

Stony Brook University



OFFICIAL COPY

The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.

© All Rights Reserved by Author.

**Optimal Designs for Segmented Polynomial Regression Models
And Web-based Implementation of Optimal Design Software**

A Dissertation Presented

by

XiangFeng Wu

to

The Graduate School

In Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

December 2007

Stony Brook University

The Graduate School

XiangFeng Wu

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Wei Zhu –Dissertation Advisor

Professor

Department of Applied Mathematics and Statistics

Nancy Mendell - Chairperson of Defense

Professor

Department of Applied Mathematics and Statistics

Stephen Finch

Professor

Department of Applied Mathematics and Statistics

Weng Kee Wong

Professor of Biostatistics

University of California, Los Angeles

This dissertation is accepted by the Graduate School

Lawrence Martin

Dean of the Graduate School

Abstract of the Dissertation
**Optimal Designs for Segmented Polynomial Regression Models
And Web-based Implementation of Optimal Design Software**

by

XiangFeng Wu

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

2007

The segmented polynomial regression model is a special type of polynomial models in that it typically consists of piecewise polynomial sub models. Little research has been done on the optimal designs for segmented polynomial models, particularly for models with heteroscedastic variances. The primary objective of this paper is presenting the up-to-date findings in the optimal design theory for segmented polynomial models and their applications. It is shown that under certain conditions, D-optimal designs for the entire segmented polynomial models are associated with the individual D-optimal designs for the piecewise polynomial sub models. The relation between the overall D-optimal designs and individual D-optimal designs is established subsequently. In addition, we show that in some cases the locally D-optimal designs for a class of segmented polynomial models with nonlinear parameterization are the same as the D-optimal designs for another class of segmented polynomial models with linear parameterization. Secondly, we present the construction of a unique optimal-design website, on which various web-based optimal design software are incorporated. The ideas and technologies on the implementation are also covered in details in this paper.

Dedication

To my parents and the memorable times I spent in Stony Brook.

TABLE OF CONTENTS

List of Figures	vi
List of Tables	vii
Acknowledgements	viii
Introduction	1
1 Optimal Experimental Design Fundamentals	3
1.1 Model-Based Optimal Experimental Designs	3
1.2 Continuous and Exact Designs	4
1.3 Design Objectives and Optimal Criteria	5
1.4 Bayesian Approach and Frequentist Approach.....	7
1.5 Multiple-Objective Optimal Design and Design Constraints	8
1.6 Design Evaluation	9
2 Numerical Optimization	14
2.1 Overview	14
2.2 Sequential Quadratic Programming.....	15
2.3 Nelder-Mead Method.....	17
2.4 The Random Search Method	19
2.5 Performance Comparisons of Optimization Methods.....	20
3 Optimal Designs for Segmented Polynomial Models	23
3.1 Model Definition.....	23
3.2 Reparameterization	24
3.3 Optimal Designs When Join Points Are Known.....	26
3.4 Optimal Designs When Join Points Have to Be Estimated	44
3.5 Applications in Agriculture Studies.....	47
4 Implementation of Web-Based Optimal Design Software	54
4.1 System Overview	55
4.2 Technologies and Development Tools.....	56
4.3 Some Technical Details	58
5 Discussion and Future Work	67
5.1 Unequal Interests in Individual Segments	67
5.2 Future Development of Optimal Design Software	68
References	71

List of Figures

Figure 3.1. Checking condition plots verify the optimality of the D-optimal designs in Table 3.3.	40
Figure 3.2. Checking condition plots verify the optimality of the D-optimal designs in Table 3.4.	41
Figure 3.3. Checking condition plots verify the optimality of the D-optimal designs in Table 3.5.	43
Figure 3.4. Checking condition plots verifies the optimality of the locally D-optimal design for the linear-quadratic polynomial model.	51
Figure 4.1. System architecture and data flow illustration.	56
Figure 4.2. Designing a web page using the Visual Studio.NET Integrated Development Environment.	60

List of Tables

Table 3.1 Analytical support points of the D-optimal design for a homogeneous polynomial regression model of degree m on the interval $[a, b]$	38
Table 3.2 Support points of the D-optimal design for the model in Example 3.1 by numerical methods	38
Table 3.3 Support points of the individual D-optimal designs for the models in Example 3.2.....	39
Table 3.4 Support points of the D-optimal design for the entire segmented model in Example 3.2	41
Table 3.5 Support points of the D-optimal design for the model in Example 3.3	42
Table 4.1 C Sharp Data Communication Code.....	61
Table 4.2 Matlab code showing web based programming	64
Table 4.3 Matlab code for plotting graphs.....	65

Acknowledgements

I would like to thank my advisor, Professor Wei Zhu, for her mentoring and support over the last few years. I would also like to thank Professor Weng Kee Wong for his expert guidance throughout my dissertation research.

Many people at Stony Brook University assisted and encouraged me in various ways during my course of studies. My special thanks go to Professor Stephen Finch and Professor Nancy Mendell for their comments and suggestions on my dissertation.

I am grateful to my parents for giving me the spirit of excellence. I am thankful to my sister, my brothers, my relatives and my friends for their encouragement and patience over the years. Without them, I could never have come so far.

Introduction

Experimenting is one of the scientific approaches we learn about the world. Because experiments cannot avoid random errors, statistical methods are crucial for the design and analysis of experiments. Often interests are to fit mathematical models in experiments, and the fitted models can then be used for prediction. It is well known that the variances of estimated model parameters and predictions using the estimated model depend on the experimental design, and thus can be minimized by a well-designed experiment.

This paper primarily addresses some aspects of optimal experimental design theory, particularly for segmented polynomial regression models. The focus of this paper is the establishment of two new theorems in optimal design theory, specifically for segmented polynomial models. For the benefits of the optimal experimental design community, we also present the implementation of a web based optimal design platform incorporating various optimal design computer programs.

This paper is organized in five chapters. The first chapter opens with a review of the general model based optimal designs theory. This chapter covers the fundamental concepts including Fisher information, optimality criteria, optimality equivalence theorem, and so on. Optimization is a very important component in optimal design theory, so we dedicate Chapter 2 to this topic. Although some optimal designs may be found analytically, in most of applications, numerical optimization methods have to be used.

Chapter 2 introduces some numerical optimization methods. In addition we also embedded some discussion on adapting these numerical methods to accommodate more complicated optimization problems. Chapter 3 is the focus of this paper. In Chapter 3 a thorough discussion on the optimal designs for segmented polynomial regression models is given. The segmented polynomial regression model typically consists of polynomial sub models defined on independent sub intervals with join points (also called transition points). The motivation of this study is the hypothesis that there is some relationship between optimal designs for the sub polynomial models and the optimal design for the entire segmented polynomial model. It is shown that, in some cases, optimal designs for the entire segmented polynomial model can be derived from the optimal designs for the regular sub polynomial models on their sub intervals. Two relevant new theorems are established in this chapter and applications for agriculture studies are also given. Chapter 4 shows how we construct a unique optimal design platform – the website www.optimal-design.org. The website hosts a variety of optimal design computer programs, which can be invoked remotely. This type of functional website is new to the optimal experimental design community and is able to draw some attentions. This paper is closing with discussions on future research topics from optimal experimental design theories to optimal design software development in Chapter 5.

It is my sincere hope that this paper will be resourceful and is a piece of contribution to the optimal design theory. Your interest in exploring this paper in more details is appreciated.

1 Optimal Experimental Design Fundamentals

1.1 Model-Based Optimal Experimental Designs

The fundamental assumption in the optimal design theory is that the statistical model relating the response factor to the explanatory factor is known. For example, the experimenter should know at least the structure of the model, such as a one dimensional polynomial regression model or a logistic regression model, and so on. To the extent that the assumption on the underlying regression model is wrong, the resulting optimal design can have less value. In optimal designs for non-linear models, the experimenter should also have some knowledge about the model parameters.

Because the knowledge about the underlying regression model is fundamental, some use the term model-based to emphasize this fundamental assumption. In practice, prior to planning an optimal design, a pilot study is usually necessary to acquire information about the regression model to be further explored. What an optimal design can do is to help the experimenter to establish the regression model more precisely.

Following this concept, on www.optimal-design.org, we classified optimal designs by the type of models that the optimal design programs work for. For linear models, we have optimal design programs for polynomial regression models. For nonlinear models, we have optimal design programs for Quantal-dose response models, a class of kinetics models, exponential models and Fourier regression models.

1.2 Continuous and Exact Designs

Continuous designs are represented by the measure ξ over χ . If the design has trials at n distinct points in χ , we write

$$\xi = \begin{Bmatrix} x_1 & x_2 & \dots & x_n \\ w_1 & w_2 & \dots & w_n \end{Bmatrix}, \quad (1.1)$$

here the first line gives the values of the design variable at the design points and the second line w_i denotes the associated design weights ($0 \leq w_i \leq 1$ and $\sum_{i=1}^n w_i = 1$).

Continuous design is also called normalized design in some texts.

In practice, all designs are exact. The number of trials in an experiment must be an integer N , and the number of trials at any design point must be an integer as well. An exact design can be written as

$$\xi = \begin{Bmatrix} x_1 & x_2 & \dots & x_n \\ r_1/N & r_2/N & \dots & r_n/N \end{Bmatrix}, \quad (1.2)$$

where r_i denotes the integer number of trials at x_i and $\sum_{i=1}^n r_i = N$.

If N is not very large, a good exact design can frequently be found by integer approximation to the optimum continuous measure ξ^* . Often, for simple models with p parameters, there will be p design points with equal weight $1/p$. So that the exact design with $N=p$ trials is optimum. Such a possible such rounding scheme can be found in Pukelsheim (1993, Section 12.4). If the design weights are not rational, it will not be possible to find an exact design that is identical with the continuous optimum design.

The mathematical problem of finding the optimum design is simplified by considering only the continuous design, thus ignoring the constraint that the number of

trials at any design point must be an integer. So, for simplicity, we mean optimal continuous design when saying optimal design throughout this dissertation.

1.3 Design Objectives and Optimal Criteria

1.3.1 Aims of a good experiment

There are several criteria in the choice of an experimental design. For the moment, of the many criteria, we are interested in deriving an optimal design for estimating the underlying regression model. That is, the specific aim of the experiments is to estimate the model parameters with “high precision”.

Here “high precision” means that the estimated model parameters must have minimum variance and covariance. That is,

$$\text{Cov}(\hat{\theta}) \text{ or } M^{-1}(\xi, \theta) \text{ with respect to the design } \xi \text{ is minimized,}$$

where $M(\xi, \theta)$ is the Fisher Information Matrix and its inverse is proportional to the covariance matrix of the parameter estimates $\hat{\theta}$. In Chapter 3, we will derive this matrix for segmented polynomial regression explicitly.

But the question is, since the variance covariance of $\hat{\theta}$ is a matrix, how would we minimize a matrix? In general, there is no such thing as minimizing a matrix. However, we can minimize a real valued function of $M(\xi, \theta)$ to make matrices comparable. We now can introduce a few popular optimal design criteria, which all are functions of $M(\xi, \theta)$.

1.3.2 Optimal Design Criteria

Kiefer (1959) introduced the alphabetical design criteria. Here we introduce those types of optimal designs mentioned on www.optimal-design.org web site, namely D-, C-, E- and Ψ_k – optimal designs.

The most important design criterion in application is D -optimality, in which the generalized variance, or its logarithm $-\log |M(\xi, \theta)|$, is minimized. The D -optimality reflects the ellipsoidal confidence regions for the parameters of the model. A D -optimal design minimizes the content of this confidence region and so minimizes the volume of the ellipsoid. See A. C. Atkinson and A. N. Donev. (1992).

The D_s -optimal design is appropriate when the interest is in estimating a subset of the parameters precisely. Formal definition and an example for D_s -optimal designs will be provided in Chapter 5, where we discuss this type of optimal designs for segmented polynomial models.

If the interest of experiments is to estimate a function of the model parameters $c(\theta)$ with minimum variance, the design criterion to be minimized is thus

$$\text{var}(c(\hat{\theta})) \propto (\nabla c(\theta))^T M^{-1}(\xi, \theta) \nabla c(\theta),$$

where $\nabla c(\theta)$ is the gradient of $c(\theta)$. This is called C -Optimal design, which is typically used, for example, to estimate one or more percentiles of a dose-response. See Zhu et al. (2001) and Baek et al. (2006).

In E-optimality, the variance of the least well-estimated contrast $a^T \beta$ is minimized subject to $a^T a = 1$. That is E-optimality minimizes $\max(1/\lambda_i)$, where λ_i s are

the eigenvalues of the information matrix. E-optimal design is less common than D- or C-optimal design. Examples are on the website, and a good reference is Dette et al. (2003).

Kiefer (1975) introduced Ψ_k – Optimality, which is a generalization of the above optimality,

$$\psi_k(\xi) = \left(p^{-1} \sum_{i=1}^p \lambda_i^{-k} \right)^{1/k}, \text{ where } 0 \leq k < \infty$$

When limiting operations are properly defined, Ψ_0 – and Ψ_∞ – are D-optimal and E-optimal respectively.

1.4 Bayesian Approach and Frequentist Approach

An issue relating to the construction of optimal design for nonlinear models is that the information matrix $M(\theta, \xi)$ usually depends on the unknown parameters θ of that model. One approach, which is called the frequentist approach, is to adopt a best guess for the parameters and to construct optimal designs for that guess, termed ‘locally optimal’ (Chernoff, H 1953). An alternative is to adopt a Bayesian approach to design by optimizing the average of a function of the information matrix over a prior distribution placed on the unknown parameters.

For example, the Bayesian *D*-Optimal design minimizes:

$$\phi_D(\xi) = E_\theta \left(\log |M^{-1}(\xi, \theta)| \right) = \int_\theta \log |M^{-1}(\xi, \theta)| g(\theta) d\theta.$$

The Bayesian *C*-Optimal design minimizes:

$$\phi_c(\xi) = E_\theta \left((\nabla c(\theta))^T M^{-1}(\xi, \theta) \nabla c(\theta) \right) = \int_\theta (\nabla c(\theta))^T M^{-1}(\xi, \theta) \nabla c(\theta) g(\theta) d\theta,$$

where $g(\theta)$ denotes the prior probability density, which can be constructed from previous studies. The Bayesian approach is more robust to misspecified prior estimates of unknown parameters. However, it requires more complicated and intensive computation upon the design criteria.

Minimax optimal design is another approach to optimal design problems for nonlinear models. Essentially, the minimax optimal design seeks to minimize the maximum loss in some sense. For example, the variance of an estimated function of parameters in a nonlinear model depends on the parameters. A minimax optimal design can be constructed to minimize the largest possible variance over all plausible values of the unknown parameters. In Chapter 3, the minimax method will be introduced for the optimal designs for segmented models when the join points are unknown.

1.5 Multiple-Objective Optimal Design and Design Constraints

An experiment may often have several, competing objectives, which cannot be easily, characterized by only one of the standard optimality criteria (Zhu and Wong 1998). Suppose there are m objectives in the study and each of them is represented by a convex criterion ϕ_i , $i=1 \dots m$. Two approaches have been suggested for constructing a multiple-objective optimal design. One approach is to create a new objective function based on a weighted average of several design criteria and has been termed either a compound or a weighted design problem (Lauter 1976; Cook and Nachtsheim 1982 ;). The compound optimal design is the design which minimizes the convex combination $\sum_{i=1}^m \lambda_i \phi_i(\xi)$, where each $\lambda_i \in [0,1]$ is user-selected weight and $\sum_{i=1}^m \lambda_i = 1$. An alternative approach is

to optimize one primary design criterion, subject to the constraints of the other design criteria.

