D-OPTIMAL DESIGN FOR THE 5PL-1P MODEL IN CHEMICAL TOXICITY ASSESSMENT

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

By

Jenna Lynn MacDonald

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

Major Department:
Statistics

May 2016

Fargo, North Dakota

North Dakota State University
Graduate School

Title

D-OPTIMAL DESIGN FOR THE 5PL-1P MODEL
IN CHEMICAL TOXICOLOGY ASSESSMENT

By

Jenna MacDonald

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. Yarong Yang

Dr. Wonwoo Byun

Approved:

May 9, 2016
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

The five-parameter logistic minus one-parameter model is a hybrid between the five-parameter model and the four-parameter model used for the relationship between concentration and response. The four-parameter model includes the maximum and minimum concentration, slope, and the median concentration level EC_{50}. The five-parameter model add an asymmetric factor which is important due to asymmetry of the sigmoid curve. This model, however, is more difficult to fit due to the addition of the fifth parameter, which is why the 5PL-1P model is used so that the asymmetric factor is taken into account but has less parameters. For the 5PL-1P model, D-optimal designs are obtained to estimate the model parameters effectively. Then we compare the D-optimal designs to the designs that are used to study the 5PL-1P model in real toxicity assessment and show that they work better than the original designs by comparing their efficiencies and comparing MSEs through simulation studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

[5PL-1P].....Five parameter model minus one parameter

[5PL] .....Five-parameter logistic

[4PL] .....Four-parameter logistic

[BRAN].....Bromoacetonitrile

[CLAN].....Chloroacetonitrile

1. INTRODUCTION

Experimental design is imperative when planning an experiment, particularly in science and medical research. Research in these areas are notorious for their high costs, and in medical research, the potential harm to its subjects. An optimal design can greatly minimize these and still provide accurate results and potentially even save time. However, an optimal design is based off of an original design on which it improves upon and thus, is not the initial design of the study. This is important because, even though an initial experiment is conducted with a chosen design, future experiments can be done more effectively. One popular experiment in biostatistics is the study between a dose or concentration and its response. There are a variety of models used to study this relationship, each of which takes on slightly different parameters or number of parameters.

The five-parameter logistic minus one-parameter model for the relationship between concentration level and response is the focus of this paper and is a hybrid between both the five-parameter model and the four-parameter model. The five-parameter logistic model takes into account the minimum dose, maximum dose, slope, median effective dose concentration, and the asymmetric factor, but due to its five parameters, is more difficult to fit to a model. The four-parameter logistic model is much easier to fit, however, it does not take the asymmetric factor into account. This is a problem due to the classic sigmoid curve being associated with logistic models that are not always symmetric. The five-parameter logistic minus one-parameter model has four parameters but differs from the four-parameter logistic function by having an asymmetric parameter instead of the minimum parameter found in the 4PL (Dawson et al., 2012). This is still a relatively new model with very little research.

Dawson et al. (2012) studied the differences between the 4PL and 5PL-1P model in relationship to various chemicals, each with seven to eight concentration levels in the design, to see the effect of adding in an asymmetric factor. In the past, it has been assumed that the curves associated with the concentration-response relationship are symmetric. However, there may be new evidence that some curves may be asymmetric, which can affect the model results. The 4PL and 5PL-1P models each have four parameters, the only difference being the minimum dose in the 4PL model and the asymmetric factor in the 5PL-1P model. In the study, it was found that the 5PL-1P models had higher r^2 and median effective concentration EC_{50} values, and lower residual sum of square values. This all indicates that the addition of an asymmetric factor does impact the results, and due to the high r^2 and low residual sum of squares, it may provide more accurate results than the model without this parameter.

One important research area, which pertains to the data in this paper, is toxicology, or the research of harmful substances. The two compounds that were chosen from the 72 original datasets are bromoacetonitrile, abbreviated as BRAN, and chloracetonitrile, abbreviated as CLAN (Dawson et al., 2010). Each of these chemicals have three separate exposure times: 15, 30, and 45 minutes, which creates a total of six separate scenarios or datasets. These chemicals are considered haloacetonitriles which are the by-products from disinfecting water and is known to have an impact on DNA (Bromoacetonitrile, 2016). The tests were conducted using a machine called the Microtox® which utilizes bioluminescent bacteria in order to determine the effect of the bromoacetonitrile and chloracetonitrile compounds (Dawson et al., 2010). These two chemicals, which were previously in Dawson et al. (2012), used to analyze the addition of an asymmetric factor in the model, are not the only chemicals that have exhibited asymmetry. Another study, which looked at the toxicity of insecticides on a biological system, found an

asymmetric sigmoid curve for the relationship between insecticide level and percentage of insect mortality (Yu, 2015). A nutritional study regarding the relationship between xenobiotic concentration and response, also presented a sigmoid curve that was not symmetric (Hathcock, 1982). Because some toxicology studies are proving to have some asymmetry, the 5PL-1P model could have a large impact in the future.

The D-optimal design for the 5PL-1P model is studied in this paper due to the interest in finding the parameter estimates. This is found by maximizing the determinant of the Fisher information matrix of the model parameters. The D-optimal design is then compared to the original design in the motivational study (Dawson et al., 2012) to see how the D-optimal design works better than the original design for studying the 5PL-1P model. This will be evaluated by looking at the efficiency between the two designs and then with the parameter mean squared errors from simulations for both the D-optimal and original design. The D-optimal design could provide a design with fewer concentration levels, which could provide more accurate results at a smaller cost. Chapter 2 will go through the optimality criterion, more specifically the D-optimality criterion, then the general equivalence theorem and the V-algorithm which will be used in an algorithm to find the D-optimal designs. Chapter 3 will discuss this model further and provide the parameter estimates for all six combinations of the two chemicals and the three exposure times. In chapter 4, we will find the D-optimal designs for the 5PL-1P model using an algorithm that encompasses the V-algorithm, general equivalence theorem, and the Newton Raphson method. In chapter 5, we will examine the efficiency of the D-optimal designs over the original designs. In chapter 6, there will be a simulation of the study using the D-optimal designs and original designs, and their mean square errors will be compared. Finally, a conclusion will be given at the end.

2. BACKGROUND

This section will go over the background of optimal designs, more specifically D-optimal designs, that will be later used in this paper, in regards to a toxicology study. In toxicology, researchers focus their studies on hazardous chemicals. Optimal designs are used in order to estimate parameters while reducing sample size and minimizing the variance for each parameter. This is important because when researchers are setting up an experiment it allows for a smaller sample to be used which lowers costs while restricting the usage of these harmful chemicals that are being tested.

2.1. Optimal Design Criteria

Using an optimal design allows a researcher to use smaller samples with a greater precision in estimating the parameters. This can reduce costs and time significantly. A design is represented by

$$\xi = \begin{pmatrix} x_i \\ w_i \end{pmatrix}_{i=1}^{k}$$

where x_i refers to the ith concentration level and $w_i = \frac{n_i}{N}$ is the weight for each of those concentrations. In real studies, the closest integer to $w_i N$ will become the replication size n_i for the ith concentration level. Optimal designs minimize or maximize the optimality criterion $\psi\{M(\xi, \boldsymbol{\theta})\}$, where $M(\xi, \boldsymbol{\theta})$ denotes the Fisher information matrix for the parameters $\boldsymbol{\theta}$.

2.2. D-optimal Design

D-optimal design is the focus of this paper and is often used to estimate the parameters of a model. This design is beneficial due to it minimizing the variance for the models' parameters in some sense. The idea of D-optimal designs is to maximize the determinant of the Fisher information matrix. The D-optimality criterion is

$$\psi = \max_{\xi} |M(\xi, \boldsymbol{\theta})|$$

2.3. The General Equivalence Theorem

The general equivalence theorem is use to validate optimal designs and in this paper's case, specifically the D-optimal design (Kiefer, 1958). Using the Fisher information matrix $M(\xi, \boldsymbol{\theta})$, the directional derivative of convex function $\psi\{\ \}$ with regards to $\bar{\xi}$ is

$$\phi(x_i, \xi) = \lim_{\alpha \to 0^+} \frac{1}{\alpha} \left[\psi\left\{(1-\alpha)M(\xi, \boldsymbol{\theta}) + \alpha M\left(\bar{\xi}, \boldsymbol{\theta}\right)\right\} - \psi\{M(\xi, \boldsymbol{\theta})\}\right]$$

where ξ represents the design matrix and $\boldsymbol{\theta}$ is the parameter vector. To solve this, first the derivative of the design matrix must be calculated by

$$\xi' = (1-\alpha)\xi + \alpha\bar{\xi}$$

where $\bar{\xi}$ is the measure on which unit mass is put at the point x_i. When the derivative of the design matrix ξ' is put into the Fisher information matrix instead of just the design ξ, then the following is obtained

$$M(\xi', \boldsymbol{\theta}) = (1-\alpha)M(\xi, \boldsymbol{\theta}) + \alpha M\left(\bar{\xi}, \boldsymbol{\theta}\right)$$

This theorem has the following assumptions for a given design ξ^*:

1. ξ^* minimizes $\psi\{M(\xi, \boldsymbol{\theta})\}$.
2. ξ^* maximizes the minimum concentration x_i over the design space $\phi(x_i, \xi)$.
3. The minimum over the design space $\phi(x_i, \xi)$ equals zero when the points support the given design ξ^*.

This theorem will later be applied to the V-algorithm.

2.4. V-Algorithm

The V-algorithm is an iterative method that is often used to calculate D-optimal designs and was introduced by Fedorov, Klimko, and Studden (1972). This algorithm starts with some

initial design ξ_0. For the initial design ξ_0, a uniform design with $s+1$ concentration levels can be used. Here s represents the number of parameters in the model. At the nth iteration, this algorithm maximizes the standardized variance d_n by following the General Equivalence Theorem,

$$d_n(x, \boldsymbol{\theta}) = g(x)^T M^{-1}(\xi_n, \boldsymbol{\theta}) g(x)$$

where $x \in \chi$ and $g(x)$ is the gradient of the mean response function. A point x is chosen from the design space χ so that it maximizes the standardized variance

$$\bar{d}_n(x, \boldsymbol{\theta}) = \max_x d_n(x, \boldsymbol{\theta})$$

The chosen point x will then be used in the next iteration. After each iteration, the Fisher information matrix is updated:

$$M(\xi_{n+1}, \boldsymbol{\theta}) = (1 - \alpha_{n+1}) M_n(\xi_n, \boldsymbol{\theta}) + \alpha_{n+1} g(x) g(x)^T$$

where $\alpha_{n+1} = \frac{1}{n+1}$. The iterations will stop and ξ_n becomes the D-optimal design when

$$d_n(x, \boldsymbol{\theta}) = g(x)^T M^{-1}(\xi_n, \boldsymbol{\theta}) g(x) - s \leq \varepsilon$$

where s is the number of model parameters and $\varepsilon = 10^{-6}$.

3. MODEL

In order to obtain D-optimal designs for nonlinear models, the parameters of the model will need to first be estimated. This section will cover the five-parameter minus one-parameter model and its parameter estimates based on the six combinations of chemical compounds and exposure times as well as its Fisher information matrix.

3.1. The General Model

The five-parameter logistic minus one-parameter function differs from the four-parameter logistic function by having an asymmetric parameter instead of the minimum parameter found in the 4PL (Dawson et al., 2012). The continuous response (effect) for the ith concentration level and jth replication is modeled by

$$Y_{ij} = f(x_i, \boldsymbol{\theta}) + \varepsilon_{ij}$$

where

$$\varepsilon_{ij} \sim N(0, \sigma^2)$$

$$i = 1, 2, \dots, k$$

$$j = 1, 2, \dots, n_i$$

$$\sum_{i=1}^{k} n_i = N$$

and the variance σ^2 is assumed to be unknown. The mean response at the ith concentration level is following the 5PL-1P model:

$$f(x_i, \boldsymbol{\theta}) = \frac{\theta_1}{\left[1 + \left(\frac{\theta_2}{x_i}\right)^{\theta_3}\right]^{\theta_4}}$$

where $\theta_1 =$ maximum effect, $\theta_3 =$ slope, $\theta_4 =$ asymmetric factor, and $x_i = i$th concentration level. The parameter θ_2 represents a function of EC_{50}:

$$\theta_2 = f(EC_{50}) = EC_{50} * 10^{\left[\frac{1}{\theta_3}\log\left(2^{1/\theta_4}-1\right)\right]}$$

that when the asymmetric factor $\theta_4 = 1$, then $\theta_2 = EC_{50}$. The EC_{50} is the median concentration level and represents the concentration level corresponding to 50% of the difference between the maximum and minimum effect.

3.2. Model Fitting

In order to show the dose-response relationship under the 5PL-1P model, the model was fit to the dataset collected in Dawson et al. (2010). This was done using the *nls* function in R, which is often used to fit nonlinear models. Among the many different chemicals, two chemical compounds were used: BRAN and CLAN, and the dose response was recorded at three different times: 15, 30, and 45 minutes, there will end up being six different sets of parameter estimates corresponding to all the combinations of chemical compound and time recorded. Each concentration level had two replications so the sample size for each of the six scenarios is 14. The design that was originally used in the Dawson et al. (2010) is in Table 1, where the top row represents the concentration levels for each compound and the bottom row represents the weight for each of those levels. Since the weight is the same for each concentration level, the same number of replications will be used for each.

Table 1. Original Designs for the Two Compounds.

$$\xi_O^{BRAN} = \begin{pmatrix} 0.1655 & 0.3089 & 0.5765 & 1.0762 & 2.0089 & 3.75 & 7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \end{pmatrix}$$

$$\xi_O^{CLAN} = \begin{pmatrix} 8.273 & 15.44 & 28.83 & 53.81 & 100.5 & 187.5 & 350 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \end{pmatrix}$$

The *nls* function in R requires estimated starting values for the parameters. The starting values for the BRAN concentration in the nls function would be $\theta_2 = 1$, $\theta_3 = 1$, $\theta_4 = 1$, and

then the maximum concentration θ_1 came from the maximum dose response for each time. The parameter estimates for BRAN are in Table 2.

Table 2. Parameter Estimates for the Compound BRAN.

Exposure Time (minutes)	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
15	128.1528	2.3244	0.9791	1.5470
30	103.2062	1.6336	1.5402	0.8235
45	100.97883	1.08130	1.70242	0.71926

Since θ_2 represents a function of the median concentration level, EC_{50}, the parameter estimates were used in order to calculate the EC_{50} as well and displayed in Table 3.

Table 3. EC_{50} Estimates for the Compound BRAN.

Exposure Time (minutes)	Median Concentration Level EC_{50}
15	4.162388
30	1.363926
45	0.8140715

The starting values for the CLAN concentration in the nls function would be $\theta_2 = 65$, $\theta_3 = 1$, $\theta_4 = 1$, and then the maximum concentration θ_1 coming from the maximum dose response for each time. Due to the difference range of dose response for the CLAN compound, the second parameter θ_2 had to increase. All values in the dose range where tested to find the optimum starting value for this parameter and it was found that 65 was the best value to use for all three times. The parameter estimates for CLAN are in Table 4.

Table 4. Parameter Estimates for the Compound CLAN.

