# Topics on the Design and Analysis of Computer Experiments

By

Youngdeok Hwang

A dissertation submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

University of Wisconsin–Madison

2012

Date of final oral examination: 07/26/2012

The dissertation is approved by the following members of the final oral exam committee:

Peter Z. G. Qian, Associate Professor, Statistics
Kam-Wah Tsui, Professor, Statistics
Jun Zhu, Professor, Statistics
Bret Hanlon, Assistant Professor, Statistics
Qing Liu, Assistant Professor, Marketing

Topics on the Design and Analysis of Computer Experiments

Youngdeok Hwang

Under the supervision of Professor Peter Z. G. Qian

At the University of Wisconsin–Madison

# Abstract

This dissertation addresses several issues on the design and analysis of computer experiments. First, we propose a new type of design, called a sliced orthogonal array based Latin hypercube design, intended for running multiple computer experiments. The proposed designs achieve both one- and two-dimensional stratification while each slice possesses univariate uniformity. Sampling properties of the proposed designs are derived. Second, we develop two procedures for randomizing a new class of nested space-filling designs. Third, we propose a statistical approach to building an accurate metamodel by exploiting the quality of high-accuracy simulation data and the abundance of low-accuracy simulation data of a mechanical dynamics system. It makes use of Gaussian processes and natural cubic splines. The effectiveness of the proposed methodology is illustrated with an example for studying the dynamics of a slider-crank system.

*To My Family*

# Acknowledgments

I would like to acknowledge the suggestions, friendship, advice and support of numerous people who helped me during my time as a graduate student in the Department of Statistics at University of Wisconsin–Madison.

First, I would like to thank my advisor, Peter Zhiguang Qian, for his patient involvement and invaluable advice in this work. Peter has given me the opportunities to present my work to other researchers and helped me to grow as an independent researcher in statistics. He has guided me in a better direction not only on the research front but also on the life.

I am also thankful to Dr. Kam-Wah Tsui, Dr. Rick Nordheim, Dr. Jun Zhu, Dr. Bret Hanlon and Dr. Qing Liu for serving on my committee and for their comments and suggestions. I would like to acknowledge Dr. Bret Hanlon for sharing his experience as a junior researcher and an uncle. I also appreciate Dr. Douglas Bates for helping me find a job.

My thanks also go to my fellow graduate students in the department. In particular, Sangbum Choi, my two-year neighbor, helped me a lot, especially in the early years in Madison. I would also like to thank my fellow students Jee Young Moon, Lisa Chung, Kevin Eng, Andrew Thurman, Sokol Vako and Taeri Uhm. I also thank my research group members, Ben Haaland, Jun Li, Xu Xu, Qiong Zhang, Jiajie Chen and Yan Chen.

My family have been my biggest and strongest supporters in the past five years.

My parents have been great role models for me as good father and mother, good husband and wife, good son and daughter, good brother and sister, good citizens and good people. My brother and his family also deserve a huge acknowledgment. Finally, but most importantly, I thank my wife, Jiae, who was very tolerant and supportive until I arrived here.

This dissertation is dedicated to my family.

# Contents

# Chapter 1

# Introduction

Over the last few decades, computer experiments have emerged as a critical tool in science and engineering. This dissertation addresses the issues related to the design and analysis of computer experiments. A computer experiment is a computational simulation for a physical process using a complex mathematical model (Fang et al., 2005; Santner et al., 2003). A computer experiment can be treated as a function producing an output $y$ for a given set of inputs $x$, i.e., $y = f(x)$ with a function relating $x$ to $y$. The defining characteristic of a computer experiment, in contrast to the traditional physical experiment, is that the experiment yields a *deterministic* answer for a given set of input conditions (Santner et al., 2003); a computer experiment produces identical output when the experiment is run twice with the same inputs.

If the computer model can be evaluated with little computational cost, it is straightforward to use it for studying the physical process of interest. However, it is usually not possible to use large number of simulation runs, because the computer experiments are often computationally expensive. Accordingly, one problem is choosing a good set of design points at which to run the experiments. Chapters 2 and 3 relate

to the design problems. The other problem is analyzing the obtained data. Chapter 4 relates to building a statistical emulator for a computer experiment.

In Chapter 2, we propose an approach for constructing a new type of design, called a sliced orthogonal array based Latin hypercube design, intended for running multiple computer experiments. This approach exploits a slicing structure of orthogonal arrays with strength two and makes use of sliced random permutations. Such a design achieves one- and two-dimensional uniformity and can be divided into smaller Latin hypercube designs with one-dimensional uniformity. Sampling properties of the proposed designs are derived. Examples are given for illustrating the construction method and corroborating the derived theoretical results.

In Chapter 3, we propose two elaborate randomization methods to shuffle the levels of an asymmetric nested orthogonal array to produce a pair of asymmetric nested lattice samples. The constructed designs have a desirable nested structure, possess attractive space-filling properties and allow different axes to be divided at different scales of fineness. For multi-fidelity computer experiments, sequential evaluations, multi-step functional fitting and linking parameters, the asymmetric feature is appealing for situations where some factors are believed to be more important or deserve more attention than the other factors. The proposed designs are also useful for problems in these applications where different factors, by nature, require dividing their axes at different levels of fineness.

In Chapter 4, we propose a statistical approach to building an accurate metamodel of multibody dynamics simulations. Such a simulation is often available in a time-consuming but accurate version and an expeditious but approximate version for studying the same dynamics system. Our approach exploits the quality of high-

accuracy simulation data and the abundance of low-accuracy simulation data by using Gaussian processes and natural cubic splines. The corresponding experimental design issue is also discussed. The effectiveness of the proposed methodology is illustrated with an example for studying the dynamics of a slider-crank system.

# Chapter 2

# Designs for Multiple Computer Experiments

## 2.1 Motivation

Multiple computer experiments, based on the same or similar mathematics, are becoming popular for studying the same complex system (Williams et al., 2009; Storlie and Reich, 2011). Throughout, we do not consider multi-fidelity computer experiments like three finite element analysis codes with different mesh sizes.

Qian (2012) proposed *sliced Latin hypercube designs* for running multiple computer experiments. A sliced Latin hypercube design is a special Latin hypercube design (McKay et al., 1979) that can be partitioned into smaller Latin hypercube designs associated with different computer experiments. Figure 2.1 presents a sliced Latin hypercube design of 16 runs that is divided into four smaller Latin hypercube designs of four runs, denoted by ○, △, ✡ and □, respectively. In this figure, the whole design achieves maximum uniformity in any one dimension with respect to the

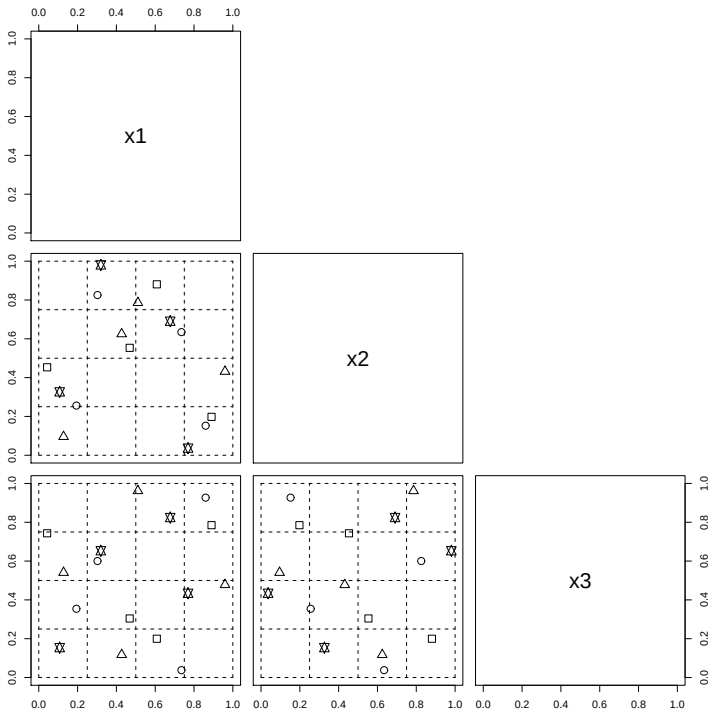


Fig. 2.1.— A sliced Latin hypercube design of 16 runs that is divided into four small Latin hypercube designs of four runs, denoted by $\bigcirc, \triangle, \hexagram, \square$, respectively, where the whole design achieves one-dimensional stratification with respect to the 16 equally spaced intervals of $(0, 1]$ and each slice achieves one-dimensional stratification with respect to the four equally spaced intervals of $(0, 1]$.

16 equally spaced intervals of $(0, 1]$ and each slice achieves maximum uniformity in any one dimension with respect to the four equally spaced intervals of $(0, 1]$. Since the whole design has 16 points while each slice has only four points, one may wonder the possibility of making the former achieve uniformity beyond one-dimensional stratification. Inspired by this curiosity, we propose a new type of design, called a *sliced orthogonal array based Latin hypercube design*, to achieve better stratification than a sliced Latin hypercube design. The proposed designs, referred to as sliced U designs hereinafter, achieves both *one- and two-dimensional* stratification while each

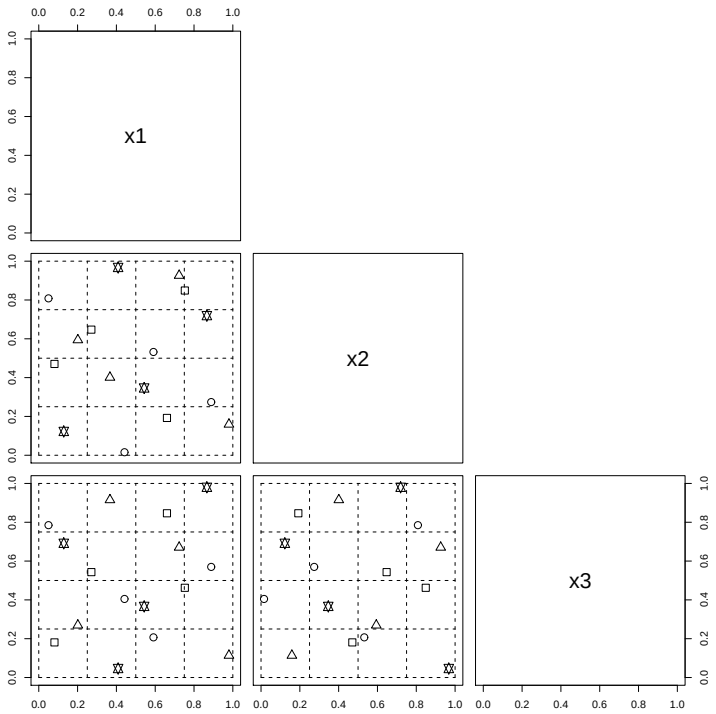


Fig. 2.2.— A sliced U design of 16 runs that is divided into four small Latin hypercube designs of four runs, denoted by $\bigcirc, \triangle, \hexagram, \square$, respectively, where the whole design achieves one-dimensional stratification with respect to the 16 equally spaced intervals of $(0, 1]$ and two-dimensional stratification with respect to the $4 \times 4$ grids, and each slice achieves one-dimensional stratification with respect to the four equally spaced intervals of $(0, 1]$.

slice possesses univariate uniformity. Although the proposed designs have better uniformity than sliced Latin hypercube designs, the latter are more flexible in run size and have no restriction on the number of factors. Figure 2.2 depicts a sliced U design, where the whole design achieves one-dimensional uniformity with respect to the 16 equally spaced intervals of $(0, 1]$ and two-dimensional uniformity with respect to the $4 \times 4$ grids, and each slice achieves one-dimensional uniformity with respect to the four equally spaced intervals of $(0, 1]$. The underlying idea of constructing a sliced U design

is to elaborately divide an orthogonal array of strength two into smaller orthogonal arrays with strength one and then randomize them using *sliced permutations* to form Latin hypercubes after some level-mapping. Note that Tang (1993) proposed orthogonal arrays based Latin hypercubes that have better stratification than ordinary Latin hypercube designs constructed in McKay et al. (1979). The designs constructed by Tang (1993) are referred to as *ordinary U designs* hereinafter. Related work in this direction includes Owen (1992b) and Patterson (1954). With respect to the existing work, the main contribution of our work is to construct new orthogonal array-based Latin hypercube designs with an appealing slicing structure and derive their sampling properties.