Compound optimal designs are easier to generate. Cook and Wong (1994) considered linear models with two objectives and established that every constrained optimal design is a compound optimal design and vice versa. Clyde and Chaloner (1996) extended this equivalence result to non-linear models with two or more objectives when Bayesian optimal designs are sought. This result enables us to find the constrained optimal design from the more easily computed compound optimal designs.

A good multiple-objective optimal design example relating to the above theory can be found on www.optimal-design.org for binary dose response models. A discussion on whether a compound design from piecewise polynomial is appropriate for segmented polynomial model is given in Chapter 5.

1.6 Design Evaluation

Once an optimal design is proposed, it is important to confirm that this design is really optimal. For some numerically derived optimal designs, there may be numeric errors. Each run of numerical optimization algorithm could possibly produce a slightly different solution. So, we need a method to compare “optimal designs”, as well as compare a user specified design with the optimal design.

1.6.1 The General Equivalence Theory

The General Equivalence Theorem (Kiefer 1961) provides three different but equivalent conditions to verify the optimality condition of an optimal design. In

continuous designs, we consider minimization of the general measure of the asymptotic variance function ϕ of the estimates. Under very mild assumptions, the most important of which are the compactness of χ and the convexity and differentiability of ϕ , designs that minimize ϕ also satisfy a second criterion.

The General Equivalence Theorem can be viewed as an application of the result that the derivatives are zero at a minimum of a function. However, the function depends on the measure ξ in the information matrix $M(\xi)$. Let the measure $\bar{\xi}$ put unit mass at the point x and let the measure ξ' be given by

$$\xi' = (1 - \alpha)\xi + \alpha\bar{\xi}, \quad (1.3)$$

Then,

$$M(\xi') = (1 - \alpha)M(\xi) + \alpha M(\bar{\xi}). \quad (1.4)$$

Accordingly, the derivative of ϕ in the direction $\bar{\xi}$ is

$$\varphi(x, \xi) = \lim_{\alpha \rightarrow 0^+} \frac{1}{\alpha} [\phi\{(1 - \alpha)M(\xi) + \alpha M(\bar{\xi})\} - \phi\{M(\xi)\}]. \quad (1.5)$$

The General Equivalence Theorem then states the equivalence of the following three conditions on the optimal design ξ^* :

- (1) The design ξ^* minimizes $\phi\{M(\xi)\}$;
- (2) The minimum of $\varphi(x, \xi^*) \geq 0$;
- (3) The derivative $\varphi(x, \xi^*)$ achieve its minimum at the points of the design

The Equivalence Theorem provides methods for the construction and checking conditions of optimum designs. The second and the third condition are often used to check the optimality of a design that satisfies the first condition.

An application of the equivalence theorem to polynomial regression models will be given in Chapter 3 when we discuss D-optimal designs for segmented polynomial regression models. A nice application of the theorem to logistic regression models can be found in Haines et al. (2003), where Bayesian D-optimal designs for Logistic regression models were discussed. The Bayesian D-optimal is to

minimize $\phi\{M(\xi)\} = E_{\theta}(\log|M^{-1}(\xi, \theta)|)$. By direct calculation, the directional derivative function $\varphi(x, \xi)$ of $\phi\{M(\xi)\}$ is

$$\varphi(x, \xi) = p - E_{\theta} \text{tr} M^{-1}(\xi, \theta) m(x, \theta),$$

where $m(x, \theta)$ is the information matrix at x and $p = \text{dim}(\theta)$. If ξ^* is the optimal design, then according to the condition (2) and (3) of the theorem,

$$\varphi(x, \xi^*) \geq 0 \text{ Or } E_{\theta} \text{tr} M^{-1}(\xi^*, \theta) m(x, \theta) \leq p$$

The equality sign establishes when x are the design points.

If the underlying regression model is very complicated, the information matrix or design criterion can be complicated. Thus, it may be very difficult to derive the directional derivative in a closed form. In that case, simulation of experiments, as an alternative, can be used to check the optimality conditions of a resulting design.

1.6.2 Simulation of Experiments

In many applications, such as optimal designs for nonlinear mixed effects models, the optimality criteria are very complicated functions, which are very difficult to compute for directional derivatives. Therefore, simulation technique may be a practical alternative approach in the evaluation of the performance of optimal designs.

The principle idea of the simulation approaches is, we simulate experiments based on various designs including the optimal design, and analyze simulated data and compare the regression results. Depending on the objectives of the experiments, comparisons can be made on the interested parameters. For example, comparisons on the standard errors of the parameter estimates usually are made for comparing a D-optimal design with an arbitrary design. Simulation techniques are widely used in optimal designs for industrial applications, such as pharmacokinetic and pharmacodynamic studies.

1.6.3 Design Efficiency

Denote the optimal design as ξ^* , and suppose an arbitrary design is ξ . Then the D -efficiency of the arbitrary design is defined as

$$D_{eff} = \left\{ \frac{|M(\xi)|}{|M(\xi^*)|} \right\}^{1/p}, \text{ where } p \text{ is the number of model parameters.}$$

Bayesian D-efficiency (Clyde and Chaloner 1996) is defined as

$$\varepsilon_D(\xi) = \exp(\phi_D(\xi^*) - \phi_D(\xi)) / p$$

For a Bayesian C-Optimal design, the design efficiency is defined as the ratio of asymptotic variances of the estimated function of model parameters

$$\varepsilon_C = \frac{\phi_C(\xi^*)}{\phi_C(\xi)}.$$

Design efficiency can be used to compare user specified design to the optimal design. Theoretically, design efficiency should be between 0 and 1, since the values of asymptotic variances are positive and these values are supposed to be smaller at optimal designs than at any other designs.

2 Numerical Optimization

2.1 Overview

Optimal experimental design is a two-part problem. The first part is regression analyses based on the assumptions of the underlying regression model. This part gives the asymptotic variance of the estimates from a design. The second part is optimizing a design criterion function with respect to the design variables. It would be nice if we could optimize a function analytically. Indeed, for some optimal designs, analytical optimization solutions do exist. A nice example can be found in Biedermann et al. (2006), in which they find optimization solutions via geometric approaches for optimal designs for logistic regression models. However, in many cases, it is extremely difficult to find optimal designs analytically. Thus, we introduce the role of numerical optimizations in this chapter.

Numerical optimization algorithms have been evolving rapidly. With the development of advanced computer hardware and software, more and more efficient numerical optimization algorithms are developed. These numerical optimization algorithms can be roughly classified into two categories, deterministic algorithms and stochastic algorithms. As their names suggest, the two types of optimization algorithms are differ in that if the algorithm has perfect information or not to determine the search

direction at every step. A deterministic algorithm converges to the same solution with the same initial information, while a stochastic algorithm doesn't.

Optimization algorithms, such as the sequential quadratic programming, Newton's methods and the Nelder-Mead method are deterministic, while simulated annealing, random search and modified exchange algorithms fall into stochastic category. Due to space limit, we introduce some of the numerical optimization algorithms we feel very useful in the context of optimal experimental design. We also adapt these algorithms to deliver better performance.

2.2 Sequential Quadratic Programming

Optimization problems in optimal designs context are usually nonlinear constrained optimization problems. A primary optimal criterion function is to be optimized subject to some constrained functions. The primary criterion function is nonlinear, and the constrained functions can be linear or nonlinear.

In constrained optimization, the general approach is to transform the problem into an easier sub problem that can then be solved and used as the basis of an iterative process. A common feature of a large class of early methods is the translation of the constrained problem to a basic unconstrained problem by using a penalty function for constraints that are near or beyond the constraint boundary. In this way the constrained problem is solved using a sequence of parameterized unconstrained optimizations, with the limit of the sequence converging to the constrained problem.

These methods are now considered relatively inefficient and have been replaced by methods that have focused on the solution of the Kuhn-Tucker (KT) equations. The

KT equations are necessary conditions for optimality for a constrained optimization problem. If the problem is a so-called convex programming problem, that is, $f(x)$ and $G_i(x), i = 1, 2, \dots, m$ are convex functions, then the KT equations are both necessary and sufficient for a global solution point. The Kuhn-Tucker equations can be stated as

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \cdot \nabla G_i(x^*) = 0$$

$$\lambda_i^* \cdot G_i(x^*) = 0 \quad i = 1, \dots, m$$

$$\lambda_i^* \geq 0 \quad i = m_e + 1, \dots, m$$

The solution of the KT equations forms the basis to many nonlinear programming algorithms. These algorithms attempt to compute the Lagrange multipliers directly.

Constrained quasi-Newton methods guarantee super linear convergence by accumulating second order information regarding the KT equations using a quasi-Newton updating procedure. These methods are commonly referred to as Sequential Quadratic Programming (SQP), in which a Quadratic Programming (QP) sub problem is solved at each major-iteration. For more details, users are referred to Fletcher (1987).

SQP method represents the state of the art in nonlinear programming methods. It has been proved that it outperforms every other tested method in terms of efficiency, accuracy, and percentage of successful solutions, over a large variety of test problems. However, the biggest disadvantage of the Sequential Quadratic Programming method is that it sometimes converges to a local minimum instead of a global minimum when the objective function is not a unimodal function. The algorithm requires an initial point to start, and its performance is sensitive to the initial point. This convergence property may be improved by integrating SQP with a grid search method to be discussed below.

The grid search algorithm is a straightforward method that may be applicable in any case. The principle idea is partitioning the optimization domain into a grid, and evaluating the objective function at each cell of the grid. Literally, the grid search method can guarantee an optimal solution as long as computing cost is not a problem. It can be very inefficient in some circumstances. For example, if the objective function has a cubic design space, then to evaluate 10 possible values on each direction, 10^3 function evaluations are required. As more variables are involved, the number of function evaluations increases exponentially.

Since at each major-iteration in SQP, the Quasi-Newton algorithm for finding the solution to the quadratic sub problem requires a good starting point, the overall optimization performance is very dependent on these starting points. Therefore, it is very critical to find a good starting point at each major-iteration. It is helpful to use a grid search method to locate a starting point at each major-iteration in SQP. Thus we propose to use the combination of SQP and the grid search method. It is a tradeoff between accuracy and efficiency. The combined optimization method outperformed SQP or grid search alone in some studies [See Wu and Tensfeldt (2006)].

2.3 Nelder-Mead Method

The Nelder-Mead (1965) method (also called the downhill simplex method) is another commonly used nonlinear optimization algorithm. The method approximately finds a locally optimal solution to a problem with N variables when the objective function varies smoothly.

The method uses the concept of a simplex, a shape with $N+1$ vertex in N dimensions (that is a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space and so on). The method works in an iterative manner. At each step, Nelder-Mead generates a new set of vertices (a simplex) and evaluates the objective function at the vertices arranged as a simplex. The algorithm then chooses to replace the vertex that yields the worst function value with a new vertex generated by reflection through the centroid of the rest of vertices. A new simplex is formed and the algorithm progresses. In this manner, the algorithm generates a sequence of simplex, for which the function values at the vertices get smaller and smaller. The size of the simplex reduced and the coordinates of the optimum are found.

At each step, the reflection method works with some rules. If the new vertex generated is better than the current best vertex, then the algorithm expands the simplex and reflects further; if the new vertex is just a good one, the algorithm reflects again; if the new vertex is the worst vertex, the algorithm compresses the simplex and reflect closer. Like many other multi-dimensional optimization algorithms, Nelder-Mead sometimes converges to local minima. But with some standard approaches, such as restarting itself with a new simplex, the algorithm may escape from some local minima.

Nelder-Mead method does not include handling constraints. However a general technique is to add penalty functions to the objective functions, as mentioned in the previous section. A penalty function is a small fraction of violation of a constraint. Thus, while minimizing the objective function, the algorithm therefore also minimizes the constraint violation.

2.4 The Random Search Method

SQP and The Nelder-Mead method are considered as deterministic optimization algorithms, meaning they are guaranteed to converge to the same local/global minima if they start from the same initial points. Unlike the SQP and Nelder-Mead method, the random search method, as its name suggests, converges to different local minima at each run.

Random search is perhaps the simplest method of stochastic optimization and can be quite effective in many problems. The random search method has a number of advantages relative to most of other search methods. These include relative ease of coding in software, the need only to obtain measurements (versus gradients or other auxiliary information), reasonable computational effort, and applicability to almost any non-trivial loss functions, and a strong theoretical foundation.

The basic random search method can be implemented in two ways. One is batch mode, and the other is recursive. In batch mode, a number of points in design space are randomly generated at once, and those points yielding the lowest value of the objective function are taken as the estimated optimum. The recursive random search algorithm is described below.

- *Step 0 (Initialization)* Choose initial values of the design ξ , say ξ_0 , either randomly or deterministically. (If random, usually a uniform distribution on ξ is used). Calculate the objective function $\phi(\xi)$. Set $\min \phi(\xi) = \phi(\xi_0)$, $\xi^* = \xi_0$.
- *Step 1* Generate a new independent value ξ' , according to the chosen probability distribution. If $\phi(\xi') < \min \phi(\xi)$, set $\min \phi(\xi) = \phi(\xi')$, $\xi^* = \xi'$. Otherwise, go to *Step 2*.

- *Step 2* Stop if the maximum number of function evaluations has been reached or the user is otherwise satisfied with the current estimate for the minimum via appropriate stopping criteria. Otherwise return to *Step 1*.

The above algorithm converges almost surely to optimal solutions under very general conditions. [See Spall (2003), pp. 40-41]. To evaluate the algorithm, it is also interesting to examine the rate of convergence. The convergence rate shows how close ξ^* is likely to be the optimal solution for a given cost of search. While random search is a reasonable algorithm when the design space is low dimensional, it can be very slow for even moderately dimensioned design spaces. [See Spall (2003), pp. 42-43]. This is a direct consequence of the exponential increase in the size of the search space as the dimension increases.

In addition, objective functions are usually subject to some constraints in experimental design settings. Thus, the generated samples from the design space must also satisfy these constraints. Generally, constraints are implemented as penalty functions imposed on the primary objective function, and those penalty functions can be incorporated in Step 1 on a case-by-case basis.

2.5 Performance Comparisons of Optimization Methods

In optimal design context, the complexities of the objective functions to be optimized vary from case to case, depending on the underlying regression models. Generally, the more variables an objective function has the more complex the optimization problem will be. In Bayesian optimal designs, computing the Bayesian integral also complicates the optimization problem.

To evaluate the performance of an optimization algorithm, we usually look at two aspects of an optimization algorithm. One is the convergence of the algorithm, in other words, if the algorithm is able to converge to the optimal solutions. The other is the convergence rate, which means how fast the algorithm converges to the optimal solution.

We utilized the optimization algorithms discussed in the previous sections in a few different optimal design applications. Here we give a brief comparison on these algorithms based on what we found in our studies. The comparison is not meant to be general.

In the application of optimal designs for dual-objective c-optimal designs for logistic regression models, we evaluated the random search method (RS), the Nelder-mead (NM) method and the sequential quadratic programming (SQP) method. For some input data sets, the convergence performance was $SQP > RS > NM$, while for some other input data, RS outperformed SQP. However, SQP always had higher convergence rate than RS and NM. SQP was also very dependent on its starting points, which is why it is sometimes less efficient than RS. If using SQP combining with the grid search method, the convergence performance can be improved; however, the convergence rate will be decreased. Of all three optimization algorithms, convergence rates were acceptable and thus were not an important factor in the context of optimal experimental designs.

While it appears NM didn't perform as well as SQP or RS in optimal designs for logistic regression models, it outperformed RS and SQP in the applications for segmented polynomial models. The optimization problems were partially simplified because the weights on the support points were determined to be equal, and thus optimization on the weights was not necessary. NM algorithm works very well for the examples in chapter 3.