Exposure Time (minutes)	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
15	105.7901	204.3502	1.5494	0.8279
30	100.78867	119.55175	1.89378	0.56313
45	100.73194	75.21709	1.87647	0.54536

The EC_{50} values were also calculated for the CLAN compound in Table 5.

Table 5. EC_{50} Estimates for the Compound CLAN.

Exposure Time (minutes)	Median Concentration Level EC_{50}
15	171.2787
30	74.89999
45	45.5369

3.3. Checking the Model Fit

After the parameter estimates were found, it was essential to check how those estimates fit the data. Figures 1 to 6 are plots of each compound and exposure time combination for the dose concentration, represented by *x*, and the dose response, represented by *y*, with the corresponding estimated model. It is clear by the plots that all six models are well-fit as the data closely follow each curve. One thing to note however is the change in the curves as the exposure time increases. Figures 1 and 3 for the exposure time of 15 minutes show only slight curvature in the fitted line. As the time increases to 30 minutes, the fitted line takes on a more defined curve, as seen in Figures 2 and 4. By the time of 45 minutes, the fitted lines in Figures 3 and 5 have a severe curvature. It is assumed that these parameter estimates are reasonable parameter values for the 5PL-1P model and will be used to find D-optimal designs later.

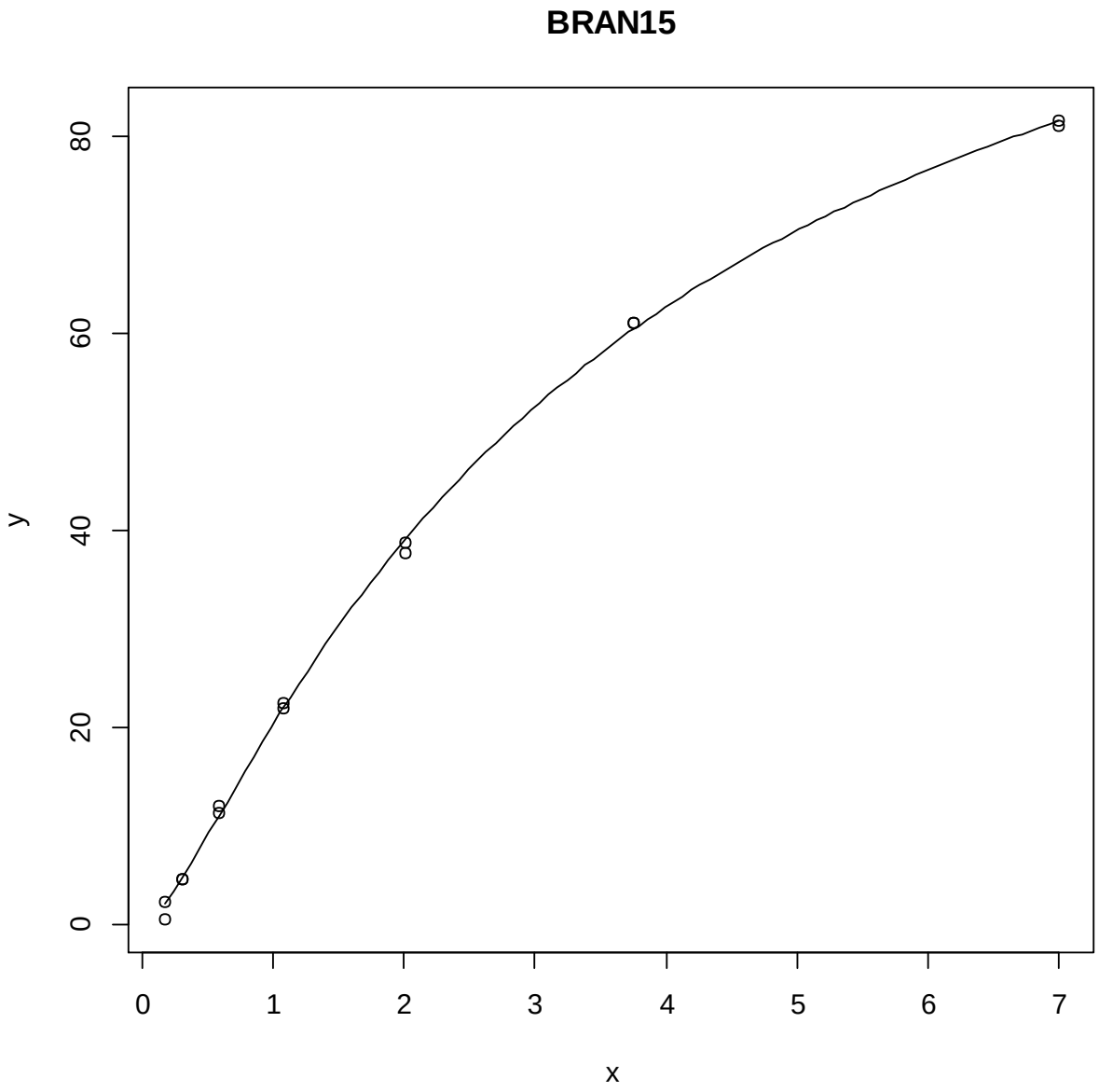

Figure 1. Fitted Curve for BRAN at Exposure Time 15.

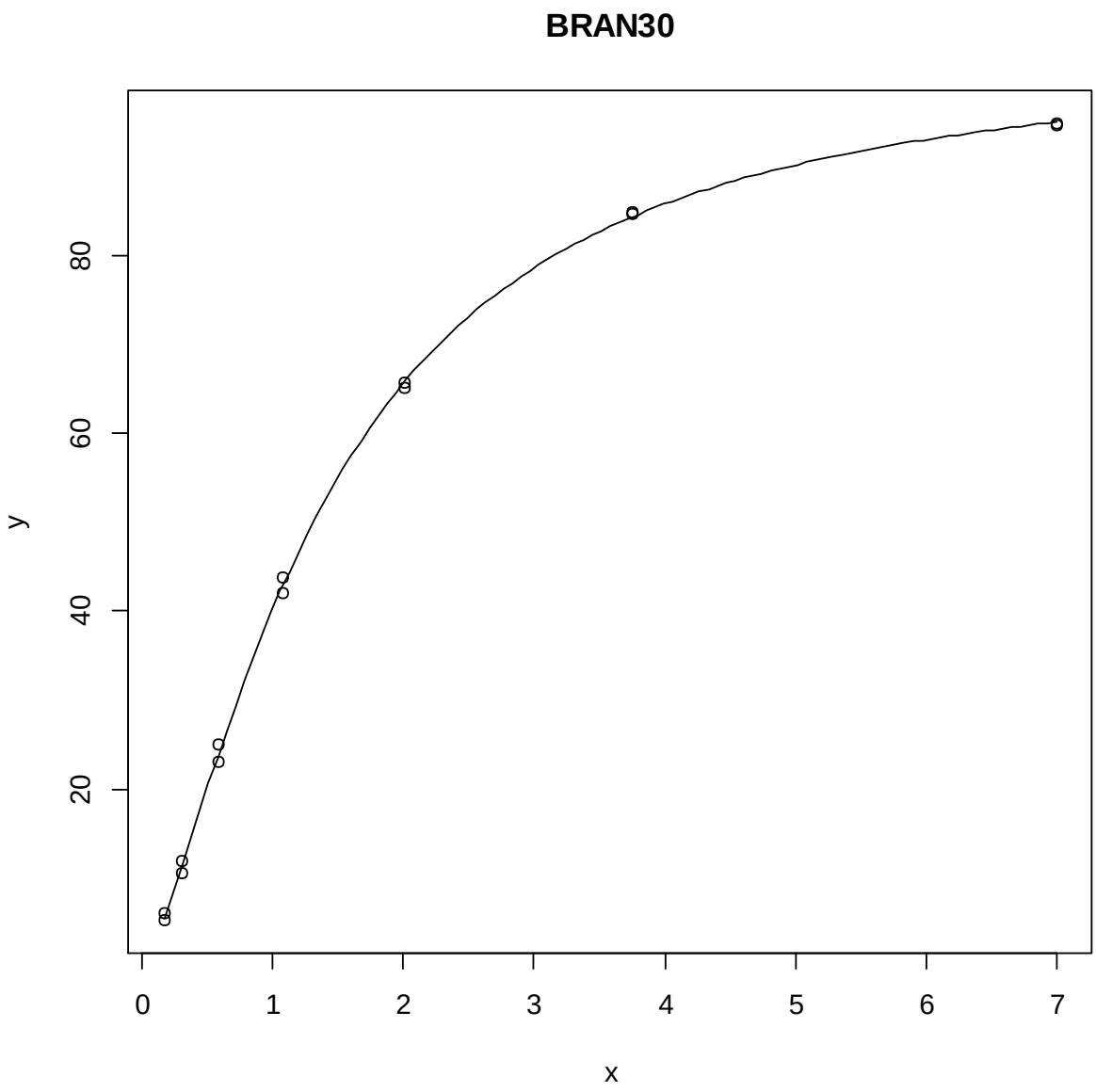

Figure 2. Fitted Curve for BRAN at Exposure Time 30.

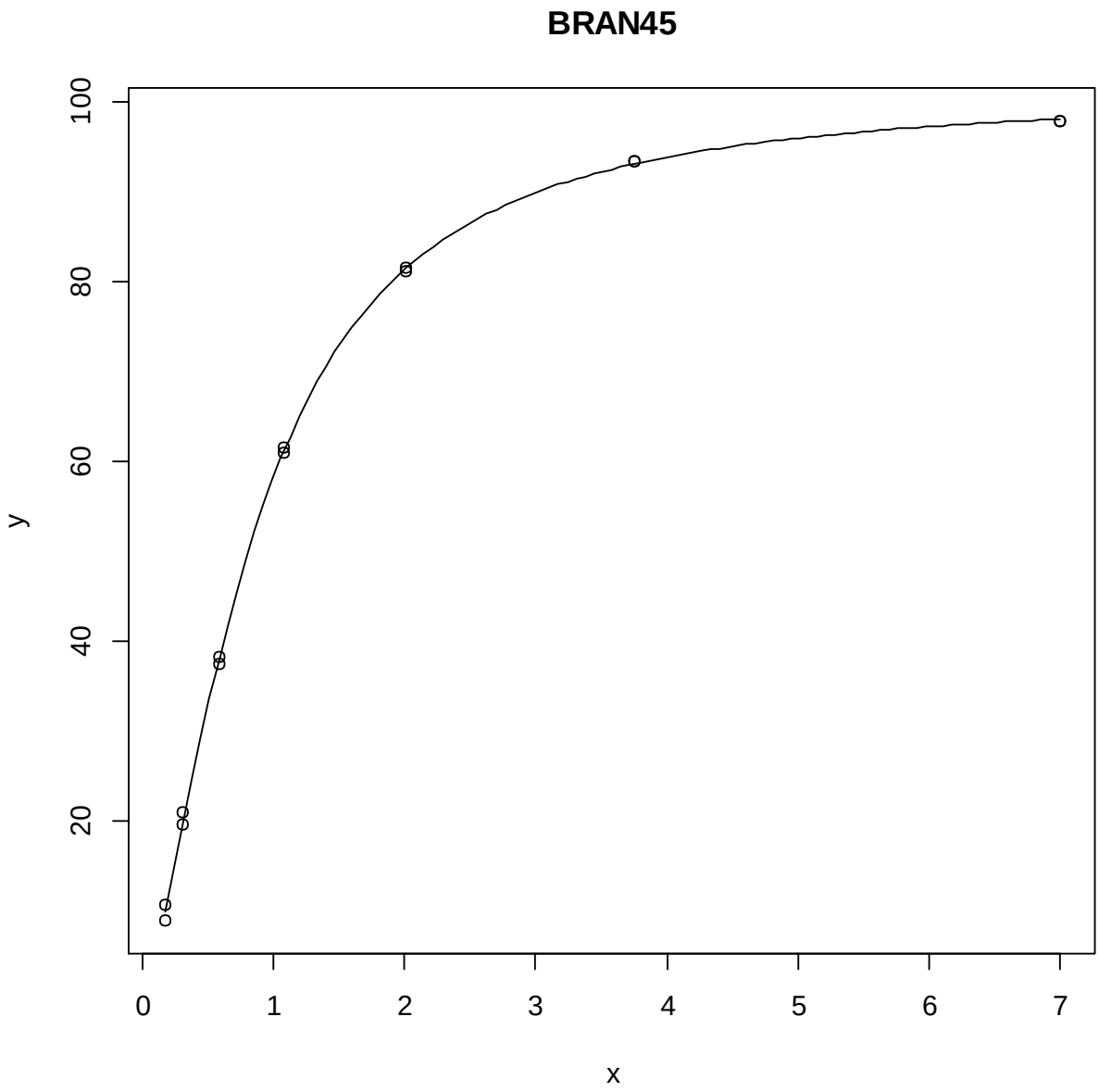

Figure 3. Fitted Curve for BRAN at Exposure Time 45.

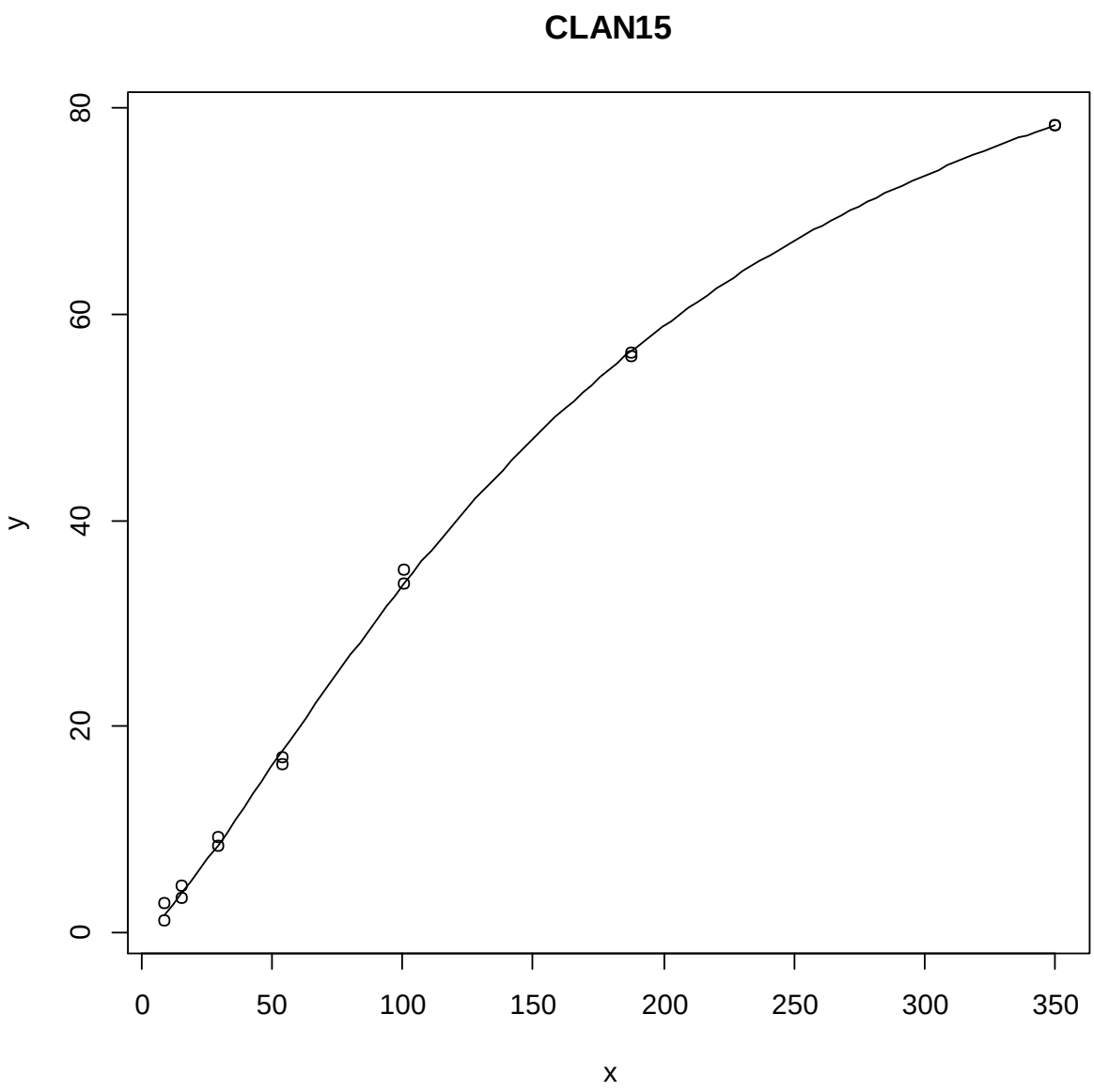

Figure 4. Fitted Curve for CLAN at Exposure Time 15.