The remainder of the chapter is organized as follows. Section 2.2 presents the proposed construction method. Section 2.3 derives sampling properties of the constructed design. Section 2.4 gives examples to corroborate the derived theoretical results. We provide some discussion in Section 2.5.

## 2.2 Construction

This section describes the proposed construction method for sliced U designs in detail. Here are some definitions and notation. A uniform permutation on a set of $p$ integers means randomly taking a permutation on the set, with all $p!$ possible permutations equally probable. Let $\lceil \cdot \rceil$ denote the ceiling function. For an integer $p$, define

$$Z_p = \{1, \ldots, p\}. \tag{2.1}$$

An *orthogonal array* (OA) of $n$ rows, $q$ columns, $s$ levels and strength $t$, denoted by $\mathrm{OA}(n, s^q, t)$, is an $n \times q$ matrix with entries from $1, \ldots, s$ such that, for every $n \times t$

submatrix, all $s^t$ level combinations occurs equally often (Hedayat et al., 1999).

Let $\mathbf{A} = (a_{ik})$ be an OA$(n_2, s^{q+1}, 2)$ with $n_2 = s^2\lambda$ and $n_1 = s\lambda$. Let $\mathbf{A}(i,:)$ and $\mathbf{A}(:,k)$ denote the $i$th row and $k$th column of $\mathbf{A}$, respectively. The key idea here is to elaborately divide $\mathbf{A}$ into $s$ smaller orthogonal arrays of $n_1$ runs with strength one and then randomize them using sliced permutations to form Latin hypercubes after some level-mapping. Divide $Z_{n_2}$ into $n_1$ disjoint blocks of $s$ elements given by

$$\mathbf{g}_{n_2}(u,v) = \left\{ z \in Z_{n_2} : \left\lceil \frac{z}{n_1} \right\rceil = u, \left\lceil \frac{z - n_1(u-1)}{s} \right\rceil = v \right\}, \text{ for } u = 1, \ldots, s, \; v = 1, \ldots, \lambda. \tag{2.2}$$

In the construction of a sliced U design, these blocks are critical for simultaneously achieving uniformity in each slice and the whole design. For $n_2 = 18$, $n_1 = 6$, $s = 3$ and $\lambda = 2$, the six disjoint blocks of $Z_{n_2}$ in (2.2) are $\mathbf{g}_{18}(1,1) = \{1,2,3\}$, $\mathbf{g}_{18}(1,2) = \{4,5,6\}$, $\mathbf{g}_{18}(2,1) = \{7,8,9\}$, $\mathbf{g}_{18}(2,2) = \{10,11,12\}$, $\mathbf{g}_{18}(3,1) = \{13,14,15\}$ and $\mathbf{g}_{18}(3,2) = \{16,17,18\}$.

Randomize the columns of $\mathbf{A}$ and then randomize the symbols in each column by a uniform permutation on $Z_s$, with the permutations carried out independently from one column to another. We now present a useful lemma from Chapter 1 of Hedayat et al. (1999).

**Lemma 2.1.** *For an OA$(n_2, s^{q+1}, t)$ with $n_2 = s^2\lambda$ and $n_1 = s\lambda$, collecting the $n_1$ runs that has the same symbol in one column and deleting the column yields an OA$(n_1, s^q, t-1)$.*

This lemma is well known in design of experiments and has been used in the construction of various designs before. See, for example, Lin (1993), Xu (2005), Xu and Wu (2005) and He and Qian (2011).

Here we choose the $(q+1)$th column of $\mathbf{A}$ as the *slicing column*, although any other column can be used as well. Guided by Lemma 2.1, divide $\mathbf{A}$ into $s$ slices of $n_1$ runs, $\mathbf{A}_1, \ldots, \mathbf{A}_s$. For $m = 1, \ldots, s$, obtain $\mathbf{A}_m = (a_{m,ik})$ by collecting the rows of $\mathbf{A}$ with entries in the slicing column being $m$ and deleting the slicing column, and randomly shuffle the rows of $\mathbf{A}_m$. Let $\mathbf{B}_1, \ldots, \mathbf{B}_s$ be $s$ $n_1 \times q$ empty matrices, and let $\mathbf{B}$ and $\mathbf{C}$ be two $n_2 \times q$ empty matrices. For $k = 1, \ldots, q$, the proposed method proceeds in two steps:

**Step 1:** For $m = 1, \ldots, s$, replace the $\lambda$ entries of $\mathbf{A}_m(:, k)$ with $a_{m,ik} = u$ with a uniform permutation on $Z_\lambda$ to obtain $\mathbf{B}_m(:, k)$, for $u = 1, \ldots, s$, with the $s$ permutations carried out independently from one to another. Obtain column $k$ of $\mathbf{B} = (b_{ik})$ by combining $\mathbf{B}_1(:, k), \ldots, \mathbf{B}_s(:, k)$.

**Step 2:** For $u = 1, \ldots, s$, $v = 1, \ldots, \lambda$, obtain column $k$ of $\mathbf{C}$ by replacing the $s$ entries of $\mathbf{C}(:, k)$ satisfying $a_{ik} = u$ and $b_{ik} = v$ with a uniform permutation on $\mathbf{g}_{n_2}(u, v)$ in (2.2). These $n_1 = s\lambda$ permutations on the $n_1$ $\mathbf{g}_{n_2}$ blocks are carried out independently from one to another.

Using $\mathbf{C} = (c_{ik})$, generate an $n_2 \times q$ design $\mathbf{D} = (d_{ik})$ through

$$d_{ik} = (c_{ik} - u_{ik})/n_2, \quad \text{for} \quad i = 1, \ldots, n_2, \ k = 1, \ldots, q, \tag{2.3}$$

where the $u_{ik}$ are $U[0, 1)$ random variables, $d_{ik}$ is the level of factor $k$ on the $i$th run, and the $u_{ik}$ and the $c_{ik}$ are mutually independent. For $m = 1, \ldots, s$, let $\mathbf{C}_m = (c_{m,ik})$ be the submatrix of $\mathbf{C}$ corresponding to $\mathbf{A}_m$, and let $\mathbf{D}_m = (d_{m,ik})$ be the submatrix of $\mathbf{D}$ corresponding to $\mathbf{C}_m$.

Proposition 2.1 presents the space-filling properties of $\mathbf{D}$ and $\mathbf{D}_1 \ldots, \mathbf{D}_s$.

**Proposition 2.1.** *Consider* $\mathbf{D}$ *with slices* $\mathbf{D}_1, \ldots, \mathbf{D}_s$ *obtained above. We have that*

*(i) the design* $\mathbf{D}$ *achieves two-dimensional stratification with respect to the* $s \times s$ *grids when projected onto any two factors and achieves one-dimensional stratification with respect to the* $n_2$ *equally spaced interval of* $(0, 1]$ *when projected onto each factor;*

*(ii) each* $\mathbf{D}_m$ *achieves one-dimensional stratification with respect to the* $n_1$ *equally spaced interval of* $(0, 1]$ *when projected onto each factor.*

Compared with the sliced U design in Proposition 2.1, a sliced Latin hypercube design (Qian, 2012) of the same size can only achieve one-dimensional stratification for the whole design. In Section 2.1, this difference was illustrated by a *vis-à-vis* comparison of a sliced U design and a sliced Latin hypercube design of 16 runs in Figures 2.1 and 2.2, respectively. The sliced U design in Figure 2.2 is generated from an $\mathrm{OA}(16, 4^4, 2)$ by using the above construction method.

The method in Tang (1993) divides $Z_{n_2}$ associated with an $\mathrm{OA}(n_2, s^q, 2)$ into $s$ groups $\mathbf{h}_{n_2}(1), \ldots, \mathbf{h}_{n_2}(s)$ given by

$$\mathbf{h}_{n_2}(u) = \{z \in Z_{n_2} : \lceil z/n_1 \rceil = u, \} \text{ for } u = 1, \ldots, s, \tag{2.4}$$

and replaces the $u$'s in each column with a uniform permutation of the $n_1$ numbers of $\mathbf{h}_{n_2}(u)$. For $\mathbf{A}$ with index $\lambda = 1$ in the construction above, $\mathbf{C}$ in (2.3) is reduced to an ordinary U design in Tang (1993) as Step 1 becomes superfluous and the double-layer blocks $\mathbf{g}_{n_2}$ in (2.2) reduce to $\mathbf{h}_{n_2}$ in (2.4). The step to divide $\mathbf{A}$ into $\mathbf{A}_1, \ldots, \mathbf{A}_s$ using Lemma 2.1 is still critical for achieving the uniformity in each slice. If an ordinary U design of $n_2$ runs is randomly divided into $s$ slices of $n_1$ runs, these slices are not

guaranteed to achieve attractive uniformity. This point will be made more clear in Proposition 2.4 in Section 2.2.

**Example 2.1.** *Let* $\mathbf{A}$ *be an* $OA(18, 3^4, 2)$ *in part (a) of Table 2.1 with* $n_2 = 18$, $n_1 = 6$, $s = 3$, $q = 3$ *and* $\lambda = 2$. *Permute the columns of* $\mathbf{A}$ *and randomize the three symbols,* $1, 2, 3$, *in each column with a uniform permutation on* $Z_3$, *giving the array in part (b) of the table. For* $m = 1, 2, 3$, *obtain a matrix* $\mathbf{A}_m$ *by collecting the rows of* $\mathbf{A}$ *with entries in column 4 being* $m$ *and then deleting column 4, and randomly shuffle the rows in each* $\mathbf{A}_m$. *Part (c) of Table 2.1 present* $\mathbf{A}_1$, $\mathbf{A}_2$ *and* $\mathbf{A}_3$ *divided by the dashed lines. For* $n_2 = 18$ *and* $n_1 = 6$, *the six disjoint blocks of* $Z_{18}$ *in (2.2) are* $\mathbf{g}_{18}(1,1) = \{1, 2, 3\}$, $\mathbf{g}_{18}(1,2) = \{4, 5, 6\}$, $\mathbf{g}_{18}(2,1) = \{7, 8, 9\}$, $\mathbf{g}_{18}(2,2) = \{10, 11, 12\}$, $\mathbf{g}_{18}(3,1) = \{13, 14, 15\}$ *and* $\mathbf{g}_{18}(3,2) = \{16, 17, 18\}$. *Below is the step-to-step randomization of column 1 of* $\mathbf{A}$.

***Step 1:*** *Obtain* $\mathbf{B}_1(:,1)$ *by replacing the two 1's in* $\mathbf{A}_1(:,1)$ *with 2, 1, respectively. Replace the two 2's in* $\mathbf{A}_1(:,1)$ *with 2, 1, respectively, and replace the two 3's in* $\mathbf{A}_1(:,1)$ *with 1, 2, respectively. In* $\mathbf{B}_2(:,1)$, *replace the two 1's in* $\mathbf{A}_2(:,1)$ *with 1, 2, respectively, replace the two 2's in* $\mathbf{A}_2(:,1)$ *with 1, 2, respectively, and replace the two 3's in* $\mathbf{A}_2(:,1)$ *with 2, 1, respectively. In* $\mathbf{B}_3(:,1)$, *replace the two 1's in* $\mathbf{A}_3(:,1)$ *with 1, 2, respectively, replace the two 2's in* $\mathbf{A}_3(:,1)$ *with 2, 1, respectively, and replace the two 3's in* $\mathbf{A}_3(:,1)$ *with 2, 1, respectively.*

***Step 2:*** *Obtain* $\mathbf{C}(:,1)$ *by combining* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$. *Since the entries 5, 7 and 15 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 1$ *and* $b_{i1} = 1$, *the entries 5, 7 and 15 of* $\mathbf{C}(:,1)$ *are taken to be 3, 2, 1, a uniform permutation of* $\mathbf{g}_{18}(1,1)$. *Because the entries 1, 8 and 16 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 1$ *and* $b_{i1} = 2$, *the entries 1,*

*8 and 16 of* $\mathbf{C}(:,1)$ *are taken to be 6, 5, 4, a uniform permutation of* $\mathbf{g}_{18}(1,2)$. *Because the entries 6, 10 and 18 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1}=2$ *and* $b_{i1}=1$, *the entries 6, 10 and 18 of* $\mathbf{C}(:,1)$ *are taken to be 8, 7, 9, a uniform permutation of* $\mathbf{g}_{18}(2,1)$. *Because the entries 4, 11 and 13 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1}=2$ *and* $b_{i1}=2$, *the entries 4, 11 and 13 of* $\mathbf{C}(:,1)$ *are taken to be 12, 10, 11, a uniform permutation of* $\mathbf{g}_{18}(2,2)$. *Because the entries 2, 12 and 17 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1}=3$ *and* $b_{i1}=1$, *the entries 2, 12 and 17 of* $\mathbf{C}(:,1)$ *are taken to be 15, 13, 14, a uniform permutation of* $\mathbf{g}_{18}(3,1)$. *Because the entries 3, 7 and 14 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1}=3$ *and* $b_{i1}=2$, *the entries 3, 7 and 14 of* $\mathbf{C}(:,1)$ *are taken to be 18, 17, 16, a uniform permutation of* $\mathbf{g}_{18}(3,2)$.