In summary, although the numerical optimization plays a very important role in the context of optimal experimental designs, it should be noticed that the performance of numerical optimization varies from case to case. More importantly, optimal designs found by numerical methods are usually not necessarily theoretical optimum. So a good practice should be using multiple numerical optimization algorithms for the same application, and using the best one of the numerical optimum as the result.

3 Optimal Designs for Segmented Polynomial Models

Optimal design methods are model oriented. The underlying regression model determines the way we derive optimal designs. In this chapter, we concentrate on the aspects of optimal designs for heteroscedastic segmented polynomial regression models. The segmented polynomial model is a special class of polynomial models. If the join points are known, it is linear. Otherwise, it is nonlinear. Fitting segmented polynomial has been discussed. Extensive literature search yields only a few papers regarding optimal designs for segmented polynomial models or similar regression models, such as spline functions. Park (1978) discussed D-optimal designs for fitting segmented polynomial regression models, assuming the errors are identically normal distributed with constant variances. Other discussions mainly focus on the spline function, which is a special segmented polynomial function. [See, e.g. Dette et al. (2006)].

In this chapter, we consider segmented regression models with heteroscedastic variance, which is a more general assumption on the random errors. We now continue our discussion with the explicit construction of D-optimal designs for heteroscedastic segmented polynomial regression models.

3.1 Model Definition

The one variable polynomial function $\eta(x)$ is defined as

$$\eta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_n x^n$$

In some regression analysis, a desirable choice of a regression model is

$$y_i = \eta(x_i) + e_i \tag{3.1}$$

where

$$\begin{aligned} \eta(x) &= \theta_{10} + \theta_{11}x + \dots + \theta_{1q_1}x^{q_1}, & a \leq x \leq \alpha_1 \\ &= \theta_{20} + \theta_{21}x + \dots + \theta_{2q_2}x^{q_2}, & \alpha_1 \leq x \leq \alpha_2 \\ &\quad \vdots \\ &= \theta_{r0} + \theta_{r1}x + \dots + \theta_{rq_r}x^{q_r}, & \alpha_{r-1} \leq x \leq b \end{aligned}$$

$$e_i \sim N(0, \sigma^2 \lambda^{-1}(x_i)),$$

in which $\lambda(x)$ is called the efficiency function of an experiment by Fedorov (1972). Thus (3.1) defines a heteroscedastic segmented polynomial regression model. When $\lambda(x) \equiv 1$, the regression model is the case with homoscedastic variances. α_i are called join points, also called free knots or transition points.

The function $\eta(x)$ has desirable properties as an approximating function, where the piecewise sub models are polynomials and $\eta(x)$ is continuous in x over the entire interval $[a, b]$ are imposed. If the order of each piecewise polynomial in (3.1) has the same degree m (i.e., $q_i = m, i = 1, 2, \dots, r$), and $\eta(x)$ is a continuous function with $m-1$ continuous derivatives over the entire interval $[a, b]$, then $\eta(x)$ is called a spline function of degree m .

3.2 Reparameterization

It can be shown that the segmented polynomial in (3.1) can be represented as a simple linear combination of the functions below [See e.g. Gallant and Fuller (1973)]:

$$1, x, x^2, \dots, x^q, T_0(x - a_i), T_1(x - a_i), \dots, T_q(x - a_i) \quad (3.2)$$

where

$$\begin{aligned} T_j(x - a_i) &= 0 && \text{if } x \leq a_i, \\ &= (x - a_i)^j && \text{if } x > a_i, \\ &&& \text{for } i = 1, 2, \dots, r-1. \text{ and } q = \max\{q_i : i = 1, 2, \dots, r\} \end{aligned}$$

The functions (3.2) form a basis for the segmented polynomial model. In addition, the function $\eta(x)$ in (3.1) can be written as a single equation (Fuller 1969):

$$\eta(x) = \sum_{i=0}^{q_1} \theta_{1i} x^i + \sum_{j=0}^{q_2} \theta_{2j} T_j(x - \alpha_1) + \dots + \sum_{j=0}^{q_r} \theta_{rj} T_j(x - \alpha_{r-1}) \quad (3.3)$$

Moreover, it has been shown that, in equation (3.2), the continuity condition at the join points will delete $T_0(x - a_i)$ term; the first derivative condition on the entire interval will delete $T_1(x - a_i)$; similarly for the second order derivative condition, and so forth [See Fuller (1969) and Rice (1969)].

If the join points are known, $\eta(x)$ is linear in

$$\theta^T = (\theta_{10}, \theta_{11}, \dots, \theta_{1q_1}, \theta_{20}, \theta_{21}, \dots, \theta_{2q_2}, \dots, \theta_{r0}, \theta_{r1}, \dots, \theta_{rq_r}).$$

We can define $f(x)^T = \nabla \eta(x, \theta)$. So that the regression model in (3.1) is

$$E(y | x) = \eta(x) = \theta^T f(x) \text{ and } Var(y | x) = \sigma^2 \lambda^{-1}(x) \quad (3.4)$$

If the join points are unknown, $\eta(x)$ is nonlinear in the parameters, and $f(x)^T = \nabla \eta(x, \theta, \alpha)$, where $\alpha^T = (\alpha_1, \alpha_2, \dots, \alpha_{r-1})$. Taylor expansion of $\eta(x)$ yields a linearization that is similar to the regression function in (3.4).

3.3 Optimal Designs When Join Points Are Known

When the join points α_i are known, the regression function in the model (3.4) is linear in the unknown parameters θ . If the join points are unknown and must be estimated from data, the regression function becomes nonlinear in the join points α_i . We will explore methods including locally D-optimal designs and minimax optimal designs later this chapter, for optimal designs for segment models with unknown join points. Based on the regression model assumption defined in (3.4), we can find the best linear unbiased estimate of θ and the associated variance and covariance matrix of the estimate using the following theorem.

Theorem 3.1 *if*

$$E(y | x) = \theta^T f(x) \text{ and } Var(y | x) = \sigma^2 \lambda^{-1}(x)$$

, then the best unbiased linear estimate for θ will be

$$\hat{\theta} = M^{-1}Y \tag{3.5}$$

where

$$M = \sum_{i=1}^n \lambda(x_i) f(x_i) f(x_i)^T \tag{3.6}$$

$$Y = \sum_{i=1}^n \lambda(x_i) y_i f(x_i) \tag{3.7}$$

The variance of the estimate $\hat{\theta}$ is equal to

$$D(\hat{\theta}) = M^{-1} \quad (3.8)$$

Proof:

I) we show (3.5) is an unbiased estimate first:

$$E(\hat{\theta}) = E(M^{-1}Y) = M^{-1}E(Y) \quad (3.9)$$

$$\begin{aligned} E(Y) &= E\left[\sum_{i=1}^n \lambda(x_i)y_i f(x_i)\right] = \sum_{i=1}^n \lambda(x_i)f(x_i)E(y_i) \\ &= \sum_{i=1}^n \lambda(x_i)f(x_i)f^T(x_i)\theta \\ &= M\theta \end{aligned} \quad (3.10)$$

Setting (3.10) into (3.9), we have:

$$E(\hat{\theta}) = M^{-1}M\theta = \theta, \text{ so that the estimate } \hat{\theta} \text{ is unbiased.}$$

II) We now prove that the estimate is the best among all unbiased estimates in that $D(\hat{\theta})$

is the smallest. We consider an arbitrary linear unbiased estimate

$$\tilde{\theta} = Ty \quad (3.11)$$

Since $\tilde{\theta}$ is unbiased,

$$E(\tilde{\theta}) = E(Ty) = \theta \quad (3.12)$$

$$E(Ty) = TE(y) = TF'\theta = \theta, \quad (3.13)$$

where

$$F = (f(x_1), f(x_2), \dots, f(x_n)). \quad (3.14)$$

From (3.13), it follows that,

$$TF' = I, \quad (3.15)$$

where I is the identity matrix. Also,

$$D(\tilde{\theta}) = TD(y)T', \quad (3.16)$$

where

$$D(y) = \sigma^2 \begin{pmatrix} \lambda^{-1}(x_1) & 0 & \cdots & 0 \\ 0 & \lambda^{-1}(x_2) & \cdots & 0 \\ \vdots & \vdots & \cdots & \\ 0 & 0 & \cdots & \lambda^{-1}(x_n) \end{pmatrix}. \quad (3.17)$$

Let the matrices

$$\phi = F\Sigma^{-1} \text{ and } \varphi = T\Sigma, \quad (3.18)$$

where

$$\Sigma = \sigma \begin{pmatrix} \lambda^{-1/2}(x_1) & 0 & \cdots & 0 \\ 0 & \lambda^{-1/2}(x_2) & \cdots & 0 \\ \vdots & \vdots & \cdots & \\ 0 & 0 & \cdots & \lambda^{-1/2}(x_n) \end{pmatrix}, \quad \Sigma\Sigma' = D(y).$$

Thus, equations (3.15) and (3.16) can be rewritten as

$$\varphi\phi' = I \quad (3.19)$$

and

$$D(\tilde{\theta}) = \varphi\varphi'. \quad (3.20)$$

It can be verified that

$$D(\tilde{\theta}) = \varphi\varphi' \equiv \{(\phi\phi')^{-1}\phi\}\{(\phi\phi')^{-1}\phi\}^T + \{\varphi - (\phi\phi')^{-1}\phi\}\{\varphi - (\phi\phi')^{-1}\phi\}^T. \quad (3.21)$$

Each term on the right-hand side of (3.21) is a matrix of the type AA^T , of which the

diagonal elements are nonnegative. Thus, $D(\tilde{\theta})$ will have a smallest value if

$$\varphi = (\phi\phi')^{-1}\phi. \quad (3.22)$$

That is,

$$T = [FD^{-1}(y)F']^{-1}FD^{-1}(y)$$

Thus

$$\tilde{\theta} = Ty = [FD^{-1}(y)F']^{-1}FD^{-1}(y)y = M^{-1}Y = \hat{\theta},$$

where M and Y are defined by Equations (3.6) and (3.7) respectively. This concludes that $\hat{\theta}$ defined by Equation (3.8) is the best-unbiased linear estimate.

The fundamental idea in the proof of Theorem 3.1 is identical to that in Fedorov (1972) p25-26, in which the author proves a similar theorem with slightly different assumptions on the underlying regression model.

The matrix M defined by equation (3.6) is called Fisher information matrix. If M is not singular, its inverse, equation (3.8), is the variance and covariance matrix of $\hat{\theta}$. The matrices M or M^{-1} form the basis of most of the optimal design criteria, i.e., the D -optimal criterion is a function of the Fisher information matrix M .

Equation (1.1) defines a continuous design to simplify the problem of finding optimal designs. We calculate the explicit form of the information matrix M with a continuous design. Suppose there are N total subjects of the design ξ , it follows that

$$M = N \sum_{i=1}^n w_i \lambda(x_i) f(x_i) f(x_i)^T .$$

For simplicity, we omit N in the formula since it is just a constant, and is irrelevant to our discussion. So we denote,

$$M(\xi) = \sum_{i=1}^n w_i \lambda(x_i) f(x_i) f(x_i)^T . \quad (3.23)$$

The most popular design criterion in applications, is that of D -optimality, which minimizes,

$$\log |M^{-1}(\xi)| \quad (3.24)$$

Or equivalently maximizes $|M(\xi)|$.

where ξ stands for a continuous design. We denote ξ^* as the optimal design that satisfies the optimal criterion. We apply the General Equivalence Theorem to derive a secondary equivalent optimality condition.

Using the definition in (3.24), the D-optimal criterion function is

$\phi\{M(\xi)\} = \log|M^{-1}(\xi)|$. According to the condition (1) of the General Equivalence theory, the optimal design ξ^* minimizes ϕ . Using the definitions (1.3), (1.4) and (1.5), the directional derivative at design ξ^* in the direction of $\bar{\xi}$ is calculated as follows:

$$\begin{aligned}
\varphi(x, \xi^*) &= \lim_{\alpha \rightarrow 0^+} \frac{1}{\alpha} \left[\phi\left((1-\alpha)M(\xi^*) + \alpha M(\bar{\xi})\right) - \phi\left(M(\xi^*)\right) \right] \\
&= \frac{d}{d\alpha} \log|M^{-1}(\xi')|_{\alpha=0} = -\frac{d}{d\alpha} \log|M(\xi')|_{\alpha=0} \\
&= \lim_{\alpha \rightarrow 0^+} -\text{Tr}M^{-1}(\xi') \frac{d}{d\alpha} M(\xi') \\
&= \lim_{\alpha \rightarrow 0^+} -\text{Tr}M^{-1}(\xi') [M(\bar{\xi}) - M(\xi^*)] \\
&= \lim_{\alpha \rightarrow 0^+} -\text{Tr}\left[(1-\alpha)M^{-1}(\xi^*) + \alpha M^{-1}(\bar{\xi})\right] [M(\bar{\xi}) - M(\xi^*)] \\
&= m - \text{Tr}M(\bar{\xi})M^{-1}(\xi^*)
\end{aligned} \tag{3.25}$$

In (3.25), m is the rank of $M(\xi)$ and is also the number of unknown parameters in the regression model or can be viewed as the dimension of $f(x)$, and the design $\bar{\xi}$ consists of one support point x with unit mass on it. We then obtain

$$\begin{aligned}
\varphi(x, \xi^*) &= m - \text{Tr}M^{-1}(\xi^*) \lambda(x) f(x) f(x)^T \\
&= m - \lambda(x) f(x)^T M^{-1}(\xi^*) f(x)
\end{aligned} \tag{3.26}$$

Equation (3.26) gives the explicit form of $\varphi(x, \xi^*)$, which is a polynomial function in the variable x . $\varphi(x, \xi^*)$ should be equal to zero at the design points and be greater than zero at all other points on the design interval according to the condition (2) and the

condition (3) of the General Equivalence Theorem. We will use this equation to check these conditions for the D-optimal designs in following examples.

By Theorem 3.1 and equation (3.23), it is not difficult to have the D -optimal criterion function specified for numerical optimization procedures. However, it is always desirable to derive some properties from the criterion function that may lead to analytical solutions to the optimal designs. In fact, the following corollaries show that the determinant of the information matrix does have some interesting properties that can facilitate the derivation of D-optimal designs analytically.

Corollary 3.1 The number of support points, n must be greater or equal to m , the number of unknown parameters or the dimension of $f(x)$.

Proof: Since

$$M(\xi) = \sum_{i=1}^n w_i \lambda(x_i) f(x_i) f(x_i)^T = F \begin{pmatrix} w_1 \lambda(x_1) & 0 & \cdots & 0 \\ 0 & w_1 \lambda(x_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_1 \lambda(x_1) \end{pmatrix} F^T, \quad (3.27)$$

where F is defined in (3.14). F is an m by n matrix. By Cauchy-Binet formula, it is easy to see that n must be greater or equal to m . Otherwise, $|M(\xi)|$ will be zero.

Corollary 3.2 When $n=m$ (the number of support points equals to the number of parameters), the determinant of the information matrix (3.23) is given by:

$$|M(\xi)| = \prod_{i=1}^n w_i \prod_{i=1}^n \lambda(x_i) |F|^2$$

Proof: Using equation (3.27) and by Cauchy –Binet formula, the result follows.

Corollary 3.3 Subject to the restriction $\sum_{i=1}^n w_i = 1$, $\prod_{i=1}^n w_i$ is maximized

when $w_1 = w_2 = \dots = w_k = 1/n$. It follows that $|M(\xi)|$ is maximized when the same number of observations is taken at each of the design points when $n=m$.