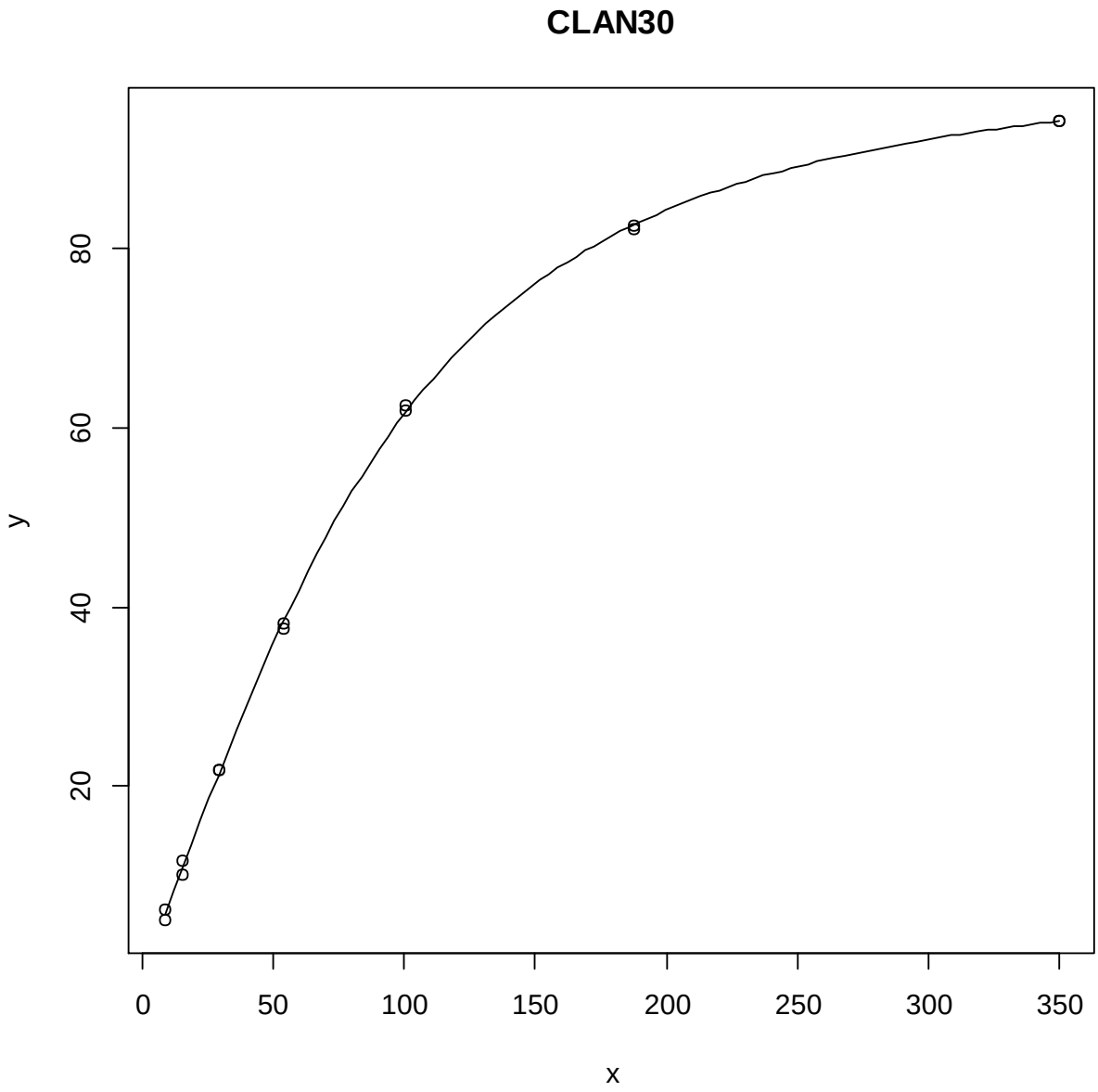

Figure 5. Fitted Curve for CLAN at Exposure Time 30.

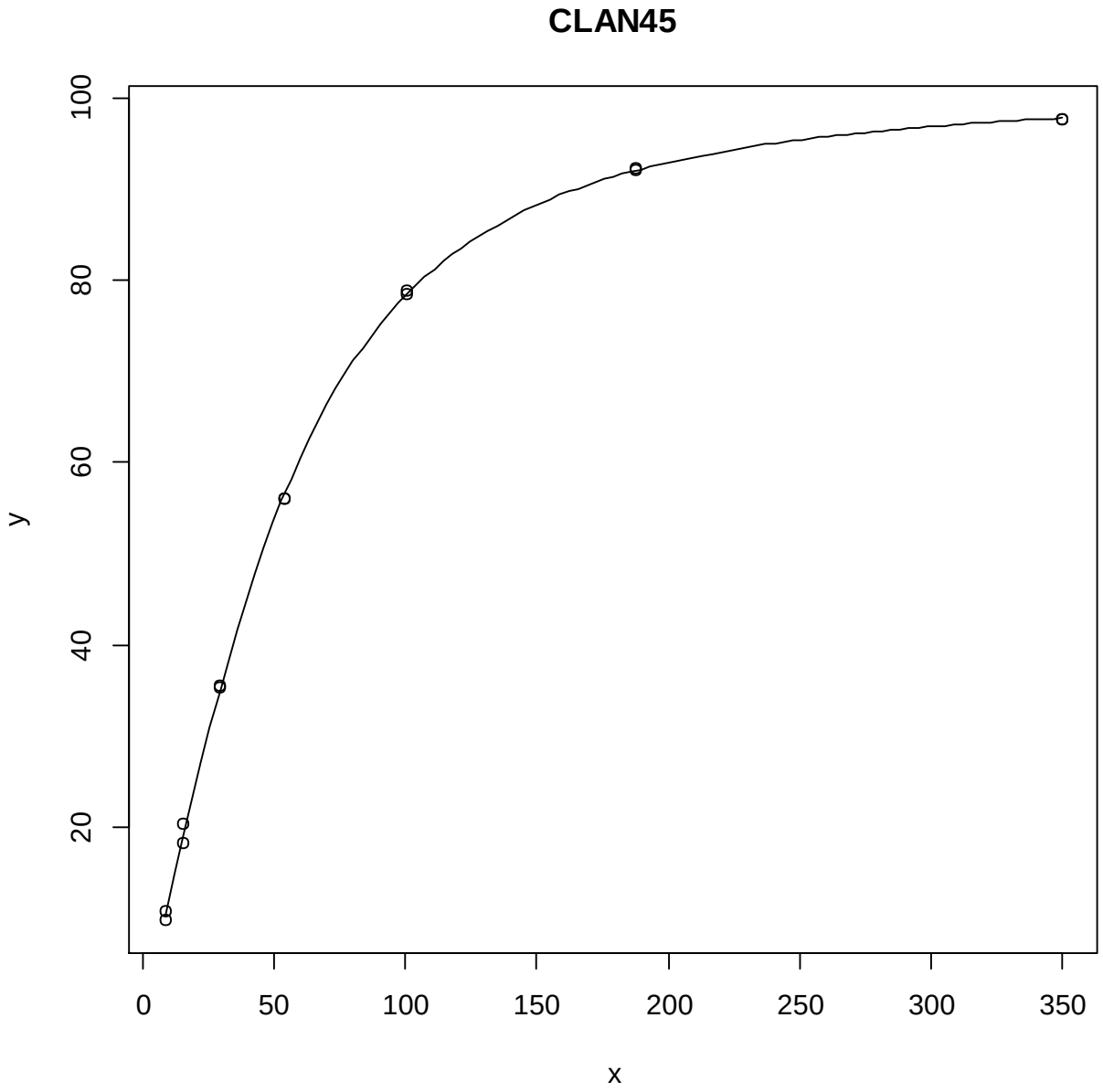

Figure 6. Fitted Curve for CLAN at Exposure Time 45.

3.4. Fisher Information Matrix

The Fisher information matrix is an essential component in finding the D-optimal design. The initial step to finding the Fisher information matrix is by calculating the gradient of the mean function $f(x_i, \boldsymbol{\theta})$:

$$g(x_i) = \left(\frac{f(x_i, \boldsymbol{\theta})}{d\theta_1} \quad \frac{f(x_i, \boldsymbol{\theta})}{d\theta_2} \quad \frac{f(x_i, \boldsymbol{\theta})}{d\theta_3} \quad \frac{f(x_i, \boldsymbol{\theta})}{d\theta_4} \right)^T$$

$$= \begin{pmatrix} \dfrac{1}{\left[1 + \left(\frac{\theta_2}{x_i}\right)^{\theta_3}\right]^{\theta_4}} \\ \dfrac{-\theta_1\theta_3\theta_4 \left(\frac{\theta_2}{x_i}\right)^{\theta_3} \left[1 + \left(\frac{\theta_2}{x_i}\right)^{\theta_3}\right]^{\theta_4}}{\theta_2} \\ -\theta_1\theta_4 \left(\frac{\theta_2}{x_i}\right)^{\theta_3} log\left(\frac{\theta_2}{x_i}\right) \left[1 + \left(\frac{\theta_2}{x_i}\right)^{\theta_3}\right]^{-\theta_4-1} \\ -\theta_1\theta_4 \left(\frac{\theta_2}{x_i}\right)^{\theta_3} \left[1 + \left(\frac{\theta_2}{x_i}\right)^{\theta_3}\right]^{-\theta_4} \log\left[1 + \left(\frac{\theta_2}{x_i}\right)^{\theta_3}\right] \end{pmatrix}$$

It is then used so that the Fisher information matrix,

$$M(\xi, \boldsymbol{\theta}) = \sum w_i g(x_i)\, g(x_i)^T$$

under the five-parameter minus one-parameter model on page 6, can be obtained, where *i* corresponds to each concentration level and ξ is all values of weighted values w_i and concentrations x_i. Note that this form is only true when the error ε follows a Normal distribution with a mean of zero and a variance of σ^2. The Fisher information matrix plays an important role to obtain the D-optimal designs since the D-optimal design maximizes the determinant of the Fisher information matrix. This matrix can be seen on the next page.

$$M(\xi,\theta)=\sum w_i\begin{bmatrix}
\frac{1}{\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{2\theta_4}} & \frac{-\theta_1\theta_3\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-1}}{\theta_2} & -\theta_1\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\log\left(\frac{\theta_2}{x}\right)\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-1} & -\theta_1\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4}\log\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right] \\
\frac{-\theta_1\theta_3\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-1}}{\theta_2} & \left(\frac{-\theta_1\theta_3\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-\theta_4-1}}{\theta_2}\right)^2 & \theta_1^2\theta_3\theta_4^2\left(\frac{\theta_2}{x}\right)^{2\theta_3}\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-2} & \frac{\theta_1^2\theta_3\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-2}\log\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]}{\theta_2} \\
-\theta_1\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\log\left(\frac{\theta_2}{x}\right)\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-1} & \theta_1^2\theta_3\theta_4^2\left(\frac{\theta_2}{x}\right)^{2\theta_3}\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-2} & \left(-\theta_1\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\log\left(\frac{\theta_2}{x}\right)\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-\theta_4-1}\right)^2 & \theta_1^2\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\log\left(\frac{\theta_2}{x}\right)\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-2}\log\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right] \\
-\theta_1\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4}\log\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right] & \frac{\theta_1^2\theta_3\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-2}\log\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]}{\theta_2} & \theta_1^2\theta_4\left(\frac{\theta_2}{x}\right)^{\theta_3}\log\left(\frac{\theta_2}{x}\right)\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-2\theta_4-2}\log\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right] & \left(-\theta_1\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]^{-\theta_4}\log\left[1+\left(\frac{\theta_2}{x}\right)^{\theta_3}\right]\right)^2
\end{bmatrix}$$

4. D-OPTIMAL DESIGN

This section will look at the D-optimal designs for the six combinations of the chemical compound and the exposure times in the 5PL-1P model. The D-optimal designs is used very efficiently when the goal is to estimate the model parameters so the researcher can study the dose-response relationship. But in order to find the D-optimal designs, for nonlinear models, the model parameter values $\boldsymbol{\theta}$ must be known or prespecified because the Fisher information matrix for nonlinear models strongly depends on the parameter values $\boldsymbol{\theta}$. So here, it is assumed that the parameter estimates from the six scenarios are the true values of $\boldsymbol{\theta}$ and are used to find the D-optimal designs for the 5PL-1P model.

D-optimal designs are found by maximizing the Fisher information matrix of the 5PL-1P Model. So the D-optimal design is $\xi_D = \max_{\xi} |M(\xi, \boldsymbol{\theta})|$, and the general equivalence states that ξ_D is the D-optimal design if and only if

$$\frac{g(x)^T M^{-1}(\xi_D, \boldsymbol{\theta}) g(x)}{4} \leq 1$$

where the equality holds when x is one of the concentration levels in ξ_D. This equivalence theorem is used in an algorithm to obtain the D-optimal design numerically. In this study, the algorithm in Hyun, Wang, and Yang (2015) is used in R and it utilizes the V-algorithm and the Newton Raphson Method. They modified the YBT algorithm created by Yang, Biedermann and Tang (2013), which finds optimal designs for nonlinear models very efficiently. However, the YBT algorithm can be a slow procedure when the initially selected concentration levels are far from the optimal concentration levels. The modified algorithm uses the V-algorithm in order to select concentration levels closer to the optimal levels, which speeds up the process.

