
C[i]=x0[i]^2*A[i]
E[i]=x0[i]^3*A[i]
F[i]=x0[i]^4*A[i]
}
M0=matrix(c(sum(A),sum(B),sum(C),sum(B),sum(C),sum(E),sum(C),sum(E),sum(F)),nrow=
3,ncol=3,byrow=F)

#Find dn
f<-function(x)
{matrix(c(exp((-1/2)*(alpha+beta*x+gamma*x^2))^2)
,x*exp((-1/2)*(alpha+beta*x+gamma*x^2))^2)
,x^2*exp((-1/2)*(alpha+beta*x+gamma*x^2))^2),nrow=3,ncol=1,byrow=F)}
#p=Difference between dn and k
p=1
while(p>.005){
#design space(x1)
x1=seq(-14,-4,.01)
n1=length(x1)
dn=rep(0,n1)
for (j in 1:n1)
{dn[j]=t(f(x1[j]))%%solve(M0)%%f(x1[j])}

#Choose x(n+1) that maximizes dn
for (j in 1:n1)

```

```

{if(max(dn)==dn[j])x1[j]=x1[j] else x1[j]=NA}
newX=na.omit(x1)
newdn=max(dn)

#Find alpha(n+1)
an=(newdn-k)/(k*(newdn-1))
p<-newdn-k
#Get M(n+1)
newM=(1-an)*M0+an*f(newX)%*%t(f(newX))
M0<-newM

#Design at each iteration
newW=(1-an)*D[2,]
W=c(newW,an)
X=c(D[1,],newX)
newD=rbind(X,W)
D=newD
print(p)}

#Summarize the result
D_optimal=by(D[2,],D[1,],FUN=sum)
D_optimal

#Verify D-optimal design
#Number of parameters
k=3
#Value of parameters
alpha=4.6359
beta=1.2327
gamma=0.0720

#D-optimal design
x=c(-12.73,-9.21,-7.91,-4.39)
n=length(x)
w=c(.33,.17,.17,.33)
D=rbind(x,w)
#Information matrix
A1<-rep(0,n)
B1<-rep(0,n)
C1<-rep(0,n)
E1<-rep(0,n)
F1<-rep(0,n)
for (i in 1:n)
{
A1[i]=w[i]*exp(-1*(alpha+(beta*x[i])+(gamma*x[i]^2))^2)
B1[i]=x[i]*A1[i]
C1[i]=x[i]^2*A1[i]
E1[i]=x[i]^3*A1[i]
F1[i]=x[i]^4*A1[i]
}

```

```
M=matrix(c(sum(A1),sum(B1),sum(C1),sum(B1),sum(C1),sum(E1),sum(C1),sum(E1),sum(F1)),nrow=3,ncol=3,byrow=F)
```

```
#Find dn
```

```
f<-function(x)
```

```
{matrix(c(exp((-1/2)*(alpha+beta*x+gamma*x^2)^2)
```

```
,x*exp((-1/2)*(alpha+beta*x+gamma*x^2)^2)
```

```
,x^2*exp((-1/2)*(alpha+beta*x+gamma*x^2)^2)),nrow=3,ncol=1,byrow=F)}
```

```
p=3
```

```
x1=seq(-14,-4,.01)
```

```
n1=length(x1)
```

```
dn=rep(0,n1)
```

```
for (j in 1:n1)
```

```
{dn[j]=t(f(x1[j]))%%solve(M)%%f(x1[j])}
```

```
plot(x1,dn)
```

## APPENDIX D. R-CODE FOR D-OPTIMAL DESIGN FOR THE REDUCED MODEL

```

#Number of parameters
k=2
#Value of parameters
alpha=-2.0381
beta=-0.1926

#Initial design
x0=c(-14,-6)
n0=length(x0)
w=rep(1/n0,n0)
D=rbind(x0,w)
#Initial Information matrix
A<-rep(0,n0)
B<-rep(0,n0)
C<-rep(0,n0)
for (i in 1:n0)
{
A[i]=(w[i]*exp(-1*(alpha+(beta*x0[i]))^2))
B[i]=x0[i]*A[i]
C[i]=x0[i]^2*A[i]
}
M0=matrix(c(sum(A),sum(B),sum(B),sum(C)),nrow=2,ncol=2,byrow=F)

#Find dn

f<-function(x)
{matrix(c(exp((-1/2)*(alpha+beta*x)^2)
, x*exp((-1/2)*(alpha+beta*x)^2)),nrow=2,ncol=1,byrow=F)}
#p=Difference between dn and k
p=1
while(p>.005){
#design space(x1)
x1=seq(-14,-6,.01)
n1=length(x1)
dn=rep(0,n1)
for (j in 1:n1)
{dn[j]=t(f(x1[j]))%%solve(M0)%%f(x1[j])}