3.3.1 The regression function $\eta(x)$ is continuous at the known join points

If the regression function $\eta(x)$ is continuous and there are no other restrictions at the join points, $\eta(x)$ can be written in the following form

$$\eta(x) = \sum_{i=0}^{q_1} \theta_{1i} x^i + \sum_{j=1}^{q_2} \theta_{2j} T_j(x - \alpha_1) + \dots + \sum_{j=1}^{q_r} \theta_{rj} T_j(x - \alpha_{r-1}). \quad (3.28)$$

There are $m = 1 + q_1 + q_2 + \dots + q_r$ unknown parameters given the join points are known.

Condition 3.1 Denote $\psi(x, \xi^*) = m - \gamma - \lambda(x) f(x)^T M^{-1}(\xi^*) f(x)$, where γ is a small number. $\psi(x, \xi^*)$ has no more than $2q_1 + 1$ roots on $[a, \alpha_1]$, and has no more than $2q_i$ roots on $(\alpha_i, \alpha_{i+1}]$ ($i = 1, 2, \dots, r-1; \alpha_r = b$).

Corollary 3.4 If Condition 3.1 is satisfied, then the D-optimal design for the regression model (3.4) with regression function (3.28) has exactly $m = 1 + q_1 + q_2 + \dots + q_r$ support points with equal weights $\frac{1}{m}$ on each support point.

Proof: By the General Equivalence Theory, at the D-optimal design points, it follows that

$$m - \lambda(x^*)f(x^*)^T M^{-1}(\xi^*)f(x^*) = 0,$$

and at all other points,

$$m - \lambda(x)f(x)^T M^{-1}(\xi^*)f(x) > 0.$$

If there are more than m support points on the entire interval, then there exists at least one interval that has more than $1 + q_1$ support points for $[a, \alpha_1]$ or has more than q_i support points for $(\alpha_i, \alpha_{i+1}] (i \geq 1)$. As a consequence, $\psi(x, \xi^*)$ would have more than $2q_1 + 1$ roots on $[a, \alpha_1]$ or have more than $2q_i$ roots on $(\alpha_i, \alpha_{i+1}] (i \geq 1)$. Both cases would yield a contradiction to Condition 3.1.

Moreover, if Condition 3.1 is satisfied, same argument would yield that the D-optimal design consists of $1 + q_1$ support points on $[a, \alpha_1]$ and q_i support points on $(\alpha_i, \alpha_{i+1}] (i \geq 1)$.

Condition 3.2 The D-optimal design for a polynomial model of degree q_i on $[\alpha_{i-1}, \alpha_i]$, ($i = 1, 2, \dots, r, \alpha_0 = a, \alpha_r = b$) has exactly $q_i + 1$ support points and two of them are α_{i-1} and α_i .

Note Condition 3.2 is more restrict than Condition 3.1. Indeed it implies that the function $\psi(x, \xi^*)$ has exactly $2q_i$ roots on $[\alpha_{i-1}, \alpha_i]$, and two roots are α_{i-1} and α_i . Condition 3.2 also implies that if it is satisfied, the support points of the D-optimal design on the entire interval include all join points, and the scheme of the support point locations is:

$$\begin{aligned} a = x_{1,1}^* < \dots < x_{1,q_1+1}^* &= \alpha_1 \\ \alpha_1 < x_{2,1}^* < \dots < x_{2,q_2}^* &= \alpha_2 \\ \vdots & \\ \alpha_{r-1} < x_{r,1}^* < \dots < x_{r,q_r}^* &= b \end{aligned}$$

Theorem 3.2 Consider the segmented polynomial regression model (3.4) with regression function $\eta(x)$ is (3.28). If Condition 3.2 is satisfied, there exists a unique D -optimal

design ξ_D^* with exactly $m = 1 + \sum_{i=1}^r q_i$ support points with equal weights $\xi_D^*(x_i) = 1/m$.

Moreover, the supports points are the same set of support points of the individual D -optimal designs for the polynomial sub models on the corresponding sub intervals.

Proof: If Condition 3.2 is satisfied, Condition 3.1 is satisfied automatically, then by

Corollary 3.4, the D -optimal design has exactly $m = 1 + \sum_{i=1}^r q_i$ support points with equal

weight on each support point. This implies:

$$\begin{aligned} |M| &= \prod_{i=1}^m w_i \prod_{i=1}^m \lambda(x_i) |F|^2 = \left(\frac{1}{m}\right)^m \prod_{i=1}^m \lambda(x_i) \prod_{i=1}^r |F_i|^2 \\ &= \left(\frac{1}{m}\right)^m \prod_{i=1}^{q_1+1} \lambda(x_{1i}) |F_1|^2 \prod_{i=1}^{q_2} \lambda(x_{2i}) |F_2|^2 \cdots \prod_{i=1}^{q_r} \lambda(x_{ri}) |F_r|^2 \end{aligned} \quad (3.29)$$

where $F = \begin{pmatrix} F_1 & \cdots & \cdots & \cdots \\ 0 & F_2 & \cdots & \cdots \\ 0 & 0 & \ddots & \cdots \\ 0 & 0 & 0 & F_r \end{pmatrix}$ is an upper triangle matrix, and F_1 is a $q_1 \times q_1$ classical

Vandermonde matrix and

$$F_i = \begin{pmatrix} x_{i1} - \alpha_{i-1} & x_{i2} - \alpha_{i-1} & \cdots & \alpha_i - \alpha_{i-1} \\ (x_{i1} - \alpha_{i-1})^2 & (x_{i2} - \alpha_{i-1})^2 & \cdots & (\alpha_i - \alpha_{i-1})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (x_{i1} - \alpha_{i-1})^{q_i} & (x_{i2} - \alpha_{i-1})^{q_i} & \cdots & (\alpha_i - \alpha_{i-1})^{q_i} \end{pmatrix} (i = 2, 3, \dots, r).$$

Notice that in equation (3.29), the determinant of the information matrix can be maximized by maximizing each multiplication component on the right hand side separately. The first multiplication component is proportional to the determinant of the information matrix for a polynomial regression model of degree q_1 , and it is maximized

at the support points of the D-optimal design on $[a, \alpha_1]$. For all other components, it follows

$$\begin{aligned} & \prod_{j=1}^{q_i} \lambda(x_j) |F_i|^2 \\ &= \lambda(\alpha_i) \prod_{j=1}^{q_i-1} \lambda(x_{ij}) (\alpha_i - \alpha_{i-1})^2 \prod_{j=1}^{q_i-1} (x_{ij} - \alpha_{i-1})^2 (x_{ij} - \alpha_i)^2 \prod_{1 < s < t \leq q_i-1} (x_s - x_t)^2 \end{aligned} \quad (3.30)$$

When Condition 3.2 is satisfied, the determinant of the information matrix for the individual polynomial regression of degree q_i on the interval $[\alpha_{i-1}, \alpha_i]$ is given by

$$\begin{aligned} \prod_{j=1}^{q_i+1} \lambda(x_j) |F|^2 &= \prod_{j=1}^{q_i+1} \lambda(x_j) \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha_{i-1} & x_{i1} & \cdots & \alpha_i \\ \vdots & \vdots & & \vdots \\ \alpha_{i-1}^{q_i} & x_{i1}^{q_i} & \cdots & \alpha_i^{q_i} \end{vmatrix}^2 \\ &= \lambda(\alpha_{i-1}) \lambda(\alpha_i) \prod_{j=1}^{q_i-1} \lambda(x_{ij}) (\alpha_{i-1} - \alpha_i)^2 \prod_{j=1}^{q_i-1} (\alpha_{i-1} - x_{ij})^2 (x_{ij} - \alpha_i)^2 \prod_{1 < s < t \leq q_i-1} (x_{is} - x_{it})^2 \end{aligned} \quad (3.31)$$

Comparing the equation (3.30) and (3.31), it is easy to see that (3.30) is maximized at the same points at which the equation (3.31) is maximized. Consequently, the support points of the D-optimal design for the entire segmented polynomial model are the same set of support points of the individual D-optimal designs for individual polynomial models on corresponding sub intervals. The theorem is proved.

Remarks. Note that Condition 3.2 is very general. It is dependent on the function $\lambda(x)$ as well as the design interval. For instance, it is known that the D-optimal design for a polynomial model of degree q with homoscedastic variance ($\lambda(x) \equiv 1$) has exactly $q+1$ support points and two of them are on the boundaries of the design interval. So Condition 3.2 is always satisfied when $\lambda(x) \equiv 1$. However, for any other $\lambda(x) \neq 1$, the condition is also dependent on the design interval.

We now give examples to illustrate the application of Theorem 3.2. For homoscedastic segmented polynomial models, the following theorem along with Theorem 3.2 shows that the D-optimal designs can be obtained analytically.

Theorem 3.3 (Fedorov 1972) if $E(y | x) = \sum_{\alpha=1}^m \theta_{\alpha} f_{\alpha}(x)$, $Var(y | x) = \sigma^2$, where

$f_{\alpha}(x) = x^{\alpha-1}$ ($\alpha = 1, 2, \dots, m$), and $-1 \leq x \leq 1$; Then the D-optimal design is unique and is concentrated at m points with equal weights $p_i = m^{-1}$. Moreover, the points are the roots of the polynomial $(1 - x^2)P'_{m-1}(x)$, where $P_m(x)$ is the m^{th} Legendre polynomial;

Remarks. Theorem 3.3 states the design problem on the restricted interval $[-1, 1]$; however, it is also applicable on any other bounded intervals. For an arbitrary bounded interval, the support points can be obtained by a simple linear transformation from the support points on $[-1, 1]$.

Example 3.1 Consider a segmented regression model with a single join point α , $\alpha \in (-1, 1)$ and

$$E(Y | x) = \theta_{10} + \theta_{11}x + \theta_{12}x^2 + \theta_{21}T_1(x - \alpha) + \theta_{22}T_2(x - \alpha) + \theta_{23}T_3(x - \alpha),$$

$Var(y | x) = \sigma^2 \lambda^{-1}(x)$, where $\lambda(x) = 1$. The problem is to find the D-optimal design on the interval $[-1, 1]$.

Analytical Solution:

The segmented regression model consist of two piecewise polynomial models of degree 2 and of degree 3 on intervals $[-1, \alpha]$ and $[\alpha, 1]$ respectively. So first we find the

D-optimal design for each polynomial model separately. Once we have the two individual D-optimal designs, we can then derive the D-optimal design for the segmented model on the entire design interval by Theorem 3.2.

Based on the assumptions in this example, it is easy to determine the locations of the support points. By Theorem 3.3, the support points for polynomial model of degree m on the interval $[-1, 1]$ are the roots of the equation:

$$(1 - x^2)P'_{m-1}(x) = 0, \quad (3.32)$$

where $P_m(x)$ is the m^{th} Legendre polynomial. The Legendre polynomial can be constructed using the three term recurrence relations

$$mP_m = (2m - 1)xP_{m-1} - (m - 1)P_{m-2}$$

with $P_0 = 1$ and $P_1 = x$. It is easy to solve the equation (3.32) for the support points.

Since the support points are defined on interval $[-1, 1]$, a simple linear transform on these support points have to be applied in order to get the support points for the same model defined on any arbitrary intervals, say $[a, b]$. The linear transform is:

$$x' = a + \frac{x - (-1)}{1 - (-1)}(b - a)$$

Moreover, it is not difficult to find the close-form roots of a Legendre polynomial equation using MATHEMATICA. We list the D-optimal design support points for polynomials of degree m , ($m=1, 2, 3$), in Table 3.1.

Now, getting back to the problem, the segmented polynomial consists of a polynomial of degree 2 and a polynomial of degree 3. Using the results in Table 3.1 and Theorem 3.2, we find that the 6 support points of the D-optimal design for the model with the join point α on $(-1, 1)$ are given by

$$\left(-1, \frac{-1+\alpha}{2}, \alpha, \alpha + \frac{(1-\alpha)(\sqrt{5}-1)}{2\sqrt{5}}, \alpha + \frac{(1-\alpha)(\sqrt{5}+1)}{2\sqrt{5}}, 1\right) \quad (3.33)$$

Table 3.1 Analytical support points of the D-optimal design for a homogeneous polynomial regression model of degree m on the interval $[a, b]$

Degree m	D-optimal support points x^*
1	(a, b)
2	$(a, \frac{a+b}{2}, b)$
3	$(a, a + \frac{(b-a)(\sqrt{5}-1)}{2\sqrt{5}}, a + \frac{(b-a)(\sqrt{5}+1)}{2\sqrt{5}}, b)$

Numeric Solution:

On the other hand, we calculated the D-optimal designs numerically and list the results (rounded in four decimal places) in Table 3.2.

Table 3.2 Support points of the D-optimal design for the model in Example 3.1 by numerical methods

Join Point α	Support points of D-optimal design ξ^*	$\frac{ M(\xi_N^*) }{ M(\xi_A^*) }$
-0.4	$(-1, -0.7000, -0.4, -0.0131, 0.6129, 1)$	1
-0.2	$(-1, -0.5993, -0.2, 0.1313, 0.6683, 1)$	1
0.0	$(-1, -0.5, 0, 0.2763, 0.7236, 1)$	1
0.2	$(-1, -0.3999, 0.2, 0.4212, 0.7790, 1)$	1
0.4	$(-0.9818, -0.3071, 0.4, 0.5609, 0.8314, 0.9998)$	1

Note that ξ_N^* and ξ_A^* are the D-optimal designs given by numerical solutions and analytical solutions respectively. The ratios in the last column indicate that the design ξ_N^* and the design ξ_A^* are equivalent. If we further compare the support points in Table 3.2 with the support points calculated using (3.33) – the analytical solutions, one by one

for each join point α , we can see that the analytical method and the numerical method yield identical results.

Example 3.1 shows that the D-optimal design for a homoscedastic segmented polynomial model can be obtained analytically from individual D-optimal designs for the individual polynomial models on the corresponding subintervals.

Example 3.2 Consider a quadratic-quadratic segmented polynomial regression model with heteroscedastic variance:

$$E(y | x) = \eta(x) = \theta_{10} + \theta_{11}x + \theta_{12}x^2 + \theta_{21}T_1(x - \alpha) + \theta_{22}T_2(x - \alpha)$$

$$Var(y | x) = \sigma^2 / \exp(-x), \quad -1 \leq x \leq 1.$$

D) First, we find the D-optimal designs for both individual quadratic polynomials. Table 3.3 listed the D-optimal designs for both individual models with selected join points. The checking condition plots of the functions in (3.26) verify that all D-optimal designs listed in Table 3.3 are optimal. Moreover, the support points of both individual D-optimal designs include their boundaries. This implies that Condition 3.2 is satisfied. Consequently, by Theorem 3.2, we know that the support points of the overall D-optimal design must be the same support points of the two individual D-optimal designs.

