

**Optimal Regression Design under Second-Order Least Squares
Estimator: Theory, Algorithm and Applications**

by

Chi-Kuang Yeh

B.Sc. (Hons.), University of Victoria, 2016

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Mathematics and Statistics

© Chi-Kuang Yeh, 2018

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.

**Optimal Regression Design under Second-Order Least Squares
Estimator: Theory, Algorithm and Applications**

by

Chi-Kuang Yeh

B.Sc. (Hons.), University of Victoria, 2016

Supervisory Committee

Dr. Julie Zhou, Supervisor
(Department of Mathematics and Statistics)

Dr. Xuekui Zhang, Departmental Member
(Department of Mathematics and Statistics)

ABSTRACT

In this thesis, we first review the current development of optimal regression designs under the second-order least squares estimator in the literature. The criteria include A- and D-optimality. We then introduce a new formulation of A-optimality criterion so the result can be extended to c-optimality which has not been studied before. Following Kiefer's equivalence results, we derive the optimality conditions for A-, c- and D-optimal designs under the second-order least squares estimator. In addition, we study the number of support points for various regression models including Peleg models, trigonometric models, regular and fractional polynomial models. A generalized scale invariance property for D-optimal designs is also explored. Furthermore, we discuss one computing algorithm to find optimal designs numerically. Several interesting applications are presented and related MATLAB code are provided in the thesis.

Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
List of Abbreviations	ix
Acknowledgements	x
1 Introduction	1
1.1 Optimal regression design problem	2
1.2 Second-order least squares estimator	5
1.3 Research problem	6
1.4 Main Contributions	6
2 Optimal Regression Designs Under SLSE	8
2.1 Design criteria under SLSE	8
2.2 Equivalence theorem for optimal designs under SLSE	12
2.3 Results on the number of support points	14
2.4 Scale invariance property of D-optimal design	23

3	Numerical Algorithm and Applications	27
3.1	Optimization problem	27
3.2	Computational Algorithm	29
3.3	Applications	29
3.4	Remarks	40
4	Discussions	44
	Appendix	46
A	Guidance of using CVX	46
B	MATLAB Code	49
B.1	Main function description	49
B.2	Main functions	50
B.3	Gompertz Growth model (Example 4)	58
B.4	Fractional polynomial (Example 6)	59
B.5	Michaelis-menton model example (Example 7)	59
B.6	Peleg model (Example 8)	60
B.7	Spline model (Example 9)	60
B.8	Mixture model (Example 10)	61
B.9	E _{max} model (Example 11)	63
	Bibliography	65

List of Tables

Table 3.1	D-optimal designs for Gompertz growth model with $t = 0.0$, $S = [0.0, 10.0]$, $\theta_o = (1, 1, 1)^T$ and various values of N	31
Table 3.2	A- and c-optimal design for Gompertz growth model with $\theta_o = (1, 1, 1)^T$, $\mathbf{c}_1 = (2, 0.5, 1)^T$, $N = 2001$ and various values of t . . .	33
Table 3.3	A- and c-optimal design for Michaelis-Menton model with $S = [0, 4]$, $\mathbf{c}_1 = (1, 1)^T$ and $N = 1001$	36
Table 3.4	Loss functions and D-optimal designs for Peleg model with $\theta_o = (0.5, 0.05)^T$, $S = [0, 180]$ and $N = 1001$ for $t = 0.0, 0.3, 0.5, 0.7$ and 0.9	36
Table 3.5	Loss functions and A-optimal designs for Peleg model with $\theta_o = (0.5, 0.05)^T$, $S = [0, 180]$ and $N = 1001$ for $t = 0.0, 0.3, 0.5, 0.7$ and 0.9	37
Table 3.6	Loss functions and c-optimal designs for Peleg model with $\theta_o = (0.5, 0.05)^T$, $S = [0, 180]$, $\mathbf{c}_1 = (1, 1)^T$ and $N = 1001$ for $t = 0.0, 0.3, 0.5, 0.7$ and 0.9	37
Table 3.7	D-optimal designs for the spline regression models in (2.22) on both scaled design space $S^V = [0, 1]$ and the original design space $S = [0, 10]$ with $N = 1001$	38
Table 3.8	D-optimal design for mixture model with 23424 grid points for $t = 0.0$ and 0.7	39
Table 3.9	D-efficiency for Gompertz growth model, Peleg model and Michaelis-Menton model.	43

List of Figures

Figure 1.1 A D-optimal design for Gompertz growth model with $\boldsymbol{\theta}_o = (1, 1, 1)^T$ and $S = [0, 10]$, which has three support points and each point has weight $\frac{1}{3}$	4
Figure 2.1 Plots of $d_A(x, \xi_A^*, t)$ and $d_D(x, \xi_A^*, t)$ in (2.10b) and (2.10a) for Example 1. (a) $d_A(x, \xi_A^*, t)$ with $t = 0.3$, (b) $d_A(x, \xi_A^*, t)$ with $t = 0.7$, (c) $d_D(x, \xi_A^*, t)$ with $t = 0.3$, (d) $d_D(x, \xi_A^*, t)$ with $t = 0.7$	18
Figure 3.1 Flow chart for computing optimal designs through CVX programming.	30
Figure 3.2 Loss function ($\log(\phi_D)$) for D-optimal design versus the number of points (N) for Gompertz growth model.	32
Figure 3.3 Plots of D- and A-optimal designs for polynomial model of order 5 without intercept under the SLSE with different values of t and $N = 2001$. (a) D-optimal design with $t = 0.00$, (b) D-optimal design with $t = 0.70$, (c) A-optimal design with $t = 0.00$, (d) A-optimal design with $t = 0.70$	34
Figure 3.4 Plots of the distribution of the weights for D- and A-optimal designs for fractional polynomial model (2.15) with various values of t and $N = 1001$: (a) D-optimal design with $t = 0.00$, (b) D-optimal design with $t = 0.90$, (c) A-optimal design with $t = 0.00$, (d) A-optimal design with $t = 0.90$	35
Figure 3.5 Plot of computation time versus the number of design points (N) for finding D-optimal design for Emax model with $\boldsymbol{\theta}_o = (1, 2, 1)^T$	41

Figure 3.6 Plots of the change of computation time versus the level of asymmetry of finding designs for Emax model with $\boldsymbol{\theta} = (1, 2, 1)^T$. (a) A-optimal design with $N = 1001$, (b) A-optimal design with $N = 5001$, (c) D-optimal design with $N = 1001$, (d) D-optimal design with $N = 5001$ 42

List of Abbreviations

BLUE	Best Linear Unbiased Estimator
CVX	A MATLAB-based modeling system for convex optimization
FPM	Fractional Polynomial Model
GSI	Generalized Scale Invariance
MATLAB	A script language that uses matrix as default format and a computing environment, which is developed by MathWorks
OLSE	Ordinary Least Squares Estimator
SDP	Semi-Definite Programming
SLSE	Second-Order Least Squares Estimator

ACKNOWLEDGEMENTS

I would like to thank:

My supervisor, Professor Julie Zhou, for your mentorship, support, encouragement, and patience. I would not be able to be here without you and your help. You helped me to understand the meaning of many things.

My parents, Chih-Chiang (Richard) Yeh and Minghui (Maria) Liu, for your endless support and unconditional love since childhood. I cannot thank you enough.

My sisters, Ya-Hsin (Christine) Yeh and Ya-Sung (Margarita) Yeh, for the time we lived together.

Friends, for all the unforgettable memories we had together.

Professors, Gary MacGillivray, Junling Ma and Laura Cowen who helped me dramatically during my studies in all aspects. Moreover I want to thank Professor David Giles, Mary Lesperance, Peter Dukes and Wu-Sheng Lu who made huge impacts towards my graduate studies via the coursework. The ways you delivered knowledge were fun and engaging.

All the staff at the University of Victoria, for your generosity and the countless help you provided.

The University of Victoria, for the comfortable, convenient and lovely studying environment. I always feel like the University of Victoria is warm and friendly, just like home.

Others, for your appearance in my life.

Chapter 1

Introduction

Design of experiment is a sub-field in statistics that has a long history in its developments. After Sir Ronald A. Fisher's pioneering work on the analysis of variance and fractional factorial design concepts, while working in Rothamsted Experimental Station in the 1920s and 1930s (Montgomery 2012, p. 21), many statisticians worked in this research area and made significant contributions. Berger and Wong (2009) gave many vital examples in the development of optimal design theory over the years, including Chernoff (1953), Kiefer (1959, 1974), Kiefer and Wolfowitz (1959), Silvey (1980) and Atkinson and Donev (1992). Their inputs to this field have had huge impacts towards today's design framework. One century later, design techniques have been found to be effective and now are widely used in other disciplines, such as agriculture, engineering, environmental science, biomedical and pharmaceutical studies. See examples in Crary et al. (2000), Crary (2002), Haines et al. (2003), Zhou et al. (2003), Jain and Tiwari (2003), Brosi et al. (2008), and Dette et al. (2017). Its primary objective is to study the cause and effect between variables under some systematic approaches and procedures.

Over the decades, many theoretical results and algorithms have been developed to construct different kinds of optimal designs, which include factorial design, fractional factorial design, response surface design, and regression design. In this thesis, we will focus on optimal regression design under a recently proposed statistical estimator,

second-order least squares estimator (SLSE) in Wang and Leblanc (2008).

1.1 Optimal regression design problem

“Regression analysis is a statistical technique for investigating and modeling the relationship between variables“ (Montgomery 2012, p. 1). It is one of the most widely used statistical methods to explore the relationship between variables based on observed data. Although there are different kinds of regression models, we mainly focus on one response linear and nonlinear regression models.

Suppose we want to study the relationship between $\mathbf{x} \in \mathbb{R}^p$ (a vector of explanatory variables) and y (a response variable). Consider a general regression model for (\mathbf{x}_i, y_i) ,

$$y_i = g(\mathbf{x}_i; \boldsymbol{\theta}) + \epsilon_i, \quad i = 1, \dots, n, \quad (1.1)$$

where $\boldsymbol{\theta} \in \mathbb{R}^q$ is an unknown parameter vector, n is the sample size, $g(\cdot)$ is a known linear or nonlinear expectation function depending on $\boldsymbol{\theta}$, and ϵ_i is a random error of the regression model. The random error ϵ_i 's are assumed to be independent identically distributed with zero mean and variance σ^2 . The random error terms are further assumed to be homoscedastic in this thesis. Since \mathbf{x}_i and y_i are observed data and $g(\cdot)$ is also known, the only unknown component is $\boldsymbol{\theta}$. A question naturally comes in mind is how to estimate $\boldsymbol{\theta}$ efficiently.

Suppose $\hat{\boldsymbol{\theta}}$ is an estimator of $\boldsymbol{\theta}$. The design problem aims to get the most information about $\boldsymbol{\theta}$ or $g(\mathbf{x}_i; \boldsymbol{\theta})$ by selecting the best probability distribution on $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ that maximizes some scalar functions of the Fisher's information matrix of $\hat{\boldsymbol{\theta}}$. The sample points \mathbf{x}_i and space are called design points and design space, respectively. It is known that Fisher's information matrix is proportional to the inverse of the variance-covariance matrix of $\hat{\boldsymbol{\theta}}$. Thus, the design problem aims to minimize some scalar functions of the variance-covariance matrix of $\hat{\boldsymbol{\theta}}$, which are called objective functions or loss functions. The resulted probability measure ξ contains two components that are the support points and the corresponding probabilities

associated with these points. The choice of the loss functions is determined based on the design interest. Various design criteria have been studied in the literature, such as A- and D-optimality criteria. D-optimality is one of the most widely used design criterion that minimizes the determinant of the variance-covariance matrix. The most desired property of D-optimal design is its scale invariant property. A-optimality minimizes the trace that leads to minimize the sum of the variances of the estimated parameters. See Fedorov (1972), Silvey (1980), Pukelsheim (1993), Berger and Wong (2009), and Dean et al. (2015) for other optimality criteria.

Let us consider Gompertz growth model to illustrate a D-optimal design. The model is given by

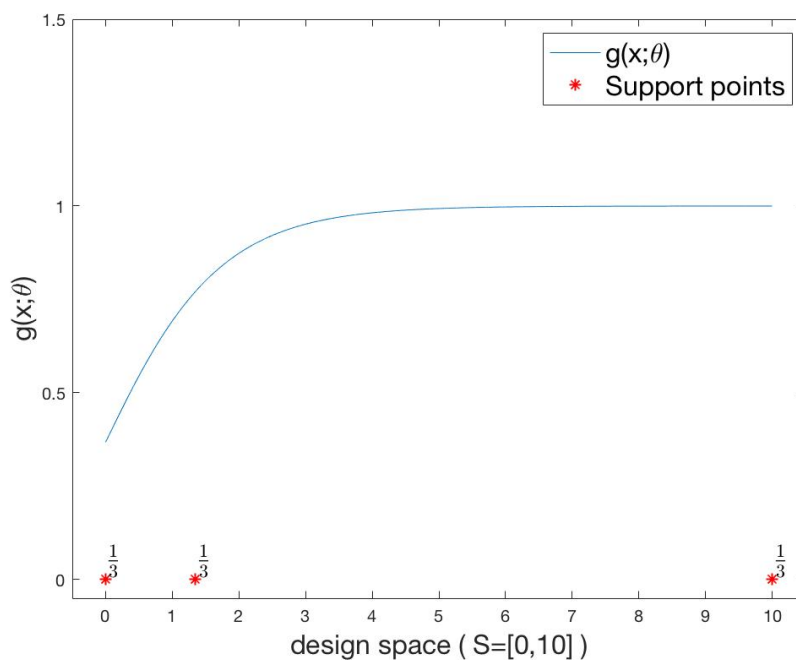
$$y_i = \theta_1 e^{-\theta_2 e^{-\theta_3 x_i}} + \epsilon_i, \quad i = 1, 2, \dots, n, \quad \boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^T, \quad x_i \in S,$$

where θ_1 describes the maximum growing capacity, θ_2 explains the initial status of the subject, θ_3 determines the growth rate, y is the overall growth at the current time point and x is the time. Note that x , θ_1 , θ_2 and θ_3 are assumed to be positive in this context. We want to study how one subject's total growth associated with time. The model has broad applications in biological science and cancer studies. See some examples in Laird (1964) and Kozusko and Bajer (2003). Suppose the design space S is $[0, 10]$ (i.e. $x \in [0, 10]$) and the true parameters of $\boldsymbol{\theta}$ is given by $\boldsymbol{\theta}_o = (1, 1, 1)^T$. For this model, D-optimal design aims to select the probability measure on S that minimizes $\det(\hat{\boldsymbol{\theta}})$, where $\hat{\boldsymbol{\theta}}$ is the SLSE. The details of the SLSE will be discussed in Section 1.2 and Chapter 2. The resulted probability measure under D-optimality is

$$\xi_D^* = \begin{bmatrix} 0.000 & 1.350 & 10.000 \\ 0.333 & 0.333 & 0.333 \end{bmatrix}, \quad (1.2)$$

where the top row represents the support points and the second row describes the corresponding probabilities on the points. The resulted design has three support points at $x = 0.000$, 1.350 and 10.000 having equal weight ($\frac{1}{3}$). The interpretation of (1.2) is that we will distribute resources evenly at three points, 0.000 , 1.350 and

Figure 1.1: A D-optimal design for Gompertz growth model with $\theta_o = (1, 1, 1)^T$ and $S = [0, 10]$, which has three support points and each point has weight $\frac{1}{3}$.



10.000. For instance, if the maximum number of runs available is fifteen due to the scarcity of resources, the researcher will make five observations at each of the three points. In Figure 1.1, the line represents the behavior of expectation function $g(\mathbf{x}; \boldsymbol{\theta})$ in the design space S , and the $*$ represents the support point.

Many studies are conducted by using ordinary least squares estimator (OLSE) as the estimator ($\hat{\boldsymbol{\theta}}$) in optimal regression design framework. OLSE is the best linear unbiased estimator (BLUE) in the regression context. However, if the error distribution is asymmetric, the SLSE is more efficient than OLSE from Wang and Leblanc (2008), which is reviewed in next section.

1.2 Second-order least squares estimator

We first discuss the relationship between OLSE and SLSE, as well as the advantages of SLSE over OLSE. OLSE is an estimator to estimate the parameter vector $\boldsymbol{\theta}$ in regression model (1.1), which is defined to be

$$\hat{\boldsymbol{\theta}} := \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - g(\mathbf{x}_i; \boldsymbol{\theta}))^2. \quad (1.3)$$

The assumptions for using OLSE are: the error terms are assumed to be homoscedastic and independently identically distributed with zero mean and finite constant variance. It has many desired properties such as consistency and it is the BLUE, which is widely used. In practice, however, other estimators might outperform OLSE in some scenarios. If the error distribution is asymmetric, Wang and Leblanc (2008) has shown that SLSE is asymptotically more efficient than OLSE. When the random errors are symmetrically distributed, SLSE and OLSE have the same asymptotic efficiency. SLSE has caught attentions in optimal regression design context due to these reasons.

We now review some properties of the SLSE in the regression model (1.1). SLSE is defined as

$$(\hat{\boldsymbol{\theta}}^T, \hat{\sigma}^2)^T := \operatorname{argmin}_{\boldsymbol{\theta}, \sigma^2} \sum_{i=1}^n \begin{pmatrix} y_i - g(\mathbf{x}_i; \boldsymbol{\theta}) \\ y_i^2 - g^2(\mathbf{x}_i; \boldsymbol{\theta}) - \sigma^2 \end{pmatrix}^T W(\mathbf{x}_i) \begin{pmatrix} y_i - g(\mathbf{x}_i; \boldsymbol{\theta}) \\ y_i^2 - g^2(\mathbf{x}_i; \boldsymbol{\theta}) - \sigma^2 \end{pmatrix}. \quad (1.4)$$

Note that $W(\mathbf{x}_i)$ is a 2×2 non-negative semi-definite matrix which may or may not depend on \mathbf{x}_i (Wang and Leblanc 2008). It is clear that SLSE is a natural extension of the OLSE which is defined based on the first-order difference function (i.e. $y_i - \mathbb{E}[y_i] = y_i - g(\mathbf{x}_i; \boldsymbol{\theta})$). On the other hand, SLSE is defined using not only the first-order difference function, but also second-order difference function (i.e. $y_i^2 - \mathbb{E}[y_i^2] = y_i^2 - (g^2(\mathbf{x}_i; \boldsymbol{\theta}) + \sigma^2)$). One might think about the downsides of the SLSE after talking about the advantages of SLSE over OLSE. SLSE does have its disadvantages indeed. It is not a linear estimator and there is no closed-form solution. It requires more

computational resources compared to the OLSE due to the nonlinearity. However, numerical results can be easily computed for SLSE nowadays. As the result, SLSE is a powerful alternative estimator to be considered in research studies and real-life applications.