#Choose x(n+1) that maximize dn
for (j in 1:n1)
{if(max(dn)==dn[j])x1[j]=x1[j] else x1[j]=NA}
newX=na.omit(x1)
newdn=max(dn)

#Find alpha(n+1)
an=(newdn-k)/(k*(newdn-1))

```

```

p<-newdn-k
#Get M(n+1)
newM=(1-an)*M0+an*f(newX)%*%t(f(newX))
M0<-newM

#Design at each iteration
newW=(1-an)*D[2,]
W=c(newW,an)
X=c(D[1,],newX)
newD=rbind(X,W)
D=newD
print(p)}

#Summarize the result
D_optimal=by(D[2,],D[1,],FUN=sum)
D_optimal

#Verify D-optimal design
#Number of parameters
k=2
#Value of parameters
alpha=-2.0381
beta=-0.1926

#D-optimal design
x=c(-14,-6.84)
n=length(x)
w=c(.5,.5)
D=rbind(x,w)
#Information matrix
A1<-rep(0,n)
B1<-rep(0,n)
C1<-rep(0,n)
for (i in 1:n)
{
A1[i]=(w[i]*exp(-1*(alpha+(beta*x[i]))^2))
B1[i]=x[i]*A1[i]
C1[i]=x[i]^2*A1[i]
}
M=matrix(c(sum(A1),sum(B1),sum(B1),sum(C1)),nrow=2,ncol=2,byrow=F)

#Find dn
f<-function(x)
{matrix(c(exp((-1/2)*(alpha+beta*x)^2)
, x*exp((-1/2)*(alpha+beta*x)^2)),nrow=2,ncol=1,byrow=F)}
p=3
x1=seq(-14,-6,.01)
n1=length(x1)
dn=rep(0,n1)

```

```
for (j in 1:n1)
{dn[j]=t(f(x1[j]))**solve(M)**f(x1[j])}
plot(x1,dn)
```

## APPENDIX E. R-CODE FOR ROBUST D-OPTIMAL DESIGN

```

#Number of parameters
ka=3
kb=2
#Value of parameters
alphaa=4.6359
betaa=1.2327
gammaa=0.0720
alphab=-2.0381
betab=-0.1926

#Weight of interest in model A
r=.5

#Initial design
x0=c(-14,-9,-4)
n0=length(x0)
w=rep(1/n0,n0)
D=rbind(x0,w)

#Initial Information matrix
A<-rep(0,n0)
B<-rep(0,n0)
C<-rep(0,n0)
E<-rep(0,n0)
F<-rep(0,n0)
G<-rep(0,n0)
H<-rep(0,n0)
I<-rep(0,n0)
for (i in 1:n0)
{
A[i]=w[i]*exp(-1*(alphaa+(betaa*x0[i])+(gammaa*x0[i]^2))^2)
B[i]=x0[i]*A[i]
C[i]=x0[i]^2*A[i]
E[i]=x0[i]^3*A[i]
F[i]=x0[i]^4*A[i]
G[i]=w[i]*exp(-1*(alphab+(betab*x0[i]))^2)
H[i]=x0[i]*G[i]
I[i]=x0[i]^2*G[i]
}
Ma0=matrix(c(sum(A),sum(B),sum(C),sum(B),sum(C),sum(E),sum(C),sum(E),sum(F)),nrow
=3,ncol=3,byrow=F)
Mb0=matrix(c(sum(G),sum(H),sum(H),sum(I)),nrow=2,ncol=2,byrow=F)

#Find dn
fa<-function(x)
{matrix(c(exp((-1/2)*(alphaa+betaa*x+gammaa*x^2))^2)
,x*exp((-1/2)*(alphaa+betaa*x+gammaa*x^2))^2)

```

```
,x^2*exp((-1/2)*(alphaa+betaa*x+gammaa*x^2)^2)),nrow=3,ncol=1,byrow=F}}
```

```
fb<-function(x)
{matrix(c(exp((-1/2)*(alphab+betab*x)^2)
,x*exp((-1/2)*(alphab+betab*x)^2)),nrow=2,ncol=1,byrow=F)}
```

```
#p=Difference between dn and k
```

```
p=1
```

```
q=1
```

```
t=2
```

```
while(q>.00001){
```

```
#design space(x1)
```

```
x1=seq(-14,-4,.1)
```

```
n1=length(x1)
```

```
dn=rep(0,n1)
```

```
for (j in 1:n1)
```

```
{dn[j]=((r/ka)*t(fa(x1[j]))%%solve(Ma0)%fa(x1[j]))+((1-
r)/kb)*t(fb(x1[j]))%%solve(Mb0)%fb(x1[j]))}
```