*Part (e) of Table 2.1 presents the matrix* $\mathbf{C}$ *with three slices* $\mathbf{C}_1, \mathbf{C}_2$ *and* $\mathbf{C}_3$ *divided by the dashed lines. Figure 2.3 presents the bivariate projections of* $\mathbf{D}$ *of 18 runs generated from* $\mathbf{C}$. *The whole design of* $\mathbf{D}$ *achieves one-dimensional stratification with respect to the 18 equally spaced intervals of* $(0,1]$ *and two-dimensional stratification with respect to the* $3 \times 3$ *grids displayed in dashed lines. In any one-dimensional projection of* $\mathbf{D}$, *each of the 18 equally spaced intervals of* $(0,1]$ *contains exactly one point. In any two-dimensional projections of* $\mathbf{D}$, *each of the nine reference squares of* $(0,1]^2$ *contains exactly two points. The design* $\mathbf{D}$ *is divided into three Latin hypercube designs of six runs* $(\bigcirc, \triangle, \hexagram)$, *each having exactly one point in each of the six equally spaced intervals of* $(0,1]$.

**Example 2.2.** *Let* $\mathbf{A}$ *be an* $OA(32, 4^4, 2)$ *in part (a) of Table 2.2 with* $n_2 = 32$, $n_1 = 8$, $s = 4$, $q = 3$ *and* $\lambda = 2$. *Permute the columns of* $\mathbf{A}$ *and randomize the four symbols,* $1, 2, 3, 4$, *in each column with a uniform permutation on* $Z_4$, *giving the*

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| 3 | 2 | 2 | 2 |
| 3 | 2 | 1 | 1 |
| 3 | 1 | 1 | 2 |
| 3 | 3 | 2 | 3 |
| 3 | 1 | 3 | 3 |
| 3 | 3 | 3 | 1 |
| 2 | 3 | 1 | 3 |
| 2 | 3 | 3 | 2 |
| 2 | 2 | 3 | 3 |
| 2 | 1 | 1 | 1 |
| 2 | 2 | 2 | 1 |
| 2 | 1 | 2 | 2 |
| 1 | 1 | 3 | 1 |
| 1 | 1 | 2 | 3 |
| 1 | 3 | 2 | 1 |
| 1 | 2 | 3 | 2 |
| 1 | 3 | 1 | 2 |
| 1 | 2 | 1 | 3 |

(a)

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| 3 | 2 | 2 | 1 |
| 3 | 3 | 3 | 1 |
| 1 | 2 | 3 | 1 |
| 2 | 1 | 2 | 1 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 3 | 1 |
| 2 | 1 | 3 | 2 |
| 2 | 2 | 1 | 2 |
| 3 | 1 | 1 | 2 |
| 1 | 3 | 3 | 2 |
| 3 | 3 | 2 | 2 |
| 1 | 2 | 2 | 2 |
| 1 | 3 | 1 | 3 |
| 1 | 1 | 2 | 3 |
| 2 | 3 | 2 | 3 |
| 3 | 2 | 1 | 3 |
| 2 | 2 | 3 | 3 |
| 3 | 1 | 3 | 3 |

(b)

| $x_1$ | $x_2$ | $x_3$ |
|---|---|---|
| 1 | 1 | 1 |
| 3 | 3 | 3 |
| 3 | 2 | 2 |
| 2 | 2 | 3 |
| 1 | 2 | 3 |
| 2 | 1 | 2 |
| 1 | 2 | 2 |
| 1 | 3 | 3 |
| 3 | 1 | 1 |
| 2 | 2 | 1 |
| 2 | 1 | 3 |
| 3 | 3 | 2 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 1 | 1 | 2 |
| 3 | 1 | 3 |
| 3 | 3 | 1 |
| 2 | 2 | 3 |

(c)

| $x_1$ | $x_2$ | $x_3$ |
|---|---|---|
| 2 | 1 | 2 |
| 1 | 1 | 2 |
| 2 | 1 | 2 |
| 2 | 2 | 1 |
| 1 | 2 | 1 |
| 1 | 2 | 1 |
| 1 | 2 | 2 |
| 2 | 1 | 2 |
| 2 | 2 | 2 |
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 1 | 2 | 1 |
| 2 | 2 | 2 |
| 2 | 2 | 1 |
| 1 | 1 | 2 |
| 2 | 2 | 1 |
| 1 | 1 | 2 |
| 1 | 1 | 1 |

(d)

| $x_1$ | $x_2$ | $x_3$ |
|---|---|---|
| 6 (2) | 1 (1) | 6 (2) |
| 15 (5) | 13 (5) | 16 (6) |
| 18 (6) | 8 (3) | 11 (4) |
| 12 (4) | 17 (6) | 2 (1) |
| 3 (1) | 12 (4) | 14 (5) |
| 8 (3) | 5 (2) | 9 (3) |
| 2 (1) | 11 (4) | 12 (4) |
| 5 (2) | 14 (5) | 17 (6) |
| 17 (6) | 4 (2) | 4 (2) |
| 7 (3) | 7 (3) | 1 (1) |
| 10 (4) | 3 (1) | 15 (5) |
| 13 (5) | 16 (6) | 8 (3) |
| 11 (4) | 18 (6) | 10 (4) |
| 16 (6) | 10 (4) | 3 (1) |
| 1 (1) | 15 (5) | 5 (2) |
| 4 (2) | 6 (2) | 7 (3) |
| 14 (5) | 2 (1) | 18 (6) |
| 9 (3) | 9 (3) | 13 (5) |

(e)

Table 2.1: (a) An OA$(18, 3^4, 2)$ denoted by $\mathbf{A}$, (b) $\mathbf{A}$ after column and symbol permutations, (c) divide $\mathbf{A}$ into submatrices $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ (indicated by the dashed lines) according to different symbols in column 4 of $\mathbf{A}$ and deleting column 4 and randomly shuffle the rows in each slice, (d) $\mathbf{B}$ with submatrices $\mathbf{B}_1, \mathbf{B}_2$ and $\mathbf{B}_3$ obtained in Step 1 of the construction, (e) $\mathbf{C} = (c_{ik})$ obtained in Step 2 of the construction, where each slice is a Latin hypercube of six runs taking values in $Z_6$ after every entry $c_{ik}$ is collapsed according to level-mapping $\lceil c_{ik}/3 \rceil$

*array in part (b) of the table. For* $m = 1, 2, 3, 4$*, obtain* $\mathbf{A}_m$ *by collecting the rows of* $\mathbf{A}$ *with entries in column 4 being* $m$ *and then deleting column 4, and randomly shuffle the rows in* $\mathbf{A}_m$*. Part (c) of Table 2.2 present* $\mathbf{A}_1$*,* $\mathbf{A}_2$*,* $\mathbf{A}_3$ *and* $\mathbf{A}_4$ *divided by the dashed lines. For* $n_2 = 32$ *and* $n_1 = 8$*, the eight disjoint blocks of* $Z_{32}$ *in (2.2) are* $\mathbf{g}_{32}(1,1) = \{1, 2, 3, 4\}$*,* $\mathbf{g}_{32}(1,2) = \{5, 6, 7, 8\}$*,* $\mathbf{g}_{32}(2,1) = \{9, 10, 11, 12\}$*,*

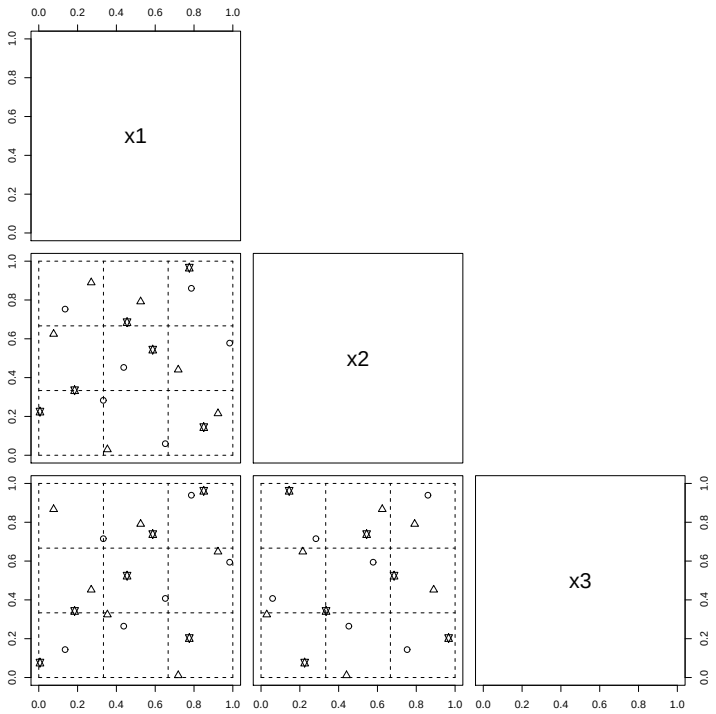


Fig. 2.3.— Bivariate projections of a sliced U design $\mathbf{D}$ with slices $\mathbf{D}_1$, $\mathbf{D}_2$ and $\mathbf{D}_3$ in Example 2.1. Each of the $3 \times 3$ squares in the dashed lines has exactly two points, and each of the 18 equally spaced intervals of $(0, 1]$ contains exactly one point. The array $\mathbf{D}$ is divided into three Latin hypercube designs of six runs ($\bigcirc, \triangle, \hexagram$), each containing exactly one point in each of the six equally spaced intervals of $(0, 1]$.

$\mathbf{g}_{32}(2,2) = \{13, 14, 15, 16\}$, $\mathbf{g}_{32}(3,1) = \{17, 18, 19, 20\}$, $\mathbf{g}_{32}(3,2) = \{21, 22, 23, 24\}$, $\mathbf{g}_{32}(4,1) = \{25, 26, 27, 28\}$ *and* $\mathbf{g}_{32}(4,2) = \{29, 30, 31, 32\}$. *Below is the step-to-step randomization of column 1 of* $\mathbf{A}$.