1.3 Research problem

As introduced in the previous section, Wang and Leblanc (2008) showed that SLSE is asymptotically more efficient than OLSE when the error distribution is asymmetric. Optimal designs under the SLSE was proposed in Gao and Zhou (2014). Bose and Mukerjee (2015), Yin and Zhou (2017), and Gao and Zhou (2017) did further investigations, including the convexity analysis, numerical methods, transformation and scale invariance properties for both A- and D-optimality criteria. There are other commonly used design criteria under the SLSE that have not been studied in the literature. Our goal is to fill this gap by extending the results to other design criteria, as well as exploring and deriving more theoretical results for the optimal designs under the SLSE.

The rest of the thesis is organized as follows. Chapter 2 describes the detailed formulation of optimal regression designs under SLSE. We derive several analytical results including equivalence theorem and the number of support points in optimal designs. Chapter 3 explains how to use numerical algorithms to solve the proposed optimal regression design problems via convex programming. We also present several interesting applications of the optimal designs studied in this thesis. Chapter 4 provides concluding remarks and discusses possible future research topics. MATLAB code are given in the Appendix.

1.4 Main Contributions

Here is a summary of the main contributions in this thesis.

1. We have studied the c-optimality design criterion under the SLSE, which has not been studied before.
2. We have applied Kiefer's equivalence theorem (1974) to obtain the conditions for the A-, c- and D-optimal designs under the SLSE.
3. We have obtained the number of support points in optimal designs under the SLSE analytically for various regression models.
4. We have studied the generalized scale invariance property of D-optimal designs under the SLSE.
5. We have given one efficient and effective computing algorithm based on the program in MATLAB for computing optimal designs under the SLSE on discrete design spaces.

Chapter 2

Optimal Regression Designs Under SLSE

In this chapter, we first review the results and properties of optimal regression designs under the SLSE. We then derive several new analytical results for the optimal designs under the SLSE. We begin with recalling the formulation of the optimal regression designs under SLSE based on three different criteria, A-, D-, and c-optimality. The formulation of A-, and D-optimality under the SLSE was first proposed in Gao and Zhou (2014) while A-optimality was further investigated in Yin and Zhou (2017). We formulate optimal design problems under A-optimality differently so that the properties can be extended to c-optimality which has not been studied yet. Equivalence results for verifying optimal designs are also obtained. In addition, analytical results are derived for the number of support points for several regression models.

2.1 Design criteria under SLSE

Let us introduce the notations first. Assume σ_o and $\boldsymbol{\theta}_o$ are the true parameter values of σ and $\boldsymbol{\theta}$, respectively. Let $S \subset \mathbb{R}^p$ be the design space for \boldsymbol{x} . Let $\text{tr}()$ and $\text{det}()$ be the trace and determinant functions of a matrix, respectively. Moreover, let Ξ denote

the class of all probability measures on S . Define, for any $\xi \in \Xi$,

$$\mathbf{g}_1 = \mathbf{g}_1(\xi, \boldsymbol{\theta}_o) = \mathbb{E}_\xi \left[\frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_o} \right] \quad (2.1)$$

and

$$\mathbf{G}_2 = \mathbf{G}_2(\xi, \boldsymbol{\theta}_o) = \mathbb{E}_\xi \left[\frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_o} \right]. \quad (2.2)$$

For a discrete probability measure $\xi \in \Xi$, we write it as

$$\xi = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ p_1 & p_2 & \cdots & p_m \end{bmatrix},$$

where $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ are the support points in S , and p_1, \dots, p_m are the probabilities associated with those points. From Gao and Zhou (2014), the asymptotic variance-covariance matrix of $\hat{\boldsymbol{\theta}}$ under the SLSE is given by

$$\mathbb{V}(\hat{\boldsymbol{\theta}}) = \sigma_o^2(1-t)(\mathbf{G}_2 - t\mathbf{g}_1\mathbf{g}_1^T)^{-1}, \quad (2.3)$$

where $t = \frac{\mu_3^2}{\sigma_o^2(\mu_4 - \sigma_o^4)}$, $\mu_3 = \mathbb{E}[\epsilon_i^3 | \mathbf{x}]$ and $\mu_4 = \mathbb{E}[\epsilon_i^4 | \mathbf{x}]$. Note that Gao and Zhou (2014) discussed that $t \in [0, 1)$ for any error distributions, and $t = 0$ for symmetric error distributions. Define a matrix $\mathbf{J} = \mathbf{J}(\xi, \boldsymbol{\theta}_o, t) = \mathbf{G}_2 - t\mathbf{g}_1\mathbf{g}_1^T$. It is clear that matrix \mathbf{J} is proportional to the inverse of the variance-covariance matrix (2.3). For the rest of the thesis, we will be working on the design problems using matrix \mathbf{J} .

As discussed in Chapter 1, we aim to minimize the loss functions in optimal design problems, and the loss functions for D-, A- and c-optimality criteria under SLSE can be expressed as

$$\begin{aligned} \phi_D(\xi, \boldsymbol{\theta}_o, t) &= \det(\mathbf{J}^{-1}(\xi, \boldsymbol{\theta}_o, t)), \\ \phi_A(\xi, \boldsymbol{\theta}_o, t) &= \text{tr}(\mathbf{J}^{-1}(\xi, \boldsymbol{\theta}_o, t)), \\ \phi_c(\xi, \boldsymbol{\theta}_o, t) &= \mathbf{c}_1^T \mathbf{J}^{-1}(\xi, \boldsymbol{\theta}_o, t) \mathbf{c}_1, \end{aligned} \quad (2.4)$$

when \mathbf{J} is non-singular, and \mathbf{c}_1 is a given vector in \mathbb{R}^q . If \mathbf{J} is singular, all the three loss functions are defined to be $+\infty$. We use ξ_D^* , ξ_A^* and ξ_c^* to denote for D-, A- and c-optimal designs, respectively. For two measures ξ_1 and $\xi_2 \in \Xi$, define $\xi_\alpha = (1 - \alpha)\xi_1 + \alpha\xi_2$ for $\alpha \in [0, 1]$.

Lemma 1. $\log(\phi_D(\xi_\alpha, \boldsymbol{\theta}_o, t))$, $\phi_A(\xi_\alpha, \boldsymbol{\theta}_o, t)$ and $\phi_c(\xi_\alpha, \boldsymbol{\theta}_o, t)$ are convex functions of α .

The convexity results are discussed in Boyd and Vandenberghe (2004) and Wong et al. (2017). Similar convexity results are given in Bose and Murkerjee (2015). We will use $\log(\det(\mathbf{J}^{-1}(\xi, \boldsymbol{\theta}_o, t)))$ for D-optimal design for the rest of the thesis as $\log()$ is a monotonic increasing function which does not change the optimality.

Although we have formulated the loss functions, there are some issues associated with the formulation in (2.4). The reason is that \mathbf{J} is lacking of linearity. From construction of \mathbf{J} , $\mathbf{J}(\xi_\alpha, \boldsymbol{\theta}_o, t)$ is not a linear combination of $\mathbf{J}(\xi_1, \boldsymbol{\theta}_o, t)$ and $\mathbf{J}(\xi_2, \boldsymbol{\theta}_o, t)$. Thus, it is difficult to obtain the theoretical results using \mathbf{J} . To solve this issue, Gao and Zhou (2017) proposed an alternative expression for characterizing the loss functions. The key is to define a matrix

$$\mathbf{B}(\xi) = \mathbf{B}(\xi, \boldsymbol{\theta}_o, t) = \begin{pmatrix} 1 & \sqrt{t}\mathbf{g}_1^T \\ \sqrt{t}\mathbf{g}_1 & \mathbf{G}_2 \end{pmatrix}, \quad (2.5)$$

which plays an important role in the following formulation. Note $\mathbf{B}(\xi_\alpha)$ is now an affine function of α , i.e.,

$$\mathbf{B}(\xi_\alpha) = (1 - \alpha)\mathbf{B}(\xi_1) + \alpha\mathbf{B}(\xi_2).$$

This fact ultimately makes \mathbf{B} much more useful than \mathbf{J} to study optimal designs under SLSE. The inverse of \mathbf{B} is given as

$$\mathbf{B}^{-1}(\xi) = \mathbf{B}^{-1}(\xi, \boldsymbol{\theta}_o, t) = \begin{pmatrix} \frac{1}{r} & \frac{-\sqrt{t}}{r}\mathbf{g}_1^T\mathbf{G}_2^{-1} \\ \frac{-\sqrt{t}}{r}\mathbf{G}_2^{-1}\mathbf{g}_1 & \mathbf{J}^{-1} \end{pmatrix}, \quad (2.6)$$

where $r = 1 - t\mathbf{g}_1^T\mathbf{G}_2^{-1}\mathbf{g}_1$. Note that if \mathbf{J} is invertible, \mathbf{G}_2 must also be invertible since $\mathbf{G}_2 = \mathbf{J} + t\mathbf{g}_1\mathbf{g}_1^T$ and $t\mathbf{g}_1\mathbf{g}_1^T$ is positive semi-definite. Consequently, \mathbf{B}^{-1} exists from (2.6). Now, we are going to present the following lemmas to characterize the loss functions for A-, c- and D-optimal design problems. Lemma 2 is slightly different from a result in Yin and Zhou (2017), Lemma 3 is a result from Gao and Zhou (2017), and Lemma 4 is a new result.

Lemma 2. *If \mathbf{J} is invertible, then*

$$\phi_A(\xi, \boldsymbol{\theta}_o, t) = \text{tr}(\mathbf{J}^{-1}) = \text{tr}(\mathbf{C}^T \mathbf{B}^{-1} \mathbf{C}),$$

where $\mathbf{C} = 0 \oplus \mathbf{I}_q$, \mathbf{I}_q denotes for the $q \times q$ identity matrix, and \oplus denotes for matrix direct sum operator.

Proof. From (2.6) and $\mathbf{C} = 0 \oplus \mathbf{I}_q$, we get

$$\begin{aligned} \mathbf{C}^T \mathbf{B}^{-1} \mathbf{C} &= \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{I}_q \end{pmatrix}^T \begin{pmatrix} \frac{1}{r} & \frac{-\sqrt{t}}{r} \mathbf{g}_1^T \mathbf{G}_2^{-1} \\ \frac{-\sqrt{t}}{r} \mathbf{g}_1 \mathbf{G}_2^{-1} & \mathbf{J}^{-1} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{I}_q \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{J}^{-1} \end{pmatrix}, \end{aligned}$$

which implies $\text{tr}(\mathbf{C}^T \mathbf{B}^{-1} \mathbf{C}) = \text{tr}(\mathbf{J}^{-1})$. □

Lemma 3. *If \mathbf{J} is invertible, then*

$$\phi_D(\xi, \boldsymbol{\theta}_o, t) = \det(\mathbf{J}^{-1}) = \det(\mathbf{B}^{-1}).$$

Proof. From (2.5), we have

$$\begin{aligned} \det(\mathbf{B}) &= \det(1 - t \mathbf{g}_1^T \mathbf{G}_2^{-1} \mathbf{g}_1) \det(\mathbf{G}_2) \\ &= \det(\mathbf{I} - t \mathbf{g}_1 \mathbf{g}_1^T \mathbf{G}_2^{-1}) \det(\mathbf{G}_2) \\ &= \det(\mathbf{G}_2 - t \mathbf{g}_1 \mathbf{g}_1^T) \\ &= \det(\mathbf{J}), \end{aligned}$$

which gives $\det(\mathbf{J}^{-1}) = \det(\mathbf{B}^{-1})$. □

Lemma 4. *If \mathbf{J} is invertible, then*

$$\phi_c(\xi, \boldsymbol{\theta}_o, t) = \mathbf{c}_1^T \mathbf{J}^{-1} \mathbf{c}_1 = \mathbf{c}^T \mathbf{B}^{-1} \mathbf{c},$$

where \mathbf{c}_1 is a vector in \mathbb{R}^q and $\mathbf{c}^T = (0, \mathbf{c}_1^T)$.

Proof. From (2.6), we obtain

$$\begin{aligned} \mathbf{c}^T \mathbf{B}^{-1} \mathbf{c} &= \begin{pmatrix} 0, & \mathbf{c}_1^T \end{pmatrix} \begin{pmatrix} \frac{1}{r} & \frac{-\sqrt{t}}{r} \mathbf{g}_1^T \mathbf{G}_2^{-1} \\ \frac{-\sqrt{t}}{r} \mathbf{g}_1 \mathbf{G}_2^{-1} & \mathbf{J}^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{c}_1 \end{pmatrix} \\ &= \mathbf{c}_1^T \mathbf{J}^{-1} \mathbf{c}_1. \end{aligned}$$

□

Thus, by Lemmas 2, 3 and 4, the alternative expressions for the loss functions in (2.4) are

$$\begin{aligned} \phi_D(\xi, \boldsymbol{\theta}_o, t) &= \det(\mathbf{B}^{-1}(\xi, \boldsymbol{\theta}_o, t)), \\ \phi_A(\xi, \boldsymbol{\theta}_o, t) &= \text{tr}(\mathbf{C}^T \mathbf{B}^{-1}(\xi, \boldsymbol{\theta}_o, t) \mathbf{C}), \\ \phi_c(\xi, \boldsymbol{\theta}_o, t) &= \mathbf{c}^T \mathbf{B}^{-1}(\xi, \boldsymbol{\theta}_o, t) \mathbf{c}, \end{aligned} \tag{2.7}$$

where $\mathbf{C} = 0 \oplus \mathbf{I}_q$, $\mathbf{c}_1 \in \mathbb{R}^q$ and $\mathbf{c}^T = (0, \mathbf{c}_1^T)$. If \mathbf{B} is singular, all the three loss functions are defined to be $+\infty$.

2.2 Equivalence theorem for optimal designs under SLSE

In this section we derive the optimality conditions for the optimal designs under the SLSE which follows from the equivalence theorem in Kiefer and Wolfowitz (1959) and Kiefer (1974). We also analyze the minimum number of support points in optimal designs for various regression models, and theoretical results are obtained. Note we study approximate designs in this thesis. The advantages of working with approximate designs instead of exact design are well documented in Kiefer (1985).

Define a vector $f(\mathbf{x}, \boldsymbol{\theta}_o) = \left. \frac{\partial g(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_o} \in \mathbb{R}^q$ and a matrix

$$\mathbf{M}(\mathbf{x}) = \mathbf{M}(\mathbf{x}, \boldsymbol{\theta}_o, t) = \begin{pmatrix} 1 & \sqrt{t} f^T(\mathbf{x}, \boldsymbol{\theta}_o) \\ \sqrt{t} f(\mathbf{x}, \boldsymbol{\theta}_o) & f(\mathbf{x}, \boldsymbol{\theta}_o) f^T(\mathbf{x}, \boldsymbol{\theta}_o) \end{pmatrix}_{(q+1) \times (q+1)}. \tag{2.8}$$

Then $\mathbf{B}(\xi, \boldsymbol{\theta}_o, t) = \mathbb{E}_\xi[\mathbf{M}(\mathbf{x}, \boldsymbol{\theta}_o, t)]$. Define dispersion functions

$$\begin{aligned} d_D(\mathbf{x}, \xi, t) &= \text{tr}(\mathbf{B}^{-1}\mathbf{M}(\mathbf{x})) - (q+1), \\ d_A(\mathbf{x}, \xi, t) &= \text{tr}(\mathbf{M}(\mathbf{x})\mathbf{B}^{-1}\mathbf{C}^T\mathbf{C}\mathbf{B}^{-1}) - \text{tr}(\mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T), \\ d_c(\mathbf{x}, \xi, t) &= \mathbf{c}^T\mathbf{B}^{-1}\mathbf{M}(\mathbf{x})\mathbf{B}^{-1}\mathbf{c} - \mathbf{c}^T\mathbf{B}^{-1}\mathbf{c}, \end{aligned} \quad (2.9)$$

where $\mathbf{B} = \mathbf{B}(\xi, \boldsymbol{\theta}_o, t)$ is invertible.

Theorem 1. *We suppose all the dispersion functions are evaluated at $\boldsymbol{\theta}_o$. If ξ_D^* , ξ_A^* and ξ_c^* are the optimal probability measures for D-, A- and c- optimality, respectively, then \mathbf{B} is invertible and for any $\mathbf{x} \in S$,*

$$d_D(\mathbf{x}, \xi_D^*, t) \leq 0, \quad (2.10a)$$

$$d_A(\mathbf{x}, \xi_A^*, t) \leq 0, \quad (2.10b)$$

$$d_c(\mathbf{x}, \xi_c^*, t) \leq 0. \quad (2.10c)$$

Proof. In this proof, we use $\mathbf{B}(\xi)$ for $\mathbf{B}(\xi, \boldsymbol{\theta}_o, t)$ and $\mathbf{M}(\mathbf{x})$ for $\mathbf{M}(\mathbf{x}, \boldsymbol{\theta}_o, t)$. Suppose ξ^* is an optimal design to a criterion. Define $\xi_\alpha = (1 - \alpha)\xi^* + \alpha\xi$ where ξ is an arbitrary probability measure. This proof is based on Kiefer's general equivalence theorem (Kiefer, 1974), and the optimal condition can be derived from $\frac{\partial \phi(\xi_\alpha)}{\partial \alpha} \Big|_{\alpha=0} \geq 0$ for any measure $\xi \in \Xi$, where ϕ is a loss function.