```
#Choose x(n+1) that maximize dn
```

```
for (j in 1:n1)
```

```
{if(max(dn)==dn[j])x1[j]=x1[j] else x1[j]=NA}
```

```
newX=na.omit(x1)
```

```
newdn=max(dn)
```

```
#Find alpha(n+1)
```

```
an=1/t
```

```
newp=(newdn-(r*ka+(1-r)*kb))
```

```
q<-(abs(p-newp))
```

```
p<-newp
```

```
#Get Mx(n+1)
```

```
newMa=(1-an)*Ma0+an*fa(newX)%t(fa(newX))
```

```
Ma0<-newMa
```

```
newMb=(1-an)*Mb0+an*fb(newX)%t(fb(newX))
```

```
Mb0<-newMb
```

```
#Design at each iteration
```

```
newW=(1-an)*D[2,]
```

```
W=c(newW,an)
```

```
X=c(D[1,],newX)
```

```
newD=rbind(X,W)
```

```
D=newD
```

```
t=t+1
```

```
print(q)}
```

```
#Summarize the result
```

```
D_optimal=by(D[2,],D[1,],FUN=sum)
```

```
D_optimal
```

```

#Verify D-optimal design
#Number of parameters
ka=3
kb=2
#Value of parameters
alphaa=4.6359
betaa=1.2327
gammaa=0.0720
alphab=-2.0381
betab=-0.1926

#D-optimal design
x=c(-12.9,-10.5,-7.5,-4.4)
n=length(x)
w=c(.4,.03,.29)
w=c(w,1-sum(w))
D=rbind(x,w)

#Information matrix
A<-rep(0,n)
B<-rep(0,n)
C<-rep(0,n)
E<-rep(0,n)
F<-rep(0,n)
G<-rep(0,n)
H<-rep(0,n)
I<-rep(0,n)
for (i in 1:n)
{
A[i]=w[i]*exp(-1*(alphaa+(betaa*x[i])+(gammaa*x[i]^2))^2)
B[i]=x[i]*A[i]
C[i]=x[i]^2*A[i]
E[i]=x[i]^3*A[i]
F[i]=x[i]^4*A[i]
G[i]=w[i]*exp(-1*(alphab+(betab*x[i]))^2)
H[i]=x[i]*G[i]
I[i]=x[i]^2*G[i]
}
Ma=matrix(c(sum(A),sum(B),sum(C),sum(B),sum(C),sum(E),sum(C),sum(E),sum(F)),nrow=
3,ncol=3,byrow=F)
Mb=matrix(c(sum(G),sum(H),sum(H),sum(I)),nrow=2,ncol=2,byrow=F)

fa<-function(x)
{matrix(c(exp((-1/2)*(alphaa+betaa*x+gammaa*x^2))^2)
,x*exp((-1/2)*(alphaa+betaa*x+gammaa*x^2))^2)
,x^2*exp((-1/2)*(alphaa+betaa*x+gammaa*x^2))^2),nrow=3,ncol=1,byrow=F)}

fb<-function(x)
{matrix(c(exp((-1/2)*(alphab+betab*x)^2)
,x*exp((-1/2)*(alphab+betab*x)^2)),nrow=2,ncol=1,byrow=F)}

```

```
p=3
x1=seq(-14,-4,.01)
n1=length(x1)
dn=rep(0,n1)
for (j in 1:n1)
{dn[j]=((r/ka)*t(fa(x1[j]))%%solve(Ma0)%*%fa(x1[j])+((1-
r)/kb)*t(fb(x1[j]))%%solve(Mb0)%*%fb(x1[j]))}
plot(x1,dn)
```

## APPENDIX F. R-CODE FOR RELATIVE EFFICIENCIES

```
#Number of parameters
ka=3
kb=2

#Value of parameters
alphaa=4.6359
betaa=1.2327
gammaa=0.0720
alphab=-2.0381
betab=-0.1926

#D-optimal Design for Reduced Model
xr=c(-14,-6.82)
nr=length(xr)
wr=c(.5,.5)

#Alternative D-optimal Design for Reduced Model
xra=c(-14,-6.84,-6.2)
nra=length(xra)
wra=c(.5,.25,.25)

#D-optimal Design for Complete Model
xc=c(-12.73,-9.21,-7.91,-4.39)
nc=length(xc)
wc=c(.33,.17,.17,.33)

#Robust D-optimal Design
x=c(-12.9,-10.5,-7.5,-4.4)
n=length(x)
w=c(.4,.03,.29,.28)