***Step 1:*** *Obtain* $\mathbf{B}_1(:,1)$ *by replacing the two 1's in* $\mathbf{A}_1(:,1)$ *with 1, 2, respectively. Replace the two 2's in* $\mathbf{A}_1(:,1)$ *with 1, 2, respectively, replace the two 3's in* $\mathbf{A}_1(:,1)$ *with 1, 2, respectively, and replace the two 4's in* $\mathbf{A}_1(:,1)$ *with 1, 2, respectively. In* $\mathbf{B}_2(:,1)$*, replace the two 1's in* $\mathbf{A}_2(:,1)$ *with 1, 2, respectively, replace the two 2's in* $\mathbf{A}_2(:,1)$ *with 1, 2, respectively, replace the two 3's in*

$\mathbf{A}_2(:,1)$ *with 2, 1, respectively, and replace the two 4's in* $\mathbf{A}_2(:,1)$ *with 1, 2, respectively. In* $\mathbf{B}_3(:,1)$*, replace the two 1's in* $\mathbf{A}_3(:,1)$ *with 2, 1, respectively, replace the two 2's in* $\mathbf{A}_3(:,1)$ *with 1, 2, respectively, replace the two 3's in* $\mathbf{A}_3(:,1)$ *with 2, 1, respectively, and replace the two 4's in* $\mathbf{A}_3(:,1)$ *with 1, 2, respectively. In* $\mathbf{B}_4(:,1)$*, replace the two 1's in* $\mathbf{A}_4(:,1)$ *with 1, 2, respectively, replace the two 2's in* $\mathbf{A}_4(:,1)$ *with 2, 1, respectively, replace the two 3's in* $\mathbf{A}_4(:,1)$ *with 2, 1, respectively, and replace the two 4's in* $\mathbf{A}_4(:,1)$ *with 2, 1, respectively.*

***Step 2:*** *Obtain* $\mathbf{C}(:,1)$ *by combining* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$*. Since the entries 1, 11, 21 and 29 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 1$ *and* $b_{i1} = 1$*, the entries 1, 11, 21 and 29 of* $\mathbf{C}(:,1)$ *are taken to be 1, 2, 4, 3, a uniform permutation of* $\mathbf{g}_{32}(1,1)$*. Because the entries 5, 16, 20 and 32 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 1$ *and* $b_{i1} = 2$*, the entries 5, 16, 20 and 32 of* $\mathbf{C}(:,1)$ *are taken to be 7, 6, 8, 5, a uniform permutation of* $\mathbf{g}_{32}(1,2)$*. Because the entries 3, 9, 23 and 30 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 2$ *and* $b_{i1} = 1$*, the entries 3, 9, 23 and 30 of* $\mathbf{C}(:,1)$ *are chosen to be 12, 11, 10, 9, a uniform permutation of* $\mathbf{g}_{32}(2,1)$*. Because the entries 4, 15, 24 and 26 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 2$ *and* $b_{i1} = 2$*, the entries 4, 15, 24 and 26 of* $\mathbf{C}(:,1)$ *are taken to be 13, 14, 16, 15, a uniform permutation of* $\mathbf{g}_{32}(2,2)$*. Because the entries 2, 14, 22 and 31 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 3$ *and* $b_{i1} = 1$*, the entries 2, 14, 22 and 31 of* $\mathbf{C}(:,1)$ *are taken to be 18, 19, 17, 20, a uniform permutation of* $\mathbf{g}_{32}(3,1)$*. Because the entries 8, 13, 19 and 28 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1} = 3$ *and* $b_{i1} = 2$*, the entries 8, 13, 19 and 28 of* $\mathbf{C}(:,1)$ *are taken to be 21, 22, 24, 23, a uniform permutation*

*of* $\mathbf{g}_{32}(3,2)$*. Because the entries 6, 10, 17 and 27 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1}=4$ *and* $b_{i1}=1$*, the entries 6, 10, 17 and 27 of* $\mathbf{C}(:,1)$ *are taken to be 25, 27, 26, 28, a uniform permutation of* $\mathbf{g}_{32}(4,1)$*. Because the entries 7, 12, 18 and 25 in* $\mathbf{A}(:,1)$ *and* $\mathbf{B}(:,1)$ *have* $a_{i1}=4$ *and* $b_{i1}=2$*, the entries 7, 12, 18 and 25 of* $\mathbf{C}(:,1)$ *are taken to be 29, 32, 31, 30, a uniform permutation of* $\mathbf{g}_{32}(4,2)$*.*

*Part (e) of Table 2.2 presents the matrix* $\mathbf{C}$ *with four slices* $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ *and* $\mathbf{C}_4$ *divided by the dashed lines. Figure 2.4 presents the bivariate projections of* $\mathbf{D}$ *of 32 runs generated from* $\mathbf{C}$*. The whole design of* $\mathbf{D}$ *achieves one-dimensional stratification with respect to the 32 equally spaced intervals of* $(0,1]$ *and two-dimensional stratification with respect to the* $4 \times 4$ *grids displayed in dashed lines. In any one-dimensional projection of* $\mathbf{D}$*, each of the 32 equally spaced intervals of* $(0,1]$ *contains exactly one point. In any two-dimensional projections of* $\mathbf{D}$*, each of the 16 reference squares of* $(0,1]^2$ *contains exactly two points. The design* $\mathbf{D}$ *is divided into four Latin hypercube designs of eight runs* $(\bigcirc, \triangle, \text{✡}, \square)$*, each having exactly one point in each of the eight equally spaced intervals of* $(0,1]$*.*

## 2.3 Sampling Properties

In this section, we derive sampling properties of sliced U designs. Let $F$ denote the uniform measure on the unit hypercube $(0,1]^q$. Let $f : (0,1]^q \to \mathbb{R}$ be a measurable function with $\int f(\mathbf{x})^2 dF < \infty$. Express $dF$ as $\prod_{k=1}^{q} dF_k$, where $F_k$ is the uniform measure of the $k$th dimension. The continuous ANOVA decomposition (Owen, 1994; Loh, 1996) of $f$ is

$$f = \sum_{u \in \mathcal{Q}} f_u, \tag{2.5}$$

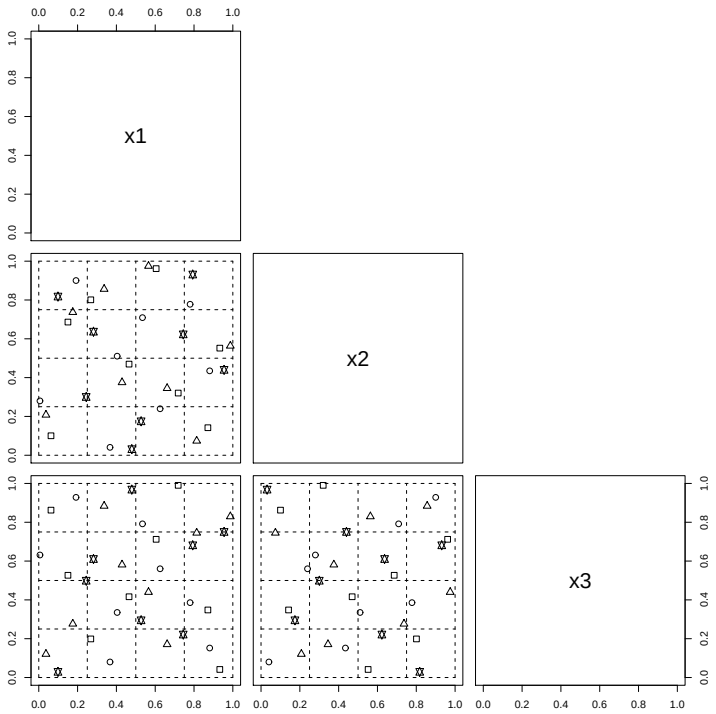


Fig. 2.4.— Bivariate projections of a sliced U design $\mathbf{D}$ with slices $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$ and $\mathbf{D}_4$ in Example 2.2. Each of the $4 \times 4$ squares in the dashed lines has exactly two points, and each of the 32 equally spaced intervals of $(0, 1]$ contains exactly one point. The array $\mathbf{D}$ is divided into four Latin hypercube designs of eight runs ($\bigcirc, \triangle, \hexagram, \square$), each containing exactly one point in each of the eight equally spaced intervals of $(0, 1]$.

where $\mathcal{Q}$ represents the set of all axes of $(0, 1]^q$. For any $u \in \mathcal{Q}$, $f_u$ can be defined via

$$\int \left( f(\mathbf{x}) - \sum_{v \subset u} f_v(\mathbf{x}) \right) dF_{\mathcal{Q}/u}, \tag{2.6}$$

where $dF_{\mathcal{Q}/u} = \prod_{k \notin u} dF_k$. For the empty set $\emptyset$, $f_\emptyset$ denotes the grand mean $\mu = \int f dF$. The variance of $f$, denoted by $\sigma^2 = \int (f - \mu)^2 dF$, can be decomposed as

$$\sigma^2 = \sum_{|u|>0} f_u^2 dF. \tag{2.7}$$

**Lemma 2.2.** *Consider* $\mathbf{C}$ *of* $n_2$ *runs from (2.3) based on an* $OA(n_2, s^{q+1}, 2)$ *with slices* $\mathbf{C}_1, \ldots, \mathbf{C}_s$ *of* $n_1$ *runs each, with* $n_1 = s\lambda$ *and* $n_2 = sn_1$. *Then (i) the columns of* $\mathbf{C}$

*and* $\mathbf{C}_m$, $m = 1, \ldots, s$, *are exchangeable; (ii) for* $m = 1, \ldots, s$, *the rows of* $\mathbf{C}_m$ *are exchangeable.*

*Proof.* Part (i) follows immediately by the column permutation of $\mathbf{A}$ prior to the construction method in Section 2.2. Part (ii) follows by the row permutation of each slice of $\mathbf{A}$ in the construction method. □

**Proposition 2.2.** *For* $m = 1, \ldots, s$, *let* $c_{m,ik}$ *denote the* $(i,k)$*th entry of* $\mathbf{C}_m$ *in Lemma 2.2. Then for* $k = 1, \ldots, q$,

*(i) the probability mass function for* $c_{m,ik}$, $i = 1, \ldots, n_1$, *is*

$$P(c_{m,ik} = x) = \ {n_2}^{-1}, \quad x \in Z_{n_2}. \tag{2.8}$$

*(ii) the joint probability mass function for* $c_{m,ik}$ *and* $c_{m,jk}$, $i \neq j$, *is*

$$P(c_{m,ik} = x, c_{m,jk} = y) = \begin{cases} [n_2(n_2 - s)]^{-1} & \lceil x/s \rceil \neq \lceil y/s \rceil, \ x, y \in Z_{n_2} \\ 0 & \textit{otherwise.} \end{cases} \tag{2.9}$$

*(iii) the joint probability mass function for* $c_{m_1,ik}$ *and* $c_{m_2,jk}$, $m_1 \neq m_2$, *is*

$$P(c_{m_1,ik} = x, c_{m_2,jk} = y) = \begin{cases} n_2^{-2} & \lceil x/s \rceil \neq \lceil y/s \rceil, \ x, y \in Z_{n_2} \\ [n_2(n_2 - n_1)]^{-1} & \lceil x/s \rceil = \lceil y/s \rceil, \ x \neq y, \ x, y \in Z_{n_2} \\ 0 & \textit{otherwise.} \end{cases} \tag{2.10}$$

Following He and Qian (2011), express the $(i,k)$th entry $c_{ik}$ of $\mathbf{C}$ in Proposition 2.2 as

$$c_{ik} \ = n_1\alpha_{ik} - s\beta_{ik} + \gamma_{ik}, \tag{2.11}$$

where $\alpha_{ik}$, $\beta_{ik}$, and $\gamma_{ik}$ are uniform random variables on $Z_s$, $Z_\lambda$ and $Z_s$, respectively. Similarly, express the $(i,k)$th entry $c_{m,ik}$ of $\mathbf{C}_m$ in Proposition 2.2 as

$$c_{m,ik} \ = n_1\alpha_{m,ik} - s\beta_{m,ik} + \gamma_{m,ik}. \tag{2.12}$$

The components in this expression correspond to the steps of the construction method in Section 2.2, with $\alpha_{ik}$ associated with the symbol permutation on $Z_s$, $\beta_{ik}$ the permutation on $Z_\lambda$ in Step 1 and $\gamma_{ik}$ the permutation on $Z_s$ in Step 2. Expressions of (2.11) and (2.12) will be used in Proposition 2.3. For row $i$ in $\mathbf{C}_m$, let $\tau_{m,i}^{kl}$ be the number of rows $j$ in $\mathbf{C}_m$ satisfying $\alpha_{m,ik} = \alpha_{m,jk}$ and $\alpha_{m,il} = \alpha_{m,jl}$. Let

$$\tau_m = n_1^{-1}[q(q-1)]^{-1} \sum_{i=1}^{n_1} \sum_{k \neq l} \tau_{m,i}^{kl} \tag{2.13}$$

denote the average of $\tau_{m,i}^{kl}$ values over all row and column pairs in $\mathbf{C}_m$. Averaging the $\tau_m$ values over all $s$ slices of $\mathbf{C}$ gives

$$\tau = s^{-1} \sum_{m=1}^{s} \tau_m. \tag{2.14}$$

Table 2.3 presents a grouping scheme originally introduced in He and Qian (2011) for nested U designs. A pair of nested U design constructed in He and Qian (2011) are two space-filling designs that the larger design contains a small space-filling design as a subset. The construction method for nested U designs is also motivated by Lemma 2.1, but works in a different fashion and for a different purpose by collecting the runs of an orthogonal array with strength two with the same symbol in one chosen column to form a small orthogonal array and then simultaneously randomizing the nested small array and the remaining runs of the large array. Table 2.3 was derived for grouping the entries in two rows and two columns of a pair of nested U designs. Though the probability mass functions of a sliced U design and a nested U design are different, these groups are still useful here as they are characteristics of the underlying OA.