Table 3.3 Support points of the individual D-optimal designs for the models in Example 3.2

Join Point α	Support points of D-optimal design ξ_1^*	Support points of D-optimal design ξ_2^*
-0.6	(-1, -0.81, -0.6)	(-0.6, -0.0459, 1)
-0.4	(-1, -0.7244, -0.4)	(-0.4, -0.1810, 1)
-0.2	(-1, -0.6396, -0.2)	(-0.2, -0.3119, 1)
0.0	(-1, -0.5616, 0)	(0, 0.4384, 1)
0.2	(-1, -0.4881, 0.2)	(0.2, 0.5604, 1)

Table 3.3 (Continued)

0.4	(-1, -0.4190, 0.4)	(0.4, 0.6776, 1)
0.5	(-1, -0.3860, 0.5)	(0.5, 0.7344, 1)
0.6	(-1, -0.3541, 0.6)	(0.6, 0.79, 1)

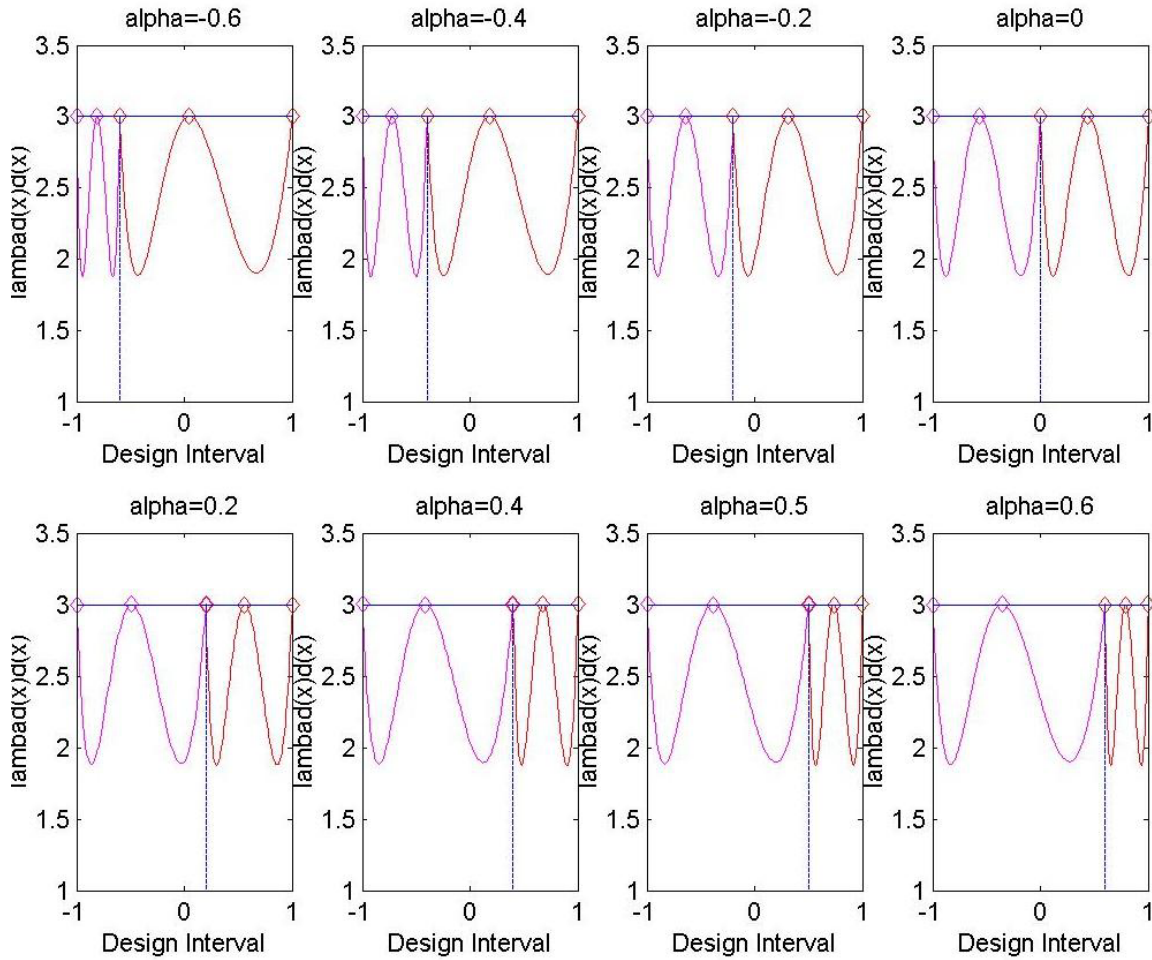


Figure 3.1. Checking condition plots verify the optimality of the D-optimal designs in Table 3.3.

II). Next consider the segmented polynomial as a single one piece, and find the D-optimal design directly over the entire interval. Table 3.4 listed the support points of the D-optimal design found numerically, and the checking condition plots in Figure 3.2 show the designs are D-optimal.

Table 3.4 Support points of the D-optimal design for the entire segmented model in Example 3.2

Join Point α	Support points of D-optimal design ξ^*	$ M(\xi^*) $
-0.6	(-1, -0.81, -0.6, 0.046, 1)	0.0009
-0.4	(-1, -0.7224, -0.4, 0.181, 1)	0.0033
-0.2	(-1, -0.6396, -0.2, 0.3119, 1)	0.0049
0.0	(-1, -0.5616, 0, 0.4384, 1)	0.0042
0.2	(-1, -0.4881, 0.2, 0.5604, 1)	0.0022
0.4	(-1, -0.4207, 0.4, 0.6770, 1)	0.0007
0.5	(-1, -0.3860, 0.5, 0.7344, 1)	0.0003
0.6	(-1, -0.3537, 0.6, 0.7900, 1)	0.0000

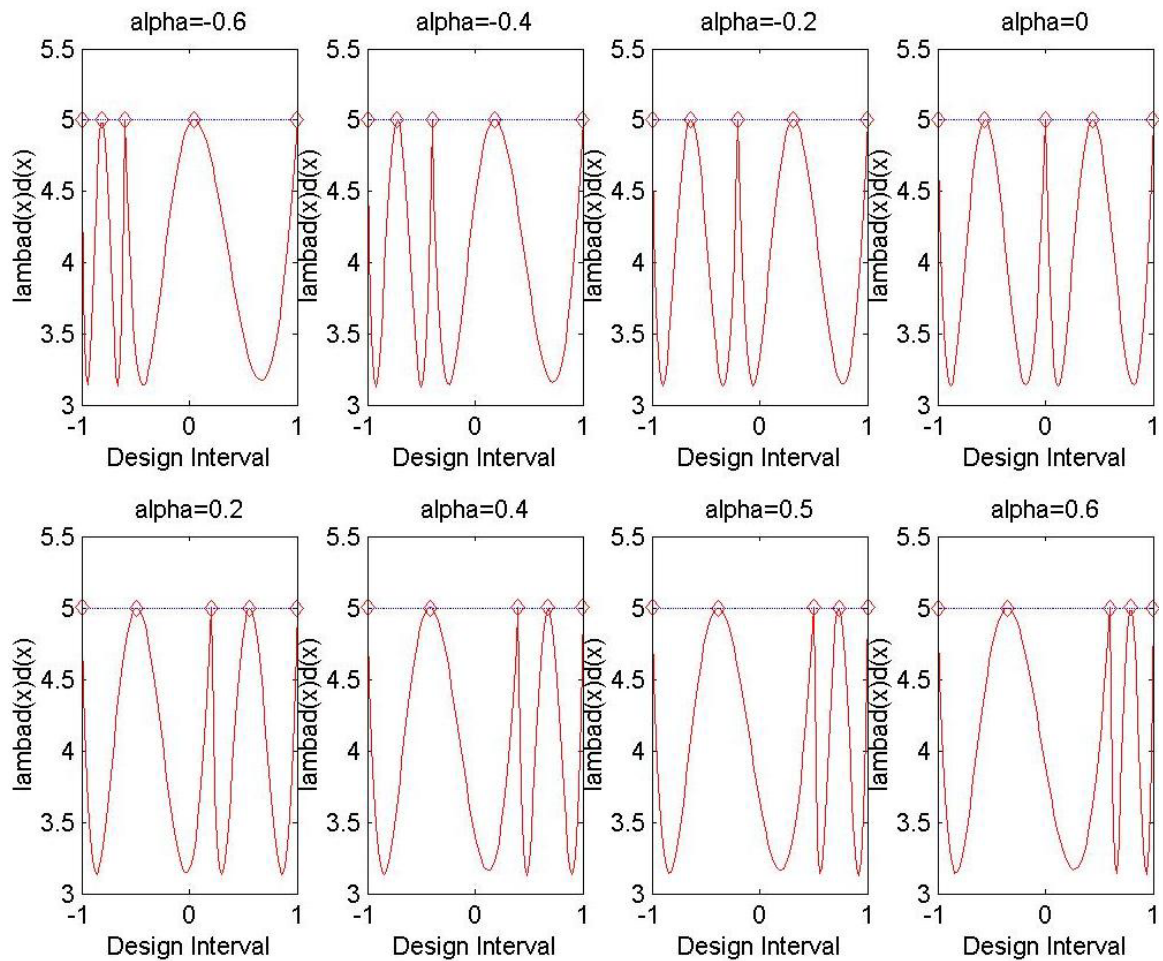


Figure 3.2. Checking condition plots verify the optimality of the D-optimal designs in Table 3.4.

III). After rounding off, it is easy to see that the support points of the two individual D-optimal designs in Table 3.3 for each join point α are identical to those in the corresponding overall D-optimal design in Table 3.4. This finding is agreed with Theorem 3.2.

3.3.2 The function $\eta(x)$ is differentiable at the known join points

If the regression function $\eta(x)$ is also differentiable at the known join points, the D-optimal designs for the segmented models can be found numerically. For example, if $\eta(x)$ is continuous and once differentiable at all join points, the regression $\eta(x)$ has the following form,

$$\eta(x) = \sum_{i=0}^{q_1} \theta_{1i} x^i + \sum_{j=2}^{q_2} \theta_{2j} T_j(x - \alpha_1) + \dots + \sum_{j=2}^{q_r} \theta_{rj} T_j(x - \alpha_{r-1}) \quad (3.34)$$

There are $k = 1 + q_1 + q_2 + \dots + q_r - (r - 1)$ unknown parameters. The following example shows efficient D-optimal designs for segmented polynomial models of this type can be found numerically. However, a relation between the overall D-optimal and the individual D-optimal designs was not demonstrated.

Example 3.3 Consider a quadratic-quadratic segmented regression

model $\eta(x) = \theta_{10} + \theta_{11}x + \theta_{12}x^2 + \theta_{22}T_2(x - \alpha)$, $\lambda(x) = \exp(-x)$, $-1 \leq x \leq 1$. The D-optimal designs found for a series of α , are tabulated in Table 3.5.

Table 3.5 Support points of the D-optimal design for the model in Example 3.3

Join Point α	Support points of D-optimal design ξ^*	$ M(\xi^*) $
-0.6	(-1, -0.7194, -0.0141, 1)	0.0590
-0.4	(-1, -0.6299, 0.0811, 1)	0.1341
-0.2	(-1, -0.5572, 0.1778, 1)	0.1843

Table 3.5 (Continued)

0.0	(-1, -0.4953, 0.2804, 1)	0.1815
0.2	(-1, -0.4402, 0.3920, 1)	0.1327
0.4	(-1, -0.3890, 0.5158, 1)	0.0685
0.5	(-1, -0.3641, 0.5833, 1)	0.0412
0.6	(-1, -0.3394, 0.6553, 1)	0.0207

Again, the D-optimality of these designs was checked by the General Equivalence Theorem as shown in the plots in Figure 3.3.

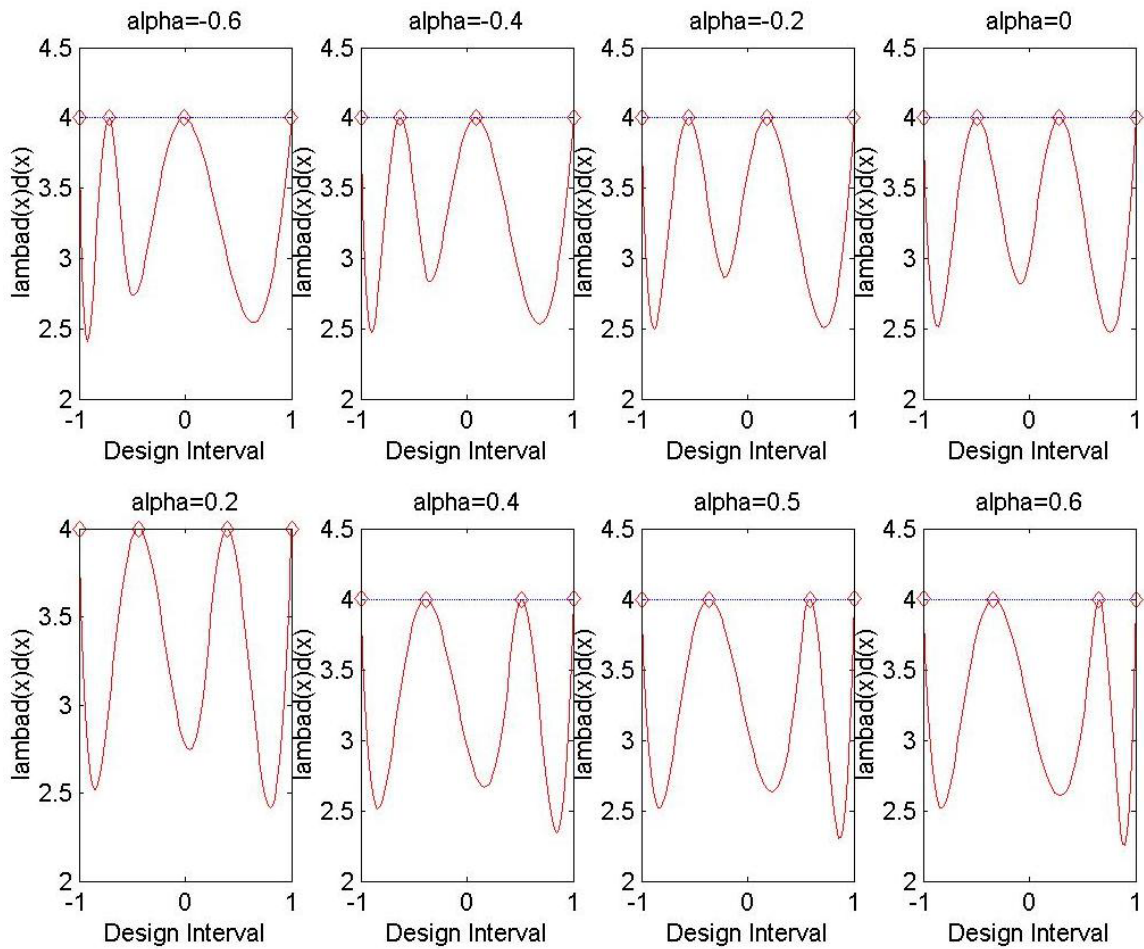


Figure 3.3. Checking condition plots verify the optimality of the D-optimal designs in Table 3.5.

In this example, the segmented polynomial model also consists of two individual quadratic polynomial models on the subintervals. As we can see in Table 3.5, for each known join point α , the support points of the D-optimal design consists of 4 support points, and these support points are different from those of corresponding individual D-optimal designs on the subintervals as listed in Table 3.4.

3.4 Optimal Designs When Join Points Have to Be Estimated

When the join points in a segmented polynomial model are unknown and have to be estimated from the data, the model becomes nonlinear in the join points α . The methods for fitting a segmented polynomial with join points have to be estimated has been discussed, for example see Gallant et al. (1973). The fundamental idea behind fitting nonlinear models is to linearise the model using Taylor series with respect to the unknown parameters and then to proceed iteratively using linear least squares. The estimates obtained in this way are referred to as quasi-least square estimates in some sense. It has been shown that the quasi-least squares estimates are consistent estimates of the model parameters. So essentially, the optimal design approaches for segment polynomial models with unknown join points are similar to those for segmented polynomial models with known join points, except that their information matrices are slight different.

The information matrix for a nonlinear model usually has the unknown model parameter in it. Consequently, the optimal design criterion is a function of both the design and the unknown model parameters. One simple approach is plugging the best guess on the unknown parameters in the criterion function, which is termed as “locally” optimal

designs. If a prior joint distribution on the unknown parameters can be established, the Bayesian approach is a more robust approach. When there are multiple unknown parameters in the regression model, a concrete joint prior distribution is usually difficult to be established. Often, ranges of possible values of the unknown parameters are easier to be obtained than joint distributions. In that case, the minimax optimal design is appropriate.