We first prove (2.10a). Let ξ_D^* be the optimal measure under D-optimality. We have

$$\begin{aligned} \frac{\partial \log(\phi_D(\xi_\alpha))}{\partial \alpha} \Big|_{\alpha=0} &= -\text{tr}(\mathbf{B}^{-1}(\xi_D^*)(-\mathbf{B}(\xi_D^*) + \mathbf{B}(\xi))) \\ &= -\text{tr}(\mathbf{B}^{-1}(\xi_D^*)\mathbf{B}(\xi)) + \text{tr}(\mathbf{I}_{q+1}) \\ &= -\text{tr}(\mathbf{B}^{-1}(\xi_D^*)\mathbf{B}(\xi)) + (q+1) \\ &= -\text{tr}(\mathbf{B}^{-1}(\xi_D^*)\mathbb{E}_\xi[\mathbf{M}(\mathbf{x})]) - (q+1) \\ &= -\mathbb{E}_\xi[d_D(\mathbf{x}, \xi_D^*, t)] \\ &\geq 0, \text{ for any } \xi \text{ on } S, \end{aligned}$$

which implies $d_D(\mathbf{x}, \xi_D^*, t) \leq 0$, for all $\mathbf{x} \in S$.

To prove (2.10b), let ξ_A^* be the optimal measure under A-optimality. We have

$$\begin{aligned}
\left. \frac{\partial \phi_A(\xi_\alpha)}{\partial \alpha} \right|_{\alpha=0} &= -\text{tr}(\mathbf{C}^T \mathbf{B}^{-1}(\xi_A^*) [\mathbf{B}(\xi) - \mathbf{B}(\xi_A^*)] \mathbf{B}^{-1}(\xi_A^*) \mathbf{C}) \\
&= -\text{tr}(\mathbf{C}^T \mathbf{B}^{-1}(\xi_A^*) \mathbf{B}(\xi) \mathbf{B}^{-1}(\xi_A^*) \mathbf{C}) + \text{tr}(\mathbf{C}^T \mathbf{B}^{-1}(\xi_A^*) \mathbf{C}) \\
&= -\text{tr}(\mathbf{C}^T \mathbf{B}^{-1}(\xi_A^*) \mathbb{E}_\xi [\mathbf{M}(\mathbf{x})] \mathbf{B}^{-1}(\xi_A^*) \mathbf{C}) + \text{tr}(\mathbf{C}^T \mathbf{B}^{-1}(\xi_A^*) \mathbf{C}) \\
&= -(\text{tr}(\mathbb{E}_\xi [\mathbf{M}_\xi(\mathbf{x}, \boldsymbol{\theta}_o)] \mathbf{B}^{-1}(\xi_A^*) \mathbf{C} \mathbf{C}^T \mathbf{B}^{-1}(\xi_A^*)) - \text{tr}(\mathbf{C}^T \mathbf{B}^{-1}(\xi_A^*) \mathbf{C})) \\
&= -\mathbb{E}_\xi [d_A(\mathbf{x}, \xi_A^*, t)] \\
&\geq 0, \text{ for any } \xi \text{ on } S,
\end{aligned}$$

which implies $d_A(\mathbf{x}, \xi_A^*, t) \leq 0$, for all $\mathbf{x} \in S$.

Lastly, to prove (2.10c), let ξ_c^* be the optimal measure under c-optimality. We have

$$\begin{aligned}
\left. \frac{\partial \phi_c(\xi_\alpha)}{\partial \alpha} \right|_{\alpha=0} &= -\mathbf{c}^T \mathbf{B}^{-1}(\xi_c^*) [-\mathbf{B}(\xi) + \mathbf{B}(\xi_c^*)] \mathbf{B}^{-1}(\xi_c^*) \mathbf{c} \\
&= -\mathbf{c}^T \mathbf{B}^{-1}(\xi_c^*) \mathbf{B}(\xi) \mathbf{B}^{-1}(\xi_c^*) \mathbf{c} + \mathbf{c}^T \mathbf{B}^{-1}(\xi_c^*) \mathbf{B}(\xi_c^*) \mathbf{B}^{-1}(\xi_c^*) \mathbf{c} \\
&= -\mathbf{c}^T \mathbf{B}^{-1}(\xi_c^*) \mathbb{E}_\xi [\mathbf{M}(\mathbf{x})] \mathbf{B}^{-1}(\xi_c^*) \mathbf{c} + \mathbf{c}^T \mathbf{B}(\xi_c^*)^{-1} \mathbf{c} \\
&= -\mathbb{E}_\xi [\mathbf{c}^T \mathbf{B}^{-1}(\xi_c^*) \mathbf{M}(\mathbf{x}) \mathbf{B}^{-1}(\xi_c^*) \mathbf{c} + \mathbf{c}^T \mathbf{B}(\xi_c^*)^{-1} \mathbf{c}] \\
&= -\mathbb{E}_\xi [d_c(\mathbf{x}, \xi_c^*, t)] \\
&\geq 0, \text{ for any } \xi \text{ on } S,
\end{aligned}$$

which implies $d_c(\mathbf{x}, \xi_c^*, t) \leq 0$, for all $\mathbf{x} \in S$. □

2.3 Results on the number of support points

Using the results in Theorem 1, we can explore the properties of the optimal designs. In Yin and Zhou (2017) and Gao and Zhou (2017), there are some discussions about the number of support points based on computational results. However, there is little discussion on the number of support points theoretically in Gao and Zhou (2017), and there is still a large gap to be filled in. Hence we derive several results about the

number of support points for various models, including polynomial models, fractional polynomial models, Michaelis-Menton model, Peleg model and trigonometric models.

A polynomial regression model of degree q ($q \geq 1$) without intercept is given by

$$y_i = \theta_1 x_i + \theta_2 x_i^2 + \cdots + \theta_q x_i^q + \epsilon_i, \quad x_i \in S = [-1, +1], \quad i = 1, 2, \dots, n. \quad (2.11)$$

Polynomial regression models are widely used when the response and regressors have curvilinear relationship. Complex nonlinear relationships can be well approximated by polynomials over a small range of the explanatory variables (Montgomery et al. 2012, p. 223). There are different kinds of polynomial models such as orthogonal polynomial models, multi-variable polynomial models, and one variable polynomial models. Polynomial models are often used in design of experiment for the response surface methodology, and there are many applications in industry. For example, see Box and Draper (1987), Box et al. (1978) and Khuri and Cornell (1996).

A- and D-optimal designs for (2.11) under SLSE are symmetric on S (Yin and Zhou (2017), Gao and Zhou (2017)). In (2.11), we have $\mathbf{f}(x, \boldsymbol{\theta}) = (x, x^2, \dots, x^q)^T$ and

$$\mathbf{M}(x, \boldsymbol{\theta}_o, t) = \begin{pmatrix} 1 & \sqrt{t}x & \sqrt{t}x^2 & \dots & \sqrt{t}x^q \\ \sqrt{t}x & x^2 & x^3 & \dots & x^{q+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sqrt{t}x^q & x^{q+1} & \dots & \dots & x^{2q} \end{pmatrix}_{(q+1) \times (q+1)}. \quad (2.12)$$

Theorem 2. *Let n_A and n_D denote the minimum number of support points in A- and D-optimal designs under SLSE, respectively. For (2.11), we have*

$$n_A = q \text{ or } q+1, \quad (2.13)$$

and

$$n_D = q \text{ or } q+1. \quad (2.14)$$

Proof. The proof includes the following three parts.

(i). From (2.9) and (2.12), we can see that $d_A(x, \xi_A^*, t)$ and $d_D(x, \xi_D^*, t)$ are polynomial functions of x with highest degree $2q$. By fundamental theorem of algebra, there are

exactly $2q$ roots for x in equations $d_A(x, \xi_A^*, t) = 0$ and $d_D(x, \xi_D^*, t) = 0$. However, we have at most $2q$ real roots.

(ii). By the construction, the determinant of \mathbf{B} matrix is not zero if and only if the determinant of \mathbf{G}_2 is not zero. Therefore, there are at least q support points in ξ .

(iii). Both boundary points are the support points, so the number of support points are at most $2q - 2$ in the interval $(-1, +1)$. From the equivalence theorem, we know that the dispersion functions are all less or equal to zero (i.e. $d_A(x, \xi_A^*, t) \leq 0$ and $d_D(x, \xi_D^*, t) \leq 0$), so all those support points in $(-1, +1)$ have a multiplicity of two. In total, we have at most $2 + \frac{(2q-2)}{2} = q + 1$ distinct support points.

Thus, the number of support points in ξ_A^* and ξ_D^* is either q or $q + 1$. \square

Example 1. Consider (2.11) with $q = 2$ and $S = [-1, +1]$. Let $\eta_m = \mathbb{E}[x^m]$ be the m th moment of distribution $\xi(x)$. Gao and Zhou (2014) showed that the D -optimal design is symmetric in this case (i.e. $\eta_1 = \eta_3 = 0$), and is given by

$$\xi_D^* = \begin{cases} \begin{bmatrix} -1 & +1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, & \text{for } t \in [0, \frac{2}{3}), \\ \begin{bmatrix} -1 & 0 & +1 \\ \frac{1}{3t} & \frac{3t-2}{3t} & \frac{1}{3t} \end{bmatrix}, & \text{for } t \in [\frac{2}{3}, 1). \end{cases}$$

We now study the A -optimal design. By taking the advantage of the symmetric result in Yin and Zhou (2017), we have

$$\mathbf{B}(\xi) = \begin{pmatrix} 1 & 0 & \sqrt{t}\eta_2 \\ 0 & \eta_2 & 0 \\ \sqrt{t}\eta_2 & 0 & \eta_4 \end{pmatrix}, \text{ and } \mathbf{B}^{-1}(\xi) = \begin{pmatrix} \frac{\eta_4}{\eta_4 - t\eta_2} & 0 & \frac{\sqrt{t}\eta_2}{t\eta_2^2 - \eta_4} \\ 0 & \frac{1}{\eta_2} & 0 \\ \frac{\sqrt{t}\eta_2}{t\eta_2^2 - \eta_4} & 0 & \frac{1}{\eta_4 - t\eta_2^2} \end{pmatrix}.$$

From Dette and Sudden (1997), on $S = [-1, +1]$, the even moments of any distributions must satisfy $0 \leq \eta_2^2 \leq \eta_4 \leq \eta_2 \leq 1$. Then our loss function can be expressed as

$$\phi_A(\xi) = \text{tr}(\mathbf{C}^T \mathbf{B}^{-1} \mathbf{C}) = \frac{1}{\eta_2} + \frac{1}{\eta_4 - t\eta_2^2}.$$

The optimal design problem can be written as

$$\begin{aligned} \min_{\eta_2, \eta_4} \quad & \frac{1}{\eta_2} + \frac{1}{\eta_4 - t\eta_2^2} \\ \text{s.t.} \quad & 0 \leq \eta_2^2 \leq \eta_4 \leq \eta_2 \leq 1. \end{aligned}$$

In order to minimize the loss function, we first fix η_2 . Then it is trivial to see we should make η_4 as big as possible, which leads to $\eta_4 = \eta_2$, its ceiling. Now the question becomes

$$\begin{aligned} \min_{\eta_2} \quad & \frac{1}{\eta_2} + \frac{1}{\eta_2 - t\eta_2^2} \\ \text{s.t.} \quad & 0 \leq \eta_2 \leq 1. \end{aligned}$$

It yields to $\eta_2 = \frac{2-\sqrt{2}}{t}$ or $\eta_2 = \frac{2+\sqrt{2}}{t}$ where $t \neq 0$. Note $t \in [0, +1)$ and $\eta_2 \in [0, +1]$, so the latter solution should be excluded as it is not within the feasible region.

Consequently,

$$\eta_2 = \begin{cases} \min\left(\frac{2-\sqrt{2}}{t}, 1\right), & t \in (0, 1), \\ 1, & t = 0. \end{cases}$$

By the symmetric property, the A -optimal design is

$$\xi_A^* = \begin{bmatrix} -1 & 0 & +1 \\ \frac{\eta_2}{2} & 1 - \eta_2 & \frac{\eta_2}{2} \end{bmatrix}$$

Here we have three cases:

Case 1 When $t = 0$,

$$\xi_A^* = \begin{bmatrix} -1 & +1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Case 2 When $t \in (0, 2 - \sqrt{2})$,

$$\xi_A^* = \begin{bmatrix} -1 & +1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Case 3 When $t \in [2 - \sqrt{2}, 1)$,

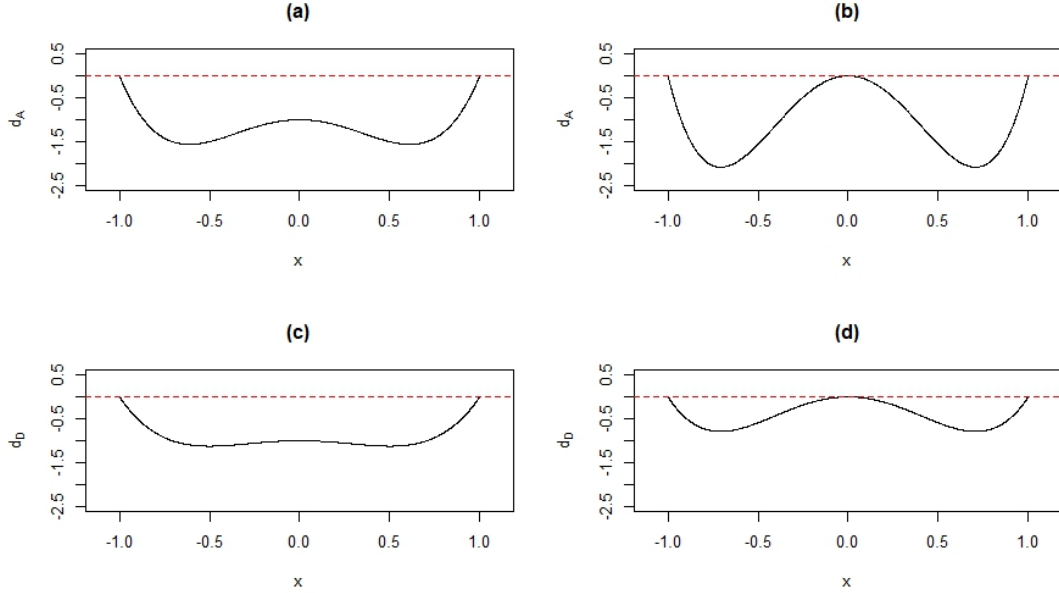
$$\xi_A^* = \begin{bmatrix} -1 & 0 & +1 \\ \frac{2-\sqrt{2}}{2t} & 1 - \frac{2-\sqrt{2}}{t} & \frac{2-\sqrt{2}}{2t} \end{bmatrix}.$$

In summary, the A -optimal design for (2.11) when $q = 2$ is

$$\xi_A^* = \begin{cases} \begin{bmatrix} -1 & +1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, & \text{for } t \in [0, 2 - \sqrt{2}), \\ \begin{bmatrix} -1 & 0 & +1 \\ \frac{2-\sqrt{2}}{2t} & \frac{2t+\sqrt{2}-2}{t} & \frac{2-\sqrt{2}}{2t} \end{bmatrix}, & \text{for } t \in [2 - \sqrt{2}, 1). \end{cases}$$

From the results above, it is clear to observe that A - and D -optimal designs have either 2 or 3 support points which is consistent with Theorem 2. \square

Figure 2.1: Plots of $d_A(x, \xi_A^*, t)$ and $d_D(x, \xi_A^*, t)$ in (2.10b) and (2.10a) for Example 1. (a) $d_A(x, \xi_A^*, t)$ with $t = 0.3$, (b) $d_A(x, \xi_A^*, t)$ with $t = 0.7$, (c) $d_D(x, \xi_A^*, t)$ with $t = 0.3$, (d) $d_D(x, \xi_A^*, t)$ with $t = 0.7$.



To show the result in Theorem 1, we plot $d_A(x, \xi^*, t)$ and $d_D(x, \xi^*, t)$ in Figure 2.1 using the optimal designs in Example 1. It is clear that $d_A(x, \xi^*, t) \leq 0$ and $d_D(x, \xi^*, t) \leq 0$ for all $x \in S$, which is consistent with Theorem 1. Also $d_A(x, \xi_A^*, t) = 0$ at the support points of ξ_A^* and $d_D(x, \xi_D^*, t) = 0$ at the support points of ξ_D^* . As t

increases, the origin becomes a support point which changes the number of support points from 2 to 3. This is again consistent with Theorem 2.

When the design space S is asymmetric, say $S = [-1, b]$, $b \in (-1, 1)$, the number of support points under A- and D-optimality are either q or $q + 1$ for all $q \in \mathbb{Z}^+$. When $t = 0$, $n_D = q$ for all $q \in \mathbb{Z}^+$. The derivation is similar to that of Theorem 2 and is omitted. We will show some computational results in Chapter 3.

Fractional polynomial model (FPM) is given by

$$y = \theta_1 x^{q_1} + \theta_2 x^{q_2} + \cdots + \theta_p x^{q_p} + \epsilon, \quad q_i \in \mathbb{Q}, \quad \forall i, \quad (2.15)$$

which provides more flexible parameterization with wide range of applications in many disciplines. For instance, Cui et al. (2009) used FPM for longitudinal epidemiological studies, and Royston and Altman (1994) applied this model to analyze medical data. This model has also been studied for optimal designs. For example, Torsney and Alahmadi (1995) used FPM to study the effects of concentration x to viscosity y and this model is given by

$$y = \theta_1 x + \theta_2 x^{\frac{1}{2}} + \theta_3 x^2 + \epsilon, \quad x \in S = (0.0, 0.2]. \quad (2.16)$$

Let $z = x^{1/2}$, then (2.16) becomes a polynomial model of order 4 with the new design space $S' = (0.0, \sqrt{0.2}]$. Now, we can apply the result for the polynomial model (2.11) and conclude that n_A and n_D are at most 4 or 5. Moreover, notice that there are only three parameters in model (2.16). In order to have an invertible matrix $\mathbf{G}_2(\xi, \theta_o)$ which is a 3×3 matrix, we need at least 3 support points. Thus, the number of support point for both A- and D-optimal design is at least 3. In summary, the number of support points in A- and D-optimal designs are either 3, 4, or 5. The purpose of this example is for illustrating that FPM can be transformed into a polynomial model so we can use the result for the number of support points in polynomial models.