Here I will introduce a brief idea about the modified algorithm. The modified algorithm begins with the V-algorithm which is used to find the initial concentration levels. This is done

by running the V-algorithm r times, where $r > s + 1$, and then selecting the last $s + 1$ distinct concentration levels among all selected concentration levels, where r is a prespecified number, such as 10 or 30, and s represents the number of parameters. Since there are four parameters in the 5PL-1P model, the last five concentration levels will be used as the initial concentration levels and will move on to the next step. This next step is the Newton Raphson Method, which will use those initial concentration levels to find the optimum weights corresponding to them.

$$\xi = \begin{pmatrix} x_1 & \dots & x_{s+1} \\ w_1^* & \dots & w_{s+1}^* \end{pmatrix}$$

It is good to note that some concentration values may drop out of the design for having a weight close to zero. It will then find a value of x_* such that

$$\max_{x_*} \frac{g(x)^T M^{-1}(\xi, \boldsymbol{\theta}) g(x)}{4}$$

where $g(x)$ represents the gradient of the mean response function and $M^{-1}(\xi, \boldsymbol{\theta})$ is the inverse of the Fisher information matrix using the initial design. The final step is to see if, by the general equivalence theorem,

$$\frac{g(x_*)^T M^{-1}(\xi, \boldsymbol{\theta}) g(x_*)}{4} - 1 \leq \varepsilon$$

where ε is equal to 10^{-6}. If this is satisfied, then the iteration will stop and the resulting design ξ will equal the D-optimal design ξ_D. However, if this is not satisfied, then it will go back to the Newton Raphson Method with the x^* value added to the design,

$$\xi = \begin{pmatrix} x_1 & \dots & x_{s+1} & x_* \\ w_1^* & \dots & w_{s+1}^* & w_*^* \end{pmatrix}$$

and find another concentration value to maximize the formula above. This will continue until the condition is satisfied.

The original designs from Douglas Dawson (2007) consisted of seven concentration levels with only two different designs: one for the BRAN compound and one for the CLAN compound. The design did not change for the three exposure times. Under the given parameter values in Tables 2 and 4, the D-optimal designs for the 5PL-1P model have four concentration levels with equal weight. Since the selected concentration levels were changed by the different parameter values, each of the six different combinations of the chemical compound and exposure time generated its own unique D-optimal design. The resulting D-optimal designs are in Table 6 where the top row corresponds to the concentration levels and the bottom row correspond to their given weight.

Table 6. D-Optimal Designs for the Six Combinations.

$\xi_D^{BRAN-15} = \begin{pmatrix} 0.33 & 1.33 & 3.78 & 7.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{CLAN-15} = \begin{pmatrix} 24.0 & 90.8 & 212.7 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{BRAN-30} = \begin{pmatrix} 0.26 & 1.01 & 2.84 & 7.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{CLAN-30} = \begin{pmatrix} 14.7 & 63.9 & 161.7 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{BRAN-45} = \begin{pmatrix} 0.18 & 0.70 & 2.03 & 7.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{CLAN-45} = \begin{pmatrix} 9.8 & 42.1 & 116.8 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$

For example, for the chemical compound BRAN at a 15-minute exposure time, the D-optimal design assigns 25% of the trials conducted to each of the concentration levels: 0.33, 1.33, 378, and 7.00. This will then minimize the variance of estimating the model parameters $\boldsymbol{\theta}$ so the dose-response relationship can be accurately estimated.

To verify that these designs follow the D-optimality criteria, their sensitivity functions,

$$d(x_i, \boldsymbol{\theta}) = g(x_i)^T M^{-1}(\xi, \boldsymbol{\theta}) g(x_i)$$

where ξ_D is the obtained D-optimal design, are plotted in Figures 7 to 12. The maxima of $d(x_i, \boldsymbol{\theta})$ in $x_i \, \epsilon \, \chi$ is bounded horizontally above by one. So the curve in each of these plots reaches their peak at the concentration levels x_i in their D-optimal design.

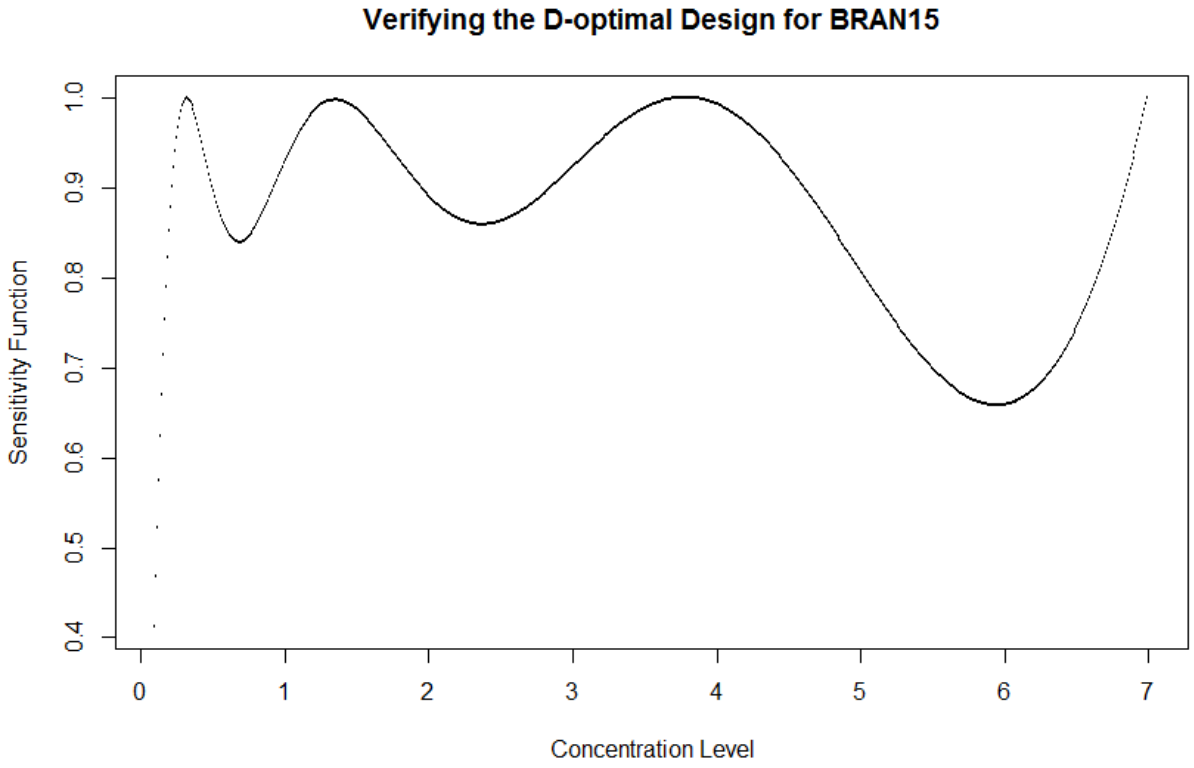

Figure 7. Sensitivity Function Plot for BRAN at Exposure Time 15.

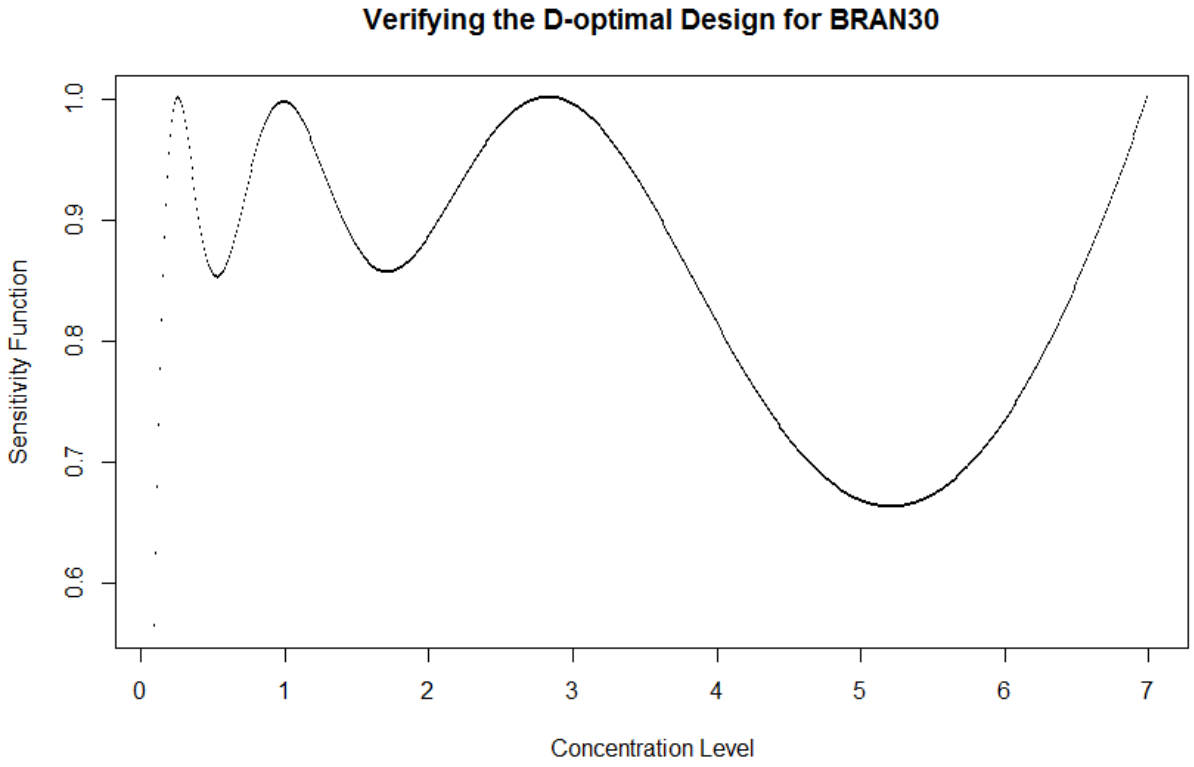

Figure 8. Sensitivity Function Plot for BRAN at Exposure Time 30.

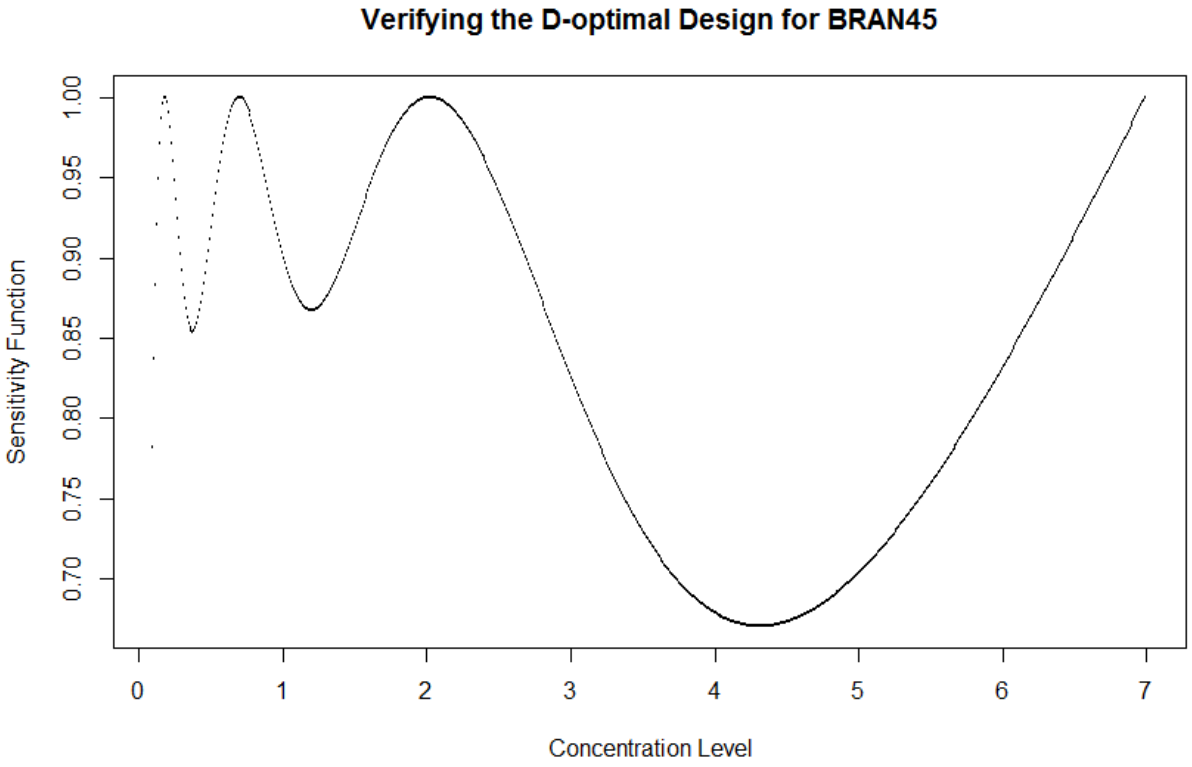

Figure 9. Sensitivity Function Plot for BRAN at Exposure Time 45.

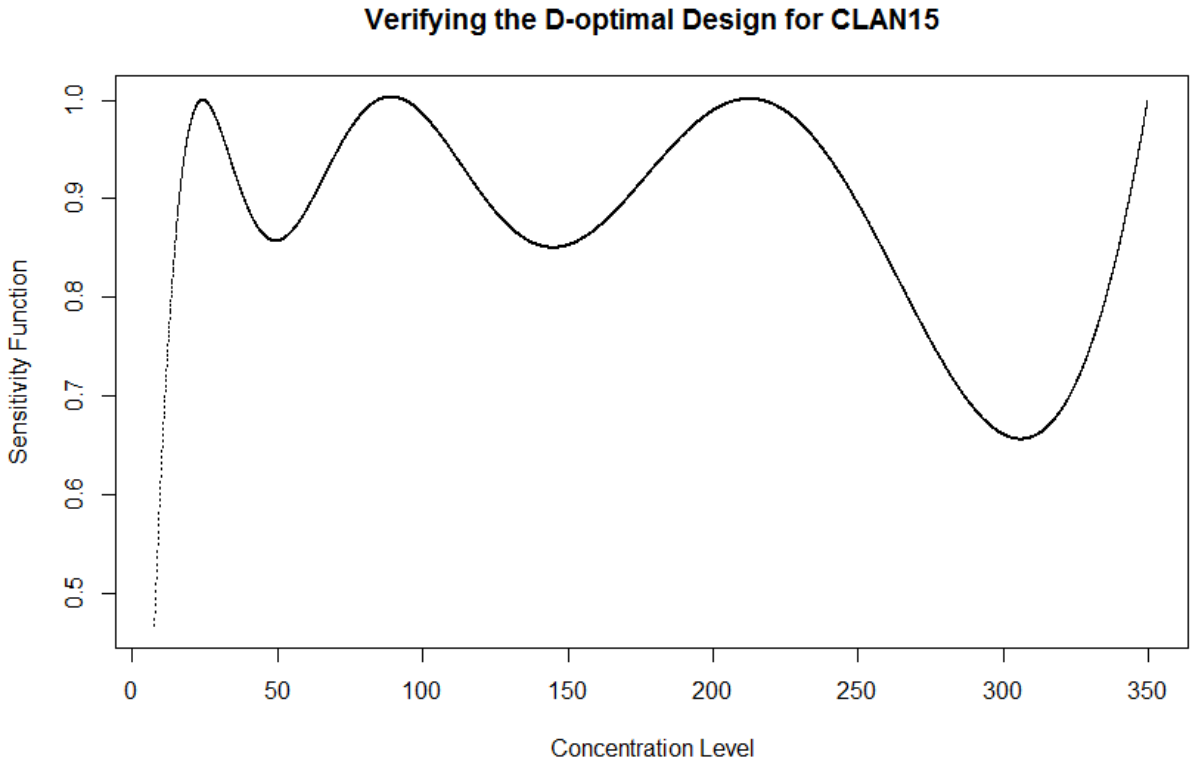

Figure 10. Sensitivity Function Plot for CLAN at Exposure Time 15.