A Minimax approach is to search the optimal design by minimizing the maximum of an asymptotic loss function, where the maximum is taken over all plausible locations of the unknown parameters. Formally, the Minimax D-optimal criterion is defined as:

$$\text{Min}_{\xi} \max_{\alpha} |M^{-1}(\xi, \alpha)|,$$

where α is the vector of unknowns parameters, taking over all possible values.

Bayesian optimal designs and minimax optimal designs are robust comparing to the locally optimal designs. However, they have to be found numerically most of the time. The following discussion in this section is solely on the locally D-optimal designs for segmented polynomial models with unknown join points, for which we may find the locally D-optimal designs analytically in some cases.

3.4.1 $\eta(x)$ is continuous and exactly once differentiable at all join points

If the regression function $\eta(x)$ is continuous and has exactly one derivative at all join points, it has the form as in (3.34). With the join points have to be estimated, $\eta(x)$ is a nonlinear function in the following parameters:

$$\theta^T = (\theta_{10}, \theta_{11}, \dots, \theta_{1q_1}, \alpha_1, \theta_{22}, \theta_{23}, \dots, \theta_{2q_2}, \alpha_2, \dots, \alpha_{r-1}, \theta_{r2}, \theta_{r3}, \dots, \theta_{rq_r})$$

Then,

$$\begin{aligned}
f^T(x) = \nabla^T \eta(x, \theta) = & (1, x, \dots, x^{q_1}, \\
& - \sum_{i=2}^{q_2} \theta_{2i} T_{i-1}(x - \alpha_1), T_2(x - \alpha_1), \dots, T_{q_2}(x - \alpha_1), \\
& \vdots \\
& - \sum_{i=2}^{q_r} \theta_{ri} T_{i-1}(x - \alpha_{r-1}), T_2(x - \alpha_{r-1}), \dots, T_{q_r}(x - \alpha_{r-1}))
\end{aligned} \tag{3.35}$$

It is easy to see that $f(x)$ in (3.35) can be transformed to the following form

$$\begin{aligned}
U^{-1}(\theta) f^T(x) = & g(x)^T \\
& (1, x, \dots, x^{q_1}, T_1(x - \alpha_1), T_2(x - \alpha_1), \dots, T_{q_2}(x - \alpha_1), \\
& \dots, T_1(x - \alpha_{r-1}), T_2(x - \alpha_{r-1}), \dots, T_{q_r}(x - \alpha_{r-1}))^T
\end{aligned} \tag{3.36}$$

where $U(\theta)$ is an upper triangle matrix and its non-trivial elements are the unknown model parameters. The right hand side of the equation, $g(x)$ is the same vector of functions for the model with regression function (3.28) for join points are known cases

Theorem 3.4 For segmented polynomial model with regression function (3.34), if Condition 3.1 is satisfied, or in other word, the locally D-optimal design has exactly $m = 1 + \sum q_i$ support points, then the locally D-optimal design is the same as the D-optimal design for the segmented polynomial model with regression function (3.28) assuming the join points are the same in both cases.

Proof: when Condition 3.1 is satisfied, by corollary 3.2, the determinant of the information matrix for both models has the same form

$$|M(\xi, \theta)| = \prod_{i=1}^n w_i \prod_{i=1}^n \lambda(x_i) |F|^2$$

Here, using the transformation (3.36) we obtain

$$\begin{aligned}
F &= (f(x_1), f(x_2), \dots, f(x_n)) \\
&= (U(\theta) f'(x_1), U(\theta) f'(x_2), \dots, U(\theta) f'(x_n)), \\
&= U(\theta) F'
\end{aligned}$$

where F' is the design matrix for the model with regression function (3.28) when join point are known. It is clear that the locally D-optimal criterion for the model with regression function (3.34) is proportional to the D-optimal criterion function for the model with regression function (3.28), thus, the locally D-optimal design for (3.34) must be the same as the D-optimal design for (3.28) if both models have the same set of join points.

In addition, it is also conceivable that there exists a one-one relationship between locally D-optimal designs for segmented polynomial models with unknown join points and D-optimal designs for segmented polynomial models with known join points under certain conditions.

Dette et al. (2006) has shown explicit locally D-optimal designs for homoscedastic spline functions with exactly one continuous derivative at the knots (join points). It is easy to derive the same result from Theorem 3.2, Theorem 3.3, and Theorem 3.4 for $\lambda(x) \equiv 1$.

3.5 Applications in Agriculture Studies

In this section, we present the optimal designs for segmented polynomial models in an agriculture study. Shuai et al. (2003) fitted segmented regression models for reflecting soil Manganese (Mn) changes under different soil pH conditions. In their paper, a three-segment linear-linear-linear model and a two-segment linear-quadratic model were established, and both were statistically good fit for the data. The fitted linear-linear-linear polynomial model was:

$$E(Mn|pH) = \begin{cases} -0.5175pH + 2.7470 & pH \leq 5.1281 \\ -0.1040pH + 0.6265 & 5.1281 \leq pH \leq 5.9823 \\ 0.004349 & pH \geq 5.9823 \end{cases} \quad (3.37)$$

And the fitted linear-quadratic model was:

$$E(Mn|pH) = \begin{cases} -0.5050pH + 2.6886 & pH \leq 5.1000 \\ 0.0735pH^2 - 0.9286pH + 2.9372 & pH \geq 5.1000 \end{cases} \quad (3.38)$$

The error term was assumed to be of a constant in both models. We checked the data for constant variance. The assumption was found valid. That is $\lambda(pH) = 1$ in the variance function in the definition (3.4).

If further investigation on Manganese changes to pH values is of interest, e.g. to confirm the above established models or to precisely estimate the model, a well-designed experiment is highly desirable. Suppose we are planning for the next study, a few issues should be taken into account: 1) which model we will design for? 2) Is (are) the join point(s) to be estimated from the data? 3) What if the experimenter prefers one model to the other? We respond to each of these questions with the discussions that follow.

3.5.1 The join points are known

If there is no need to estimate the join points from the data, the D-optimal designs for both models can be easily obtained. The regression functions are continuous on the entire interval $- [4.7, 6.3]$, they are not differentiable at the join points, and they are homoscedastic. By Theorem 3.2 and Theorem 3.3, we obtain the D-optimal designs as follows:

$$\xi_{D3.37}^* = \begin{pmatrix} 4.7 & 5.1281 & 5.9823 & 6.3 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \text{ for model 3.37,}$$

$$\xi_{D3.38}^* = \begin{pmatrix} 4.7 & 5.1 & 5.7 & 6.3 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \text{ for model 3.38.}$$

A characteristic in the D-optimal designs is that the boundary points and the join points are included in the support points. Also, as a comparator, we should mention here that the original design ξ^0 used in the paper was:

$$\xi^0 = \begin{pmatrix} 4.7 & 4.9 & 5.1 & 5.3 & 5.5 & 5.7 & 5.9 & 6.1 & 6.3 \\ 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 \end{pmatrix}.$$

This is obviously an empirical design and is not optimal. The efficiencies of ξ^0 to $\xi^*_{D3.37}$ and $\xi^*_{D3.38}$ were

$$\left\{ \frac{|M_{3.37}(\xi^0)|}{|M_{3.37}(\xi^*_{D3.37})|} \right\}^{1/4} = 0.479 \text{ and } \left\{ \frac{|M_{3.5.2}(\xi^0)|}{|M_{3.5.2}(\xi^*_{D3.5.2})|} \right\}^{1/4} = 0.502 \text{ respectively.}$$

3.5.2 The join points have to be estimated from the data

In the original paper, the join points along with the model parameters were estimated from the data using a non-linear regression method. In order to use the optimal design methods developed earlier, we need to convert the regression functions (3.37) and (3.38) to be of the following simpler forms:

$$E(Mn|pH) = \theta_0 + \theta_1 pH + \theta_2 T_1(pH - \alpha_1) + \theta_3 T_1(pH - \alpha_2) \quad 4.7 \leq pH \leq 6.3 \quad (3.39)$$

$$E(Mn|pH) = \theta_0 + \theta_1 pH + \theta_2 T_1(pH - \alpha_1) + \theta_3 T_2(pH - \alpha_1) \quad 4.7 \leq pH \leq 6.3 \quad (3.40)$$

If the join points have to be estimated, the models are non-linear in the parameters:

$$(\theta_0, \theta_1, \theta_2, \theta_3, \alpha_1, \alpha_2)$$

The gradient of the regression functions (3.39) and (3.40) with respect to the unknown parameters by direct partial differentiations are given below:

$$f^T(pH) = (1 \quad pH \quad T_1(pH - \alpha_1) \quad -\theta_2 I_{1+} \quad T_1(pH - \alpha_2) \quad -\theta_3 I_{2+}) \quad (3.41)$$

$$f^T(pH) = (1 \quad pH \quad T_1(pH - \alpha_1) \quad -\theta_2 I_{1+} \quad T_2(pH - \alpha_1) \quad -2\theta_3 T_1(pH - \alpha_1)) \quad (3.42)$$

$$\text{where } I_{1+} = \begin{cases} 0 & pH < \alpha_1 \\ 1 & pH \geq \alpha_1 \end{cases} \text{ and } I_{2+} = \begin{cases} 0 & pH < \alpha_2 \\ 1 & pH \geq \alpha_2 \end{cases}$$

Notice that (3.41) and (3.42) have unknown parameters, thus the optimal criterion functions would have these unknown parameters as well. As an appropriate prior joint distribution or a range of possible values of the unknown parameters cannot be derived from the original paper, we consider locally D-optimal designs here. It is easy to show that the best guesses for the unknown parameters in (3.39) and in (3.40) are

$$\left\{ \begin{array}{l} \theta_0 = -0.5175 \\ \theta_1 = 2.7470 \\ \theta_2 = 0.4135 \\ \theta_3 = 0.1040 \\ \alpha_1 = 5.1281 \\ \alpha_2 = 5.9823 \end{array} \right. \text{ and } \left\{ \begin{array}{l} \theta_0 = 2.6886 \\ \theta_1 = -0.5050 \\ \theta_2 = 0.3261 \\ \theta_3 = 0.0735 \\ \alpha_1 = 5.1 \end{array} \right. , \text{ respectively.}$$

The locally D-optimal design for the model with regression function (3.39) is calculated numerically below:

$$\xi_{D_{3.39}}^* = \begin{pmatrix} 4.7006 & 5.1 & 5.1001 & 5.9771 & 5.9823 & 6.2981 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

Interestingly, the above design implies the join points are hot spots if those join points are to be estimated from the data. Comparing the empirical design to $\xi_{D_{3.39}}^*$, the design efficiency is

$$D_{eff} = \left\{ \frac{|M(\xi^0)|}{|M(\xi_{D_{3.39}}^*)|} \right\}^{1/6} = 0.865.$$

Likewise, the locally D-optimal design found numerically for the model with regression function (3.40) is:

$$\xi_{D_{3.40}}^* = \begin{pmatrix} 4.7 & 5.1 & 5.1104 & 5.7242 & 6.2997 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{pmatrix}.$$

Coincidentally, when comparing $\xi_{D_{3,40}}^*$ and ξ^0 , we found they were equally efficient. In figure 3.4, we plot the dispersion of the predicted response variable Mn along with the support points of ξ^0 and $\xi_{D_{3,40}}^*$

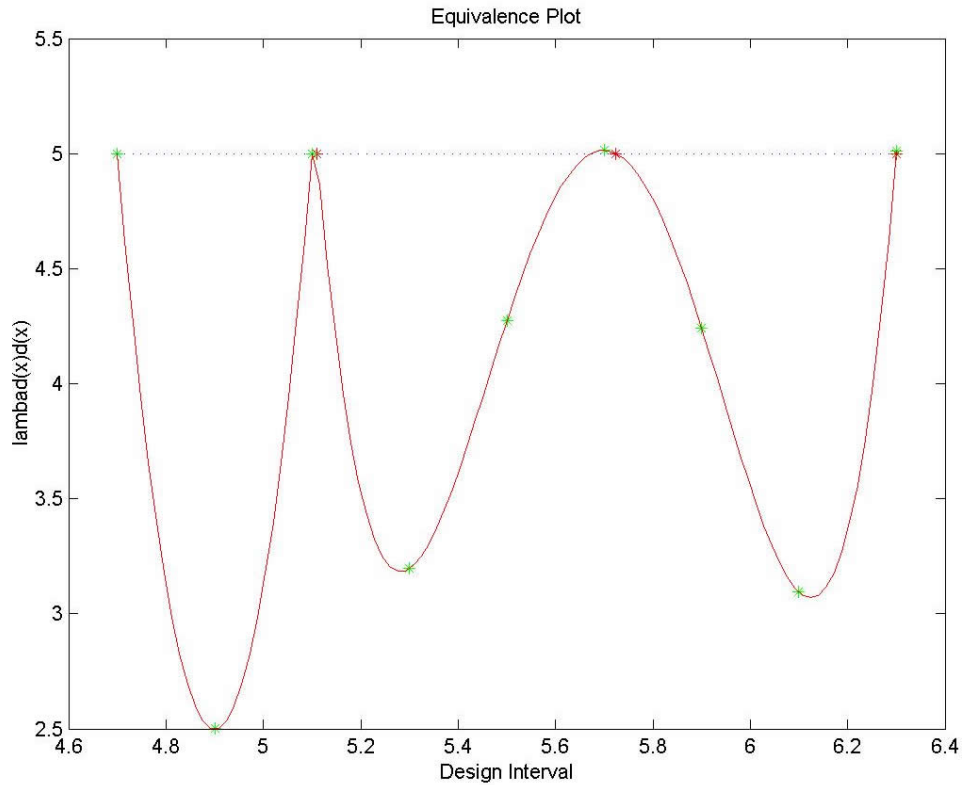


Figure 3.4. Checking condition plots verifies the optimality of the locally D-optimal design for the linear-quadratic polynomial model.

On Figure 3.4, the support points of $\xi_{D_{3,40}}^*$ are the stars in red, and the support points of ξ^0 are stars in green. ξ^0 has more points than $\xi_{D_{3,40}}^*$, which means $\xi_{D_{3,40}}^*$ is still a better design than ξ^0 although they are equally efficient. Indeed, ξ^0 covers $\xi_{D_{3,40}}^*$, however, additional support points don't increase design efficiency in this case.

3.5.3 Robust D-optimal designs and Multiple-objective D-optimal designs

In the end of the original paper, the authors concluded that based on the goodness of fit tests both models fit the data well. Although they prefer one model to the other from realistic perspectives, challenges do arise in planning of efficient experiments, such as on which model should the optimal design be based? On the other hand, the authors also modeled the relationships between pH values and other soil nutrient parameters, such as the amount of Calcium and Protactinium. This also induced a multiple-objective optimal design problem, e.g., if the experimenter wants an optimal design for modeling all the relationships together.

The optimal design methods we have been talking about are model based. The fundamental assumption is that the regression model is known when designing experiments. If the underlying model is misspecified, the resulting optimal designs can be completely meaningless. As we can see that the locally D-optimal designs $\xi_{D3.39}^*$ and $\xi_{D3.40}^*$ are totally different, and they don't even have the same number of support points. On the other hand, the problem comes from the multiple-objective optimal design is essentially similar to the model misspecification problem. In light of these similar problems, here we introduce a model robust optimal design approach as follows.

The robust optimal design criterion function is straightforward. Denote ϕ_1 as the D-optimal design criterion function for the linear-linear-linear polynomial model, and ϕ_2 as the D-optimal design criterion function for the quadratic-quadratic model. Let γ ($0 \leq \gamma \leq 1$) reflect the experimenter's preference or confidence about the linear-linear-linear polynomial model. For example, if the experimenter believes both models are equally important, he may set $\gamma = 1/2$. If the experimenter has no interest in the quadratic-quadratic model, he can set $\gamma = 1$. Thus, for seeking a design that allows good

estimates of all the parameters in both models, it is natural we propose the following optimality criterion:

$$\phi(\xi) = \gamma\phi_1(\xi) + (1 - \gamma)\phi_2(\xi) \quad (3.43)$$

The fundamental idea underlying this criterion has been introduced in several papers, for different but essentially similar problems, such as, optimal robust designs, multiple-objective optimal designs. For example, readers who want more details are referred to Zhu et al. (1998) and Dette (1992).