Michaelis-Menton model is one of the best-known models proposed by Leonor Michaelis and Maud Menten (Michaelis and Menten, 1913) that studies the enzyme reactions between the enzyme and the substrate concentration. This model is given

by

$$y_i = \frac{\theta_1 x_i}{\theta_2 + x_i} + \epsilon_i, \quad i = 1, 2, \dots, n, \quad x_i \in (0, k_o], \quad \theta_1, \theta_2 \geq 0. \quad (2.17)$$

Here, $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$. Define $z = \frac{1}{\theta_2 + x}$. Then $\mathbf{f}(x, \boldsymbol{\theta}) = \left(\frac{x}{\theta_2 + x}, \frac{-\theta_1 x}{(\theta_2 + x)^2}\right)^T = (1 - z\theta_2, -z\theta_1 + \theta_1\theta_2 z^2)^T$, and

$$\mathbf{M}(x, \boldsymbol{\theta}_o, t) = \begin{pmatrix} 1 & \sqrt{t}(1 - z\theta_2) & \sqrt{t}(-\theta_1 z(1 - z\theta_2)) \\ \sqrt{t}(1 - z\theta_2) & (1 - z\theta_2)^2 & -z\theta_1(1 - z\theta)^2 \\ \sqrt{t}(-\theta_1 z(1 - z\theta_2)) & -z\theta_1(1 - z\theta)^2 & (z\theta_1(1 - z\theta_2))^2 \end{pmatrix}. \quad (2.18)$$

We can clearly see that, after the transformation, $z \in S' = [\frac{1}{\theta_1 + k_o}, \frac{1}{\theta_1}]$, and since $\mathbf{f}(x, \boldsymbol{\theta}_o)$ and $\mathbf{M}(x, \boldsymbol{\theta}, t)$ are polynomial functions of z , ϕ_D and ϕ_A can be described as polynomial functions with highest degrees 4. Now, similar to the discussion for model (2.11), we can find the results for n_A and n_D for model (2.17) and they are presented in the following Lemma.

Lemma 5. *For Michaelis-Menton model in (2.17), the number of support points for D-optimal design and A-optimal design are either 2 or 3 for $t \in [0, 1)$. Moreover, there are 2 support points under D-optimality when $t = 0$.*

The proof of Lemma 5 is similar to that of Theorem 2 and is omitted. It is easy to observe that $d_D(0, \xi_D^*, 0) < 0$, so the boundary point, 0, is not a support point under D-optimality which leads to the conclusion $n_D = 2$. Moreover, $n_c \leq 3$ for $t \in [0, 1)$.

Peleg model is a statistical model used to investigate the relationship of water absorption for various kinds of food. This model is given by

$$y_i = y_o + \frac{x_i}{\theta_1 + \theta_2 x_i} + \epsilon_i, \quad i = 1, 2, \dots, n, \quad (2.19)$$

where y_o represents the initial moisture of the food, y_i is the current level of moisture at current time x_i , θ_1 is the Peleg's moisture rate constant, and θ_2 is the asymptotic moisture as time increases. Note that parameters θ_1 and θ_2 must be positive. Optimal experimental designs have been studied for this model using OLSE, for example, Paquet-Durand et al. (2015). For (2.19), $\mathbf{f}(x, \boldsymbol{\theta}) = \left(\frac{-x}{(\theta_1 + \theta_2 x_i)^2}, \frac{-x^2}{(\theta_1 + \theta_2 x)^2}\right)^T$. We let

$z = \frac{1}{\theta_1 + \theta_2 x}$. Then for $S = [0, d]$, the new design space after the transformation is $S' = [\frac{1}{\theta_1 + \theta_2 d}, \frac{1}{\theta_1}]$. Now, $\mathbf{f}(x, \boldsymbol{\theta}) = (\frac{-z(1-\theta_1 z)}{\theta_2}, \frac{-(1-\theta_1 z)^2}{\theta_2^2})^T$. Since θ_1 and θ_2 are parameters, \mathbf{f} depends on z only. Hence, $\mathbf{M}(x, \boldsymbol{\theta}, t)$ becomes a polynomial function of z of degree 4. Therefore, the number of support points, n_A and n_D are also either 2 or 3 for the Peleg model. In addition, n_c is either 1, 2 or 3 for $t \in [0, 1)$.

Next we consider trigonometric models without intercept. If the intercept term is included, it has been proven that optimal designs under SLSE and OLSE are the same. The k th order trigonometric model is given by

$$y_i = \sum_{j=1}^k [\cos(jx_i)\theta_{1j} + \sin(jx_i)\theta_{2j}] + \epsilon_i, \quad i = 1, 2, \dots, n, \quad (2.20)$$

where $x_i \in S_b = [-b\pi, b\pi]$, $0 < b \leq 1$. Let $\boldsymbol{\theta} = (\theta_{11}, \theta_{12}, \dots, \theta_{1k}, \theta_{21}, \dots, \theta_{2k})^T$. From (2.8), we get

$$\mathbf{M}(x, \boldsymbol{\theta}_o, t) = \begin{pmatrix} 1 & \sqrt{t}\cos(x) & \dots & \sqrt{t}\sin(kx) \\ \sqrt{t}\cos(x) & \cos^2(x) & \dots & \cos(x)\sin(kx) \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{t}\sin(kx) & \cos(x)\sin(kx) & \dots & \sin^2(kx) \end{pmatrix}_{(2k+1) \times (2k+1)}. \quad (2.21)$$

We consider two cases of design space, (i) $b = 1$, full circle and (ii) $0 < b < 1$, the partial circle.

For case (i), the following Lemma is helpful for finding the number of support points for (2.20).

Lemma 6. *For trigonometric functions, $j, u = 1, 2, \dots, k$, $u \neq j$, we have*

1. $\int_{-\pi}^{\pi} \cos(jx)dx = \int_{-\pi}^{\pi} \sin(jx)dx = 0$,
2. $\int_{-\pi}^{\pi} \cos^2(jx)dx = \int_{-\pi}^{\pi} \sin^2(jx)dx = \pi$,
3. $\int_{-\pi}^{\pi} \cos(jx)\cos(ux)dx = \int_{-\pi}^{\pi} \sin(jx)\sin(ux)dx = 0$,
4. $\int_{-\pi}^{\pi} \sin(jx)\cos(ux)dx = \int_{-\pi}^{\pi} \cos(jx)\sin(ux)dx = 0$. *This also holds for $u = j$.*

Theorem 3. For model (2.20) with design space $S = [-\pi, \pi]$, the uniform distribution on S is both A- and D-optimal designs.

Proof. For the uniform distribution ξ^* , we have $\mathbf{B}(\xi^*, \boldsymbol{\theta}_o, t) = 1 \oplus \frac{1}{2} \mathbf{I}_{2k}$ by Lemma 6. From (2.21), we obtain

$$\begin{aligned}
& \text{tr}(\mathbf{M}(x, \boldsymbol{\theta}_o, t) \mathbf{B}^{-1}(\xi^*, \boldsymbol{\theta}_o, t)) - (q + 1) \\
&= 1 + 2\cos^2(x) + 2\cos^2(2x) + \cdots + 2\sin^2(kx) - (2k + 1) \\
&= 1 + 2[\cos^2(x) + \sin^2(x)] + \cdots + 2[\cos^2(kx) + \sin^2(kx)] - (2k + 1) \\
&= 1 + 2k - (2k + 1) \\
&= 0, \quad \text{for all } x \in S.
\end{aligned}$$

This implies that $d_D(x, \xi^*, t) = 0$ for all $x \in S$. By Theorem 1, ξ^* is a D-optimal design.

For A-optimality, it is easy to get

$$\begin{aligned}
& \text{tr}(\mathbf{M}(x, \boldsymbol{\theta}_o, t) \mathbf{B}^{-1}(\xi^*, \boldsymbol{\theta}_o, t) \mathbf{C}^T \mathbf{C} \mathbf{B}^{-1}(\xi^*, \boldsymbol{\theta}_o, t)) - \text{tr}(\mathbf{C} \mathbf{B}^{-1}(\xi^*, \boldsymbol{\theta}_o, t)) \\
&= 4\cos^2(x) + 4\cos^2(2x) + \cdots + 4\sin^2(kx) - 4k \\
&= 4[\cos^2(x) + \sin^2(x)] + \cdots + 4[\cos^2(kx) + \sin^2(kx)] - 4k \\
&= 4k - 4k \\
&= 0, \quad \text{for all } x \in S,
\end{aligned}$$

which gives $d_A(x, \xi^*, t) = 0$ for all $x \in S$. Thus, by Theorem 1, ξ^* is an A-optimal design. \square

For case (ii), $0 < b < 1$, the partial circle $S = [-b\pi, +b\pi]$, let $z = \cos(x)$. Now, instead of using x directly, we study the number of support point of x through z in $S' = [\cos(b\pi), 1]$. Note that cosine is an even function, so each point of $z \in S'$ corresponds to two symmetric points around 0, $\pm x \in S$. Gao and Zhou (2017) discussed that all the elements in $\mathbf{M}(x, \boldsymbol{\theta}_o, t)$ in (2.21) can be written as polynomial functions of z . Hence, functions $d_A(x, \xi, t)$, $d_D(x, \xi, t)$ and $d_c(x, \xi, t)$ are all polynomial functions of z with degree $2k$. Using the similar arguments for the polynomial models, A-, D- and c-optimal designs, in terms of $z \in S'$, cannot exceed $k + 1$ support points.

2.4 Scale invariance property of D-optimal design

D-optimality is one of the most used design criteria due to its many advantages. One good property of this criterion is that it is invariant under some scaling of the independent variables using OLSE (Berger and Wong 2009, p. 40). Recently there are some discussions about the invariance properties including scale invariance and shift invariance for the optimal designs under the SLSE. Examples can be found in Gao and Zhou (2014) and Yin and Zhou (2017). In this section, we focus on finding a new property of scale invariance of D-optimal designs under the SLSE for nonlinear regression models.

For linear regression models, D-optimal designs are often scale invariant. On the other hand, if the model is nonlinear, the scale invariance property is no longer available. The optimal designs for nonlinear models are called locally optimal, since they depend on the true parameter vector $\boldsymbol{\theta}_o$. Thus, D-optimal designs have to be constructed for each $\boldsymbol{\theta}_o$ and for each S . Wong and Zhou (2018) proposed a generalized scale invariance (GSI) concept for studying scale invariance property of D-optimal designs for nonlinear models and generalized linear models. The GSI property is also useful for studying D-optimal designs under the SLSE such that the designs may be constructed on a scaled design space S^V instead of the original design space S .

Denote the D-optimal design for a given model, with true parameter vector $\boldsymbol{\theta}_o$ on a design space S by $\xi_D^*(S, \boldsymbol{\theta}_o)$. Also, denote the scaled design space from the original design space S by $S^V = \{\mathbf{V}\mathbf{x} | \mathbf{x} \in S\}$.

Definition 1 (Scale matrix). *The matrix \mathbf{V} is called scale matrix which is a diagonal matrix defined as $\mathbf{V} = \text{diag}(v_1, v_2, \dots, v_p)$, where all v_i are positive.*

Definition 2 (Scale invariant). *Transforming \mathbf{x} to $\mathbf{V}\mathbf{x}$ is called scale transformation.*

Definition 3 (Generalized scale Invariant). *$\xi_D^*(S, \boldsymbol{\theta}_o)$ is said to be scale invariant for a model if there exists a parameter vector $\tilde{\boldsymbol{\theta}}_o$ such that the D-optimal design $\xi_D^*(S^V, \tilde{\boldsymbol{\theta}}_o)$ can be obtained from $\xi_D^*(S, \boldsymbol{\theta}_o)$ using some scale transformations.*

This property of the design, $\xi_D^*(S, \boldsymbol{\theta}_o)$, is defined as a generalized scale invariance. Note that $\tilde{\boldsymbol{\theta}}_o$ and $\boldsymbol{\theta}_o$ are closed related through the elements v_1, \dots, v_p in the scale matrix \mathbf{V} . When the model is linear, GSI of $\xi_D^*(S, \boldsymbol{\theta}_o)$ becomes the traditional scale invariance since it does not depend on $\boldsymbol{\theta}_o$. For many nonlinear regression models, the GSI property holds for D-optimal designs under both OLSE and SLSE. The following lemma provides a condition to check for this property.

Lemma 7. *For a given model and a scale matrix \mathbf{V} , if there exists a parameter vector $\tilde{\boldsymbol{\theta}}_o$ and an invertible diagonal matrix \mathbf{K} which does not depend on \mathbf{x} , such that $\mathbf{f}(\mathbf{V}\mathbf{x}, \tilde{\boldsymbol{\theta}}_o) = \mathbf{K}\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}_o)$ for all $\mathbf{x} \in S$, then the design $\xi_D^*(S, \boldsymbol{\theta}_o)$ has the generalized scale invariance property.*

Proof. For design space S and parameter vector $\boldsymbol{\theta}_o$, $\xi_D^*(S, \boldsymbol{\theta}_o)$ minimizes $\phi_D^*(\xi, \boldsymbol{\theta}_o) = \det(\mathbf{B}^{-1}(\xi, \boldsymbol{\theta}_o))$ where $\mathbf{B}(\xi, \boldsymbol{\theta}_o) = \mathbb{E}[\mathbf{M}(\mathbf{x}, \boldsymbol{\theta}_o)]$ and $\mathbf{M}(\mathbf{x}, \boldsymbol{\theta}_o)$ is given by (2.8), and the expectation is taken with respect to $\xi(\mathbf{x})$ on S .

For design space $S^V = \{\mathbf{V}\mathbf{x} | \mathbf{x} \in S\}$ and parameter vector $\tilde{\boldsymbol{\theta}}_o$, we minimize the following function to find $\xi_D^*(S^V, \tilde{\boldsymbol{\theta}}_o)$, $\phi_D^*(\xi, \tilde{\boldsymbol{\theta}}_o) = \det(\mathbf{B}^{-1}(\xi, \tilde{\boldsymbol{\theta}}_o))$, where $\mathbf{B}(\xi, \tilde{\boldsymbol{\theta}}_o) = \mathbb{E}[\mathbf{M}(\mathbf{z}, \tilde{\boldsymbol{\theta}}_o)]$, and the expectation is taken with respect to $\xi(\mathbf{z})$ on S^V . Each $\mathbf{z} \in S^V$ can be written as $\mathbf{V}\mathbf{x}$ followed by definition, where $\mathbf{x} \in S$. Thus, we have $\mathbf{M}(\mathbf{z}, \tilde{\boldsymbol{\theta}}_o) = \mathbf{M}(\mathbf{V}\mathbf{x}, \tilde{\boldsymbol{\theta}}_o)$.

By the assumption of Lemma 7 and (2.8), we get

$$\mathbf{M}(\mathbf{V}\mathbf{x}, \tilde{\boldsymbol{\theta}}_o) = (1 \oplus \mathbf{K})\mathbf{M}(\mathbf{x}, \boldsymbol{\theta}_o)(1 \oplus \mathbf{K}), \text{ for all } \mathbf{x} \in S.$$

Therefore, $\mathbf{B}(\xi, \tilde{\boldsymbol{\theta}}_o) = (1 \oplus \mathbf{K})\mathbf{B}(\xi, \boldsymbol{\theta}_o)(1 \oplus \mathbf{K})$ and $\phi_D(\xi, \tilde{\boldsymbol{\theta}}_o) = \frac{\phi_D(\xi, \boldsymbol{\theta}_o)}{\det^2(\mathbf{K})}$. This implies that we can minimize $\phi_D(\xi, \boldsymbol{\theta}_o)$ to obtain $\xi_D^*(S^V, \tilde{\boldsymbol{\theta}}_o)$, where $\xi_D^*(S^V, \tilde{\boldsymbol{\theta}}_o)$ is the scale transformation from $\xi_D^*(S, \boldsymbol{\theta}_o)$. \square

Example 2. *Consider Peleg regression model in (2.19). In this model, since $p = 1$, the scale matrix $V = v_1 > 0$ is a scalar and $\boldsymbol{\theta}_o = (a, b)^T$. Now, let $\tilde{\boldsymbol{\theta}}_o = (a, \frac{b}{v_1})^T$ and*

we can obtain

$$\begin{aligned}
\mathbf{f}(\mathbf{V}\mathbf{x}, \tilde{\boldsymbol{\theta}}_o) &= \mathbf{f}(v_1x, \tilde{\boldsymbol{\theta}}_o) \\
&= \left(\frac{-v_1x}{(a+\frac{b}{v_1}v_1x)^2}, \frac{-(v_1x)^2}{(a+\frac{b}{v_1}v_1x)^2} \right)^T \\
&= \left(v_1 \frac{-x}{(a+bx)^2}, v_1^2 \frac{-x^2}{(a+bx)^2} \right)^T \\
&= \begin{pmatrix} v_1 & 0 \\ 0 & v_1^2 \end{pmatrix} \left(\frac{-x}{(a+bx)^2}, \frac{-(v_1x)^2}{(a+bx)^2} \right)^T \\
&= \mathbf{K} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}_o).
\end{aligned}$$

Hence, Peleg model has GSI property based on Lemma 7 by choosing $\mathbf{K} = \text{diag}(v_1, v_1^2)$ and $\tilde{\boldsymbol{\theta}}_o = (a, \frac{b}{v_1})^T$. \square

It is worth noting that when matrix $\mathbf{B}(\xi, \boldsymbol{\theta}_o, t)$ is ill-conditioned (i.e. the condition number of the matrix is very large), numerical algorithm computing $\mathbf{B}^{-1}(\xi, \boldsymbol{\theta}_o, t)$ may fail as the numerical inverse is inaccurate and imprecise. In these situations, the GSI property may be helpful. Here we provide one example to demonstrate the usefulness of the GSI property.