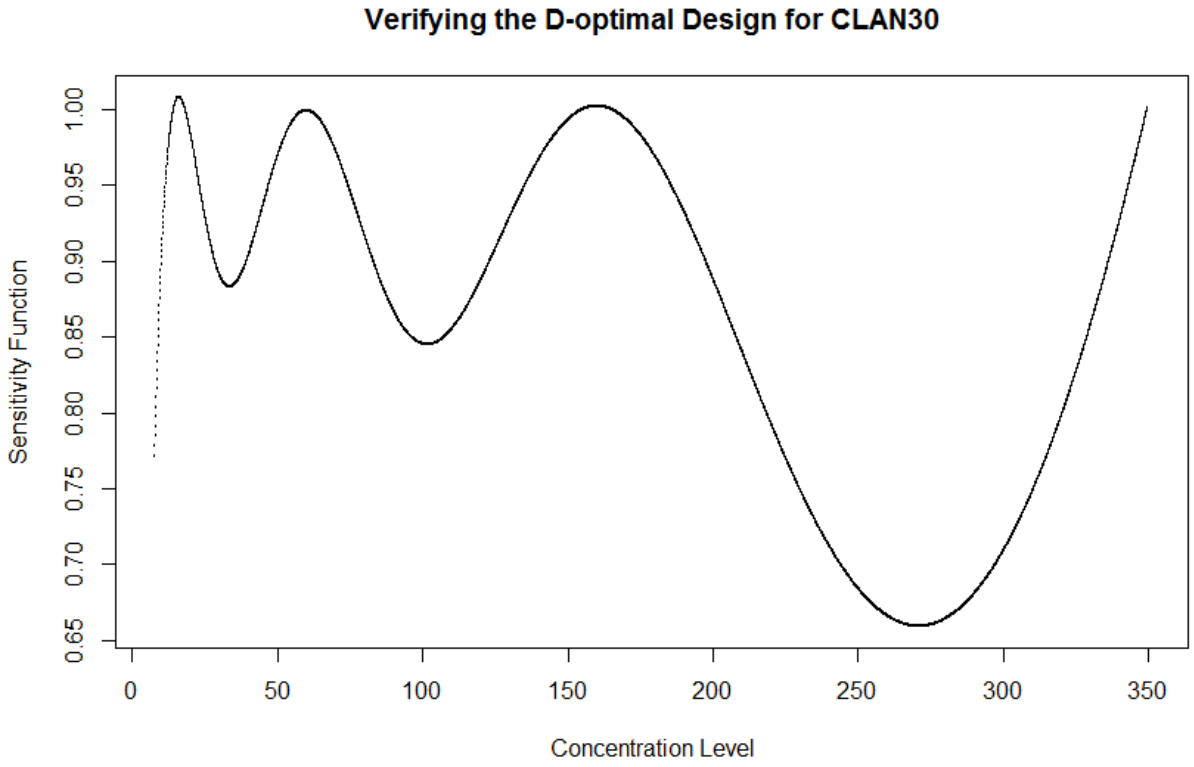

Figure 11. Sensitivity Function Plot for CLAN at Exposure Time 30.

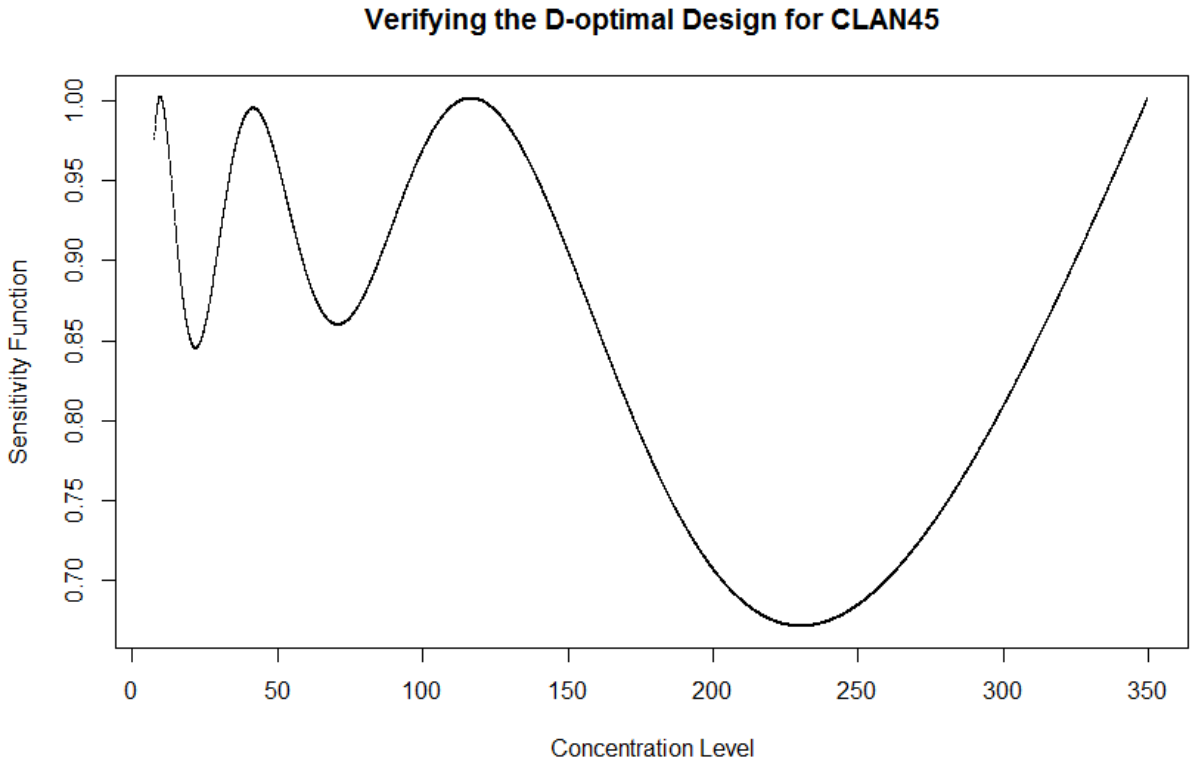

Figure 12. Sensitivity Function Plot for CLAN at Exposure Time 45.

5. EFFICIENCY

Since the parameter estimates of the chemical compounds BRAN and CLAN and their corresponding exposure times are used to obtain the D-optimal designs, we can check the efficiency of the original designs compared to the D-optimal designs. The efficiency was calculated for each of the six scenarios by dividing determinant for the original design's Fisher information matrix by the determinant for the D-optimal design's Fisher information matrix for each of the six designs and taking it to the power of 1/4:

$$eff_D = \left(\frac{\left|M\left(\xi_O, \widehat{\boldsymbol{\theta}}\right)\right|}{\left|M\left(\xi_D, \widehat{\boldsymbol{\theta}}\right)\right|}\right)^{1/4}$$

where *O* represents the original design and *D* represents the D-optimal design. The calculated efficiencies and corresponding percent of additional samples required to provide the same accuracy as the D-optimal design does are in Table 7 below.

Table 7. Efficiencies for the Six Models.

Compound	Exposure Time	Efficiency	% More Samples Needed $100\left(\frac{1}{efficiency_D} - 1\right)$
BRAN	15	0.866267	15.43785
BRAN	30	0.887015	12.73767
BRAN	45	0.8880933	12.60078
CLAN	15	0.8012226	24.80926
CLAN	30	0.8698052	14.96827
CLAN	45	0.8871641	12.71872

Based on these efficiency values, it does appear that the D-optimal design proves to be better. For the BRAN compound at time 30 and 45 as well as the CLAN compound at time 45, it would take approximately 13% more samples for the original design to match the D-optimal design. This slightly increases for the BRAN compound at time 15 and the CLAN compound at time 30 to approximately 15%. The greatest difference appears to be for the CLAN compound at

time 15 with approximately 25% more samples for the designs to be equivalent. We can observe that the original designs used to fit the 5PL-1P model for the six scenarios are not too bad, but it is obvious that the accuracy can be increased by using the D-optimal designs.

6. SIMULATION

Through simulation studies, we can check the performance of the D-optimal designs to estimate model parameters and compare it with the original designs. There were 1000 simulations for each of the six scenarios. Response values are simulated for both the original designs and the D-optimal designs under the 5PL-1P model on page 6. For $\boldsymbol{\theta}$, the estimates for each of the six scenarios in Tables 2 and 4 are used as the true values. The sample sizes 14 and 28 are used in the simulations, where 14 was the original sample size in Dawson et al. (2012), and we see how it changes when the sample size N increases to 28.

For the original design, the seven concentration levels were placed into the mean response function $f(x_i, \boldsymbol{\theta})$ to generate the seven corresponding predicted y values. Those seven predicated responses were then used to randomly generate two y values each when $N = 14$ and four y values each when $N = 28$ using a normal distribution with the *nrorm* function in R given that the standard deviations for the six scenarios are:

Table 8. Residual Standard Error for the Six Models.

	Exposure Time		
Compound	15 Minutes	30 Minutes	45 Minutes
BRAN	0.8876	0.8003	0.5917
CLAN	0.846	0.6589	0.68

For the D-optimal design, the four optimal concentration levels in Table 6 were also placed into the mean response function $f(x_i, \boldsymbol{\theta})$ to generate the corresponding predicted y values. When $N = 14$, three y values each for the lowest and highest concentration levels and four y values each for the middle two concentration levels are generated. The reason why the D-optimal designs do not have equal replications is due to the sample size 14 not being able to divide evenly by the number four. So there will be slightly higher weight this time to the middle

two concentrations. Once the response values are calculated, then each model is refit with their corresponding 14 concentrations and response effects using the *nlm* function in R. Then the mean squared error (MSE) is calculated for each design for each simulation:

$$MSE = \frac{\sum(\boldsymbol{\theta}^T - \widehat{\boldsymbol{\theta}})^2}{1000}$$

where $\boldsymbol{\theta}^T$ are the true parameter values and $\widehat{\boldsymbol{\theta}}$ are the parameter estimates for each simulation. The mean square error for the original designs' and D-optimal designs' parameters are then compared to see which design is better. The MSE shows the quality of an estimator, and the lower MSE value tells us that the estimator is more accurate. From Tables 9 to 14, all six combinations of compound and exposure time have smaller mean square error values for the D-optimal design than the original design.

Table 9. Original and D-Optimal Design Mean Squared Error for BRAN at Time 15 when the Sample Size Equals 14.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	358.59676437	0.96204253	0.06039432	0.90061684
D-Optimal	253.93142192	0.88707942	0.04677891	0.69407818

Table 10. Original and D-Optimal Design Mean Squared Error for BRAN at Time 30 when the Sample Size Equals 14.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	4.16014568	0.03877797	0.01995154	0.01933491
D-Optimal	2.67016629	0.03410884	0.01403019	0.01567006

Table 11. Original and D-Optimal Design Mean Squared Error for BRAN at Time 45 when the Sample Size Equals 14.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	0.568513261	0.006797097	0.008326326	0.005551599
D-Optimal	0.352139157	0.004189896	0.004577633	0.003249384

Table 12. Original and D-Optimal Design Mean Squared Error for CLAN at Time 15 when the Sample Size Equals 14.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	166.1844215	1285.8108881	0.1499704	0.1601099
D-Optimal	10.09408	1.084983	0.1009763	0.09916476

Table 13. Original and D-Optimal Design Mean Squared Error for CLAN at Time 30 when the Sample Size Equals 14.

	Mean Square Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	2.550605259	60.044586155	0.024602409	0.005045965
D-Optimal	1.466548658	49.918306733	0.015777633	0.003492685

Table 14. Original and D-Optimal Design Mean Squared Error for CLAN at Time 45 when the Sample Size Equals 14.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	0.800843526	23.148848292	0.01541577	0.003195462
D-Optimal	0.512629456	18.027613363	0.009551908	0.002236515

This simulation was conducted a second time but for the larger sample size of 28. Once again for the D-optimal design, the four optimal concentration levels were also placed into the mean response function $f(x_i, \boldsymbol{\theta})$ to generate the corresponding predicted y values. When $N =$

28, seven y values are generated for each concentration level. This time there are equal replications for each of the four concentration levels. Besides the difference in sample size, the simulation was done identically to the previous one. Tables 15 to 20 provide the MSE values for all six scenarios for simulations using the original and D-optimal designs. Once again, the MSE values are lower for the D-optimal designs than the original designs.

Table 15. Original and D-Optimal Design Mean Squared Error for BRAN at Time 15 when the Sample Size Equals 28.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	189.44245106	0.55950587	0.03288836	0.40609078
D-Optimal	138.50499781	0.53656172	0.02762743	0.33210241

Table 16. Original and D-Optimal Design Mean Squared Error for BRAN at Time 30 when the Sample Size Equals 28.

	Mean Square Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	1.892005791	0.020262262	0.009918884	0.009050211
D-Optimal	1.246671896	0.015844758	0.006731355	0.006794351

Table 17. Original and D-Optimal Design Mean Squared Error for BRAN at Time 45 when the Sample Size Equals 28.

	Mean Square Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	0.283391865	0.003282052	0.004144244	0.002612257
D-Optimal	0.164588317	0.002136183	0.002456894	0.001561186

Table 18. Original and D-Optimal Design Mean Squared Error for CLAN at Time 15 when the Sample Size Equals 28.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	0.1298222	1.199471	0.0001113812	0.00009968976
D-Optimal	0.05875716	0.8114535	0.00004192186	0.00005925953

Table 19. Original and D-Optimal Design Mean Squared Error for CLAN at Time 30 when the Sample Size Equals 28.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	1.167300297	26.873699455	0.011302609	0.002125242
D-Optimal	0.767176672	23.557844922	0.008181628	0.001643100

Table 20. Original and D-Optimal Design Mean Squared Error for CLAN at Time 45 when the Sample Size Equals 28.

	Mean Squared Error (MSE)			
Design	Maximum Concentration θ_1	θ_2	Hillslope θ_3	Asymmetric Factor θ_4
Original	0.391656258	11.380449782	0.006815518	0.001537013
D-Optimal	0.233079773	8.939986380	0.004815156	0.001075364

7. CONCLUSION

This paper studied D-optimal designs for the 5PL-1P model. This model is a hybrid between the 4PL and 5PL models for the relationship between concentration level and response effect. The 4PL model takes into account the minimum and maximum effect, median concentration level EC_{50}, and the slope. However, it has been found that there is some asymmetry in this relationship. So the 5PL model takes in the same parameters as the 4PL model, but adds an asymmetric factor. But when you reach five parameters in a nonlinear model, it can be difficult to fit. This is why one parameter was taken out of the 5PL model, the minimum effect factor, to create the 5PL-1P model.