Most of the time, the optimal design ξ^* that minimizes (3.43) has to be found numerically. Theoretically, using one of the optimization methods we have mentioned before, it is not difficult to find the best robust designs or the best multiple-objective designs that a numerical approach can find. Since there is no analytical derivative to demonstrate, we would not list the results here.

4 Implementation of Web-Based Optimal Design Software

In this chapter, we discuss the implementation of our web-based optimal design computer programs. In fact the web-based optimal design computer programs are integrated on this website - www.optimal-design.org. We will cover how the entire system was designed and how optimal design programs were integrated on the website.

The original idea of construction such a website comes from an unpublished research plan. This project was funded by NIHGMS. In this chapter we are intended to document the implementation details so that this documentation may serve as a handbook for future developers.

In order to present optimal design method and results in computer software, we may have two options. One is a stand-alone application. The other is the so-called web based interactive application. In recent years, the World Wide Web has become an increasingly important tool to share information and knowledge in various fields. It can provide an excellent environment for deploying statistical tools because of its easy accessibility. One major advantage by deploying web based statistical programs is that statistical design and analysis can be performed anywhere an Internet connection is available. These features along with many others makes it is very attractive to us to set up a web server and share our work and present our findings on it.

Nevertheless, there is no question that such a web site could be set up using different strategies and technologies. However, due to space limitations, I would only

discuss our design and approaches. I will not make arguments, whether why one technology is preferred to another.

4.1 System Overview

The web-based optimal design computer software system uses client-server architecture. Input data are supplied to the optimal design programs on the server side via a user-friendly interface on the web pages. Users input data on web pages, and then the data is packaged into HTTP format and sent to the Internet Information Server (IIS) at the server side. When IIS receives input data, it then transfer the data to the Web Application for further processing. The Web Application runs on IIS platform and its interface to the user are the web pages. The Web Application preprocesses a user's input and establishes a connection between the IIS and the Matlab Web Server. Once the connection is established, a user's input is then sent to the Matlab Web Server, where the user's input will be transferred to the right Optimal Design Program for final processing. The results, which typically are optimal designs along with checking condition plots, are then sent back to the user's interfaces on the web page in a reverse manner. Since a picture worth a thousand words, the following diagram may illustrate the data flow discussed above better.

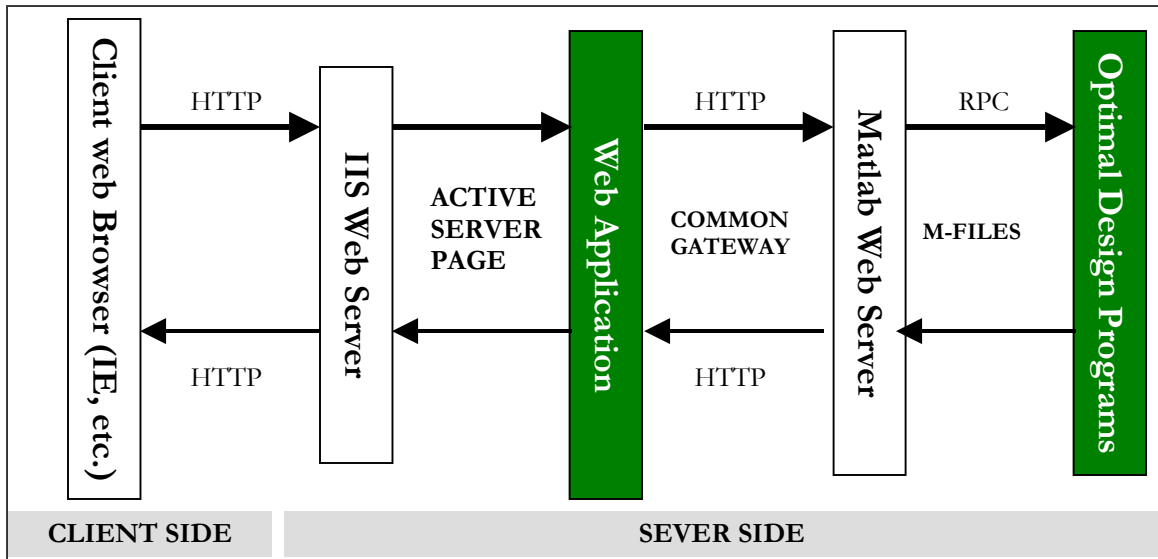


Figure 4.1. System architecture and data flow illustration.

4.2 Technologies and Development Tools

There are five components in Figure 4.1. However, software programming is needed for only two of them, the Web application and the Optimal Design Programs. The other three components are commercial available software, which can be set up by an experienced system administrator.

Web browsers usually come with any personal computer systems nowadays. The Internet Information Server comes with Microsoft Windows[®] operating system, and the Matlab Web Server is available from MathWorks. Installation and configuration for these systems have to be done by specific specialists. Interestingly, MathWorks has discontinued supporting its Matlab Web Server product. Instead, the product has been replaced with Matlab[®] Distributed Computing Engine. For more details regarding these products as well as their installation and configuration procedures, the user may refer to

Microsoft TechNet Technical References on IIS and MathWorks Distributed Computing Engine System Administrator's Guide.

For Internet users, the interfaces to the optimal design software are the web pages on which users input their requests and get results in HTML format. The Web Application interacts with the Matlab Server exchanging input and output since optimal designs are actually calculated by the Optimal Design Programs. The Web Application is developed using Microsoft ASP.NET platform and C Sharp programming language. Microsoft ASP.NET is a technology for creating dynamic web applications. ASP.NET pages (Web Forms) are compiled, providing better performance than with scripting languages. When building these pages, one can use ASP.NET server controls to create common UI (User Interface) elements and program them for common tasks. These controls allow one to build rapidly a Web Form out of reusable built-in or custom components, simplifying the coding of a page. ASP.NET provides a programming model and infrastructure to make creating scalable, secure and stable applications faster, and easier than with previous web technologies. The best development tool is Visual Studio .NET[®]. Its excellent integrated development environment makes developers' lives much easier.

The Optimal Design Programs are written in MATLAB[®]. These programs can be considered as a computing engine. All optimal design requests are processed by this engine. MATLAB[®] is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. The Optimal Design Software consists of a class of individual Matlab programs. These individual

programs are Model-based optimal design programs. That means each individual optimal design program is just computing optimal designs for a specific class of statistical models. These programs can be developed by anyone who specializes in optimal design and Matlab programming. With slight modifications, these programs can be transferred to be deployed on the web server, and they become web-based programs.

4.3 Some Technical Details

In this section, we explain some key issues in the implementation. Such as: 1) how do we create user's interfaces or web pages? 2) How do we pack user's input to Matlab server and optimal design programs for computation? 3) How dose the system output resulting optimal designs on user's interfaces or web pages? These issues will be addressed in the following discussions.

4.3.1 Creating User's Interfaces

The user's interface is a set of web pages. When a user visits the website, the website automatically delivers web pages to the web browser on the user's computer in HTML (Hyper Text Markup Language) format. These HTML web pages are interpreted by the web browser and are displayed in the web browser as texts and pictures as well as many other kinds of elements called web controls.

Once, when advanced tools were not available, web developers had to be very familiar with the HTML language. Now, there are many tools that allow developers to design and create web pages quickly. One such tool is the Visual Studio.Net from Microsoft. Visual Studio.Net has most of commonly used web controls built in its web form toolbox.

Through Figure 4.2, we can get some conception of how to create and design web pages with Visual Studio.Net. On the left of the figure, the items in the toolbox are the web controls. The window in the middle is the designing area. So basically, one drags web controls that are needed from the toolbox and drops them in the designing window. For each web control, one can then set its properties, such as font, text size, color, and position. By this way – dragging and dropping, one can develop web pages quickly.

With Visual Studio.NET, one does not need to know much about HTML. However, Visual Studio.Net does offer another option for advanced web developers. That is, one could switch to HTML view of the web design. By doing that, one can see the automatically generated HTML code by Visual Studio.Net as per the web design. This allows advanced users do some finer tune-ups on their web designs by editing the HTML code.

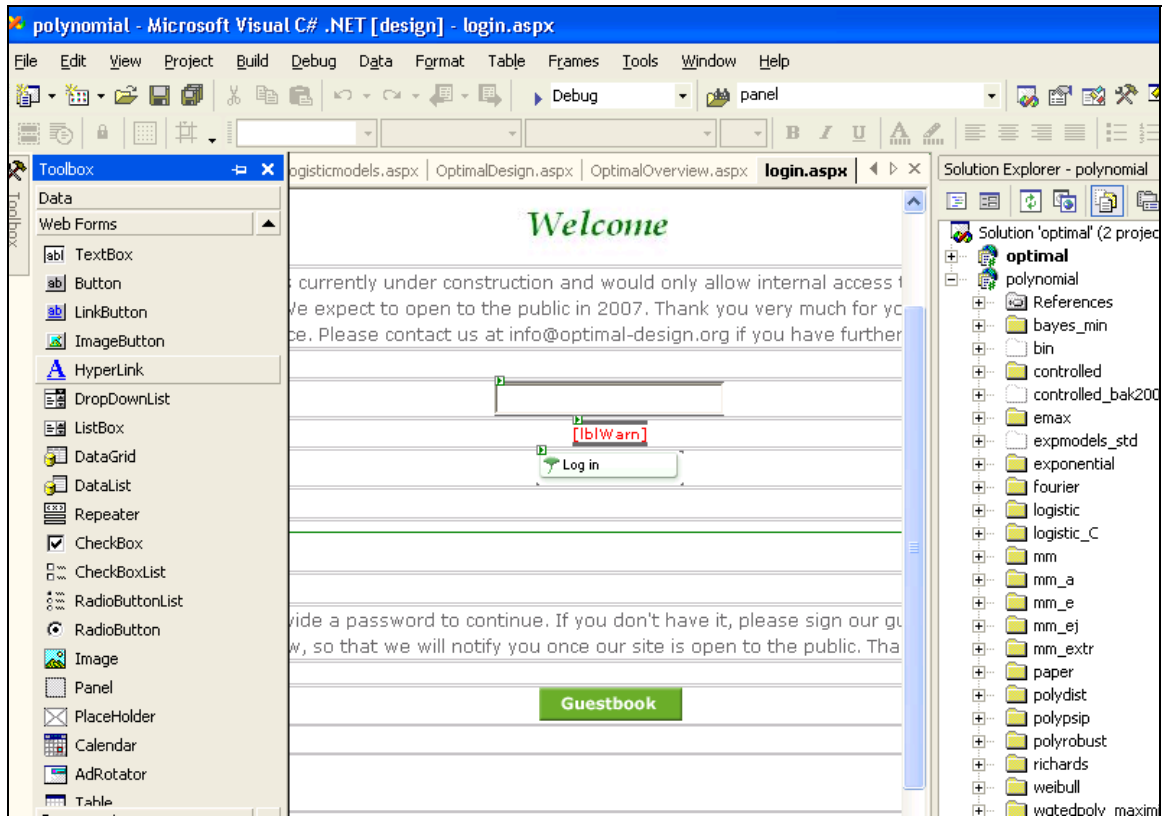


Figure 4.2. Designing a web page using the Visual Studio.NET Integrated Development Environment.

4.3.2 Data Communication between Servers

Another key technology to be discussed is how we fulfill the data communication between the Internet information server and the Matlab web server. In fact, the data communication happens between the web applications and the optimal design programs. In our case, the web application takes control of all communication processes.

The web application is a set of C sharp programs running behind the web pages. It takes user's input on the web pages, and then sends the input to optimal design programs for computation. There are a couple of techniques that can be used to accomplish this task. One way is to use Remote Procedure Call (RPC); the other one is to use Hyper Text

Transfer Protocol (HTTP). The former requires more programming. The later is straightforward and efficient. We choose HTTP as it eases implantation tasks.

Table 4.1 is the C Sharp code we used. This piece of code is the most important component in the web application. Based on the ASP.Net framework infrastructure, C Sharp programming is object-orient programming. The ASP.Net framework provides a number of fundamental Classes. The key component in this code is an ASP.Net built-in Class – HttpWebRequest, which has everything one need to facilitate an Internet HTTP communication between two applications or servers. It is not difficult to understand the code, as the C Sharp programming language is close to human language and is self explained.

Table 4.1 C Sharp Data Communication Code

```
System.Text.ASCIIEncoding encoding=new System.Text.ASCIIEncoding();

// preparing data to be transferred
string postData="mlmfile=D_Optimal";

postData+="&tbxN="+this.tbxN.Text.ToString();

postData+="&tbxalpha_l="+this.tbxalpha_l.Text.ToString();
postData+="&tbxalpha_u="+this.tbxalpha_u.Text.ToString();

postData+="&tbxbeta_l="+this.tbxbeta_l.Text.ToString();
postData+="&tbxbeta_u="+this.tbxbeta_u.Text.ToString();

postData+="&tbxlb="+this.tbxlb.Text.ToString();
postData+="&tbxub="+this.tbxub.Text.ToString();

postData+="&tbxWeightlb="+this.tbxWeightlb.Text.ToString();

//encoding data
byte[] data = encoding.GetBytes(postData);

//preparing a HTTP request
//server address, communication type, econding format

System.Net.HttpWebRequest myreq=(System.Net.HttpWebRequest)
    System.Net.HttpWebRequest.Create("http://www.optimal-design.org/optimal/polynomial/matweb.exe");
```

Table 4.1 (Continued)

```
myreq.Method="POST";
myreq.ContentType="application/x-www-form-urlencoded";
myreq.ContentLength=data.Length;

//initializing a sending request
System.IO.Stream stream=myreq.GetRequestStream();

//sending data
stream.Write(data,0,data.Length);

//closing this session
stream.Close();

string result;

//receiving results
System.Net.HttpWebResponse httpResponse = (System.Net.HttpWebResponse)myreq.GetResponse();

using (System.IO.StreamReader streamReader = new
System.IO.StreamReader(httpResponse.GetResponseStream()))
{
    result = streamReader.ReadToEnd();
    streamReader.Close();
}

//finding results from the returning stream.
this.FindValues(result);
```

4.3.3 Web based Matlab Programming

The Matlab optimal design programs compute optimal designs. Matlab is a powerful scientific computing tool. Many computationally intensive tasks are suited to Matlab as a tool. Unlike traditional programming languages, such as C, C++, and FORTRAN, Matlab has a unique feature that problems and solutions are expressed in familiar mathematical notation, which makes programming for technical computing problems easier and faster than with other traditional programming tools.

Regarding statistical computing, there are several competitive software packages available, such as SAS[®] system and Insightful S-plus[®]. Both are commercially available. Additionally R, which is the origin of S-Plus, is widely used in academia. Some software

tools are even more specialized. For example, Design-Expert[®] is a tool dedicated to finding general experiment designs. While all these software tools may be versatile and powerful, they lack the programming flexibility needed for our purpose because the model-oriented optimal designs vary in computing from one model to another.

Mathematica[®] may be a better choice because of its symbolic calculations. However, it runs in an interactive mode. It may be considered as an advanced scientific calculator, but it is not a software development tool.

It turns out Matlab is the best choice. Complete discussions about Matlab programming can be found in Matlab product documentation. Here, we only discuss important issues in our implementation with Matlab. That is, how to make a Matlab program web-based and enable it to interact with a web application?