Example 3. *Piecewise polynomial regression using knots is frequently used and has various applications. See Dette et al. (2008) and the references therein. They investigated optimal designs under OLSE for piecewise polynomial regression model with unknown knots, and obtained results for the number of support points under D-optimality and various other properties about designs. Here we consider one model they used, a cubic spline regression model with unknown knots, which is given by*

$$y = \theta_1 + \theta_2x + \theta_3x^2 + \theta_4x^3 + \theta_5(x - \lambda)_+^3 + \epsilon, \quad x \in [0, b] \quad (2.22)$$

where $(x - \lambda)_+ = \max(0, (x - \lambda))$. Model (2.22) is nonlinear with parameter vector $\boldsymbol{\theta}_o = (\theta_1, \dots, \theta_5, \lambda)^T$, $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}_o) = (1, x, x^2, x^3, (x - \lambda)_+^3, -3\theta_5(x - \lambda)_+^2)^T$. We now start to illustrate GSI property for this model. Consider a scale matrix $\mathbf{V} = v_1 > 0$ and let

$\tilde{\boldsymbol{\theta}}_{\mathbf{o}} = (\theta_1, \dots, \theta_5, v_1\lambda)^T$. Then we have, for all $x \in [0, b]$,

$$\begin{aligned}
\mathbf{f}(\mathbf{V}\mathbf{x}, \tilde{\boldsymbol{\theta}}_{\mathbf{o}}) &= \mathbf{f}(v_1x, \tilde{\boldsymbol{\theta}}_{\mathbf{o}}) \\
&= (1, v_1x, (v_1x)^2, (v_1x)^3, (v_1x - v_1\lambda)_+^3, -3\theta_5(v_1x - v_1\lambda)_+^2)^T \\
&= (1, v_1, x, (v_1x)^2, (v_1x)^3, v_1^3(x - \lambda)_+^3, -3\theta_5v_1^2(x - \lambda)_+^2)^T \\
&= \text{diag}(1, v_1, v_1^2, v_1^3, v_1^3, v_1^2)(1, x, x^2, x^3, (x - \lambda)_+^3, -3\theta_5(x - \lambda)_+^2)^T \\
&= \mathbf{K}\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}_{\mathbf{o}}), \quad \text{with } \mathbf{K} = \text{diag}(1, v_1, v_1^2, v_1^3, v_1^3, v_1^2).
\end{aligned}$$

Hence, by Lemma 7, the D -optimal design under SLSE has the GSI property. Moreover, the model is linear in $\boldsymbol{\theta}'_{\mathbf{o}} = (\theta_1, \theta_2, \dots, \theta_5)^T$, so the D -optimal designs under both OLSE and SLSE do not depend on $\boldsymbol{\theta}'_{\mathbf{o}}$ (Yin and Zhou, 2017). Thus, for $S = [0, b]$ and $S^V = [0, v_1b]$, the D -optimal designs $\xi_D^*(S, \boldsymbol{\theta}_{\mathbf{o}})$ and $\xi_D^*(S^V, \tilde{\boldsymbol{\theta}}_{\mathbf{o}})$ can be obtained from each other by using the corresponding $\boldsymbol{\theta}_{\mathbf{o}}$ and $\tilde{\boldsymbol{\theta}}_{\mathbf{o}}$. This property can help us dramatically for finding the D -optimal designs when matrix $\mathbf{B}(\xi, \boldsymbol{\theta}_{\mathbf{o}}, t)$ is ill-conditioned. Numerical results of D -optimal designs are presented in Example 9 after we discuss about numerical algorithms in Chapter 3. \square

We also want to note that in model (2.22), the intercept is included so the D -optimal designs under both OLSE and SLSE are the same (Gao and Zhou 2014). Moreover, the result may easily be extended for piecewise regression models with multiple unknown knots.

Chapter 3

Numerical Algorithm and Applications

We have studied the optimality conditions and the number of support points for optimal designs under SLSE in Chapter 2. However, it is still challenging to find optimal designs analytically, except for simple regression models. Often numerical algorithms are used to compute optimal designs, and various algorithms have been studied including Titterton (1976), Silvey et al. (1978), Dette et al. (2008), and Torsney and Martin-Martin (2009). For optimal designs under the SLSE, Bose and Mukerjee (2015) used multiplicative algorithms for finding the optimal designs whereas Gao and Zhou (2017) used convex programming and semi-definite programming (SDP) in MATLAB. Here we want to further extend the computational strategies for finding optimal designs under the SLSE. In this chapter, we discuss an effective algorithm to compute optimal designs based on CVX programming in MATLAB.

3.1 Optimization problem

We first discretize the design space $S \subset \mathbb{R}^p$, and the discretized space is denoted as $S_N \subset S$, where N is the number of points in S_N . The magnitude of N can be very large, but often the designs are highly efficient already with moderate N .

More discussions about the choice of N is given in the examples in this chapter. The discretization can be done in many different ways, but we discretize the design space using equally spaced grid points for the sake of simplicity. For an one dimensional design space, say $S = [a, b]$, the design points are $a + (i - 1)\frac{(b-a)}{N}, i = 1, 2, \dots, N$. For higher dimensional design spaces, we use equally spaced grid points for each variable, and S_N is formed by Cartesian product.

Suppose ξ^* is the optimal probability measure in Ξ_N , where Ξ_N includes all the distributions on S_N . We denote $S_N = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\} \subset S$. Any probability measure $\xi \in \Xi_N$ can be express as

$$\xi = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \\ w_1 & w_2 & \dots & w_N \end{bmatrix},$$

where w_i is the weight at \mathbf{u}_i , which is non-negative and $\sum_{i=1}^N w_i = 1$. The optimal regression design problem can be expressed as

$$\begin{aligned} \min_{w_1, \dots, w_N, \delta} \quad & \delta \\ \text{s.t.} \quad & \sum_{i=1}^N w_i = 1, \\ & -w_i \leq 0, \text{ for } i = 1, 2, \dots, N, \\ & \phi \leq \delta, \end{aligned} \tag{3.1}$$

where ϕ is a loss function, which can be ϕ_A , $\log(\phi_D)$ or ϕ_c defined in (2.7). Notice that for any $\xi \in \Xi_N$, $\mathbf{B}(\xi, \boldsymbol{\theta}_o, t) = \mathbb{E}_\xi[\mathbf{M}(\mathbf{x}, \boldsymbol{\theta}_o, t)] = \sum_{i=1}^N \mathbf{M}(\mathbf{u}_i, \boldsymbol{\theta}_o, t)w_i$, where $\mathbf{M}(\mathbf{u}_i, \boldsymbol{\theta}_o, t)$ is defined in (2.8). We also note that a design point is selected if its weight is greater than a very small number δ_1 (a small positive number, say 10^{-5}) in practice. Lastly, we report the $d_D^{\max} = \max\{d_D | \mathbf{x} \in S\}$ (d_A^{\max} , d_c^{\max}) for D-optimality (A-, c-optimality), where d_D, d_A and d_c are defined in (2.10). If $d_D^{\max} < \delta_2$ (a small positive number, say 10^{-4}), then the numerical solution to the above optimization problem is a D-optimal design by Theorem 1. Similarly we can check for A- and c-optimal designs. Problem (3.1) is a convex optimization problem. We will discuss an algorithm to solve (3.1) in the next section.

3.2 Computational Algorithm

There are some numerical algorithms which have been studied under the SLSE. Bose and Mukerjee (2015) applied multiplicative algorithms for finding the optimal designs to binary design points. Gao and Zhou (2017) computed D-optimal designs using the moments of distribution $\xi \in \Xi$. Yin and Zhou (2017) formulated the design problems as convex optimization problem and applied convex optimization programs CVX and SeDuMi in MATLAB to find D- and A-optimal designs, respectively. In their work, the A-optimal design problems are solved by SeDuMi which is hard to write the code in MATLAB. In our newly proposed formulation in (3.1), we are able to use CVX to find A-optimal designs, which is much easier to code. Moreover, the CVX programming can be also applied to find c-optimal designs, which the previous algorithms were not applied before.

Figure 3.1 illustrates a simple algorithm with six steps to apply CVX for finding optimal designs. Users input the variables, partition the design space into N grid points, and compute the \mathbf{M} matrix in the first three steps. Steps (i), (ii) and (iii) depend on the regression model and the design space. Step (iv) solves the optimal design problem by CVX program, which depends on the optimality criterion. The details of using CVX and MATLAB code can be found in Appendix A and Appendix B. We present several examples using this algorithm to show optimal designs in Section 3.3.

3.3 Applications

We compute A-, D- and c-optimal designs for various regression models. Notice that when the intercept term presents, the resulting optimal designs under the OLSE and the SLSE are the same (Gao and Zhou, 2014). Since there are many studies in optimal regression designs under the OLSE in the literature already, we only consider the models without the intercept. All the examples are computed via CVX package

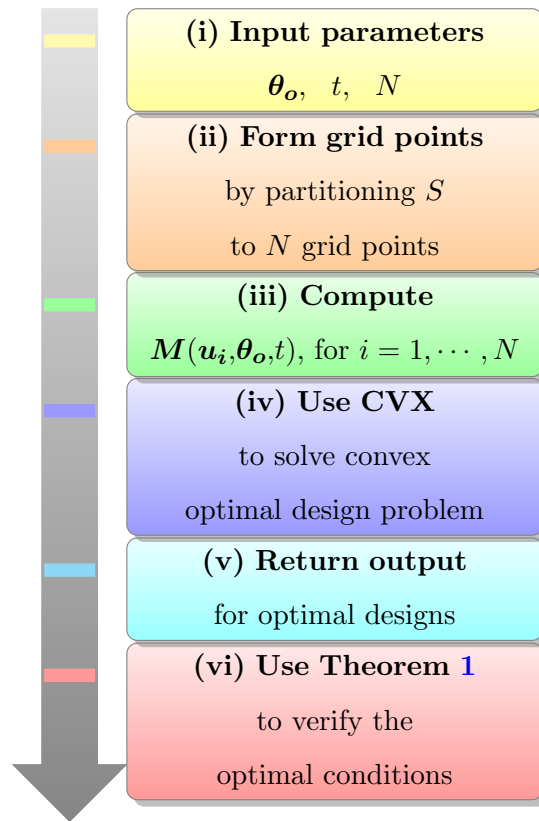


Figure 3.1: Flow chart for computing optimal designs through CVX programming.

in MATLAB. We provide the results with different values of t , where t is related to a measure of skewness of the error distribution. The related MATLAB code can be found in Appendix B.

All the examples in this thesis are computed on a PC with Intel Core I5 6500 4 cores CPU, 3.20 GHz. The MATLAB version is R2018a academic student version with CVX version 2.1 and build number 1123. The RAM and the platform for this particular machine are 16 Gigabyte and Windows 10 Professional version, respectively.

Example 4. *This example is for Gompertz growth model which was briefly described in Chapter 1 to showcase the output. Here, we use this model to show the effects of N on the D -optimal designs under SLSE with $\theta_o = (1, 1, 1)^T$, $S = [0.0, 10.0]$ and $t = 0$.*

Table 3.1 gives D -optimal designs for various values of N . Note that the D -optimal

Table 3.1: D-optimal designs for Gompertz growth model with $t = 0.0$, $S = [0.0, 10.0]$, $\theta_o = (1, 1, 1)^T$ and various values of N .

N	Support point (weight)	$\log(\phi_D)$	d^{\max}
51	0.000(0.333), 1.400(0.333), 10.00(0.333)	7.9183	2.4e-7
101	0.000(0.333), 1.300(0.169), 1.400(0.165), 10.000(0.333)	7.9175	6.6e-10
501	0.000(0.333), 1.340(0.201), 1.360(0.132), 10.000(0.333)	7.9163	8.7e-08
1001	0.000(0.333), 1.350(0.333), 10.000(0.333)	7.9162	5.9e-9
2001	0.000(0.333), 1.350(0.333), 10.000(0.333)	7.9162	1.2e-7
5001	0.000(0.333), 1.348(0.047), 1.350(0.287), 10.000(0.333)	7.9162	6.3e-9
10001	0.000(0.333), 1.349(0.333), 10.000(0.333)	7.9162	2.2e-8
20001	0.000(0.333), 1.349(0.163), 1.350(0.170), 10.000(0.333)	7.9162	4.8e-8

design in Figure 1.1 in Chapter 1 is for $N = 2001$. We can see that the optimal design changes when N increases. However, when $N \geq 1001$, the optimal designs do not change much and converge to a distribution having three support points with equal weights. We also show the change of the loss function as N changes in Figure 3.2. As $N \rightarrow +\infty$, the loss function converges too. In this example, moderate N already gives a highly efficient design as shown in Figure 3.2. Similar results are obtained for A- and c-optimal designs, and representative results are given in Table 3.2, where $\mathbf{c}_1 = (2, 0.5, 1)^T$ is used for c-optimal designs, and $t = 0.0, 0.3, 0.5, 0.7$ and 0.9 .

It is clear that $d_A^{\max} = \max_x (d_A(x, \xi_A^*, t))$ (or $d_c(x, \xi_c^*, t)$) is small, which satisfies the optimality conditions in Theorem 1. We should also point out that for some situations when two numerical support points are very close to each other, they are probably representing only one theoretical support point which is between the two numerical support points. For instance, when $t = 0.0$, the two support points 1.315 and 1.320 are very close to each other in A-optimal design. If we discretize the design space to more points (i.e. choose larger N), we can find this particular point. For instance,

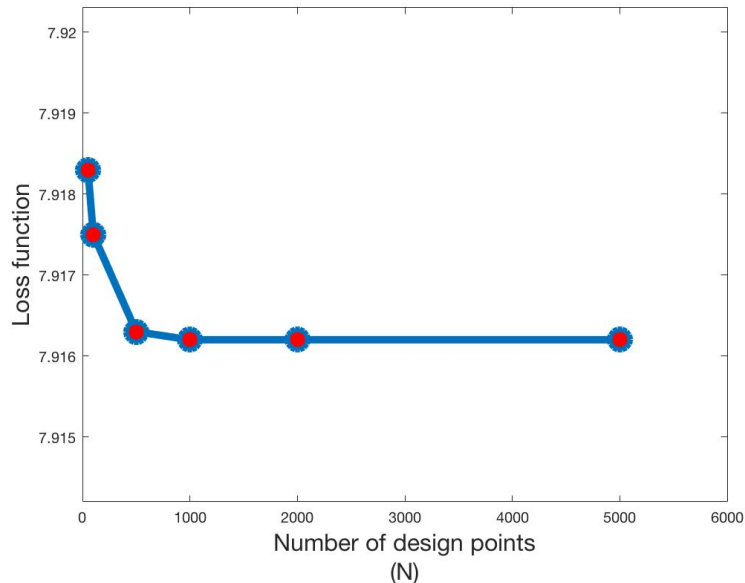


Figure 3.2: Loss function ($\log(\phi_D)$) for D-optimal design versus the number of points (N) for Gompertz growth model.

the A-optimal design with $N = 9001$ is

$$\xi_A^* = \begin{bmatrix} 0.000 & 1.318 & 10.000 \\ 0.354 & 0.385 & 0.261 \end{bmatrix},$$

which only has three support points and 1.318 is indeed in between 1.315 and 1.320. The computational time changes from 1.65 seconds to 86.16 seconds for $N = 51$ to $N = 20001$, which increases quite a bit but the computational time is still less than two minutes.

□

Example 5. We consider the polynomial regression model in (2.11) with $q = 5$. Here $\mathbf{f}(x; \boldsymbol{\theta}) = (x, x^2, \dots, x^q)^T$ which does not depend on $\boldsymbol{\theta}$. The A- and D-optimal designs are shown in Figure 3.3. With $N = 2001$, we can see that the number of support points is always six for this model. When q is even, the number of support points for D- and A-optimal designs equals to q for small t , and equals to $q + 1$ for larger t . This is consistent with Theorem 2.

Table 3.2: A- and c-optimal design for Gompertz growth model with $\boldsymbol{\theta}_0 = (1, 1, 1)^T$, $\boldsymbol{c}_1 = (2, 0.5, 1)^T$, $N = 2001$ and various values of t .

t		Support points (weights)	ϕ	d^{\max}
0.0	c-	0.000(0.444), 1.615(0.135), 1.620(0.421)	47.025	3.0e-4
	A-	0.000(0.354), 1.315(0.118), 1.320(0.267), 10.000 (0.261)	92.832	2.9e-6
0.3	c-	0.000(0.444), 1.615(0.135), 1.620(0.421)	47.025	3.0e-4
	A-	0.000(0.354), 1.320(0.385), 10.000(0.261)	94.606	5.4e-5
0.5	c-	0.000(0.444), 1.615(0.135), 1.620(0.421)	47.358	4.8e-6
	A-	0.000(0.354), 1.320(0.385), 10.000(0.261)	96.972	7.3e-7
0.7	c-	0.000(0.444), 1.615(0.135), 1.620(0.421)	48.133	9.2e-4
	A-	0.000(0.354), 1.320(0.385) 10.000(0.261)	102.493	1.8e-5
0.9	c-	0.000(0.444), 1.615(0.135), 1.620(0.421)	52.010	2.1e-4
	A-	0.000(0.353), 1.33(0.3847), 10.000(0.262)	130.087	6.3e-7

□

Example 6. One fractional polynomial model is given in (2.16). Here, we compute A- and D-optimal designs for this model, and Figure 3.4 shows representative results. When $N = 1001$, the number of support points for both A- and D-optimal designs is equal to 3 when t is small, and is equal to 4 when t is large. This is consistent with the results in Section 2.3. □

Example 7. We now turn our attention to Michealis-Menton model (2.17). Let the parameters $\theta_1 = \theta_2 = 1$, the design space $S = [0, 4]$, and $\boldsymbol{c}_1 = (1, 1)^T$. The results for both A- and D-optimal designs are given in Yin and Zhou (2017) while c-optimal designs are not studied before. In Table 3.3, the designs under A-optimality match those in Yin and Zhou (2017). For small t , there are two support points for both c- and A-optimal designs whereas for large t , there are three support points as the boundary point zero becomes a support point. This is true for D-optimal designs as well and