**Remark 2.1.** *With the same underlying OA, each slice of a sliced U design has the same distribution as the small design of a pair of nested U designs constructed in He*

*and Qian (2011). In terms of the joint distribution of the whole design, a sliced U design and a nested U design are different. A nested U design does not have a slicing structure.*

**Proposition 2.3.** *For $m = 1, \ldots, s$, $c_{m,ik}$ denotes the $(i,k)$th entry of $\mathbf{C}_m$ in Lemma 2.2. Let $P(c_{m,ik}, c_{m,il}, c_{m,jk}, c_{m,jl})$ denote the joint probability mass function for $(c_{m,ik}, c_{m,il}, c_{m,jk}, c_{m,jl})$. For $\mathbf{C}$ with slices $\mathbf{C}_1, \ldots, \mathbf{C}_s$ in Lemma 2.2, we have that*

*(i) for rows $i, j$ in $\mathbf{C}_m$, $m = 1, \ldots, s$,*

$$P(c_{m,ik}, c_{m,il}, c_{m,jk}, c_{m,jl}) = \begin{cases} \frac{\tau}{\lambda^2 s^6 (s\lambda-1)(\lambda-1)^2} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_4. \\ \frac{\lambda-1-\tau}{\lambda^3 s^6 (s\lambda-1)(\lambda-1)(s-1)} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_7. \\ \frac{s\lambda-2\lambda+1+\tau}{\lambda^4 s^6 (s\lambda-1)(s-1)^2} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_8. \\ 0 & \textit{otherwise.} \end{cases} \tag{2.15}$$

*(ii) for row $i$ in $\mathbf{C}_{m_1}$ and row $j$ in $\mathbf{C}_{m_2}$, $m_1 \neq m_2$,*

$$P(c_{m_1,ik}, c_{m_1,il}, c_{m_2,jk}, c_{m_2,jl}) = \begin{cases} \frac{\lambda-1-\tau}{\lambda^5 s^5 (s-1)^3} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_2. \\ \frac{\lambda-1-\tau}{\lambda^5 s^6 (s-1)^2} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_3. \\ \frac{\lambda-1-\tau}{\lambda^5 s^7 (s-1)} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_4. \\ \frac{\lambda s-2\lambda+1+\tau}{\lambda^5 s^6 (s-1)^3} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_6. \\ \frac{\lambda s-2\lambda+1+\tau}{\lambda^5 s^7 (s-1)^2} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_7. \\ \frac{\lambda s^2-3\lambda s+3\lambda-1-\tau}{\lambda^5 s^7 (s-1)^3} & (x_{i1}, x_{i2}, y_{j1}, y_{j2}) \in H_8. \\ 0 & \textit{otherwise.} \end{cases} \tag{2.16}$$

Consider $s$ similar computer experiments $f_1, \ldots, f_s$ having inputs $\mathbf{x} = (x_1, \ldots, x_q)$ with the uniform distribution on $(0,1]^q$. For $m = 1, \ldots, s$, let

$$\mu_m = \int f_m(\mathbf{x}) dF. \tag{2.17}$$

Define

$$\mu = s^{-1} \sum_{m=1}^{s} \mu_m. \tag{2.18}$$

Let $\mathbf{D}$ be a sliced U design of $n_2$ runs from (2.3) having $s$ slices $\mathbf{D}_1, \ldots, \mathbf{D}_s$ of $n_1$ runs. For $m = 1, \ldots, s$, an estimator of $\mu_m$ in (2.17) using $\mathbf{D}_m$ is

$$\hat{\mu}_m \;=\; {n_1}^{-1} \sum_{i=1}^{n_1} f_m\left(\mathbf{x}_{m,i}\right), \tag{2.19}$$

where $\mathbf{x}_{m,i}$ denotes the $i$th run of $\mathbf{D}_m$. Then $\mu$ in (2.18) is estimated by

$$\hat{\mu} \;=\; s^{-1} \sum_{i=1}^{s} \hat{\mu}_m, \tag{2.20}$$

which uses $n_2$ runs of $\mathbf{D}$. For $u \in \mathcal{Q}$ and $r = 1, \ldots, q$, as in Owen (1994) and He and Qian (2011), let $w_{m,ij}(u) = \{k \in u | c_{m,ik} = c_{m,jk}\}$. Define

$$M_m(u, r) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} 1_{|w_{m,ij}(u)|=r}, \text{ for } m = 1, \ldots, s. \tag{2.21}$$

Similarly, define $w_{m_1,m_2,ij}(u) = \{k \in u | c_{m_1,ik} = c_{m_2,jk}\}$ and

$$M_{m_1,m_2}(u, r) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} 1_{|w_{m_1,m_2,ij}(u)|=r}, \text{ for } m_1, m_2 = 1, \ldots, s, \; m_1 \neq m_2. \tag{2.22}$$

For $m = 1, \ldots, s$, replacing $f$ with $f_m$ in (2.5) gives

$$f_m(\mathbf{x}) = \sum_{u \in \mathcal{Q}} f_{m,u}(\mathbf{x}). \tag{2.23}$$

Theorem 2.1 gives some variance formulas for sliced U designs.

**Theorem 2.1.** *Suppose that $E\left([f_m(\mathbf{x})]^2\right)$, for $m = 1, \ldots, s$, and $E[f_{m_1}(\mathbf{x}) f_{m_2}(\mathbf{x})]$, for $m_1, m_2 = 1, \ldots, s$, are well defined and finite. Let $\tau$ be as defined in (2.14). Suppose that, for $m = 1, \ldots, s$, $f_m$ is a continuous function on $(0, 1]^q$. Then for $\hat{\mu}_m$ in (2.19)*

*and $\hat{\mu}$ in (2.20) under sliced U designs, as $s \to \infty$ with $\lambda$ fixed,*

$$(i)\quad \mathrm{var}\,(\hat{\mu}_m) = \sum_{|u|\geq 2} M_m(u,|u|) n_1^{-2} \mathrm{var}[f_{m,u}\,(\mathbf{x})] + o(n_1^{-1});$$

$$(ii)\quad \mathrm{var}\,(\hat{\mu})$$
$$= (1+\tau) n_2^{-1} s^{-1} \sum_{|u|=2} \left( \sum_{m=1}^{s} \mathrm{var}[f_{m,u}(\mathbf{x})] - (s-1)^{-1} \sum_{m_1 \neq m_2} \mathrm{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})] \right)$$
$$+ \sum_{|u|>2} n_2^{-2} \left( \sum_{m=1}^{s} M_m(u,|u|) \mathrm{var}[f_{m,u}(\mathbf{x})] + \sum_{m_1 \neq m_2} M_{m_1,m_2}(u,|u|) \mathrm{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})] \right)$$
$$+ o(n_2^{-1}).$$

Theorem 2.1 (i) implies that the slices of a sliced U design achieve variance reduction similar to those of a sliced Latin hypercube design. If $m_1 \neq m_2$, $\mathrm{cov}[f_{m_1,u}(\mathbf{x}) f_{m_2,u}(\mathbf{x})]$ are positive (e.g., $f_{m_1} = f_{m_2}$), Theorem 2.1 (ii) implies that a sliced U design as a whole achieves a similar degree of variance reduction to an ordinary U design constructed in Tang (1993). As Owen (1994), define

$$M(u,r) = \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} 1_{|w_{ij}(u)|=r},$$

similar to (2.21), where $w_{ij}(u)$ is defined in (A.5).

**Remark 2.2.** *When $f = f_1 = \cdots = f_s$, $\mathrm{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})]$ in (A.6) reduces to $\mathrm{var}[f_u(\mathbf{x})]$ and Theorem 2.1 (ii) hence reduces to*

$$\sum_{|u|\geq 3} M(u,|u|) n_2^{-2} \mathrm{var}[f_u(\mathbf{x})] + o(n_2^{-1}),$$

*which achieves a similar degree of variance reduction to an ordinary U design constructed in Tang (1993).*

For $f_{m,u}$ and $f_{m_1,u}, f_{m_2,u}$ with $u \in \mathcal{Q}$ in (2.23), define

$$\begin{array}{ll} V_u & = s^{-1}\sum_{m=1}^{s} \mathrm{var}[f_{m,u}(\mathbf{x})], \\ C_u & = [s(s-1)]^{-1}\sum_{m_1 \neq m_2} \mathrm{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})]. \end{array} \tag{2.24}$$

We now compare Theorem 2.1 with four other methods to generate $\mathbf{D}_1, \ldots, \mathbf{D}_s$ of $n_1$ runs each.

**Proposition 2.4.** *Let* $\mathbf{D}$ *denote the union of a sequence of designs* $\mathbf{D}_1, \ldots, \mathbf{D}_s$ *of* $n_1$ *runs each. As in Proposition 2.1, here* $n_2 = sn_1$*. For* $m = 1, \ldots, s$*, consider* $\hat{\mu}_m$ *in (2.19) using* $\mathbf{D}_m$ *and* $\hat{\mu}$ *in (2.20) using* $\mathbf{D}$*. Then we have the following result:*

*(i) IID: If* $\mathbf{D}_1, \ldots, \mathbf{D}_s$ *are* $s$ *IID samples of* $n_1$ *runs, we have that*

$$\begin{array}{lcl} \mathrm{var}(\hat{\mu}_m) & = & \sum_{|u|>0} n_1^{-1}\mathrm{var}[f_{m,u}(\mathbf{x})], \\ \mathrm{var}(\hat{\mu}) & = & \sum_{|u|>0} n_2^{-1} V_u. \end{array}$$

*(ii) LHD: If* $\mathbf{D}_1, \ldots, \mathbf{D}_s$ *are* $s$ *independent Latin hypercube designs of* $n_1$ *runs (McKay et al., 1979), we have that*

$$\begin{array}{lcl} \mathrm{var}(\hat{\mu}_m) & = & \sum_{|u|>1} n_1^{-1}\mathrm{var}[f_{m,u}(\mathbf{x})] + o(n_1^{-1}), \\ \mathrm{var}(\hat{\mu}) & = & \sum_{|u|>1} n_2^{-1} V_u + o(n_2^{-1}). \end{array}$$

*(iii) SLHD: If* $\mathbf{D}_1, \ldots, \mathbf{D}_s$ *are a sliced Latin hypercube design with* $s$ *slices of* $n_1$ *runs (Qian, 2012), we have that*

$$\begin{array}{lcl} \mathrm{var}(\hat{\mu}_m) & = & \sum_{|u|>1} n_1^{-1}\mathrm{var}[f_{m,u}(\mathbf{x})] + o(n_1^{-1}), \\ \mathrm{var}(\hat{\mu}) & = & \sum_{|u|>1} n_2^{-1} V_u + o(n_2^{-1}). \end{array}$$

*(iv) OU: If* $\mathbf{D}_1, \ldots, \mathbf{D}_s$ *are* $s$ *slices of* $n_1$ *runs obtained by randomly dividing an ordinary U design* $\mathbf{D}$ *based on an OA*$(n_2, s^q, 2)$*, we have that*

$$\begin{array}{l} \mathrm{var}\,(\hat{\mu}_m) = \sum_{|u|>0} n_1^{-1}\mathrm{var}[f_{m,u}(\mathbf{x})] + o(n_1^{-1}), \\ \mathrm{var}\,(\hat{\mu}) = \sum_{|u|=1,2} n_2^{-1}(V_u - C_u) + \sum_{|u|>2} n_2^{-1} V_u + o(n_2^{-1}). \end{array}$$

A comparison of Theorem 2.1 (i) and Proposition 2.4 (iii) implies that each slice of a sliced U design achieves similar variance reduction as a sliced Latin hypercube design of the same size. Proposition 2.4 (iv) implies that if an ordinary U design is randomly divided into $s$ slices of $n_1$ runs, the slices do not achieve the variance reduction as those of a sliced U design in Theorem 2.1 (i).