A stand-alone Matlab program or application runs independently, within or without Matlab integrated development environment. Input and output are usually made through the underlying operating system. A significant difference in our implementation is that the input and output of the Matlab Web Server are controlled by the web application. The process of creating a web based Matlab program involves the creation of three files:

1. An input HTML file, which is a web page. It is not difficult to do this using Microsoft Visual Studio.Net as we showed this previously.
2. An output HTML file, saving and displaying the results. In our implementation, this document stores the results only, and the results are actually displayed in the input HTML file.
3. Matlab M-file(s), which is the body carrying the entire optimal design computation.

The entire process is controlled by C Sharp programs behind an input HTML web page. The following code in Table 4.2 shows the Matlab code that illustrates what changes needs to be made from a stand-alone Matlab program accordingly. Note this is not a complete executable program, but it emphasizes the differences.

Table 4.2 Matlab code showing web based programming

```
function retstr=D_Optimal(instruct, outfile)

% STEP 1
% Initialize the return string.
retstr = char("");

% STEP 2
% Set working directory.
% The variables INSTRUCT.MLDIR and INSTRUCT.MLID are provided automatically to all MATLAB Web
% Server applications that use the matweb program.
cd(instruct.mldir);

% STEP 3
% Get the HTML form input variables
global alpha_l alpha_u alpha_width
global beta_l beta_u beta_width

alpha_l = str2double(instruct.tbalpha_l);
alpha_u = str2double(instruct.tbalpha_u);

% STEP 4
% Perform your MATLAB computations, graphics file creations,
% etc., here:

% Calling Random Search Method

global Niter
Niter=1000;
random_search;
x_w=x_p;

% STEP 5
% Put variables that you want to put into your HTML output document in an output structure. You create
% an HTML output document from OUTPUT_TEMPLATE.HTML.
modelplot;
outstruct.ModelImg=ModelGraphFileName;

designpoints_D = num2str(x_w(1), '%.3f');
for i=2:N
    designpoints_D=strcat(designpoints_D, ',' , num2str(x_w(i), '%.3f'));
end
outstruct.xi = designpoints_D;

% STEP 6
% Call the function HTMLREP with the output structure you just
% created and the filename you created from OUTPUT_TEMPLATE.HTML.
```

Table 4.2 (Continued)

```
% Replace <OUTPUT_TEMPLATE.HTML> with the name of the HTML output
% file you created using OUTPUT_TEMPLATE.HTML.
% This call fills the string RETSTR to return and optionally
% writes the output as a file if a valid filename is given as the
% second argument to the present function.
```

```
templatefile = which('logisticoutput.aspx');
if (nargin == 1)
    retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
    retstr = htmlrep(outstruct, templatefile, 'debug.html');
end
```

There are basically five STEPS in this program, and there is a brief description for each step. For more details, the users are referred to the Matlab Web Server user's manual.

Matlab has appealing graphic capability as well. At step 5 in the Table 4.2, “modelplot” is calling another Matlab program for plotting a graph. The generated graph is stored on the server with its physical address in ModelGraphFileName. This physical address is then sent back in the output HTML file, and is linked on the output page to display the graph. The following Matlab code in Table 4.3 shows how to plot the mean functions of a set of logistic regressions.

Table 4.3 Matlab code for plotting graphs

```
% modleplot.m
N_plotsamples=50;
Xx=lower_bound:((upper_bound-lower_bound)/N_plotsamples):upper_bound;
handle=round(rand(1)*100);

% STEP 1: create a new figure by calling function figure
f=figure(handle);
clf

% STEP 2: call plot function to plot graph elements on the figure
cols=length(Xx);
PIx=zeros(1,cols);
for k=1:cols
    PIx(k)=1/(1+exp(-(Xx(k)-alpha_1)/beta_1));
end;

plot(Xx,PIx,'c');
```

Table 4.3 (Continued)

```
hold on

for k=1:cols
    PIx(k)=1/(1+exp(-(Xx(k)-(alpha_1+alpha_u)/2)/((beta_1+beta_u)/2)));
end;
plot(Xx,PIx,'g');
hold on

for k=1:cols
    PIx(k)=1/(1+exp(-(Xx(k)-alpha_u)/beta_u));
end;
plot(Xx,PIx,'b');
hold on

PIxPIx=0:0.1:1;
for i=1:length(x_w)/2
    XxXx=ones(1,length(PIxPIx))*x_w(i);
    plot(XxXx,PIxPIx,'r');
    plot(XxXx,0,'r^');
end;

% STEP 3: add legned on the graph
%-----
legendstr1=strcat('alpha=',num2str(alpha_1,'%2f'),' ','beta=',num2str(beta_1,'%2f'));
legendstr2=strcat('alpha=',num2str((alpha_1+alpha_u)/2,'%2f'),' ','beta=',num2str((beta_1+beta_u)/2,'%2f'));
legendstr3=strcat('alpha=',num2str(alpha_u,'%2f'),' ','beta=',num2str(beta_u,'%2f'));
legend(legendstr1,legendstr2,legendstr3,'optimal design points');

%STEP 4: add axis labels and figure title
%-----
xlabel('dose x');
ylabel('P(x)');
title('Mean function of the postulated Logit model and optimal design points');

% STEP 5: wrtie the graph to a file.
% This file is to be used on the output webpage to disply the graph
%-----

wsprintjpeg(f,ModelGraphFileName);
close(handle);
```

5 Discussion and Future Work

5.1 Unequal Interests in Individual Segments

In the previous discussions, we assumed every segment is of equal interest. However, it is nature to bring up such a question that what if the experimenter is more interested in estimating one segment than another? In other words, how does the D-optimal design reflect experimenters' unequal interests in individual segments? In fact, the real question is how do we formulate the problem?

Suppose we have a segmented polynomial model with two segments, and we are more interested in estimating the left segment than the right assuming the join point is known. It is handy to define a variable $\gamma(0 \leq \gamma \leq 1)$ to reflect the degree of interest in the left segment.

From the experimenter's perspective, he/she could estimate the two segments from the data separately since the two segments are independent, so that from experiment designers' perspective we could provide two independent D-optimal designs, one for each segment. To reflect the unequal interests, we can allocate the number of observations for each design according to the value of γ . By doing this, we can assure that the two individual designs are D-optimal and are reflecting experiments interest adequately. For example, if the sample size is N that is given, then the rounded quantity

of $N\gamma$ will be the number of observations taken on the left interval, rounded $N(1-\gamma)$ will be the number of observations taken on the right interval.

On the other hand, the two-piece segmented polynomial model can be estimated in one regression. If this would be case, then the design obtained by combining the two individual D-optimal designs is not necessarily a D-optimal design. However, we may start to formulate the original question using optimal exact designs. In practice, all designs must be exact. For a two-piece segmented polynomial, we define an exact as

$$\xi = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1q_1} & x_{21} & x_{22} & \cdots & x_{2q_2} \\ n_{11} & n_{12} & \cdots & n_{1q_1} & n_{21} & n_{22} & \cdots & n_{2q_2} \end{pmatrix}$$

Assuming q_1 support points on the left interval and q_2 support points on the right interval, the original question can be formulated as follows:

$$\min_{\xi} \phi(\xi) \text{ subject to } \sum_{i=1}^{q_1} n_{1i} = N\gamma \text{ and } \sum_{i=1}^{q_2} n_{2i} = N(1-\gamma),$$

where $\phi(\xi)$ is the D-optimal criterion function defined using an exact design. ξ can be split to two non-overlapping designs, say ξ_1 and ξ_2 , thus this two designs can be evaluated for optimality separately. Of course, ξ is D-optimal doesn't mean ξ_1 and ξ_2 are automatically D-optimal.

These are some preliminary thoughts in response to the emerged question. Finding an exact optimal design generally is difficult and seems to be done by computer programs, thus we leave it for future studies if it is of importance.

5.2 Future Development of Optimal Design Software

The result of our current development on optimal design software is the optimal design web site – www.optimal-design.org . The website is a great accomplishment of the entire team. It is the first web-based optimal design platform. Authorized Internet user can run optimal design applications remotely on the website. To my knowledge, only this website can really achieve that remote run capability. It is also a unique website that integrated various optimal design programs for various models by its single entry point and consistent user's interfaces. It is also an excellent presentation of the collaboration among researchers in optimal design community.

The website is still under construction, and it is continuously being improved. The website is good for now, but looking into the future, it may be outdated if encountered the following scenarios:

- **Scenario I:** When practitioners run optimal design applications on the website, they anticipate real-time responses from the server. This kind of expectation is particularly difficult to fulfill due to resource constraints. First, the Internet connection may break during the calculations. Second, some optimal design problems are very complicated, and may require more intensive computing. Furthermore, if several practitioners run their applications concurrently, the server could easily be overloaded.
- **Scenario II:** Practitioners should be offered to save their input and download optimal design results in a standard format. For example practitioners would like to manage their application histories like the way they do online transactions.

To meet these increasing needs, we need to adjust the system design or to plan for future development on the exist system. In either case, we propose the following solutions:

- **Proposal I:** we may deploy a cluster of computers. On each of these computers, we may deploy a copy of the optimal design programs. Ideally, when several concurrent requests are received, they will be distributed uniformly to these individual computers. To implement this solution, we need to develop a job distribution mechanism within the web application and need sufficient resources.
- **Proposal II:** We may develop stand-alone optimal design software. The stand-alone software should be just like other commercial available software. It can be installed on any personal computers. It should include at least two important features. One is a graphic user's interface, which should be well organized to guide practitioners perform optimal designs. The other one is a regression model library. The model library should include all of the regression models that are currently supported on the website.
- **Proposal III:** This solution is in light of the Scenario II. We may create an account for each practitioner on the website. Practitioners are required to register and sign in to the website. When a practitioner runs applications on the website, any details of his/her transactions are saved in his/her account. Another benefit is that practitioners could check back the status at a later time when their jobs cannot be completed instantly. Of course, this solution requires a database, as there are massive transactional data to be stored and be retrieved frequently.

References

- Atkinson, A. C. and Cook, R.D. (1995). D-Optimum Designs for Heteroscedastic Linear Models. *Journal of the American Statistical Association*. Vol. 90, No. 429, 204-212.
- Atkinson, A. C. and Donev, A. N. (1991). *Optimum Experimental Designs*. Oxford Science Publications.
- Baek, I., Zhu, W., Wu, X., Wong, W. K. (2006). Bayesian optimal Designs for a Quantal Dose Response Study with Potentially Missing Observations. *Journal of Biopharmaceutical Statistics*. 16:679-693.
- Biedermann, S., Dette, H., Zhu, W. (2006). Optimal Designs for Dose-Response Models with Restricted Design Spaces. *Journal of the American Statistical Association*. 101(407), 749-759.
- Chernoff, H. (1953). Locally optimal designs for estimating parameters. *The Annals of Mathematical Statistics*, 24, 586-602.
- Clyde, M. and Chaloner, K. (1996). The Equivalence of Constrained and Weighted Designs in Multiple Objective Design Problems. *Journal of the American Statistical Association*. 91:1236-1244.
- Cook, R. D. and Nachtsheim, C. J. (1982). Model robust, linear-optimal designs. *Technometrics* 24, 49-54.
- Cook, R. D. and Wong, W. K. (1994). On the Equivalence of Constrained and Compound Optimal Designs. *Journal of the American Statistical Association*. 89:687-692.
- Dette, H. (1992). Experimental Designs for a Class of Weighted Polynomial Regression Models. *Computational Statistics & Data Analysis*, 14, 359-373.
- Dette, H. Meas, V.B., Pepelyshev, A. (2003). Standardized Maximin E-Optimal Designs for the Michaelis-Menten Model. *Statistica Sinica*, 12, 1147-1163.
- Dette, H., Melas, V. B., Pepelyshev, A. (2006). Optimal Designs for Free Knot Least Square Splines. In Press.
- Fedorov, V. V. and Hackl, Peter. (1997). *Model-Oriented Design of Experiments*. Springer.

- Fedorov, V. V. (1972). Theory of Optimal Experiments. Academic Press New York.
- Firth, D. and Hinde, J. P. (1997). On Bayesian D-optimum Design Criteria and the Equivalence Theorem in Non-linear Models. J.R. Statist. Soc. B 59, No.4, 793-797.
- Fletcher, R. (1987). Practical Methods of Optimization. John Wiley and Sons.
- Pukelsheim, F. (1993). Optimal Design of Experiments. New York, Wiley.
- Fuller, W.A. (1969) Grafted Polynomial as Approximating Functions. The Australian Journal of Agriculture Economics, 13, 35-46.
- Gallant, A.R. and Fuller, Wayne A. (1973). Fitting Segmented Polynomial Regression Models Whose Join Points Have to Be Estimated. Journal of the American Statistical Association. Vol. 68, No. 341, 144-147.
- Haines, L.M., Perevozskaya, I., Rossenberger, W. F. (2003). Bayesian Optimal Designs for Phase I Clinical Trials. Biometrics. 59,591-600.
- Kiefer, J. (1959). Optimum Experimental Designs. Journal of Royal Statistical Society. Ser, B, 21, 272-319.
- Kiefer, J. (1961). Optimum Designs in Regression Problems, II. The Annal of Mathematical Statistics. 32, 298-325.
- Kiefer, J. (1975). Optimal design: variation in structure and performance under change of criterion. Biometrika 62, 277-88.
- Läuter, E. (1976). Optimal multipurpose designs for regression models. Math Operationforsch, Statist 5. 379-396.
- Lim, Y. B. (1991), D-optimal designs in polynomial spline regression. The Korean Journal of Applied Statistics. 2, 849-879.
- Microsoft. Microsoft TechNet IIS Technical References.
<http://technet2.microsoft.com/windowsserver/en/technologies/featured/iis/default.mspx>
- Microsoft Developer Network. MSDN Library, Visual Studio Web Developer.
[http://msdn2.microsoft.com/en-us/library/ms178093\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms178093(VS.80).aspx)
- MathWorks. Matlab Documentation.
<http://www.mathworks.com/access/helpdesk/help/techdoc/>
- MathWorks. Matlab Web Server.
http://www.mathworks.com/access/helpdesk_r13/help/toolbox/webserver/webserver.html

Nelder, J. A., Mead, R. A. (1965). A Simplex Method for Function Minimization. *Computer Journal*. 7:308-313.

Park, Sung H. (1978). Experimental Designs for Fitting Segmented Polynomial Regression Models. *Technometrics*, Vol. 20, No. 2, 151-154.

Pukelsheim, F. (1993). *Optimal Design of Experiments*, Wiley, New York.

Rice, J. R. (1969), *The Approximation of Functions*, Vol. 2, Reading, Mass.: Addison-Wesley.

Silvey, S. D. (1980). *Optimal Design*. Chapman and Hall, London.

Spall, J. C. (2003) *Introduction to Stochastic Search and Optimization*. Wiley, Hoboken, New Jersey.

Wu, X and Tensfeldt, T (2006). *Optimal Sampling Designs for Pharmacokinetic Studies*. 2006 Pfizer Intern Workshop.

Shuai, X, Zhou, Z, Yost, R. S. (2003). Using Segmented Regression Models to Fit Soil Nutrient and Soybean Grain Yield Changes due to Liming. *Journal of Agricultural, Biological, and Environmental Statistics*, Volume 8, Number 2, 240-252.

Zhu, W., Ahn, H., Wong, W. K. (1998). Multiple-Objective Designs in Dose-Response Experiments. *Institute of Mathematical Statistics Lecture Notes --Monograph Series: New Developments and Applications in Experimental Designs*, 73-82.

Zhu, W., Wong, W. K. (2001). Bayesian Optimal Designs for Estimating a Set of Symmetrical Quantiles. *Statistics in Medicine*, 20, 123-137.