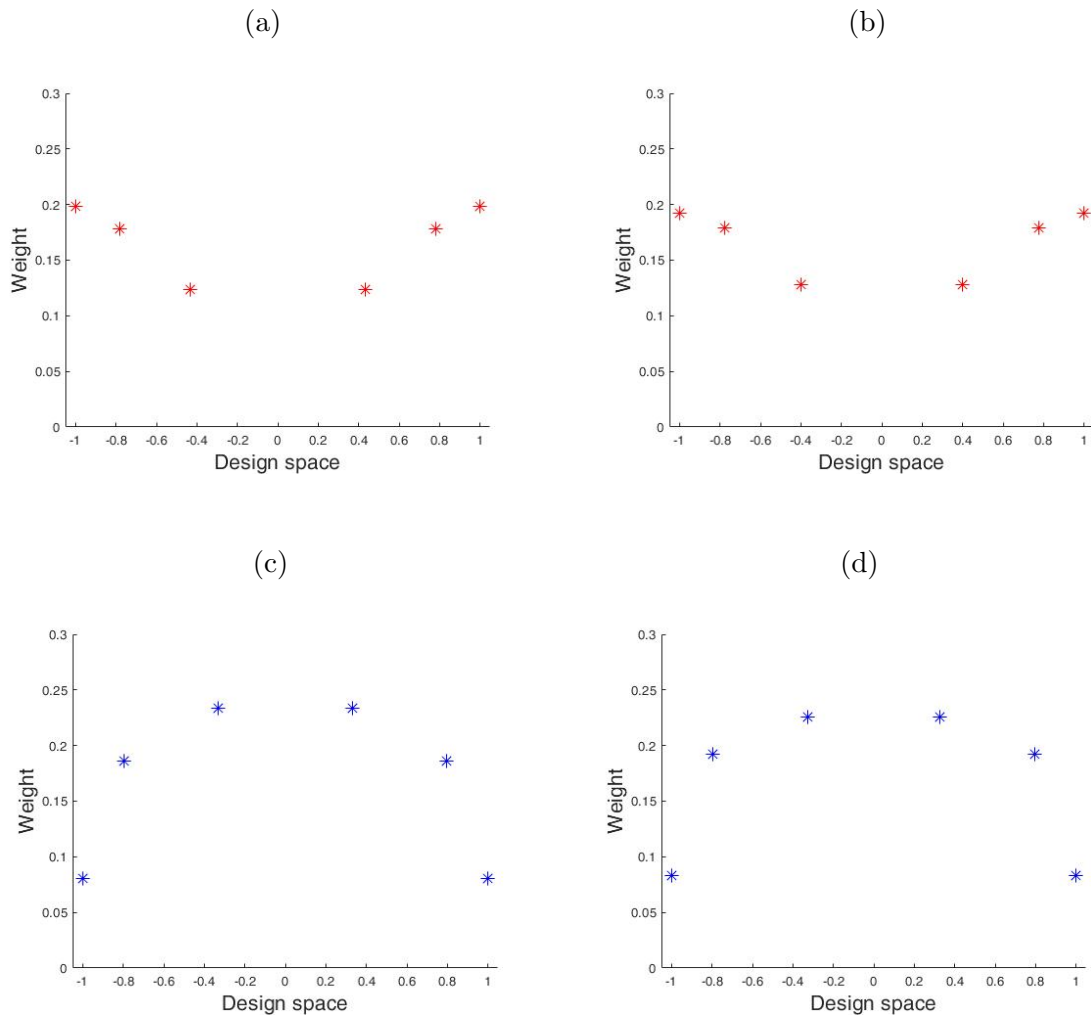


Figure 3.3: Plots of D- and A-optimal designs for polynomial model of order 5 without intercept under the SLSE with different values of t and $N = 2001$. (a) D-optimal design with $t = 0.00$, (b) D-optimal design with $t = 0.70$, (c) A-optimal design with $t = 0.00$, (d) A-optimal design with $t = 0.70$.

the results are omitted. The result on the number of support points is consistent with Lemma 5 in Section 2.3. \square

Example 8. We now study the number of support points for model (2.19), the Peleg model. The true parameter vector is $\boldsymbol{\theta}_o = (\theta_1, \theta_2)^T = (0.5, 0.05)^T$. The results for D-, A- and c-optimal designs are presented in Tables 3.4, 3.5 and 3.6, respectively.

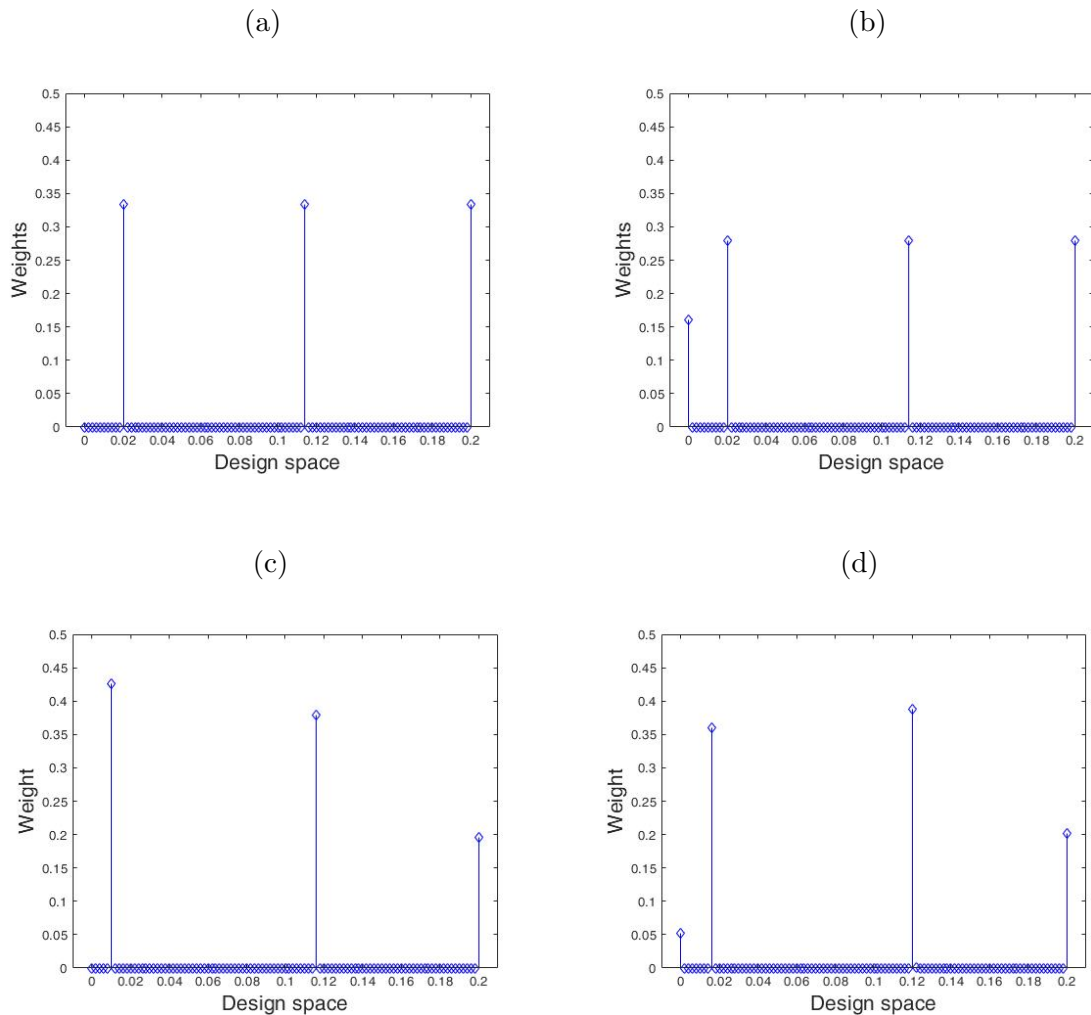


Figure 3.4: Plots of the distribution of the weights for D- and A-optimal designs for fractional polynomial model (2.15) with various values of t and $N = 1001$: (a) D-optimal design with $t = 0.00$, (b) D-optimal design with $t = 0.90$, (c) A-optimal design with $t = 0.00$, (d) A-optimal design with $t = 0.90$.

We can see from those tables that D-, A- and c-optimal designs have either 2 or 3 support points which is consistent with the discussions in the previous chapter. Note that when $t = 0$, the design under the OLSE and SLSE are the same, and we obtain the same result as in Paquet-Durand et al. (2015) for the D-optimal design when $t = 0$. Moreover, the computation times under all the three criteria are around 2-3 seconds which is really fast.

Table 3.3: A- and c-optimal design for Michaelis-Menton model with $S = [0, 4]$, $\mathbf{c}_1 = (1, 1)^T$ and $N = 1001$.

t		Support points (weights)	ϕ	d^{\max}
0.0	A-	0.504 (0.670), 4.000 (0.330)	95.550	1.4166e-5
	c-	0.496 (0.634), 4.000 (0.366)	148.311	1.1725e-5
0.3	A-	0.536 (0.662), 4.000 (0.338)	101.391	2.9874e-6
	c-	0.520 (0.629), 4.000 (0.371)	152.718	7.2097e-6
0.5	A-	0.568 (0.655), 4.000 (0.345)	108.611	5.8928e-5
	c-	0.544 (0.623), 4.000 (0.377)	158.098	7.1312e-5
0.7	A-	0.632 (0.642), 4.000 (0.358)	123.810	6.2421e-6
	c-	0.596 (0.613), 4.000 (0.388)	169.144	2.0653e-6
0.9	A-	0.000 (0.158), 0.664 (0.536), 4.000 (0.306)	156.933	1.0962e-5
	c-	0.000 (0.074), 0.668 (0.556), 4.000 (0.371)	202.501	4.4967e-5

Table 3.4: Loss functions and D-optimal designs for Peleg model with $\boldsymbol{\theta}_o = (0.5, 0.05)^T$, $S = [0, 180]$ and $N = 1001$ for $t = 0.0, 0.3, 0.5, 0.7$ and 0.9 .

t	Support points (weights)	$\log(\phi_D)$	d_D^{\max}
0.0	9.000 (0.500), 180.00 (0.500)	-14.8774	8.4365e-7
0.3	9.000 (0.500), 180.00 (0.500)	-14.5207	1.2224e-6
0.5	9.000 (0.500), 180.00 (0.500)	-14.1843	3.7681e-6
0.7	0.000 (0.048), 9.000 (0.476), 180.00 (0.476)	-13.6812	5.2797e-6
0.9	0.000 (0.259), 9.000 (0.370), 180.00 (0.370)	-13.1786	1.6091e-5

□

Example 9. For the spline regression model in (2.22), we now apply our numerical algorithm to confirm the GSI result. First of all, we compute the D-optimal design on $S = [0, 10]$ with $\lambda = 8$. Using CVX with $N = 1001$, we have the results in Table 3.7 that matrix $\mathbf{B}(\xi, \boldsymbol{\theta}_o, t)$ is ill-conditioned. Hence, the D-optimal design $\xi_D^*(S, \boldsymbol{\theta}_o)$

Table 3.5: Loss functions and A-optimal designs for Peleg model with $\boldsymbol{\theta}_o = (0.5, 0.05)^T$, $S = [0, 180]$ and $N = 1001$ for $t = 0.0, 0.3, 0.5, 0.7$ and 0.9 .

t	Support points (weights)	ϕ_A	d_A^{\max}
0.0	6.48 (0.852), 180.00 (0.148)	0.0163	9.4385e-08
0.3	7.38 (0.831), 180.00 (0.169)	0.0196	1.2534e-08
0.5	8.10 (0.815), 180.00 (0.184)	0.0236	1.0637e-08
0.7	0.00 (0.107), 9.00 (0.714), 180.00 (0.180)	0.0313	7.7097e-08
0.9	0.00 (0.305), 9.00 (0.555), 180.00 (0.138)	0.0402	2.9918e-07

Table 3.6: Loss functions and c-optimal designs for Peleg model with $\boldsymbol{\theta}_o = (0.5, 0.05)^T$, $S = [0, 180]$, $\mathbf{c}_1 = (1, 1)^T$ and $N = 1001$ for $t = 0.0, 0.3, 0.5, 0.7$ and 0.9 .

t	Support points (weights)	ϕ_c	d_c^{\max}
0.0	6.480 (0.872), 180.000 (0.128)	0.0154	1.7182e-8
0.3	7.380 (0.8513), 180.000 (0.1487)	0.0188	1.2665e-7
0.5	8.280 (0.832), 180.00 (0.168)	0.0230	1.8582e-7
0.7	0.000 (0.126), 9.000 (0.714), 180.000 (0.160)	0.0309	3.8118e-7
0.9	0.000 (0.320), 9.000 (0.556), 180.000 (0.124)	0.0397	5.6527e-8

cannot be obtained precisely using the numerical algorithm. However, we may find the optimal design on the scaled design space $S^V = [0, 1]$ with $\lambda = 0.8$ and obtain

$$\xi_D^*(S^V, \tilde{\boldsymbol{\theta}}_o) = \begin{bmatrix} 0.000 & 0.2250 & 0.5900 & 0.8200 & 0.9350 & 1.0000 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}.$$

This result is consistent with that in Dette et al. (2008). Then we may use the GSI property to obtain the D-optimal design on S which gives

$$\xi_D^*(S, \boldsymbol{\theta}_o) = \begin{bmatrix} 0.000 & 2.2500 & 5.9000 & 8.2000 & 9.3500 & 10.0000 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}.$$

□

Table 3.7: D-optimal designs for the spline regression models in (2.22) on both scaled design space $S^V = [0, 1]$ and the original design space $S = [0, 10]$ with $N = 1001$.

$S = [0, 10], \lambda = 8$		$S^V = [0, 1], \lambda = 0.8$	
Support points	Weights	Support points	Weights
0.000	0.1667	0.000	0.1664
2.230	0.0002	0.2250	0.1664
2.240	0.0012	0.5900	0.1665
2.250	0.1413	0.8200	0.1669
2.260	0.0220	0.9350	0.1673
5.920	0.0003	1.0000	0.1666
5.930	0.0001		
8.200	0.1635		
8.210	0.0030		
9.340	0.0002		
9.350	0.1663		
10.000	0.1667		
$\log(\phi_D)$	-11.6064	Loss function	39.0607
Solved?	Inaccurate/Solved	Solved?	Solved

Example 10. The algorithm in Section 3.2 can be also applied to higher dimensional design space. Here we consider a mixture regression model, which studies the yield (y) of an experiment by mixing three solutions x_1 , x_2 and x_3 . The linear model is given by

$$y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_3^2 + \theta_7 x_1 x_2 + \theta_8 x_1 x_3 + \epsilon,$$

where $(x_1, x_2, x_3) \in S = \{(x_1, x_2, x_3) | x_1 + x_2 + x_3 \leq 1, x_i \geq 0, i = 1, 2, 3\}$. It is clear that $S \subset \mathbb{R}^3$. The model does not contain the interaction term $x_2 x_3$ so new designs are obtained. We discretize the design space S by three dimensional grid points, say N_i points in x_i , $i = 1, 2, 3$. We use $N_1 = N_2 = N_3 = 51$ points for each $x_i \in [0, 1]$.

Notice that not all combinations of x_1 , x_2 and x_3 are in S . The total number of point is $N = 23424$ points in S . With a small t , the 9 support points are given in Table 3.8 whereas for large t , the design is slightly different as the first and third design points change to $(0.00, 0.00, 0.48)$ and $(0.00, 0.48, 0.00)$, and the corresponding weights remain the same. We also compute the D -optimal design for $N_1 = N_2 = N_3 = 21$, and the computation time is less than 10 seconds. There are only 1771 design points in S_N , and the support points are the exactly the same as in Table 3.8 for $0 \leq t < 1$. For $N_1 = N_2 = N_3 = 51$, the computation time is around 8 minutes. \square

Table 3.8: D -optimal design for mixture model with 23424 grid points for $t = 0.0$ and 0.7 .

Support points (x_1, x_2, x_3)	Weights	
	$t = 0.0$	$t = 0.7$
$(0.00, 0.00, 0.50)$	0.083	0.085
$(0.00, 0.00, 1.00)$	0.125	0.124
$(0.00, 0.50, 0.00)$	0.083	0.085
$(0.00, 0.50, 0.50)$	0.083	0.085
$(1.00, 0.00, 0.00)$	0.125	0.124
$(0.50, 0.00, 0.00)$	0.125	0.124
$(0.50, 0.50, 0.00)$	0.125	0.124
$(0.50, 0.00, 0.50)$	0.125	0.124
$(1.00, 0.00, 0.00)$	0.125	0.124
$\log(\phi_D)$	30.211	31.350
d_D^{\max}	4.2927e-7	1.256e-6

Example 11. We may wonder how the computational time changes in terms of the number of design points in S_N . We use one example to show the computation time. Emax model is a commonly used model in microhydrodynamical studies that is

reviewed in Derendorf and Meibohm (1999) and also discussed in Berger and Wong (2009) for optimal designs. The model is given by

$$y_i = \frac{\theta_1 x_i^{\theta_3}}{\theta_2 + x_i^{\theta_3}} + \epsilon_i, \quad i = 1, 2, \dots, n,$$

where y_i is the effects at drug concentration x_i , θ_1 is the maximum effect, $\theta_2 = C_{50}^h$ is the concentration of the drug that produces half of the maximal effect and θ_3 is a shape factor. We use the true parameter vector $\boldsymbol{\theta}_o = (1, 2, 1)^T$ and design space is $S = [0, 10]$ from Berger and Wong (2009).

In Figure 3.5, we can see that the computational time of D-optimal design increases exponentially as N increases. A- and c-optimal designs behave similarly as well. We also consider the changes of computation time in terms of the value of t . In Figure 3.6, we can clearly see that the computation time is approximately the same for $0 \leq t < 1$. The computation time for c-optimal designs behaves similarly as t changes. Therefore, we are confident to conclude that the value of t , a measure related to skewness, does not affect the computation time whereas the number of grid points has a huge effect towards the computation time.

□

3.4 Remarks

As we have shown in Section 3.3, the computation time for finding optimal designs increases as N increases. In practice, how to choose N becomes an interesting and important question to us. We know that as $N \rightarrow +\infty$, the optimal designs on S_N converge to those on S . From our numerical results, N does not have to be very large to achieve highly efficient designs.