## 2.4 Numerical Illustration

This section presents numerical examples to corroborate Theorem 2.1 and Proposition 2.4 in Section 2.3. Take $s$ functions $f_1, \ldots, f_s$ to act as $s$ similar computer models having inputs $\mathbf{x}$ with the uniform distribution on $[0,1)^q$. The goal here is to estimate (1) the expected output $\mu_m$ of $f_m$ by using a design $\mathbf{D}_m$ of $n_1$ runs for $m = 1, \ldots, s$ and (2) the overall mean $\mu = s^{-1}\sum_{m=1}^{s}\mu_m$ by combining $\mathbf{D}_1, \ldots, \mathbf{D}_s$. As in Proposition 2.1, here $n_2 = sn_1$. We compare four different methods described in Proposition 2.4 to generate $\mathbf{D}_1, \ldots, \mathbf{D}_s$ with those taken from a sliced U design.

1. **IID**: Generate $s$ IID samples of $n_1$ runs.
2. **ILHD**: Independently generate $s$ ordinary Latin hypercube designs of $n_1$ runs.
3. **SLHD**: Generate a sliced Latin hypercube design with $s$ slices of $n_1$ runs.
4. **OU**: Randomly partition an ordinary U design of $n_2$ runs based on an $\mathrm{OA}(n_2, s^q, 2)$ into $s$ slices of $n_1$ runs.
5. **SU**: Generate a sliced U design with $s$ slices of $n_1$ runs based on an $\mathrm{OA}(n_2, s^{q+1}, 2)$.

**Example 2.3.** *Let $f_1 = f_2 = f_3 = f_4$ be the borehole function (Morris et al., 1993)*

*given by*

$$\frac{2\pi x_3(x_4 - x_6)}{\log(x_2/x_1)\left[1 + 2\frac{x_7 x_3}{\log(x_2/x_1)x_1^2 x_8} + \frac{x_3}{x_5}\right]}. \tag{2.25}$$

*By using data from a Latin hypercube design with* $10^6$ *runs,* $\mu_1$ *is found to be 77.668. A sliced U design with four slices, each being a Latin hypercube design with 16 runs, is constructed by using the* $OA(64, 4^9, 2)$ *presented in Table 2.4. For each method, we compute* $\hat{\mu}_m$ *for* $m = 1, 2, 3, 4$ *in (2.19) and* $\hat{\mu}$ *in (2.20) 1000 times. Table 2.5 presents the sample means, sample standard deviations and root mean squared errors (RMSEs) over the 1000 replicates for the five methods. The SU method not only performs equally as well as the SLHD and ILHD for each* $\hat{\mu}_m$*, but also performs equally as well as the OU method for* $\hat{\mu}$*.*

**Example 2.4.** *Use* $f_1$ *in (2.25). Consider its three variants*

$$f_2 = \frac{2\pi x_3(x_4 - x_6)}{\log(x_2/x_1)\left[1 + 1.95\frac{x_7 x_3}{\log(x_2/x_1)x_1^2 x_8} + \frac{x_3}{x_5}\right]},$$
$$f_3 = \frac{2\pi x_3(x_4 - x_6)}{\log(x_2/x_1)\left[1 + 2.05\frac{x_7 x_3}{\log(x_2/x_1)x_1^2 x_8} + \frac{x_3}{x_5}\right]},$$
$$f_4 = \frac{2.05\pi x_3(x_4 - x_6)}{\log(x_2/x_1)\left[1 + 2.15\frac{x_7 x_3}{\log(x_2/x_1)x_1^2 x_8} + \frac{x_3}{x_5}\right]}.$$

*By using a Latin hypercube design with* $10^6$ *runs,* $\mu_1$*,* $\mu_2$*,* $\mu_3$ *and* $\mu_4$ *defined in (2.17) are found to be 77.668, 79.654, 75.761 and 74.076, respectively. Here,* $\mu$ *in (2.18) is 76.7901. A sliced U design with four slices, each being a Latin hypercube design with 16 runs, is constructed by using the* $OA(64, 4^9, 2)$ *in Table 2.4. For each method, we compute* $\hat{\mu}_m$ *in (2.19) for* $m = 1, 2, 3, 4$ *and* $\hat{\mu}$ *in (2.20) 1000 times. Table 2.6 presents the sample means, sample standard deviations and RMSEs over the 1000 replicates*

*for the five methods. The SU method not only performs equally as well as the SLHD and ILHD for each $\hat{\mu}_m$, but also performs equally as well as the OU method for $\hat{\mu}$.*

**Example 2.5.** *Consider the following test function from Cox et al. (2001)*

$$f_1 = \frac{x_1}{2}\left[\sqrt{1+(x_2+x_3^2)\,\frac{x_4}{x_1^2}}-1\right]+x_1+3x_4.$$

*Define its two variants*

$$f_2 = \frac{x_1}{2.1}\left[\sqrt{0.9+(x_2+x_3^2)\,\frac{0.9x_4}{x_1^2}}-1.01\right]+x_1+3x_4,$$

$$f_3 = \frac{x_1}{1.9}\left[\sqrt{1+(x_2+x_3^2)\,\frac{0.9x_4}{x_1^2}}-1.1\right]+x_1+3x_4.$$

*By using a Latin hypercube design with $10^6$ runs, $\mu_1, \mu_2$ and $\mu_3$ defined in (2.17) are found to be 2.160, 2.140 and 2.152, respectively. Here, $\mu$ in (2.18) is found to be 2.151. A sliced U design with three slices, each being a Latin hypercube design with 12 runs, is constructed by using the $OA(36, 3^5, 2)$ in Table 2.7. For each method, we compute $\hat{\mu}_m$ for $m = 1, 2, 3$ in (2.19) and $\hat{\mu}$ in (2.20) 1000 times. Table 2.8 presents the sample means, sample standard deviations and RMSEs over the 1000 replicates for the five methods. The SU method not only performs equally as well as the SLHD and ILHD for each $\hat{\mu}_m$, but also performs equally as well as the OU method for $\hat{\mu}$.*

**Example 2.6.** *Let $f_1$ be the following test function from Deb et al. (2005)*

$$10\cos(\pi x_1/2)\cos(\pi x_2/2)\cos(\pi x_3/2)[1+(x_4-0.5)^2].$$

*Consider its two variants*

$$f_2 = 10\cos(\pi x_1/2)\cos(\pi x_2/1.9)\cos(\pi x_3/1.95)[1+(x_4-0.51)^2],$$

$$f_3 = 10\cos(\pi x_1/2.1)\cos(\pi x_2/2.1)\cos(\pi x_3/2.01)[1+(x_4-0.45)^2].$$

*By using a Latin hypercube design with* $10^6$ *runs,* $\mu_1, \mu_2$ *and* $\mu_3$ *defined in (2.19) are found to be 2.798, 2.577 and 2.963, respectively. Here,* $\mu$ *in (2.18) is found to be 2.779. A sliced U design with three slices, each being a Latin hypercube design with 27 runs, is constructed by using an* $OA(81, 3^5, 2)$ *in Table 2.9. For each method, we compute* $\hat{\mu}_m$ *for* $m = 1, 2, 3$ *in (2.19) and* $\hat{\mu}$ *in (2.20) 1000 times. Table 2.10 presents the sample means, sample standard deviations and RMSEs over the 1000 replicates for the five methods. The SU method not only performs equally as well as the SLHD and ILHD for each* $\hat{\mu}_m$*, but also performs equally as well as the OU method for* $\hat{\mu}$*.*

**Example 2.7.** *Consider* $f_1, f_2, f_3$ *and* $f_4$ *used in Example 2.4. A sliced U design with four slices, each being a Latin hypercube design with 16 runs, is constructed by using an* $OA(64, 4^9, 2)$ *in Table 2.4. We estimate* $\mu_1$ *by* $\hat{\mu}_m$ *in (2.19) and* $\hat{\mu}$ *in (2.20). For each method, we compute* $\hat{\mu}_m$ *for* $m = 1, 2, 3, 4$ *and* $\hat{\mu}$ *1000 times. Table 2.11 presents the sample means, sample standard deviations and RMSEs over the 1000 replicates for the five methods. The SU method not only performs equally as well as the SLHD and ILHD for each* $\hat{\mu}_m$*, but also performs equally as well as the OU method for* $\hat{\mu}$*.*

## 2.5 Discussion

Running multiple computer experiments is a growing trend in practice. To respond to this emerging need, we have introduced a new type of design, called a sliced U design, by randomizing orthogonal arrays with sliced permutations. Such a design has a desirable sliced structure and achieves better uniformity than a sliced Latin hypercube design constructed in Qian (2012). MATLAB and R programs for constructing the proposed designs are available from the authors. Sampling properties

of these designs are derived for the purpose of estimating the expected outputs of multiple computer experiments. Compared with our proposed designs, sliced space-filling designs (Qian and Wu, 2009) and Sudoku-based space-filling designs (Xu et al., 2011) are constructed by using algebraic methods and their sampling properties are difficult to derive.

It is possible to improve the stratification of the sliced U design in Proposition 2.1 by using an OA with strength three or higher. Let $\mathbf{A}$ be an OA$(n_3, s^{q+2}, t)$ with $t \geq 3$. The key is to use Lemma 2.1 in Section 2.2 to divide $\mathbf{A}$ into $s$ slices, $\mathbf{A}_1, \ldots, \mathbf{A}_s$, each becoming an OA$(n_2, s^{q+1}, t-1)$. For $l = 1, \ldots, s$, further divide $\mathbf{A}_l$ into $s$ smaller slices, $\mathbf{A}_{l1}, \ldots, \mathbf{A}_{ls}$, each becoming an OA$(n_1, s^q, t-2)$. For $l = 1, \ldots, s$ and $m = 1, \ldots, s$, randomize $\mathbf{A}_{lm}$ to obtain a matrix $\mathbf{C}_{lm}$. For $l = 1, \ldots, s$, obtain a larger design $\mathbf{C}_l$ by combining $\mathbf{C}_{l1}, \ldots, \mathbf{C}_{ls}$ row by row. Finally, combining $\mathbf{C}_1, \ldots, \mathbf{C}_s$ row by row gives a design $\mathbf{C}$. Each $\mathbf{C}_{lm}$ is randomized by using sliced permutations more sophisticated than those in (2.2) for $\mathbf{C}$ to achieve $t$-dimensional stratification, each $\mathbf{C}_l$ to achieve $(t-1)$-dimensional stratification and each $\mathbf{C}_{lm}$ to achieve $(t-2)$-dimensional stratification.