Dawson et al. (2012) studied the 5PL-1P model to fit data in toxicology. In this paper, two chemical compounds, BRAN and CLAN, were borrowed from that study in order to obtain D-optimal designs for the 5PL-1P model and compare them with the original experimental design used in their study.

Based on the parameter estimates from the Dawson et al. (2012) study's data, the D-optimal designs were obtained using the modified algorithm from Hyun, Wang, and Yang (2015). The D-optimal designs for the 5PL-1P model had four concentration levels containing the upper bound of the concentration levels, which is 7 for the BRAN compound and 350 for the CLAN compound, and the other three levels are changed by the different parameter values. In addition, the D-optimal designs had equal weights for the four concentration levels.

It was observed that the D-optimal designs worked better than the original design by looking at the efficiency and conducting simulations. The efficiency was calculated for each of the six scenarios and remained in the 0.8 area. This means that the original design needed approximately twelve to twenty-four percent more samples in order to reach the accuracy of the

D-optimal designs. Simulations using the original design and D-optimal were conducted to compare their mean squared errors. Since all the MSE values were lower for the D-optimal designs, it suggested that they were more accurate than the original designs.

In this paper, the D-optimal designs were obtained from prespecified parameter values. In the future, this study can be extended to obtain the D-optimal designs for the 5PL-1P model that works well for unknown parameter values. For this, a Bayesian approach or multistage approach can be applied to construct a robust D-optimal design.

REFERENCES

Bromoacetonitrile. (2016). *National Center for Biotechnology Information: PubChem Compound Database.* Retrieved from https://pubchem.ncbi.nlm.nih.gov/compound/11534

Chloroacetonitrile. (2016). *National Center for Biotechnology Information: PubChem Compound Database.* Retrieved from https://pubchem.ncbi.nlm.nih.gov/compound/7856

Dawson, D. A., Jeyaratnam, J., Mooneyham, T., Pöch, G., and Schultz, T. W. (2010). "Mixture Toxicity of S_n2-Reactive Soft Electrophiles: 1. Evaluation of Mixtures Containing α-Halogenated Acetonitriles," *Arch Environ Contam Toxicol.* 59.4, 532-541.

Dawson, Douglas A., Genco, Nicole, Bensinger, Heather M., Quinn, Daphne, Il'Giovine, Zachary J., T. Wayne, Schultz, and Pöch, Gerald. (2012). "Evaulation of an asymmetry parameter for curve-fitting in single chemical and mixture toxicity assessment," *Toxicology* 292.2-3, 156-161.

Fedorov, V. V., Klimko, E.M., and Studden, W.J. (1972). "Theory of Optimal Experiments." Academic press, New York.

Hathcock, John. (1982). *Nutritional Toxicology, Volume 1,* Academic Press.

Hyun, S.W., Wong, W. K., and Yang, Y. (2015). "VNM: An R package for finding multiple objective optimal designs for the 4-parameter logistic model."

Hyun, Seung Won, and Wong, Weng Kee. (2015). "Multiple-objective optimal designs for studying the dose response function and interesting dose levels," *The Internal Journal of Biostatistics* 11.2, 253-271.

Khinkis, Leonid A., Levasseur, Laurence, Hélène, Faessel, and Greco, William R. (2003). "Optimal Design for Estimating Parameters of the 4-Parameter Hill Model." *Nonlinearity Biol Toxicol Med.* 1.3, 363-377.

Kiefer, J. (1958). "On the nonrandomized optimality and randomized nonoptimality of symmetrical designs," *Journal of Statistical Planning and Inference* 141, 559-575.

Keifer, J., Wolfowitz, J. (1960). "The equivalence of two extremum problems," *Canadian Journal of Mathematics* 12, 363-366.

Morgan, Byron J. T. (1992). *Analysis of Quantal Response Data*, Chapman & Hall.

White, Lynda V. (1973). “An extension of the General Equivalence Theorem to nonlinear Models,” *Biometrika* 60.2, 345.

Yang, M., Biedermann, S., and Tang, E. (2013). “On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm,” *Journal of American Statistical Association* 108, 1411-1420.

Yu, Simon J. (2015). *The Toxicology and Biochemistry of Insecticides*, CRC Press.

APPENDIX A. R CODES TO FIT THE MODELS

```
BRAN <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
bran15a <- c(2.303,4.668,11.33,22.55,38.73,61,81)
bran15b <- c(0.486,4.649,11.99,21.94,37.65,61.12,81.67)
bran30a <- c(6.017,10.59,23.11,42.06,65.77,84.83,94.7)
bran30b <- c(5.24,12.02,25.08,43.81,65.13,84.61,94.91)
bran45a <- c(10.58,19.58,37.37,60.87,81.42,93.35,97.92)
bran45b <- c(8.789,20.97,38.16,61.51,81.15,93.3,97.91)

CLAN <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
clan15a <- c(2.901,4.555,8.504,16.43,33.84,55.92,78.29)
clan15b <- c(1.193,3.389,9.372,17,35.28,56.17,78.36)
clan30a <-c(6.401,11.83,21.92,38.11,61.79,82.44,94.17)
clan30b <- c(5.096,10.27,21.81,37.62,62.48,82.12,94.08)
clan45a <- c(10.69,20.38,35.42,56.09,78.41,92.21,97.64)
clan45b <- c(9.76,18.15,35.24,56.1,78.94,92.04,97.74)

##BRAN-15 Parameters##
bran15 <- c(cbind(bran15a,bran15b))
x <- rep(BRAN,times=2)
y <- bran15
DataFrame <- data.frame(x,y)
Fit <-
nls(y~max/((1+(xb/x)^(hillslope))^(s)),DataFrame,start=list(max=81.67,xb=1,hillslope=1,s=1))

##BRAN-30 Parameters##
bran30 <- c(cbind(bran30a,bran30b))
x <- rep(BRAN,times=2)
y <- bran30
DataFrame <- data.frame(x,y)
Fit <-
nls(y~max/((1+(xb/x)^(hillslope))^(s)),DataFrame,start=list(max=94.91,xb=1,hillslope=1,s=1))

##BRAN-45 Parameters##
bran45 <- c(cbind(bran45a,bran45b))
x <- rep(BRAN,times=2)
y <- bran45
DataFrame <- data.frame(x,y)
Fit <-
nls(y~max/((1+(xb/x)^(hillslope))^(s)),DataFrame,start=list(max=97.92,xb=1,hillslope=1,s=1))

##CLAN-15 Parameters##
clan15 <- c(cbind(clan15a,clan15b))
x <- rep(CLAN,times=2)
y <- clan15
DataFrame <- data.frame(x,y)
```

```
Fit <-
nls(y~max/((1+(xb/x)^(hillslope))^(s)),DataFrame,start=list(max=78.36,xb=65,hillslope=1,s=1))

##CLAN-30 Parameters##
clan30 <- c(cbind(clan30a,clan30b))
x <- rep(CLAN,times=2)
y <- clan30
DataFrame <- data.frame(x,y)
Fit <-
nls(y~max/((1+(xb/x)^(hillslope))^(s)),DataFrame,start=list(max=94.17,xb=65,hillslope=1,s=1))

##CLAN-45 parameters##
clan45 <- c(cbind(clan45a,clan45b))
x <- rep(CLAN,times=2)
y <- clan45
DataFrame <- data.frame(x,y)
Fit <-
nls(y~max/((1+(xb/x)^(hillslope))^(s)),DataFrame,start=list(max=97.74,xb=65,hillslope=1,s=1))
```

APPENDIX B. R CODES FOR THE EFFICIENCIES

```
infor <- function(T,x) {
        f1 <- 1 / ((1+(T[2]/x)^T[3])^T[4])
        f2 <- -1 * T[1] * T[3] * T[4] * ((T[2]/x)^T[3]) * ((((T[2]/x)^T[3])+1)^(-1*T[4]-1))
        f3 <- -1 * T[1] * T[4] * ((T[2]/x)^T[3]) * log(T[2]/x) * ((((T[2]/x)^T[3])+1)^(-1*T[4]-1))
        f4 <- -1 * T[1] * ((((T[2]/x)^T[3])+1)^(-1*T[4])) * log(1+((T[2]/x)^T[3]))
        f = matrix(cbind(f1,f2,f3,f4))
        f%*%t(f)
}

upinfor <- function(W,T,x) {
        k = length(x)
        last_infor=infor(T,x[k])
        infor=(1-sum(W))* last_infor
        for (i in 1:(k-1)) {
                infor = infor + W[i] * infor(T,x[i])
        }
        infor
}

##BRAN - Other Design##
BRAN <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
weight1 <- c(1/7,1/7,1/7,1/7,1/7,1/7,1/7)
T15b = c(128.1528,2.3244,0.9791,1.5470)
T30b = c(103.2062,1.6336,1.5402,0.8235)
T45b = c(100.97883,1.08130,1.70242,0.71926)
M15b <- upinfor(weight1,T15b,BRAN)
M15b
detM15b <- det(M15b)
detM15b
M30b <- upinfor(weight1,T30b,BRAN)
M30b
detM30b <- det(M30b)
detM30b
M45b <- upinfor(weight1,T45b,BRAN)
M45b
detM45b <- det(M45b)
detM45b

##BRAN - D-Optimal##
weight2 <- c(0.25,0.25,0.25)
x15b <- c(0.33,1.33,3.78,7.00)
x30b <- c(0.26,1.01,2.84,7.00)
x45b <- c(0.18,0.70,2.03,7.00)
T15b = c(128.1528,2.3244,0.9791,1.5470)
T30b = c(103.2062,1.6336,1.5402,0.8235)
```

```
T45b = c(100.97883,1.08130,1.70242,0.71926)
do_M15b <- upinfor(weight2,T15b,x15b)
do_M15b
d0_detM15b <- det(do_M15b)
d0_detM15b
do_M30b <- upinfor(weight2,T30b,x30b)
do_M30b
d0_detM30b <- det(do_M30b)
d0_detM30b
do_M45b <- upinfor(weight2,T45b,x45b)
do_M45b
d0_detM45b <- det(do_M45b)
d0_detM45b

##BRAN Efficiency##
eff15b <- (detM15b/d0_detM15b)^(1/4)
eff15b
eff30b <- (detM30b/d0_detM30b)^(1/4)
eff30b
eff45b <- (detM45b/d0_detM45b)^(1/4)
eff45b



##CLAN - Other Design##
CLAN <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
weight1 <- c(1/7,1/7,1/7,1/7,1/7,1/7,1/7)
T15c = c(105.7901,204.3503,1.5294,0.8279)
T30c = c(100.78867,119.55175,1.89378,0.56313)
T45c = c(100.73194,75.21709,1.87647,0.54536)
M15c <- upinfor(weight1,T15c,CLAN)
M15c
detM15c <- det(M15c)
detM15c
M30c <- upinfor(weight1,T30c,CLAN)
M30c
detM30c <- det(M30c)
detM30c
M45c <- upinfor(weight1,T45c,CLAN)
M45c
detM45c <- det(M45c)
detM45c

##CLAN - D-Optimal##
weight2 <- c(0.25,0.25,0.25)
x15c <- c(24.0,90.8,212.7,350.0)
```

```
x30c <- c(14.7,63.9,161.7,350.0)
x45c <- c(9.8,42.1,116.8,350.0)
T15c = c(105.7901,204.3503,1.5294,0.8279)
T30c = c(100.78867,119.55175,1.89378,0.56313)
T45c = c(100.73194,75.21709,1.87647,0.54536)
do_M15c <- upinfor(weight2,T15c,x15c)
do_M15c
d0_detM15c <- det(do_M15c)
d0_detM15c
do_M30c <- upinfor(weight2,T30c,x30c)
do_M30c
d0_detM30c <- det(do_M30c)
d0_detM30c
do_M45c <- upinfor(weight2,T45c,x45c)
do_M45c
d0_detM45c <- det(do_M45c)
d0_detM45c

##CLAN Efficiency##
eff15c <- (detM15c/d0_detM15c)^(1/4)
eff15c
eff30c <- (detM30c/d0_detM30c)^(1/4)
eff30c
eff45c <- (detM45c/d0_detM45c)^(1/4)
eff45c
```