We use previous Examples to illustrate D-efficiency for $N_1 = 51$ and $N_2 = 20001$. The D-efficiency is defined by

$$D_{eff}(\xi_D^*(N_1), \xi_D^*(N_2)) = \left(\frac{\phi_D(\xi_D^*(N_1))}{\phi_D(\xi_D^*(N_2))} \right)^{\frac{1}{q}},$$

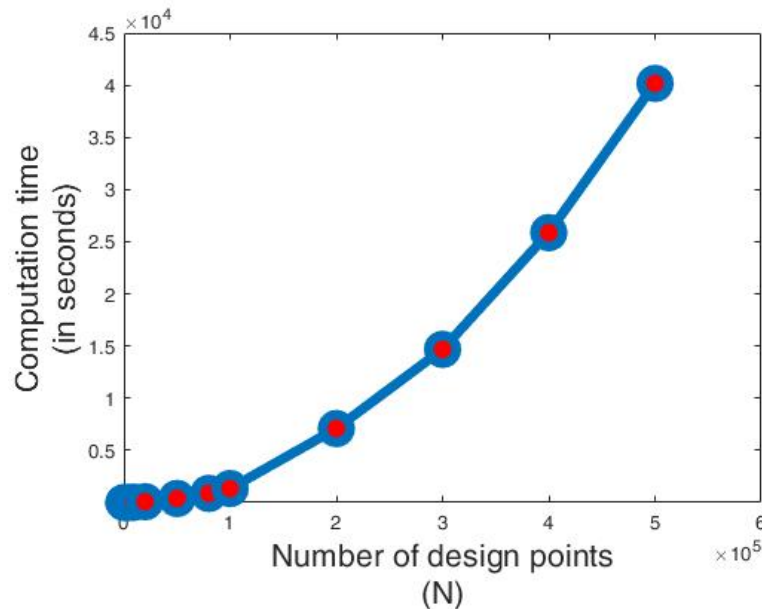


Figure 3.5: Plot of computation time versus the number of design points (N) for finding D-optimal design for Emax model with $\boldsymbol{\theta}_o = (1, 2, 1)^T$.

where q is the dimension of the parameter vector $\boldsymbol{\theta}_o$. The optimal designs $\xi_D^*(N_1)$ and $\xi_D^*(N_2)$ are D-optimal designs on S_{N_1} and S_{N_2} . In fact, they share the same value of parameter vector $\boldsymbol{\theta}_o$, design space S and regression model. The results are given in Table 3.9. We can clearly see that with very small N , the D-efficiencies are really high for Gompertz growth model, Peleg model and Michaelis-Menton model. Therefore, we can conclude that in practice, we can obtain good designs with small to moderate N . This is consistent with the results in Wong and Zhou (2018). We should also note that the D-optimal designs for all three models include the boundary points. Therefore, with a very small N , the designs are highly efficient already. For the designs that do not include the boundary point, moderate N may be needed.

For nonlinear models, optimal designs often depend on the true value of parameter vector $\boldsymbol{\theta}$. For instance, Peleg model (2.19) with parameters $\boldsymbol{\theta}_o = (\theta_1, \theta_2)^T = (1, 1)^T$, $N = 1001$ and $t = 0$, the D-optimal design on the same design space is

$$\xi_D^* = \begin{bmatrix} 0.9000 & 1.0800 & 180.0000 \\ 0.1811 & 0.3192 & 0.4998 \end{bmatrix}.$$

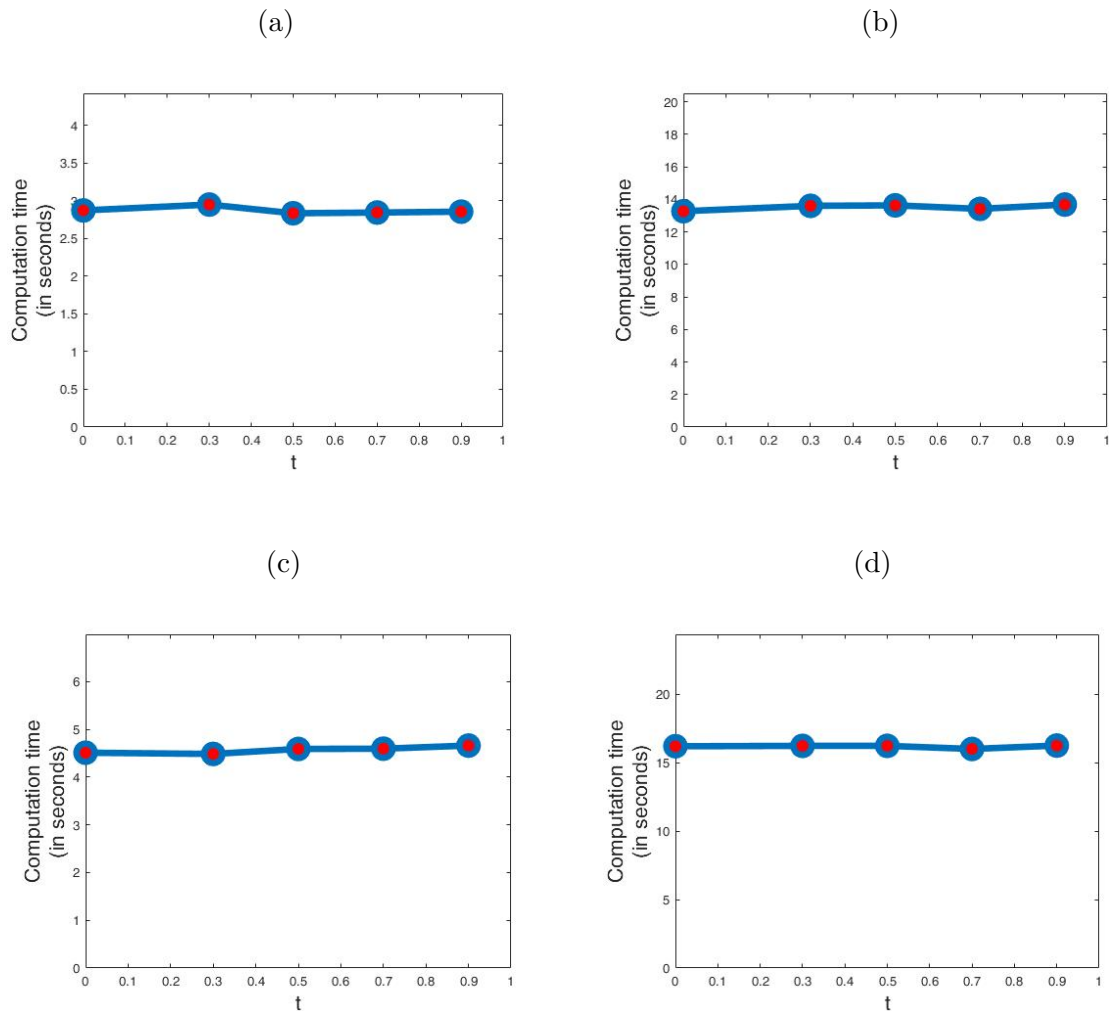


Figure 3.6: Plots of the change of computation time versus the level of asymmetry of finding designs for Emax model with $\theta = (1, 2, 1)^T$. (a) A-optimal design with $N = 1001$, (b) A-optimal design with $N = 5001$, (c) D-optimal design with $N = 1001$, (d) D-optimal design with $N = 5001$.

We can clearly see that this is different from D-optimal designs in (3.4) with parameter vector $\theta_o = (0.5, 0.05)^T$ and $t = 0$. It demonstrates that the parameter values can have inferences to the resulting designs. In practice, the values of the parameters are usually chosen based on the pilot studies, or educational guesses from experienced investigators.

Table 3.9: D-efficiency for Gompertz growth model, Peleg model and Michaelis-Menton model.

Model	t	ϕ_D on S_{N1}	ϕ_D on S_{N2}	D_{eff}
Gompertz	0.0	2.7471e+03	2.7413e+03	0.9993
	0.5	5.4939e+03	5.4830e+03	0.9993
Peleg	0.0	3.4580e-07	3.4580e-07	1.0000
	0.5	6.9157e-07	6.9157e-07	1.0000
Michaelis-Menton	0.0	246.0661	244.1298	0.9961
	0.5	492.1089	488.2854	0.9961

Chapter 4

Discussions

We have reviewed the current development of optimal regression designs under the SLSE and derived more theoretical properties for the optimal designs. We can characterize the loss function for the A-optimal design in convex optimization framework differently, so that the result can be extended to c-optimality criterion under the SLSE. We have also derived the optimality conditions based on the equivalence results for A-, D- and c-optimality criteria. Furthermore We have studied and obtained the number of support points for several regression models under the SLSE analytically.

For nonlinear models, the optimal designs often depend on the value of parameter vector θ_o , and it is difficult to solve the optimal regression design problems analytically. We have investigated a CVX based numerical algorithm to compute for the optimal designs using the properties we obtained. Using the approximate design approach by discretizing the design space into a user-specified number of grid points, we can find optimal designs efficiently and effectively with the CVX based algorithm. Several applications are given in the thesis for showcasing the effectiveness of the algorithm.

Generalized scale invariance property for D-optimality is a very useful property that is also studied in this thesis. It can be applied to both linear and nonlinear regression models. This property can be helpful in many situations. For instance,

if matrix \mathbf{B} is ill-conditioned, our numerical algorithm may fail. We may be able to use this property to rescale the design space and find the D-optimal design on the scaled design space. This scaling is helpful to avoid computing issues related to ill-conditioned matrix.

Although our algorithm is effective and efficient for the models presented in this thesis, there are still challenges to construct optimal designs for complicated models. We discuss a few of them here.

When the dimension of the design space S is high (i.e. having many design variables), for instance, 10 x 's in the mixture experiment, the algorithm may be unstable. Sometimes, since we keep the point when its weight is more than 10^{-4} by using the numerical algorithm, there can be many points having very small positive weights. Such points may not be significant in the optimal designs. Also, it is difficult to partition the high dimensional space into tiny grid points, as the computation time increases exponentially as N increases.

When matrix \mathbf{B} is ill-conditioned, the numerical algorithm will fail as it cannot find optimal designs precisely. For D-optimal designs, we have discussed that for some models, we may use the generalized scale invariant properties to rescale the ill-conditioned matrix and obtain the optimal designs. However, not all the models under D-optimality have such property. Furthermore, other design criteria, such as A- and c-, generally do not have such property. Therefore, further investigations are required to address this issue in the future.

Although we have extended the theoretical properties under the SLSE to c-optimality criterion which has not been studied before, there are still many other design criteria under the SLSE, such as E-, G- and minimax criteria that have not been studied.

Appendix A

Guidance of using CVX

CVX is a powerful tool which is MATLAB-based modeling system for convex optimization. It is developed to solve convex optimization problems (Boyd and Vandenberghe, 2004). The installation guide can be found at <http://cvxr.com/cvx/> and the user guide is at <http://web.cvxr.com/cvx/doc/>.

In the thesis, we have presented one computing algorithm in Section 3.2. Here we give more detailed MATLAB algorithms for computing various optimal designs under the SLSE. Algorithms 1 and 2 show step by step in the flow charts for D- and A-optimal designs. The only difference between them is the loss function ϕ which was described in (2.7). The algorithm to compute c-optimal is very similar and is omitted here.

D-optimality

Input: $S, N_1, N_2, \dots, N_p, \boldsymbol{\theta}_o, t$

Result: The optimal design ξ_D^*

initialization;

$N \leftarrow N_1 * N_2 * \dots * N_p$

divide S into the N grid points

CVX begins

Minimize δ

for i *in* $1:N$ **do**

$f \leftarrow \frac{\partial g(x_i; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$;

$\mathbf{g}_1 \leftarrow \mathbf{g}_1 + w(i) * f$;

$\mathbf{G}_2 \leftarrow \mathbf{G}_2 + w(i) * f * f^T$;

end

$\mathbf{B} \leftarrow \begin{pmatrix} 1 & \sqrt{t} \mathbf{g}_1^T \\ \sqrt{t} \mathbf{g}_1 & \mathbf{G}_2 \end{pmatrix}$;

$-\log \det(\mathbf{B}) \leq \delta$;

for i *in* $1:N$ **do**

$-w(i) \leq 0$;

end

$\sum(w) = 1$

CVX ends

Algorithm 1: Algorithm for computing D-optimal design

A-optimality

Input: $S, N_1, N_2, \dots, N_p, \boldsymbol{\theta}_o, t$

Result: The optimal design ξ_A^*

initialization;

$$\mathbf{C} = \mathbf{1} \oplus \mathbf{I}_q$$

$$N \leftarrow N_1 * N_2 * \dots * N_p$$

divide S into the N grid points

CVX begins

Minimize δ

for i *in* $1:N$ **do**

$$\mathbf{f} \leftarrow \frac{\partial g(x_i; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}};$$

$$\mathbf{g}_1 \leftarrow \mathbf{g}_1 + w(i) * \mathbf{f};$$

$$\mathbf{G}_2 \leftarrow \mathbf{G}_2 + w(i) * \mathbf{f} * \mathbf{f}^T;$$

end

$$\mathbf{B} \leftarrow \begin{pmatrix} 1 & \sqrt{t} \mathbf{g}_1^T \\ \sqrt{t} \mathbf{g}_1 & \mathbf{G}_2 \end{pmatrix};$$

$$\text{tr}(\mathbf{C}^T \mathbf{B}^{-1} \mathbf{C}) \leq \delta;$$

for i *in* $1:N$ **do**

$$| \quad -w(i) \leq 0;$$

end

$$\sum(w) = 1$$

CVX ends

Algorithm 2: Algorithm for computing A-optimal design

Appendix B

MATLAB Code

Here we provide the MATLAB code used to compute the optimal designs in the thesis. Users have to provide the arguments including the number of points (N), level of skewness (t), parameter vector (theta), design space (range), and partial derivative (fun) in the main function. For each different model, the function for the partial derivative is different, so it should be coded accordingly. Three main functions for A-, c- and D-optimality are coded, respectively. The detailed outputs and required arguments for the main functions are given in [B.1](#), while [B.2](#) provides the main functions and [B.3](#) - [B.8](#) present specific partial derivative functions for various models in the examples in [Chapter 3](#).

B.1 Main function description

1. Outputs

- del - the value of the loss function ϕ_A (ϕ_D, ϕ_c) under A- (D-, c-) optimality.
- ANS - supporting points and their weights for the optimal designs
- error - maximum value (d^{\max}) of the functions d_A, d_D and d_c described ([2.10a](#)), ([2.10b](#)) and ([2.10c](#)) with respect to D-, A- and c-optimality, respectively.

2. Arguments

- N - number of points in S_N
- t - level of skewness
- θ - parameter values
- range - design space
- fun - function that calculates the partial derivative of the selected model.

B.2 Main functions

D-optimal design, main function

```

1 function [del , ANS, error] = D_opt(N,t,theta ,range ,fun)
2     %% initialization
3     % discretized equally spaced space
4     u = range(1)+(range(2)-range(1))*((1:N)-1)/(N-1);
5     w = zeros(N,1); n = length(theta); del = 0 ;
6     one_vec = ones(N,1); zero_vec = zeros(N,1);
7     g1 = zeros(n,1); G2 = zeros(n);
8     sqrt = sqrt(t);
9
10    %% cvx part
11    cvx_begin quiet
12        cvx_precision best
13        variables w(N,1) del(1)
14        minimize del(1)
15        subject to
16        % constructing the B matrix
17        for i = 1:N
18            f = fun(u(i),theta);

```

```

19     g1 = g1 + w(i)*f;
20     G2 = G2 + w(i)*f*f';
21     end
22     B = [1 sqrt*g1' ; sqrt*g1 G2];
23
24     %% constrains
25     -log_det(B) <= del;
26     -w <= zero_vec;
27     one_vec' * w == 1;
28 cvx_end
29
30 %% manage the outputs
31 kk = find(w>1e-4); %
32 ANS = [u(kk);w(kk)'];
33
34 %% prepare the variables
35 phi_D = zeros(N,1);
36 for i = 1:N
37     f = fun(u(i),theta);
38     I = [1 sqrt*f' ; sqrt*f f*f'];
39     phi_D(i) = trace(B\I);
40 end
41 q = (n+1)*ones(N,1);
42
43 error = max(phi_D - q) ;
44
45 %% plotting
46 new_range = [0;0]; add_dist = (range(2)-range(1)) /20;
47 new_range(1) = range(1) - add_dist;
48 new_range(2) = range(2) + add_dist;
49

```

```

50 % first plot, plot out the design points
51 figure
52     stem( u,w, 'kd' );
53     xlim(new_range);
54     ylim([0,1]);
55     xlabel('design space','FontSize', 16) % x-axis label
56     ylabel('weights','FontSize', 16) % y-axis label
57     title('Discretized weight distribution','FontSize', 20)
58
59 % directional derivative plot
60 fx = @(x) fun(x,theta);
61 ff = @(x) trace(B\[1 sqrt.*fx(x)' ; sqrt.*fx(x) fx(x)*fx(x)'] )
        - (n+1);
62
63 mini = min(phi_D-q);
64 figure
65     h1 = plot(u,phi_D-q, '+');
66     xlim(new_range);
67     ylim([mini+mini/10,1]);
68 hold on
69     y = zeros(size(u));
70     for i = 1:length(u)
71         y(i) = ff(u(i));
72     end
73     h2 = plot(u,y, '-');
74 hold on
75     line(new_range,[0,0], 'Color','blue','LineStyle','—');
76 hold on
77     h3 = plot(u(kk),zeros(1,length(kk)), 'pg');
78     legend([h1 h2 h3], 'Discretized','d(x,\theta)', 'Supporting
        point');

```

```

79     xlabel('design space','FontSize',16)
80     ylabel('Directional Derivative','FontSize',16)
81     hold off
82 end

```

A-optimal design, main function

```

1 function [del , ANS, error] = A_opt(N,t,theta,range,fun)
2     %% initialization
3     % discretized equally spaced space
4     u = range(1) + (range(2)-range(1))*((1:N)-1)/(N-1);
5     w = zeros(N,1); n = length(theta); del = 0;
6     C = [zeros(n,1) eye(n)]';
7     g1 = zeros(n,1) ; G2 = zeros(n) ; obj_val = 0;
8     one_vec = ones(N,1); zero_vec = zeros(N,1);
9     sqrt = sqrt(t);
10    %% cvx part
11    cvx_begin quiet
12        cvx_precision best
13        variables w(N,1) del(1)
14        minimize del(1)
15        subject to
16            % constructing the B matrix
17            for i = 1:N
18                f = fun(u(i),theta);
19                g1 = g1 + w(i)*f;
20                G2 = G2 + w(i)*f*f';
21            end
22            B = [1, sqrt*g1'; sqrt*g1, G2];
23            % constraints
24            for k = 1:n