If $\mathbf{A}$ is an OA with strength three, this method constructs a sliced U design $\mathbf{C}$ of $n_3$ runs that can be divided into $\mathbf{C}_1, \ldots, \mathbf{C}_s$ of $n_2$ runs each with two-dimensional stratification, and each $\mathbf{C}_l$ can be further divided into $\mathbf{C}_{l1}, \ldots, \mathbf{C}_{ls}$ of $n_1$ runs each with one-dimensional stratification. Here, $n_1 = s\lambda$, $n_2 = sn_1$ and $n_3 = sn_2$. For illustration, let $\mathbf{A}$ be an OA$(27, 3^4, 3)$ from part (a) of Table 2.12. For $l = 1, 2, 3$, collecting the rows of $\mathbf{A}$ with entries in column 4 being $l$ and deleting column 4 yields $\mathbf{A}_l$, which is an OA$(9, 3^3, 2)$. Each $\mathbf{A}_l$ can be further divided into three slices $\mathbf{A}_{lm}$, $m = 1, 2, 3$, by collecting the rows of $\mathbf{A}_l$ with entries in column 3 being $m$ and deleting

column 3. Each $\mathbf{A}_{lm}$ is an OA$(3, 3^2, 1)$. The arrays $\mathbf{A}_{11}, \ldots, \mathbf{A}_{33}$ are randomized to give $\mathbf{C}_{11}, \ldots, \mathbf{C}_{33}$, respectively. The larger designs $\mathbf{C}_1, \mathbf{C}_2$ and $\mathbf{C}_3$ correspond to $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$, respectively, and take the union of $\mathbf{C}_1, \mathbf{C}_2$ and $\mathbf{C}_3$ to form $\mathbf{C}$. Each $\mathbf{C}_{lm}$ is randomized such that (i) $\mathbf{C}$ becomes an OA$(27, 3^2, 2)$ taking values in $Z_3$ after every element $a$ is collapsed by level mapping $\lceil a/9 \rceil$ and is a Latin hypercube taking values in $Z_{27}$, (ii) each $\mathbf{C}_l$ becomes an OA$(9, 3^2, 2)$ taking values in $Z_3$ after every element $a$ is collapsed by level mapping $\lceil a/9 \rceil$ and becomes a Latin hypercube taking values in $Z_9$ after every element $a$ is collapsed by level mapping $\lceil a/3 \rceil$, (iii) each $\mathbf{C}_{lm}$ becomes a Latin hypercube taking values in $Z_3$ after every element $a$ is collapsed by level mapping $\lceil a/9 \rceil$. Specifically, divide $Z_{27}$ into three blocks:

$$\mathbf{h}_1 = \{1, \ldots, 9\},\ \mathbf{h}_2 = \{10, \ldots, 18\},\ \mathbf{h}_3 = \{19, \ldots, 27\}.$$

For $l = 1, 2, 3$, divide $\mathbf{h}_l$ to three smaller blocks of three consecutive integers:

$$\mathbf{h}_{lm} = \left\{ z \in \mathbf{h}_l : \left\lceil \frac{z - 9(l-1)}{3} \right\rceil = m \right\},\ \text{for } m = 1, 2, 3. \tag{2.26}$$

The three symbols, $1, 2, 3$, in the first two columns of $\mathbf{A}$ are randomized to obtain $\mathbf{C}_l$ and $\mathbf{C}_{lm}$, for $l = 1, 2, 3$ and $m = 1, 2, 3$, such that exactly one element of $\mathbf{h}_{lm}$ appears in each $\mathbf{C}_l$ and exactly one element of $\mathbf{h}_l$ appears in each $\mathbf{C}_{lm}$. For example, since the entries 1, 6 and 8 contain symbol 1 in column 1 of $\mathbf{A}_1$ in part (a) in Table 2.12, these entries in column 1 of $\mathbf{C}_1$ are taken to be 3, 5 and 8, from $\mathbf{h}_{11}$, $\mathbf{h}_{12}$ and $\mathbf{h}_{13}$, respectively. Since the entries 3, 5 and 7 have symbol 2 in column 1 of $\mathbf{A}_1$, these entries in column 1 of $\mathbf{C}_1$ are taken to be 18, 14 and 11, from $\mathbf{h}_{23}$, $\mathbf{h}_{22}$ and $\mathbf{h}_{21}$, respectively. Since the entries 2, 4 and 9 have symbol 3 in column 1 of $\mathbf{A}_1$, these entries in column 1 of $\mathbf{C}_1$ are taken to be 19, 27 and 24, from $\mathbf{h}_{31}$, $\mathbf{h}_{33}$ and $\mathbf{h}_{32}$, respectively. Randomize the entries of $\mathbf{A}_2$ and $\mathbf{A}_3$ similarly to obtain $\mathbf{C}_2$ and $\mathbf{C}_3$, respectively.

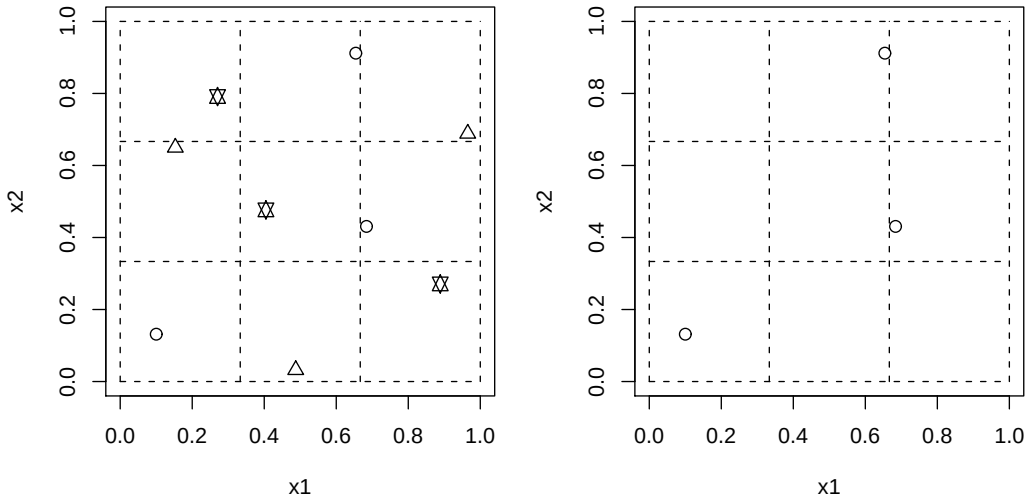


Fig. 2.5.— Bivariate projection of $\mathbf{C}_1$ of nine runs and $\mathbf{C}_{11}$ of three runs from Table 2.12, where $\mathbf{C}_1$ can be divided into three slices ($\bigcirc, \triangle, \hexagram$), each being a Latin hypercube design of three runs.

Another way to generate sliced designs with better stratification is to exploit some slicing structure in the infinite $(t, s)$ sequence in a prime power base (Niederreiter, 1992; Owen, 1995).

Sliced U designs with (nearly) orthogonal columns or with optimality according to the maximin distance criterion can be constructed by extending the methods in Ye (1998), Steinberg and Lin (2006), Bingham et al. (2009), Lin et al. (2009) and Lin et al. (2010) or the method in Morris and Mitchell (1995).

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| 3 | 3 | 1 | 4 |
| 1 | 4 | 1 | 2 |
| 2 | 2 | 1 | 1 |
| 4 | 1 | 1 | 3 |
| 4 | 3 | 1 | 4 |
| 2 | 4 | 1 | 2 |
| 1 | 2 | 1 | 1 |
| 3 | 1 | 1 | 3 |
| 2 | 3 | 2 | 3 |
| 4 | 4 | 2 | 1 |
| 3 | 2 | 2 | 2 |
| 1 | 1 | 2 | 4 |
| 1 | 3 | 2 | 3 |
| 3 | 4 | 2 | 1 |
| 4 | 2 | 2 | 2 |
| 2 | 1 | 2 | 4 |
| 3 | 3 | 3 | 1 |
| 1 | 4 | 3 | 3 |
| 2 | 2 | 3 | 4 |
| 4 | 1 | 3 | 2 |
| 4 | 3 | 3 | 1 |
| 2 | 4 | 3 | 3 |
| 1 | 2 | 3 | 4 |
| 3 | 1 | 3 | 2 |
| 2 | 3 | 4 | 2 |
| 4 | 4 | 4 | 4 |
| 3 | 2 | 4 | 3 |
| 1 | 1 | 4 | 1 |
| 1 | 3 | 4 | 2 |
| 3 | 4 | 4 | 4 |
| 4 | 2 | 4 | 3 |
| 2 | 1 | 4 | 1 |

(a)

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| 1 | 2 | 2 | 1 |
| 2 | 3 | 1 | 1 |
| 3 | 4 | 3 | 1 |
| 4 | 1 | 4 | 1 |
| 1 | 2 | 4 | 1 |
| 2 | 3 | 3 | 1 |
| 3 | 4 | 1 | 1 |
| 4 | 1 | 2 | 1 |
| 1 | 1 | 3 | 2 |
| 2 | 4 | 4 | 2 |
| 3 | 3 | 2 | 2 |
| 4 | 2 | 1 | 2 |
| 1 | 1 | 1 | 2 |
| 2 | 4 | 2 | 2 |
| 3 | 3 | 4 | 2 |
| 4 | 2 | 3 | 2 |
| 1 | 4 | 2 | 3 |
| 2 | 1 | 1 | 3 |
| 3 | 2 | 3 | 3 |
| 4 | 3 | 4 | 3 |
| 1 | 4 | 4 | 3 |
| 2 | 1 | 3 | 3 |
| 3 | 2 | 1 | 3 |
| 4 | 3 | 2 | 3 |
| 1 | 3 | 3 | 4 |
| 2 | 2 | 4 | 4 |
| 3 | 1 | 4 | 4 |
| 4 | 4 | 3 | 4 |
| 1 | 3 | 1 | 4 |
| 2 | 2 | 2 | 4 |
| 3 | 1 | 4 | 4 |
| 4 | 4 | 3 | 4 |

(b)

| $x_1$ | $x_2$ | $x_3$ |
|---|---|---|
| 1 | 2 | 2 |
| 3 | 4 | 3 |
| 2 | 3 | 1 |
| 2 | 3 | 3 |
| 1 | 2 | 4 |
| 4 | 1 | 4 |
| 4 | 1 | 2 |
| 3 | 4 | 1 |
| 2 | 4 | 4 |
| 4 | 2 | 1 |
| 1 | 1 | 1 |
| 4 | 2 | 3 |
| 3 | 3 | 2 |
| 3 | 3 | 4 |
| 2 | 4 | 2 |
| 1 | 1 | 3 |
| 4 | 3 | 4 |
| 4 | 3 | 2 |
| 4 | 3 | 2 |
| 1 | 4 | 2 |
| 1 | 4 | 4 |
| 3 | 1 | 4 |
| 2 | 2 | 3 |
| 3 | 2 | 1 |
| 4 | 4 | 3 |
| 2 | 2 | 2 |
| 4 | 1 | 2 |
| 3 | 1 | 2 |
| 1 | 3 | 1 |
| 2 | 2 | 4 |
| 3 | 1 | 4 |
| 1 | 3 | 3 |

(c)

| $x_1$ | $x_2$ | $x_3$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 2 | 2 |
| 1 | 2 | 1 |
| 2 | 1 | 1 |
| 2 | 2 | 2 |
| 1 | 2 | 1 |
| 2 | 1 | 2 |
| 2 | 1 | 2 |
| 1 | 1 | 1 |
| 1 | 2 | 1 |
| 1 | 1 | 2 |
| 2 | 1 | 1 |
| 2 | 2 | 1 |
| 1 | 1 | 2 |
| 2 | 2 | 2 |
| 2 | 2 | 2 |
| 1 | 1 | 2 |
| 2 | 2 | 2 |
| 2 | 1 | 1 |
| 2 | 2 | 1 |
| 1 | 1 | 1 |
| 1 | 2 | 2 |
| 1 | 1 | 2 |
| 2 | 2 | 1 |
| 2 | 1 | 1 |
| 2 | 2 | 2 |
| 2 | 2 | 2 |
| 1 | 2 | 2 |
| 1 | 1 | 2 |
| 1 | 2 | 1 |
| 1 | 2 | 1 |
| 2 | 2 | 2 |
| 1 | 2 | 2 |
| 2 | 1 | 2 |

(d)

| $x_1$ | $x_2$ | $x_3$ |
|---|---|---|
| 1 (1) | 10 (3) | 9 (3) |
| 18 (5) | 31 (8) | 23 (6) |
| 12 (3) | 22 (6) | 2 (1) |
| 13 (4) | 20 (5) | 17 (5) |
| 7 (2) | 14 (4) | 29 (8) |
| 25 (7) | 6 (2) | 25 (7) |
| 29 (8) | 4 (1) | 14 (4) |
| 21 (6) | 25 (7) | 8 (2) |
| 11 (3) | 27 (7) | 28 (7) |
| 27 (7) | 16 (4) | 3 (1) |
| 2 (1) | 2 (1) | 7 (2) |
| 32 (8) | 11 (3) | 19 (5) |
| 22 (6) | 24 (6) | 12 (3) |
| 19 (5) | 17 (5) | 32 (8) |
| 14 (4) | 30 (8) | 13 (4) |
| 6 (2) | 8 (2) | 24 (6) |
| 26 (7) | 19 (5) | 30 (8) |
| 31 (8) | 23 (6) | 15 (4) |
| 24 (6) | 12 (3) | 20 (5) |
| 8 (2) | 32 (8) | 10 (3) |
| 4 (1) | 26 (7) | 27 (7) |
| 17 (5) | 13 (4) | 6 (2) |
| 10 (3) | 3 (1) | 21 (6) |
| 16 (4) | 5 (2) | 1 (1) |
| 30 (8) | 28 (7) | 18 (5) |
| 15 (4) | 15 (4) | 16 (4) |
| 28 (7) | 29 (8) | 5 (2) |
| 23 (6) | 1 (1) | 11 (3) |
| 3 (1) | 21 (6) | 4 (1) |
| 9 (3) | 9 (3) | 26 (7) |
| 20 (5) | 7 (2) | 31 (8) |
| 5 (2) | 18 (5) | 22 (6) |