APPENDIX C. R CODES FOR THE SIMULATION

```
library(nplr)
######################### BRAN 15 ####################
n <- 1000
yB <- matrix(1,nrow=n,ncol=28)
yD <- matrix(1,nrow=n,ncol=28)
X1 <- matrix(1,nrow=n,ncol=28)
X2 <- matrix(1,nrow=n,ncol=28)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_B <- 0
MSE_D <- 0
mseD <- 1
mseB <- 1
theta_B <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
xD_B15 <- c(0.33,1.33,3.78,7.00)
max <- 128.1528
xb <- 2.3244
hillslope <- 0.9791
s <- 1.5470
y_B <- max/((1+(xb/xB)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_B15)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(128.1528,2.3244,0.9791,1.5470)
  RSE <- 0.8879
  xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
  xD_B15 <- c(0.33,1.33,3.78,7.00)
  m <- 0
  p <- 0
  ####Original Design####
  for(j in 1:7) {
    for(k in 1:4) {
      h <- k+m
      yB[i,h] <- rnorm(1,y_B[j],RSE)
```

```
    X1[i,h] <- xB[j]
   }
   m <- m + 4
  }
  x1 <- X1[i,]
  y1 <- yB[i,]

  ####Newton-Raphson for Original Design####
  ff_B = function(T) {
   sum((f(T,x1)-y1)^2)
  }
  result = nlm(ff_B,p=c(128,2.3,0.9,1.5))
  theta_B[i,1] <- result$estimate[1]
  theta_B[i,2] <- result$estimate[2]
  theta_B[i,3] <- result$estimate[3]
  theta_B[i,4] <- result$estimate[4]
  ####D-Optimal Design####
  for(j in 1:4) {
   for(k in 1:7) {
    h <- k+p
    yD[i,h] <- rnorm(1,y_D[j],RSE)
    X2[i,h] <- xD_B15[j]
   }
   p <- p + 7
  }
  x2 <- X2[i,]
  y2 <- yD[i,]

  ####Newton-Raphson for D-Optimal Design####
  ff_D = function(T) {
   sum((f(T,x2)-y2)^2)
  }
  result = nlm(ff_D,p=c(128,2.3,0.9,1.5))
  theta_D[i,1] <- result$estimate[1]
  theta_D[i,2] <- result$estimate[2]
  theta_D[i,3] <- result$estimate[3]
  theta_D[i,4] <- result$estimate[4]

  ####Final MSE Calculations####
  mseD = (theta_D[i,] - theta)^2
  MSE_D = MSE_D + mseD
  mseB = (theta_B[i,] - theta)^2
  MSE_B = MSE_B + mseB
}
MSE_B/1000
MSE_D/1000
```

```
######################## BRAN 30 ####################
n <- 1000
yB <- matrix(1,nrow=n,ncol=28)
yD <- matrix(1,nrow=n,ncol=28)
X1 <- matrix(1,nrow=n,ncol=28)
X2 <- matrix(1,nrow=n,ncol=28)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_B <- 0
MSE_D <- 0
mseD <- 1
mseB <- 1
theta_B <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
xD_B30 <- c(0.26,1.01,2.84,7.00)
max <- 103.2062
xb <- 1.6336
hillslope <- 1.5402
s <- 0.8235
y_B <- max/((1+(xb/xB)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_B30)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(103.2062,1.6336,1.5402,0.8235)
  RSE <- 0.8003
  xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
  xD_B30 <- c(0.26,1.01,2.84,7.00)
  m <- 0
  p <- 0

  ####Original Design####
  for(j in 1:7) {
    for(k in 1:4) {
      h <- k+m
      yB[i,h] <- rnorm(1,y_B[j],RSE)
      X1[i,h] <- xB[j]
```

```
  }
  m <- m + 4
 }
 x1 <- X1[i,]
 y1 <- yB[i,]

 ####Newton-Raphson for Original Design####
 ff_B = function(T) {
  sum((f(T,x1)-y1)^2)
 }
 result = nlm(ff_B,p=c(103,1.6,1.5,0.8))
 theta_B[i,1] <- result$estimate[1]
 theta_B[i,2] <- result$estimate[2]
 theta_B[i,3] <- result$estimate[3]
 theta_B[i,4] <- result$estimate[4]

 ####D-Optimal Design####
 for(j in 1:4) {
  for(k in 1:7) {
   h <- k+p
   yD[i,h] <- rnorm(1,y_D[j],RSE)
   X2[i,h] <- xD_B30[j]
  }
  p <- p + 7
 }
 x2 <- X2[i,]
 y2 <- yD[i,]

 ####Newton-Raphson for D-Optimal Design####
 ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
 }
 result = nlm(ff_D,p=c(103,1.6,1.5,0.8))
 theta_D[i,1] <- result$estimate[1]
 theta_D[i,2] <- result$estimate[2]
 theta_D[i,3] <- result$estimate[3]
 theta_D[i,4] <- result$estimate[4]

 ####Final MSE Calculations####
 mseD = (theta_D[i,] - theta)^2
 MSE_D = MSE_D + mseD
 mseB = (theta_B[i,] - theta)^2
 MSE_B = MSE_B + mseB
}
MSE_B/1000
MSE_D/1000
```

```
######################### BRAN 45 ####################
n <- 1000
yB <- matrix(1,nrow=n,ncol=28)
yD <- matrix(1,nrow=n,ncol=28)
X1 <- matrix(1,nrow=n,ncol=28)
X2 <- matrix(1,nrow=n,ncol=28)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_B <- 0
MSE_D <- 0
mseD <- 1
mseB <- 1
theta_B <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
xD_B45 <- c(0.18,0.70,2.03,7.00)
max <- 100.97883
xb <- 1.08130
hillslope <- 1.70242
s <- 0.71926
y_B <- max/((1+(xb/xB)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_B45)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(100.97883,1.08130,1.70242,0.71926)
  RSE <- 0.5917
  xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
  xD_B45 <- c(0.18,0.70,2.03,7.00)
  m <- 0
  p <- 0

  ####Original Design####
  for(j in 1:7) {
    for(k in 1:4) {
      h <- k+m
```

```
    yB[i,h] <- rnorm(1,y_B[j],RSE)
    X1[i,h] <- xB[j]
  }
  m <- m + 4
}
x1 <- X1[i,]
y1 <- yB[i,]

####Newton-Raphson for Original Design####
ff_B = function(T) {
  sum((f(T,x1)-y1)^2)
}
result = nlm(ff_B,p=c(100.9,1,1.7,0.7))
theta_B[i,1] <- result$estimate[1]
theta_B[i,2] <- result$estimate[2]
theta_B[i,3] <- result$estimate[3]
theta_B[i,4] <- result$estimate[4]

####D-Optimal Design####
for(j in 1:4) {
  for(k in 1:7) {
    h <- k+p
    yD[i,h] <- rnorm(1,y_D[j],RSE)
    X2[i,h] <- xD_B45[j]
  }
  p <- p + 7
}
x2 <- X2[i,]
y2 <- yD[i,]

####Newton-Raphson for D-Optimal Design####
ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
}
result = nlm(ff_D,p=c(100.9,1,1.7,0.7))
theta_D[i,1] <- result$estimate[1]
theta_D[i,2] <- result$estimate[2]
theta_D[i,3] <- result$estimate[3]
theta_D[i,4] <- result$estimate[4]

####Final MSE Calculations####
mseD = (theta_D[i,] - theta)^2
MSE_D = MSE_D + mseD
mseB = (theta_B[i,] - theta)^2
MSE_B = MSE_B + mseB
}
```

```
MSE_B/1000
MSE_D/1000


######################### CLAN 15 ####################
n <- 2
yC <- matrix(1,nrow=n,ncol=28)
yD <- matrix(1,nrow=n,ncol=28)
X1 <- matrix(1,nrow=n,ncol=28)
X2 <- matrix(1,nrow=n,ncol=28)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_C <- 0
MSE_D <- 0
mseD <- 1
mseC <- 1
theta_C <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
xD_C15 <- c(24.0,90.8,212.7,350.0)
max <- 105.7901
xb <- 204.3503
hillslope <- 1.5294
s <- 0.8279
y_C <- max/((1+(xb/xC)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_C15)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(105.7901,204.3503,1.5294,0.8279)
  RSE <- 0.846
  xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
  xD_C15 <- c(24.0,90.8,212.7,350.0)
  m <- 0
  p <- 0

  ####Original Design####
  for(j in 1:7) {
```

```
  for(k in 1:4) {
    h <- k+m
    yC[i,h] <- rnorm(1,y_C[j],RSE)
    X1[i,h] <- xC[j]
  }
  m <- m + 4
}
x1 <- X1[i,]
y1 <- yC[i,]

####Newton-Raphson for Original Design####
ff_C = function(T) {
  sum((f(T,x1)-y1)^2)
}
result1 = nlm(ff_C,p=c(105.7,204.3,1.5,0.8))
theta_C[i,1] <- result1$estimate[1]
theta_C[i,2] <- result1$estimate[2]
theta_C[i,3] <- result1$estimate[3]
theta_C[i,4] <- result1$estimate[4]

####D-Optimal Design####
for(j in 1:4) {
  for(k in 1:7) {
    h <- k+p
    yD[i,h] <- rnorm(1,y_D[j],RSE)
    X2[i,h] <- xD_C15[j]
  }
  p <- p + 7
}
x2 <- X2[i,]
y2 <- yD[i,]

####Newton-Raphson for D-Optimal Design####
ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
}
result2 = nlm(ff_D,p=c(105.7,204.3,1.5,0.8))
theta_D[i,1] <- result2$estimate[1]
theta_D[i,2] <- result2$estimate[2]
theta_D[i,3] <- result2$estimate[3]
theta_D[i,4] <- result2$estimate[4]

####Final MSE Calculations####
mseD = (theta_D[i,] - theta)^2
MSE_D = MSE_D + mseD
mseC = (theta_C[i,] - theta)^2
```

```
  MSE_C = MSE_C + mseC
}
MSE_C/1000
MSE_D/1000


######################### CLAN 30 ####################
n <- 1000
yC <- matrix(1,nrow=n,ncol=28)
yD <- matrix(1,nrow=n,ncol=28)
X1 <- matrix(1,nrow=n,ncol=28)
X2 <- matrix(1,nrow=n,ncol=28)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_C <- 0
MSE_D <- 0
mseD <- 1
mseC <- 1
theta_C <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
xD_C30 <- c(14.7,63.9,161.7,350.0)
max <- 100.78867
xb <- 119.55175
hillslope <- 1.89378
s <- 0.56313
y_C <- max/((1+(xb/xC)^(hillslope))^(s))
y_D<- max/((1+(xb/xD_C30)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(100.78867,119.55175,1.89378,0.56313)
  RSE <- 0.6589
  xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
  xD_C30 <- c(14.7,63.9,161.7,350.0)
  m <- 0
  p <- 0
  ####Original Design####
```

```
for(j in 1:7) {
  for(k in 1:4) {
    h <- k+m
    yC[i,h] <- rnorm(1,y_C[j],RSE)
    X1[i,h] <- xC[j]
  }
  m <- m + 4
}
x1 <- X1[i,]
y1 <- yC[i,]

####Newton-Raphson for Original Design####
ff_C = function(T) {
  sum((f(T,x1)-y1)^2)
}
result = nlm(ff_C,p=c(100,119,1.8,0.5))
theta_C[i,1] <- result$estimate[1]
theta_C[i,2] <- result$estimate[2]
theta_C[i,3] <- result$estimate[3]
theta_C[i,4] <- result$estimate[4]

####D-Optimal Design####
for(j in 1:4) {
  for(k in 1:7) {
    h <- k+p
    yD[i,h] <- rnorm(1,y_D[j],RSE)
    X2[i,h] <- xD_C30[j]
  }
  p <- p + 7
}
x2 <- X2[i,]
y2 <- yD[i,]

####Newton-Raphson for D-Optimal Design####
ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
}
result = nlm(ff_D,p=c(100,119,1.8,0.5))
theta_D[i,1] <- result$estimate[1]
theta_D[i,2] <- result$estimate[2]
theta_D[i,3] <- result$estimate[3]
theta_D[i,4] <- result$estimate[4]

####Final MSE Calculations####
mseD = (theta_D[i,] - theta)^2
MSE_D = MSE_D + mseD
```

```
  mseC = (theta_C[i,] - theta)^2
  MSE_C = MSE_C + mseC
}
MSE_C/1000
MSE_D/1000


########################## CLAN 45 ####################
n <- 1000
yC <- matrix(1,nrow=n,ncol=28)
yD <- matrix(1,nrow=n,ncol=28)
X1 <- matrix(1,nrow=n,ncol=28)
X2 <- matrix(1,nrow=n,ncol=28)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_C <- 0
MSE_D <- 0
mseD <- 1
mseC <- 1
theta_C <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
xD_C30 <- c(9.8,42.1,116.8,350.0)
max <- 100.73194
xb <- 75.21709
hillslope <- 1.87647
s <- 0.54536
y_C <- max/((1+(xb/xC)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_C45)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(100.73194,75.21709,1.87647,0.54536)
  RSE <- 0.68
  xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
  xD_C45 <- c(9.8,42.1,116.8,350.0)
  m <- 0
  p <- 0
```

```
####Original Design####
for(j in 1:7) {
  for(k in 1:4) {
    h <- k+m
    yC[i,h] <- rnorm(1,y_C[j],RSE)
    X1[i,h] <- xC[j]
  }
  m <- m + 4
}
x1 <- X1[i,]
y1 <- yC[i,]

####Newton-Raphson for Original Design####
ff_C = function(T) {
  sum((f(T,x1)-y1)^2)
}
result = nlm(ff_C,p=c(100,75,1.8,0.5))
theta_C[i,1] <- result$estimate[1]
theta_C[i,2] <- result$estimate[2]
theta_C[i,3] <- result$estimate[3]
theta_C[i,4] <- result$estimate[4]

####D-Optimal Design####
for(j in 1:4) {
  for(k in 1:7) {
    h <- k+p
    yD[i,h] <- rnorm(1,y_D[j],RSE)
    X2[i,h] <- xD_C45[j]
  }
  p <- p + 7
}
x2 <- X2[i,]
y2 <- yD[i,]

####Newton-Raphson for D-Optimal Design####
ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
}
result = nlm(ff_D,p=c(100,75,1.8,0.5))
theta_D[i,1] <- result$estimate[1]
theta_D[i,2] <- result$estimate[2]
theta_D[i,3] <- result$estimate[3]
theta_D[i,4] <- result$estimate[4]

####Final MSE Calculations####
```

```
  mseD = (theta_D[i,] - theta)^2
  MSE_D = MSE_D + mseD
  mseC = (theta_C[i,] - theta)^2
  MSE_C = MSE_C + mseC
}
MSE_C/1000
MSE_D/1000
```