```

```

25         obj_val = obj_val + matrix_frac(C(:,k),B) ;
26     end
27     obj_val <= del;
28     -w <= zero_vec;
29     one_vec' * w == 1;
30 cvx_end
31
32 %% manage the outputs
33 kk = find(w > 1e-4); % because the computer does not have exact
    zero
34 ANS = [u(kk) ; w(kk)'];
35
36 %% checking condition
37 % prepare the variables
38 BI = inv(B);
39 phi_A = zeros(N,1);
40 C = blkdiag( 0,eye(n));
41 for i = 1:
42     f = fun(u(i),theta);
43     I = [1 sqrt*f' ; sqrt*f f*f'];
44     phi_A(i) = trace(I*BI*C'*C*BI) ;
45 end
46
47 trac = trace(C*BI*C');
48 error = max(phi_A - trac );
49 %plotting
50 new_range = [0;0]; add_dist = (range(2)-range(1))/20;
51 new_range(1) = range(1) - add_dist;
52 new_range(2) = range(2) + add_dist;
53
54 % first plot , plot out the design points

```

```

55 figure
56     stem( u,w, 'kd' );
57     xlim(new_range);
58     ylim([0,1]);
59     xlabel('design space','FontSize', 16)
60     ylabel('weight','FontSize', 16)
61     title('Discretized weight distribution','FontSize', 20)
62
63 %directional derivative plot
64 fx = @(x) fun(x,theta);
65 ff = @(x) trace([1 sqrt*fx(x)' ; sqrt*fx(x)  fx(x)*fx(x)'] *BI*C
66             '*C*BI) - trac;
67
68 mini = min(phi_A - trac*ones(N,1));
69 figure
70     h1 = plot(u,phi_A-trac*ones(N,1), '+');
71     xlim(new_range);
72     ylim([mini+mini/10,1]);
73 hold on
74     y = zeros(size(u));
75     for i = 1:length(u)
76         y(i) = ff(u(i));
77     end
78     h2 = plot(u,y, '-');
79 hold on
80     line(new_range,[0,0], 'Color','blue','LineStyle','--');
81 hold on
82     h3 = plot(u(kk),zeros(1,length(kk)), 'pg');
83     legend([h1 h2 h3], 'Discretized','d(x,\theta)','Supporting
84             point');
85     xlabel('design space','FontSize', 16);

```

```

84     ylabel('Directional Derivative','FontSize',16);
85     hold off
86 end

```

c-optimal design, main function

```

1 function [del , ANS, error] = c_opt(N,t,theta ,range ,fun ,c)
2 %% initialization
3 u = range(1) + (range(2)-range(1))*((1:N)-1)/(N-1); %%
4     discretized equally spaced space
5 w = zeros(N,1); del = 0; n = length(theta);
6 g1 = zeros(n,1); G2 = zeros(n); C=[0,c];
7 one_vec = ones(N,1); zero_vec = zeros(N,1);
8 sqrt = sqrt(t);
9 %% cvx part
10 cvx_begin
11     cvx_precision best
12     variables w(N,1) del(1)
13     minimize del(1)
14     subject to
15         %% constructing the B matrix
16         for i = 1:N
17             f = fun(u(i),theta);
18             g1 = g1+ w(i)*f;
19             G2 = G2 + w(i)*f*f';
20         end
21     B = [1, sqrt*g1'; sqrt*g1, G2];
22     %% the three constrains
23     matrix_frac(C',B) <= del;
24     -w <= zero_vec;
25     one_vec' * w == 1;

```

```

25     cvx_end
26
27     %% checking condition
28     BI = inv(B);
29     phi_C = zeros(N,1);
30     for i = 1:N
31         f = fun(u(i), theta);
32         I = [1, sqrt*f'; sqrt*f, f*f'];
33         phi_C(i) = trace(I*BI*C'*C*BI);
34     end
35
36     term = C*BI*C';
37     error = max(phi_C - term );
38
39     %% manage the outputs
40     kk = find(w>1e-4);
41     ANS = [u(kk);w(kk)'];
42
43     %% plotting
44     new_range = [0;0]; add_dist = (range(2)-range(1))/20;
45     new_range(1) = range(1) - add_dist;
46     new_range(2) = range(2) + add_dist;
47
48     % first plot, plot out the design points
49     figure
50         stem( u,w, 'kd' )
51         xlim(new_range);
52         xlabel('design space', 'FontSize', 16)
53         ylabel('weight', 'FontSize', 16)
54         title('Discretized weight distribution', 'FontSize', 20)
55

```

```

56 % directional derivative plot
57 fx = @(x) fun(x,theta);
58 ff = @(x) trace([1 sqrt*fx(x)' ; sqrt*fx(x) fx(x)*fx(x)'] *BI*C
    '*C*...
59 BI)-term;
60
61 figure
62 h1=plot(u,phi_C-term*ones(N,1),'+'); %discretized
63 hold on
64 h2=fplot(ff,[range(1) range(2)],'-'); %function
65 hold on
66 line([range(1),range(2)],[0,0],'Color','red');
67 hold on
68 h3 = plot(u(kk),zeros(1,length(kk)),'pg'); %supporting points
69 legend([h1 h2 h3], 'Discretized','d(x,\theta)','Supporting
    point');
70 xlabel('design space','FontSize',16); % x-axis label
71 ylabel('Directional Derivative','FontSize',16); % y-axis
    label
72 hold off
73 end

```

B.3 Gompertz Growth model (Example 4)

The main command:

```
1 [d,a,e]=D_opt(10001,0,[1,1,1]',[0;10],@gompertz)
```

The function for partial derivative:

```

1 function ANS = gompertz(xi, theta)
2   po = exp(-theta(2)*exp(-theta(3)*xi)-theta(3)*xi);
3   ANS = [exp(-theta(2)*exp(-theta(3)*xi));...
4         -theta(1)*po;...
5         theta(1)*theta(2)*xi*po];
6 end

```

B.4 Fractional polynomial (Example 6)

The main command:

```

1 [d, a, e]=D_opt(10001, 0, [1, 2, 3]', [0;5], @frac)

```

The function for partial derivative:

```

1 function ANS =frac_poly_ex(xi, theta)
2   ANS = [xi, sqrt(xi), xi^2]';
3 end

```

B.5 Michaelis-menton model example (Example 7)

The main command:

```

1 [d, a, e]=D_opt(1001, 0, [0.05, 0.5]', [0;4], @mm)

```

The function for partial derivative:

```

1 function ANS = mm(xi, theta)
2   ANS = [xi/(theta(2)+xi); -(theta(1)*xi)/(theta(2)+xi).^2];
3 end

```

B.6 Peleg model (Example 8)

D-optimal design for Peleg model with 101 design points ($N = 101$), $t = 0$, $S = [0, 180]$. $\theta_o = (0.05, 0.5)^T$:

The main command:

```
1 [d,a,e]=D_opt(101,0,[0.05,0.5]',[0;180],@peleg)
```

The function for partial derivative:

```
1 function ANS = peleg(xi,theta)
2     deno = (theta(1)+xi*theta(2))^2;
3     ANS = [-xi/deno ; -xi^2/deno];
4 end
```

B.7 Spline model (Example 9)

```
1 [d,a,e]=D_opt(1001,0,[1,1,1,1,1,1,1,8]',[1;10],@spline)
```

The function for partial derivative:

```
1 function ANS = spline(xi,theta)
2     knot = max(0,(xi-theta(6)));
3     ANS = [1,xi,xi^2,xi^3,knot^3,...
4           -3*theta(5)*knot^2]';
5 end
```

B.8 Mixture model (Example 10)

D-optimal design for mixture model with 1001 design points ($N = 101$), $t = 0$, $S = [0, 180]$. $\theta_o = (0.05, 0.5)^T$:

The main command:

```
1 [d,a,e]=D_opt_mixture(21,21,21,0,[1,1,1]',[0;1],@mix_3)
```

The main function:

```
1 %% function itself
2 function [del , ANS, error ,pt] = D_opt_mixture_3(N1,N2,N3,t ,theta
    , range , fun)
3 %% initialization
4
5 counter = 0;
6 % these ones are used in solving the question
7 u1 = range(1)+(range(2)-range(1))*((1:N1)-1)/(N1-1); %
    discretized equally spaced space
8 u2 = range(1)+(range(2)-range(1))*((1:N2)-1)/(N2-1); %
    discretized equally spaced space
9 u3 = range(1)+(range(2)-range(1))*((1:N3)-1)/(N3-1); %
    discretized equally spaced space
10 for i = 1:N1
11     for j = 1:N2
12         for k = 1:N3
13             if (u1(i)+u2(j)+u3(k)<=1)
14                 x(counter+1,:) = [u1(i);u2(j);u3(k)];
15                 counter = counter + 1;
16             end
17         end
18     end
19 end
```

```

18     end
19 end
20 pt = length(x);
21 n = 8;%length(theta);
22 w = zeros(length(x),1); del = 0 ; one_vec = ones(length(x),1);
23 zero_vec = zeros(length(x),1);
24 g1 = zeros(n,1); G2 = zeros(n);
25 sqt = sqrt(t);
26
27 %% cvx part
28 cvx_begin quiet
29     cvx_precision best
30     variables w(length(x),1) del(1)%design variable and upper
31         bound
32     minimize del(1)
33     subject to
34     %   constructing the B matrix
35     for i = 1:length(x)
36         f = fun(x(i,:),theta);
37         g1 = g1 + w(i)*f;
38         G2 = G2 + w(i)*f*f';
39     end
40     B = [1 sqt*g1' ; sqt*g1 G2];
41     % constrains
42     -log_det(B) <= del;
43     %-w <= zeros(length(w),1);
44     -w <= zero_vec;
45     one_vec' * w == 1;
46 cvx_end
47 %% propose the outputs

```

```

48   kk = find(w>1e-4); % because the computer does not have exact
      zero
49   ANS = [x(kk,:)';w(kk)']; % return the answer
50
51   % prepare the variables
52   phi_D = zeros(length(x),1);
53   for i = 1:length(x)
54       f = fun(x(i,:),theta);%[x(i,1);x(i,2);x(i,3);x(i,1)^2;x(i,2)
      ^2;x(i,3)^2;x(i,1)*x(i,2);x(i,1)*x(i,3)];
55       I = [1 sqrt*f' ; sqrt*f f*f'];
56       %phi_D(i) = trace( BI*I) ;
57       phi_D(i) = trace(B\I);
58   end
59   q = (n+1)*ones(length(x),1);
60
61   % update the error
62   error = max(phi_D - q) ;
63 end

```

The function:

```

1 function ANS = mix_3(xi,theta)
2   ANS = [xi(1);xi(2);xi(3);xi(1)^2;xi(2)^2;xi(3)^2;xi(1)*xi(2);xi
      (1)*xi(3)];
3 end

```

B.9 Emax model (Example 11)

```

1 [d,a,e]=D_opt(1001,0,[1,2,1]',[1;10],@emax)

```

The function for partial derivative:

```
1 function ANS = emax(xi , theta )
2     deno = theta (2)+xi ^ theta (3);
3     x_pow = xi ^ theta (3);
4     par_1 = x_pow/deno;
5     par_2 = -theta (1)*x_pow/deno ^ 2;
6     par_3 = theta (1)*theta (2)*x_pow*log (xi )/deno ^ 2;
7     ANS = [ par_1 ; par_2 ; par_3 ];
8 end
```

Bibliography

- [1] Atkinson, A. C. and Donev, A. N. (1992). *Optimum Experimental Design*. Clarendon Press, Oxford.
- [2] Berger, M.P.F. and Wong, W.K. (2009). *An Introduction to Optimal Designs for Social and Biomedical Research*. Wiley, Chichester, UK.
- [3] Brosi, B.J., Armsworth, P.R., and Daily, G.C. (2008). Optimal design of agricultural landscapes for pollination services. *Conservation Letters*, 1, 27-36.
- [4] Bose, M. and Mukerjee, R. (2015). Optimal design measures under asymmetric errors, with application to binary design points. *Journal of Statistical Planning and Inference*, 159, 28-36.
- [5] Box, G.E.P. and Draper N.R. (1987). *Empirical Model-Building and Response Surfaces*. Wiley, New York.
- [6] Box, G.E.P., Hunter, W.G. and Hunter, J.S. (1978). *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*. Wiley, New York
- [7] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York.
- [8] Chernoff, H. (1953). Locally optimal designs for estimating parameters. *The Annals of Mathematical Statistics*, 24, 586-602.

- [9] Crary, S.B. (2002). Design of experiments for metamodel generation. Special invited issue of the *Journal on Analog Integrated Circuits and Signal Processing*, 32, 7–16.
- [10] Crary, S.B., Cousseau, P., Armstrong, D., Woodcock, D.M., Mok, E.H., Dubochet, O., Lerch, P. and Renaud, P. (2000). Optimal design of computer experiments for metamodel generation using I-OPTTM. *Computer Modeling in Engineering and Sciences*, 1, 127-139.
- [11] Cui, J., de Klerk, N, Abramson M., Del Monaco, A., Benke, G., Dennekamp, M., Musk, A.W. and Sim, M. (2009). Fractional polynomials and model selection in generalized estimating equations analysis, with an application to a longitudinal epidemiologic study in Australia. *American Journal of Epidemiology*, 169, 113-121.
- [12] Dean, A., Morris, M., Stufken, J., and Bingham, D. (2015). *Handbook of Design and Analysis of Experiments*. CRC Press, Boca Raton, Florida.
- [13] Derendorf, H. and Meibohm, B. (1999). Modeling of pharmacokinetic/pharmacodynamic (PK/PD) relationships: concepts and perspectives. *Pharmaceutical Research*, 16, 176-185.
- [14] Dette, H., Kettelhake, K., Schorning, K., Wong, W.K., and Bretz, F. (2017). Optimal designs for active controlled dose finding trials with efficacy-toxicity outcomes. *Biometrika*, 104, 1003-1010.
- [15] Dette, H., Melas, V.B., Pepelyshev, A. (2008). Optimal designs for free knot least squares splines. *Statistica Sinica*, 18, 1047-1062.
- [16] Dette, H., Pepelyshev, A. and Zhigljavsky, A. (2008). Improving updating rules in multiplicative algorithms for computing D-optimal designs. *Computational Statistics and Data Analysis*, 53, 312-320.

- [17] Fedorov, V.V. (1972). *Theory of Optimal Experiments*. Academic Press, New York.
- [18] Fedorov, V.V. (1994). Optimal experimental design: spatial sampling. *Calcutta Statistical Association Bulletin*, 44, 17-21.
- [19] Fedorov, V.V. (1996). *Design of Spatial Experiments: Model Fitting and Prediction*. Oak Ridge National Laboratory Report, ORNL/TM-13152.
- [20] Gao, L.L. and Zhou, J. (2014). New optimal design criteria for regression models with asymmetric errors. *Journal of Statistical Planning and Inference*, 149, 140-151.
- [21] Gao, L.L. and Zhou, J. (2017). D-optimal designs based on the second-order least squares estimator. *Statistical Papers*, 58, 77-94.
- [22] Haines, L.M., Perevozskaya, I. and Rosenberger, W.F. (2003). Bayesian optimal designs for Phase I clinical trials. *Biometrics*, 59, 591-600.
- [23] Jain, D. and Tiwari, G.N. (2003). Modeling and optimal design of ground air collector for heating in controlled environment greenhouse. *Energy Conversion and Management*, 44, 1357-1372.
- [24] Khuri, A.I and Cornell, J.A. (1996). *Response Surfaces: Designs and Analyses*. Second Edition. CRC press, New York.
- [25] Kiefer, J. (1959). Optimum experimental designs. *Journal of the Royal Statistical Society. Series B*, 21, 272-319.
- [26] Kiefer, J. (1974). General equivalence theory for optimum designs (Approximate theory). *The Annals of Statistics*, 2, 849-879.
- [27] Kiefer, J. (1985). *Jack Carl Kiefer Collected Papers III: Design of Experiments*. Springer Verlag, New York.

- [28] Kiefer, J. and Wolfowitz, J. (1959). Optimum designs in regression problems. *The Annals of Mathematical Statistics*, 30, 271-294.
- [29] Kozusko F. and Bajzer, Z. (2003). Combining Gompertzian growth and cell population dynamics. *Mathematical Biosciences*, 185, 153-167.
- [30] Laird, A.K. (1964). Dynamics of tumor growth. *British Journal of Cancer*, 18, 490-502.
- [31] Michaelis, L. and Menten, M.L. (1913). Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift*, 49, 333-369.
- [32] Montgomery, D.C. (2012). *Design and Analysis of Experiments*. Eighth Edition. Wiley, New York.
- [33] Montgomery, D.C., Peck, E.A. and Vining, G.G. (2013). *Introduction to Linear Regression Analysis*. Fifth Edition. Wiley, New York.
- [34] Paquet-Durand, O., zettel, V. and Hitzmann, B. (2015). Optimal experimental design for parameter estimation of the Peleg mode. *Chemometrics and Intelligent Laboratory Systems*, 140, 36-42.
- [35] Pukelsheim, F. (1993). *Optimal Design of Experiments*. Wiley, New York.
- [36] Royston, P. and Altman, D.G. (1994). Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling. *Journal of the Royal Statistical Society. Series C*, 43, 429-467.
- [37] Silvey, S.D. (1980). *Optimal Design*. Chapman and Hall, London.
- [38] Silvey, S.D., Titterington, D.M. and Torsney, B. (1978). An algorithm for optimal designs on a finite design space. *Journal Communications in Statistics - Theory and Methods*, 7, 1379-1389.
- [39] Strum, J.F. (1999). Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11, 625-653.

- [40] Titterton, D.M. (1976). Estimation of correlation coefficients by ellipsoidal trimming. *Journal of the Royal Statistical Society. Series C*, 27, 227-234.
- [41] Torsney, B. and Alahmadi, A.M. (1995). Designing for minimally dependent observations. *Statistica Sinica*, 5, 499-514.
- [42] Torsney, B. and Martin-Martin, R. (2009). Multiplicative algorithms for computing optimum designs. *Journal of Statistical Planning and Inference*, 139, 3947-3961.
- [43] Wang, L. and Leblanc, A. (2008). Second-order nonlinear least squares estimation. *Annals of the Institute of Statistical Mathematics*, 60, 883-900.
- [44] Wong, W.K. and Zhou, J. (2018). CVX based algorithms for constructing various optimal regression designs. Preprint.
- [45] Yin, Y. and Zhou, J. (2017). Optimal designs for regression models using the second-order least squares estimator. *Statistica Sinica*, 28, 1841-1856.
- [46] Zhou, X., Joseph, L., Wolfson, D. B. and Belisle, P. (2003). A Bayesian A-optimal and model robust design criterion. *Biometrics*, 59, 1082-1088.