#Information matrix: Reduced Model, Reduced D-optimal Design
A<-rep(0,nr)
B<-rep(0,nr)
C<-rep(0,nr)
for (i in 1:nr)
{
A[i]=wr[i]*exp(-1*(alphab+(betab*xr[i]))^2)
B[i]=xr[i]*A[i]
C[i]=xr[i]^2*A[i]
}
RMRD=matrix(c(sum(A),sum(B),sum(B),sum(C)),nrow=2,ncol=2,byrow=F)

#Information matrix: Reduced Model, Robust Design
D<-rep(0,n)
E<-rep(0,n)
F<-rep(0,n)
for (i in 1:n)
```

```

{
D[i]=w[i]*exp(-1*(alphab+(betab*x[i]))^2)
E[i]=x[i]*D[i]
F[i]=x[i]^2*D[i]
}
RMRoD=matrix(c(sum(D),sum(E),sum(E),sum(F)),nrow=2,ncol=2,byrow=F)

#Information matrix: Reduced Model, Complete D-optimal Design
G<-rep(0,nc)
H<-rep(0,nc)
I<-rep(0,nc)
for (i in 1:nc)
{
G[i]=wc[i]*exp(-1*(alphab+(betab*xc[i]))^2)
H[i]=xc[i]*G[i]
I[i]=xc[i]^2*G[i]
}
RMCD=matrix(c(sum(G),sum(H),sum(H),sum(I)),nrow=2,ncol=2,byrow=F)

#Information matrix: Complete Model, Reduced D-optimal Design
J<-rep(0,nra)
K<-rep(0,nra)
L<-rep(0,nra)
M<-rep(0,nra)
N<-rep(0,nra)
for (i in 1:nra)
{
J[i]=wra[i]*exp(-1*(alphaa+(betaa*xra[i])+(gammaa*xra[i]^2))^2)
K[i]=xra[i]*J[i]
L[i]=xra[i]^2*J[i]
M[i]=xra[i]^3*J[i]
N[i]=xra[i]^4*J[i]
}
CMRD=matrix(c(sum(J),sum(K),sum(L),sum(K),sum(L),sum(M),sum(L),sum(M),sum(N)),nrow=3,ncol=3,byrow=F)

#Information matrix: Complete Model, Robust D-optimal Design
O<-rep(0,n)
P<-rep(0,n)
Q<-rep(0,n)
R<-rep(0,n)
S<-rep(0,n)
for (i in 1:n)
{
O[i]=w[i]*exp(-1*(alphaa+(betaa*x[i])+(gammaa*x[i]^2))^2)
P[i]=x[i]*O[i]
Q[i]=x[i]^2*O[i]
R[i]=x[i]^3*O[i]
S[i]=x[i]^4*O[i]
}

```

```
CMRoD=matrix(c(sum(O),sum(P),sum(Q),sum(P),sum(Q),sum(R),sum(Q),sum(R),sum(S)),n
row=3,ncol=3,byrow=F)
```

```
#Information matrix: Complete Model, Complete D-optimal Design
```

```
T<-rep(0,nc)
```

```
U<-rep(0,nc)
```

```
V<-rep(0,nc)
```

```
Y<-rep(0,nc)
```

```
Z<-rep(0,nc)
```

```
for (i in 1:nc)
```

```
{
```

```
T[i]=wc[i]*exp(-1*(alphaa+(betaa*xc[i])+(gammaa*xc[i]^2))^2)
```

```
U[i]=xc[i]*T[i]
```

```
V[i]=xc[i]^2*T[i]
```

```
Y[i]=xc[i]^3*T[i]
```

```
Z[i]=xc[i]^4*T[i]
```

```
}
```

```
CMCD=matrix(c(sum(T),sum(U),sum(V),sum(U),sum(V),sum(Y),sum(V),sum(Y),sum(Z)),nro
w=3,ncol=3,byrow=F)
```

```
#D-efficiency: Reduced model, Reduced Design
```

```
(det(RMRD)/det(RMRD))^(1/kb)
```

```
#D-efficiency: Reduced model, Complete Design
```

```
(det(RMCD)/det(RMRD))^(1/kb)
```

```
#D-efficiency: Reduced model, Robust Design
```

```
(det(RMRoD)/det(RMRD))^(1/kb)
```

```
#D-efficiency: Complete model, Reduced Design
```

```
(det(CMRD)/det(CMCD))^(1/ka)
```

```
#D-efficiency: Completemodel, Complete Design
```

```
(det(CMCD)/det(CMCD))^(1/ka)
```

```
#D-efficiency: Completemodel, Robust Design
```

```
(det(CMRoD)/det(CMCD))^(1/ka)
```