APPENDIX D. R CODES FOR THE SMALL-SAMPLE SIMULATION

```
library(nplr)
######################### BRAN 15 ####################
n <- 1000
yB <- matrix(1,nrow=n,ncol=14)
yD <- matrix(1,nrow=n,ncol=14)
X1 <- matrix(1,nrow=n,ncol=14)
X2 <- matrix(1,nrow=n,ncol=14)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_B <- 0
MSE_D <- 0
mseD <- 1
mseB <- 1
theta_B <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)


xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
xD_B15 <- c(0.33,1.33,3.78,7.00)
max <- 128.1528
xb <- 2.3244
hillslope <- 0.9791
s <- 1.5470
y_B <- max/((1+(xb/xB)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_B15)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(128.1528,2.3244,0.9791,1.5470)
  RSE <- 0.8879
  xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
  xD_B15 <- c(0.33,1.33,3.78,7.00)
  m <- 0
  p <- 0
  ####Original Design####
  for(j in 1:7) {
    for(k in 1:2) {
      h <- k+m
```

```
    yB[i,h] <- rnorm(1,y_B[j],RSE)
    X1[i,h] <- xB[j]
  }
  m <- m + 2
}
x1 <- X1[i,]
y1 <- yB[i,]

####Newton-Raphson for Original Design####
ff_B = function(T) {
  sum((f(T,x1)-y1)^2)
}
result = nlm(ff_B,p=c(128,2.3,0.9,1.5))
theta_B[i,1] <- result$estimate[1]
theta_B[i,2] <- result$estimate[2]
theta_B[i,3] <- result$estimate[3]
theta_B[i,4] <- result$estimate[4]
####D-Optimal Design####
k=1
for(j in 1:4) {
  h=0
  if(j==1 || j==4) {
    for(k in 1:3){
      if(j==1) {
        h = k
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B15[j]
        K=k+1
      }
      if(j==4) {
        h = k + 11
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B15[j]
        k=k+1
      }
    }
  }
  if(j==2 || j==3) {
    for(k in 1:4) {
      if(j==2) {
        h = k + 3
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B15[j]
        k=k+1
      }
      if(j==3) {
```

```
        h = k + 7
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B15[j]
        k=k+1
       }
     }
   }
 }
 x2 <- X2[i,]
 y2 <- yD[i,]

 ####Newton-Raphson for D-Optimal Design####
 ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
 }
 result = nlm(ff_D,p=c(128,2.3,0.9,1.5))
 theta_D[i,1] <- result$estimate[1]
 theta_D[i,2] <- result$estimate[2]
 theta_D[i,3] <- result$estimate[3]
 theta_D[i,4] <- result$estimate[4]

 ####Final MSE Calculations####
 mseD = (theta_D[i,] - theta)^2
 MSE_D = MSE_D + mseD
 mseB = (theta_B[i,] - theta)^2
 MSE_B = MSE_B + mseB

}
MSE_B/1000
MSE_D/1000

######################### BRAN 30 ####################
n <- 1000
yB <- matrix(1,nrow=n,ncol=14)
yD <- matrix(1,nrow=n,ncol=14)
X1 <- matrix(1,nrow=n,ncol=14)
X2 <- matrix(1,nrow=n,ncol=14)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_B <- 0
MSE_D <- 0
mseD <- 1
mseB <- 1
theta_B <- matrix(1,nrow=n,ncol=4)
```

```
theta_D <- matrix(1,nrow=n,ncol=4)

xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
xD_B30 <- c(0.26,1.01,2.84,7.00)
max <- 103.2062
xb <- 1.6336
hillslope <- 1.5402
s <- 0.8235
y_B <- max/((1+(xb/xB)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_B30)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(103.2062,1.6336,1.5402,0.8235)
  RSE <- 0.8003
  xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
  xD_B30 <- c(0.26,1.01,2.84,7.00)
  m <- 0
  p <- 0

  ####Original Design####
  for(j in 1:7) {
    for(k in 1:2) {
      h <- k+m
      yB[i,h] <- rnorm(1,y_B[j],RSE)
      X1[i,h] <- xB[j]
    }
    m <- m + 2
  }
  x1 <- X1[i,]
  y1 <- yB[i,]

  ####Newton-Raphson for Original Design####
  ff_B = function(T) {
    sum((f(T,x1)-y1)^2)
  }
  result = nlm(ff_B,p=c(103,1.6,1.5,0.8))
  theta_B[i,1] <- result$estimate[1]
  theta_B[i,2] <- result$estimate[2]
  theta_B[i,3] <- result$estimate[3]
  theta_B[i,4] <- result$estimate[4]
```

```
####D-Optimal Design####
k=1
for(j in 1:4) {
  h=0
  if(j==1 || j==4) {
    for(k in 1:3){
      if(j==1) {
        h = k
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B30[j]
        K=k+1
      }
      if(j==4) {
        h = k + 11
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B30[j]
        k=k+1
      }
    }
  }
  if(j==2 || j==3) {
    for(k in 1:4) {
      if(j==2) {
        h = k + 3
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B30[j]
        k=k+1
      }
      if(j==3) {
        h = k + 7
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_B30[j]
        k=k+1
      }
    }
  }
}
x2 <- X2[i,]
y2 <- yD[i,]


####Newton-Raphson for D-Optimal Design####
ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
}
```

```
  result = nlm(ff_D,p=c(103,1.6,1.5,0.8))
  theta_D[i,1] <- result$estimate[1]
  theta_D[i,2] <- result$estimate[2]
  theta_D[i,3] <- result$estimate[3]
  theta_D[i,4] <- result$estimate[4]

  ####Final MSE Calculations####
  mseD = (theta_D[i,] - theta)^2
  MSE_D = MSE_D + mseD
  mseB = (theta_B[i,] - theta)^2
  MSE_B = MSE_B + mseB
}
MSE_B/1000
MSE_D/1000

######################### BRAN 45 ####################
n <- 1000
yB <- matrix(1,nrow=n,ncol=14)
yD <- matrix(1,nrow=n,ncol=14)
X1 <- matrix(1,nrow=n,ncol=14)
X2 <- matrix(1,nrow=n,ncol=14)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_B <- 0
MSE_D <- 0
mseD <- 1
mseB <- 1
theta_B <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
xD_B45 <- c(0.18,0.70,2.03,7.00)
max <- 100.97883
xb <- 1.08130
hillslope <- 1.70242
s <- 0.71926
y_B <- max/((1+(xb/xB)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_B45)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}
```

```
for(i in 1:n) {
  theta <- c(100.97883,1.08130,1.70242,0.71926)
  RSE <- 0.5917
  xB <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
  xD_B45 <- c(0.18,0.70,2.03,7.00)
  m <- 0
  p <- 0

  ####Original Design####
  for(j in 1:7) {
    for(k in 1:2) {
      h <- k+m
      yB[i,h] <- rnorm(1,y_B[j],RSE)
      X1[i,h] <- xB[j]
    }
    m <- m + 2
  }
  x1 <- X1[i,]
  y1 <- yB[i,]

  ####Newton-Raphson for Original Design####
  ff_B = function(T) {
    sum((f(T,x1)-y1)^2)
  }
  result = nlm(ff_B,p=c(100.9,1,1.7,0.7))
  theta_B[i,1] <- result$estimate[1]
  theta_B[i,2] <- result$estimate[2]
  theta_B[i,3] <- result$estimate[3]
  theta_B[i,4] <- result$estimate[4]

  ####D-Optimal Design####
  k=1
  for(j in 1:4) {
    h=0
    if(j==1 || j==4) {
      for(k in 1:3){
        if(j==1) {
          h = k
          yD[i,h] <- rnorm(1,y_D[j],RSE)
          X2[i,h] <- xD_B45[j]
          K=k+1
        }
        if(j==4) {
          h = k + 11
          yD[i,h] <- rnorm(1,y_D[j],RSE)
```

```
        X2[i,h] <- xD_B45[j]
        k=k+1
       }
      }
    }
    if(j==2 || j==3) {
     for(k in 1:4) {
      if(j==2) {
       h = k + 3
       yD[i,h] <- rnorm(1,y_D[j],RSE)
       X2[i,h] <- xD_B45[j]
       k=k+1
      }
      if(j==3) {
       h = k + 7
       yD[i,h] <- rnorm(1,y_D[j],RSE)
       X2[i,h] <- xD_B45[j]
       k=k+1
      }
     }
    }
  }
  x2 <- X2[i,]
  y2 <- yD[i,]

  ####Newton-Raphson for D-Optimal Design####
  ff_D = function(T) {
   sum((f(T,x2)-y2)^2)
  }
  result = nlm(ff_D,p=c(100.9,1,1.7,0.7))
  theta_D[i,1] <- result$estimate[1]
  theta_D[i,2] <- result$estimate[2]
  theta_D[i,3] <- result$estimate[3]
  theta_D[i,4] <- result$estimate[4]

  ####Final MSE Calculations####
  mseD = (theta_D[i,] - theta)^2
  MSE_D = MSE_D + mseD
  mseB = (theta_B[i,] - theta)^2
  MSE_B = MSE_B + mseB
}
MSE_B/1000
MSE_D/1000

######################### CLAN 15 ####################
n <- 1000
```

```
yC <- matrix(1,nrow=n,ncol=14)
yD <- matrix(1,nrow=n,ncol=14)
X1 <- matrix(1,nrow=n,ncol=14)
X2 <- matrix(1,nrow=n,ncol=14)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_C <- 0
MSE_D <- 0
mseD <- 1
mseC <- 1
theta_C <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
xD_C15 <- c(24.0,90.8,212.7,350.0)
max <- 105.7901
xb <- 204.3503
hillslope <- 1.5294
s <- 0.8279
y_C <- max/((1+(xb/xC)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_C15)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(105.7901,204.3503,1.5294,0.8279)
  RSE <- 0.846
  xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
  xD_C15 <- c(24.0,90.8,212.7,350.0)
  m <- 0
  p <- 0

  ####Original Design####
  for(j in 1:7) {
    for(k in 1:2) {
      h <- k+m
      yC[i,h] <- rnorm(1,y_C[j],RSE)
      X1[i,h] <- xC[j]
    }
    m <- m + 2
```

```
}
x1 <- X1[i,]
y1 <- yC[i,]

####Newton-Raphson for Original Design####
ff_C = function(T) {
  sum((f(T,x1)-y1)^2)
}
result1 = nlm(ff_C,p=c(105.7,204.3,1.5,0.8))
theta_C[i,1] <- result1$estimate[1]
theta_C[i,2] <- result1$estimate[2]
theta_C[i,3] <- result1$estimate[3]
theta_C[i,4] <- result1$estimate[4]

####D-Optimal Design####
k=1
for(j in 1:4) {
  h=0
  if(j==1 || j==4) {
    for(k in 1:3){
      if(j==1) {
        h = k
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C15[j]
        K=k+1
      }
      if(j==4) {
        h = k + 11
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C15[j]
        k=k+1
      }
    }
  }
  if(j==2 || j==3) {
    for(k in 1:4) {
      if(j==2) {
        h = k + 3
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C15[j]
        k=k+1
      }
      if(j==3) {
        h = k + 7
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C15[j]
```

```
        k=k+1
       }
     }
   }
  }
  x2 <- X2[i,]
  y2 <- yD[i,]


  ####Newton-Raphson for D-Optimal Design####
  ff_D = function(T) {
   sum((f(T,x2)-y2)^2)
  }
  result2 = nlm(ff_D,p=c(105.7,204.3,1.5,0.8))
  theta_D[i,1] <- result2$estimate[1]
  theta_D[i,2] <- result2$estimate[2]
  theta_D[i,3] <- result2$estimate[3]
  theta_D[i,4] <- result2$estimate[4]

  ####Final MSE Calculations####
  mseD = (theta_D[i,] - theta)^2
  MSE_D = MSE_D + mseD
  mseC = (theta_C[i,] - theta)^2
  MSE_C = MSE_C + mseC
}
MSE_C/1000
MSE_D/1000

######################### CLAN 30 ####################
n <- 1000
yC <- matrix(1,nrow=n,ncol=14)
yD <- matrix(1,nrow=n,ncol=14)
X1 <- matrix(1,nrow=n,ncol=14)
X2 <- matrix(1,nrow=n,ncol=14)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_C <- 0
MSE_D <- 0
mseD <- 1
mseC <- 1
theta_C <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
```

```
xD_C30 <- c(14.7,63.9,161.7,350.0)
max <- 100.78867
xb <- 119.55175
hillslope <- 1.89378
s <- 0.56313
y_C <- max/((1+(xb/xC)^(hillslope))^(s))
y_D<- max/((1+(xb/xD_C30)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(100.78867,119.55175,1.89378,0.56313)
  RSE <- 0.6589
  xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
  xD_C30 <- c(14.7,63.9,161.7,350.0)
  m <- 0
  p <- 0
  ####Original Design####
  for(j in 1:7) {
    for(k in 1:2) {
      h <- k+m
      yC[i,h] <- rnorm(1,y_C[j],RSE)
      X1[i,h] <- xC[j]
    }
    m <- m + 2
  }
  x1 <- X1[i,]
  y1 <- yC[i,]

  ####Newton-Raphson for Original Design####
  ff_C = function(T) {
    sum((f(T,x1)-y1)^2)
  }
  result = nlm(ff_C,p=c(100,119,1.8,0.5))
  theta_C[i,1] <- result$estimate[1]
  theta_C[i,2] <- result$estimate[2]
  theta_C[i,3] <- result$estimate[3]
  theta_C[i,4] <- result$estimate[4]

  ####D-Optimal Design####
  k=1
  for(j in 1:4) {
```

```
  h=0
  if(j==1 || j==4) {
    for(k in 1:3){
      if(j==1) {
        h = k
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C30[j]
        K=k+1
      }
      if(j==4) {
        h = k + 11
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C30[j]
        k=k+1
      }
    }
  }
  if(j==2 || j==3) {
    for(k in 1:4) {
      if(j==2) {
        h = k + 3
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C30[j]
        k=k+1
      }
      if(j==3) {
        h = k + 7
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C30[j]
        k=k+1
      }
    }
  }
}
x2 <- X2[i,]
y2 <- yD[i,]

####Newton-Raphson for D-Optimal Design####
ff_D = function(T) {
  sum((f(T,x2)-y2)^2)
}
result = nlm(ff_D,p=c(100,119,1.8,0.5))
theta_D[i,1] <- result$estimate[1]
theta_D[i,2] <- result$estimate[2]
theta_D[i,3] <- result$estimate[3]
theta_D[i,4] <- result$estimate[4]
```

```
  ####Final MSE Calculations####
  mseD = (theta_D[i,] - theta)^2
  MSE_D = MSE_D + mseD
  mseC = (theta_C[i,] - theta)^2
  MSE_C = MSE_C + mseC
}
MSE_C/1000
MSE_D/1000


######################### CLAN 45 ####################
n <- 1000
yC <- matrix(1,nrow=n,ncol=14)
yD <- matrix(1,nrow=n,ncol=14)
X1 <- matrix(1,nrow=n,ncol=14)
X2 <- matrix(1,nrow=n,ncol=14)
y1 <- 1
y2 <- 1
m <- 0
p <- 0
MSE_C <- 0
MSE_D <- 0
mseD <- 1
mseC <- 1
theta_C <- matrix(1,nrow=n,ncol=4)
theta_D <- matrix(1,nrow=n,ncol=4)

xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
xD_C30 <- c(9.8,42.1,116.8,350.0)
max <- 100.73194
xb <- 75.21709
hillslope <- 1.87647
s <- 0.54536
y_C <- max/((1+(xb/xC)^(hillslope))^(s))
y_D <- max/((1+(xb/xD_C45)^(hillslope))^(s))

T=c(max,hillslope,xb,s)

f = function(T,X) {
  (T[1])/(1+(T[2]/X)^T[3])^T[4]
}

for(i in 1:n) {
  theta <- c(100.73194,75.21709,1.87647,0.54536)
  RSE <- 0.68
```

```
xC <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
xD_C45 <- c(9.8,42.1,116.8,350.0)
m <- 0
p <- 0

####Original Design####
for(j in 1:7) {
  for(k in 1:2) {
    h <- k+m
    yC[i,h] <- rnorm(1,y_C[j],RSE)
    X1[i,h] <- xC[j]
  }
  m <- m + 2
}
x1 <- X1[i,]
y1 <- yC[i,]

####Newton-Raphson for Original Design####
ff_C = function(T) {
  sum((f(T,x1)-y1)^2)
}
result = nlm(ff_C,p=c(100,75,1.8,0.5))
theta_C[i,1] <- result$estimate[1]
theta_C[i,2] <- result$estimate[2]
theta_C[i,3] <- result$estimate[3]
theta_C[i,4] <- result$estimate[4]

####D-Optimal Design####
k=1
for(j in 1:4) {
  h=0
  if(j==1 || j==4) {
    for(k in 1:3){
      if(j==1) {
        h = k
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C45[j]
        K=k+1
      }
      if(j==4) {
        h = k + 11
        yD[i,h] <- rnorm(1,y_D[j],RSE)
        X2[i,h] <- xD_C45[j]
        k=k+1
      }
    }
```

```
    }
    if(j==2 || j==3) {
      for(k in 1:4) {
        if(j==2) {
          h = k + 3
          yD[i,h] <- rnorm(1,y_D[j],RSE)
          X2[i,h] <- xD_C45[j]
          k=k+1
        }
        if(j==3) {
          h = k + 7
          yD[i,h] <- rnorm(1,y_D[j],RSE)
          X2[i,h] <- xD_C45[j]
          k=k+1
        }
      }
    }
  }
  x2 <- X2[i,]
  y2 <- yD[i,]

  ####Newton-Raphson for D-Optimal Design####
  ff_D = function(T) {
    sum((f(T,x2)-y2)^2)
  }
  result = nlm(ff_D,p=c(100,75,1.8,0.5))
  theta_D[i,1] <- result$estimate[1]
  theta_D[i,2] <- result$estimate[2]
  theta_D[i,3] <- result$estimate[3]
  theta_D[i,4] <- result$estimate[4]

  ####Final MSE Calculations####
  mseD = (theta_D[i,] - theta)^2
  MSE_D = MSE_D + mseD
  mseC = (theta_C[i,] - theta)^2
  MSE_C = MSE_C + mseC
}
MSE_C/1000
MSE_D/1000
```