(e)

Table 2.2: (a) An OA$(32, 4^4, 2)$ denoted by $\mathbf{A}$, (b) $\mathbf{A}$ after column and symbol permutations, (c) divide $\mathbf{A}$ into submatrices $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ and $\mathbf{A}_4$ (indicated by the dashed lines) according to different symbols in column 4 of $\mathbf{A}$ and deleting column 4 and randomly shuffle the rows in each slice, (d) $\mathbf{B}$ with submatrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ and $\mathbf{B}_4$ obtained in Step 1 of the construction, (e) $\mathbf{C} = (c_{ik})$ obtained in Step 2, where each slice is a Latin hypercube of eight runs taking values in $Z_8$ after every element $c_{ik}$ is collapsed by level mapping $\lceil c_{ik}/4 \rceil$

| Group | Definition | Size |
|---|---|---|
| $H_1$ | $\alpha_{ik}=\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{ik}=\beta_{jk},\gamma_{ik}=\gamma_{jk}$, or $\alpha_{ik}=\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{il}=\beta_{jl},\gamma_{ik}=\gamma_{jk}$ | $\lambda^2 s^4(2\lambda s-1)$ |
| $H_2$ | $\alpha_{ik}=\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{ik}=\beta_{jk},\beta_{il}=\beta_{jl},\gamma_{ik}\neq\gamma_{jk},\gamma_{il}\neq\gamma_{jl}$ | $\lambda^2 s^4(s-1)^2$ |
| $H_3$ | $\alpha_{ik}=\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{ik}=\beta_{jk},\beta_{il}\neq\beta_{jl},\gamma_{ik}\neq\gamma_{jk}$, or $\alpha_{ik}=\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{ik}\neq\beta_{jk},\beta_{il}=\beta_{jl},\gamma_{il}\neq\gamma_{jl}$ | $2\lambda^2 s^5(\lambda-1)(s-1)$ |
| $H_4$ | $\alpha_{ik}=\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{ik}\neq\beta_{jk},\beta_{jl}\neq\beta_{jl}$ | $\lambda^2 s^6(\lambda-1)^2$ |
| $H_5$ | $\alpha_{ik}=\alpha_{jk},\alpha_{il}\neq\alpha_{jl},\beta_{ik}=\beta_{jk},\gamma_{ik}=\gamma_{jk}$ or $\alpha_{ik}\neq\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{il}=\beta_{jl},\gamma_{il}=\gamma_{jl}$ | $2\lambda^3 s^5(s-1)$ |
| $H_6$ | $\alpha_{ik}=\alpha_{jk},\alpha_{il}\neq\alpha_{jl},\beta_{ik}=\beta_{jk},\gamma_{ik}\neq\gamma_{jk}$, or $\alpha_{ik}\neq\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{il}=\beta_{jl},\gamma_{il}\neq\gamma_{jl}$ | $2\lambda^3 s^5(s-1)^2$ |
| $H_7$ | $\alpha_{ik}=\alpha_{jk},\alpha_{il}\neq\alpha_{jl},\beta_{ik}\neq\beta_{jk}$, or $\alpha_{ik}\neq\alpha_{jk},\alpha_{il}=\alpha_{jl},\beta_{il}\neq\beta_{jl}$ | $2\lambda^3 s^6(\lambda-1)(s-1)$ |
| $H_8$ | $\alpha_{ik}\neq\alpha_{jk},\alpha_{il}\neq\alpha_{jl}$ | $\lambda^4 s^6(s-1)^2$ |

Table 2.3: A scheme for grouping the entries in rows $i,j$ and columns $k,l$ of $\mathbf{C}$ of Lemma 2.2

| Run# | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | Run# | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 33 | 1 | 1 | 3 | 2 | 3 | 4 | 1 | 4 | 1 |
| 2 | 2 | 4 | 2 | 3 | 1 | 3 | 1 | 2 | 3 | 34 | 2 | 4 | 4 | 4 | 3 | 2 | 1 | 3 | 3 |
| 3 | 3 | 2 | 3 | 4 | 1 | 4 | 1 | 3 | 4 | 35 | 3 | 2 | 1 | 3 | 3 | 1 | 1 | 2 | 4 |
| 4 | 4 | 3 | 4 | 2 | 1 | 2 | 1 | 4 | 2 | 36 | 4 | 3 | 2 | 1 | 3 | 3 | 1 | 1 | 2 |
| 5 | 1 | 2 | 4 | 2 | 3 | 1 | 3 | 1 | 2 | 37 | 1 | 2 | 2 | 1 | 1 | 4 | 3 | 4 | 2 |
| 6 | 2 | 3 | 3 | 4 | 3 | 3 | 3 | 2 | 4 | 38 | 2 | 3 | 1 | 3 | 1 | 2 | 3 | 3 | 4 |
| 7 | 3 | 1 | 2 | 3 | 3 | 4 | 3 | 3 | 3 | 39 | 3 | 1 | 4 | 4 | 1 | 1 | 3 | 2 | 3 |
| 8 | 4 | 4 | 1 | 1 | 3 | 2 | 3 | 4 | 1 | 40 | 4 | 4 | 3 | 2 | 1 | 3 | 3 | 1 | 1 |
| 9 | 1 | 3 | 2 | 3 | 4 | 1 | 4 | 1 | 3 | 41 | 1 | 3 | 4 | 4 | 2 | 4 | 4 | 4 | 3 |
| 10 | 2 | 2 | 1 | 1 | 4 | 3 | 4 | 2 | 1 | 42 | 2 | 2 | 3 | 2 | 2 | 2 | 4 | 3 | 1 |
| 11 | 3 | 4 | 4 | 2 | 4 | 4 | 4 | 3 | 2 | 43 | 3 | 4 | 2 | 1 | 2 | 1 | 4 | 2 | 2 |
| 12 | 4 | 1 | 3 | 4 | 4 | 2 | 4 | 4 | 4 | 44 | 4 | 1 | 1 | 3 | 2 | 3 | 4 | 1 | 4 |
| 13 | 1 | 4 | 3 | 4 | 2 | 1 | 2 | 1 | 4 | 45 | 1 | 4 | 1 | 3 | 4 | 4 | 2 | 4 | 4 |
| 14 | 2 | 1 | 4 | 2 | 2 | 3 | 2 | 2 | 2 | 46 | 2 | 1 | 2 | 1 | 4 | 2 | 2 | 3 | 2 |
| 15 | 3 | 3 | 1 | 1 | 2 | 4 | 2 | 3 | 1 | 47 | 3 | 3 | 3 | 2 | 4 | 1 | 2 | 2 | 1 |
| 16 | 4 | 2 | 2 | 3 | 2 | 2 | 2 | 4 | 3 | 48 | 4 | 2 | 4 | 4 | 4 | 3 | 2 | 1 | 3 |
| 17 | 1 | 1 | 2 | 4 | 2 | 3 | 1 | 3 | 1 | 49 | 1 | 1 | 4 | 3 | 4 | 2 | 1 | 2 | 1 |
| 18 | 2 | 4 | 1 | 2 | 2 | 1 | 1 | 4 | 3 | 50 | 2 | 4 | 3 | 1 | 4 | 4 | 1 | 1 | 3 |
| 19 | 3 | 2 | 4 | 1 | 2 | 2 | 1 | 1 | 4 | 51 | 3 | 2 | 2 | 2 | 4 | 3 | 1 | 4 | 4 |
| 20 | 4 | 3 | 3 | 3 | 2 | 4 | 1 | 2 | 2 | 52 | 4 | 3 | 1 | 4 | 4 | 1 | 1 | 3 | 2 |
| 21 | 1 | 2 | 3 | 3 | 4 | 3 | 3 | 3 | 2 | 53 | 1 | 2 | 1 | 4 | 2 | 2 | 3 | 2 | 2 |
| 22 | 2 | 3 | 4 | 1 | 4 | 1 | 3 | 4 | 4 | 54 | 2 | 3 | 2 | 2 | 2 | 4 | 3 | 1 | 4 |
| 23 | 3 | 1 | 1 | 2 | 4 | 2 | 3 | 1 | 3 | 55 | 3 | 1 | 3 | 1 | 2 | 3 | 3 | 4 | 3 |
| 24 | 4 | 4 | 2 | 4 | 4 | 4 | 3 | 2 | 1 | 56 | 4 | 4 | 4 | 3 | 2 | 1 | 3 | 3 | 1 |
| 25 | 1 | 3 | 1 | 2 | 3 | 3 | 4 | 3 | 3 | 57 | 1 | 3 | 3 | 1 | 1 | 2 | 4 | 2 | 3 |
| 26 | 2 | 2 | 2 | 4 | 3 | 1 | 4 | 4 | 1 | 58 | 2 | 2 | 4 | 3 | 1 | 4 | 4 | 1 | 1 |
| 27 | 3 | 4 | 3 | 3 | 3 | 2 | 4 | 1 | 2 | 59 | 3 | 4 | 1 | 4 | 1 | 3 | 4 | 4 | 2 |
| 28 | 4 | 1 | 4 | 1 | 3 | 4 | 4 | 2 | 4 | 60 | 4 | 1 | 2 | 2 | 1 | 1 | 4 | 3 | 4 |
| 29 | 1 | 4 | 4 | 1 | 1 | 3 | 2 | 3 | 4 | 61 | 1 | 4 | 2 | 2 | 3 | 2 | 2 | 2 | 4 |
| 30 | 2 | 1 | 3 | 3 | 1 | 1 | 2 | 4 | 2 | 62 | 2 | 1 | 1 | 4 | 3 | 4 | 2 | 1 | 2 |
| 31 | 3 | 3 | 2 | 4 | 1 | 2 | 2 | 1 | 1 | 63 | 3 | 3 | 4 | 3 | 3 | 3 | 2 | 4 | 1 |
| 32 | 4 | 2 | 1 | 2 | 1 | 4 | 2 | 2 | 3 | 64 | 4 | 2 | 3 | 1 | 3 | 1 | 2 | 3 | 3 |

Table 2.4: An OA$(64, 4^9, 2)$ used in Examples 2.3, 2.4 and 2.7

| | | IID | ILHD | SLHD | OU | SU |
|---|---|---|---|---|---|---|
| | mean | 77.3183 | 77.6343 | 77.6849 | 78.2753 | 77.5661 |
| $\hat{\mu}_1$ | sd | 11.0415 | **2.4340** | **2.3614** | 9.5809 | **2.1384** |
| | RMSE | 11.0412 | 2.4329 | 2.3604 | 9.5959 | 2.1393 |
| | mean | 78.1968 | 77.6487 | 77.5819 | 77.8818 | 77.5854 |
| $\hat{\mu}_2$ | sd | 11.1352 | **2.5183** | **2.5017** | 10.1782 | **2.1752** |
| | RMSE | 11.1442 | 2.5172 | 2.5008 | 10.1763 | 2.1745 |
| | mean | 77.7801 | 77.5067 | 77.5961 | 77.2549 | 77.7175 |
| $\hat{\mu}_3$ | sd | 11.5782 | **2.4218** | **2.3523** | 9.7066 | **2.2150** |
| | RMSE | 11.5732 | 2.4247 | 2.3517 | 9.7098 | 2.2150 |
| | mean | 77.1138 | 77.6890 | 77.7036 | 77.2161 | 77.7197 |
| $\hat{\mu}_4$ | sd | 11.2919 | **2.4095** | **2.4028** | 9.8984 | **2.1064** |
| | RMSE | 11.3004 | 2.4083 | 2.4017 | 9.9043 | 2.1057 |
| | mean | 77.6022 | 77.6197 | 77.6416 | 77.6570 | 77.6472 |
| $\hat{\mu}$ | sd | 5.5519 | 1.2483 | 1.1689 | **0.4450** | **0.4363** |
| | RMSE | 5.5493 | 1.2481 | 1.1683 | 0.4448 | 0.4361 |

Table 2.5: Comparison of the sample mean, standard deviation and RMSE of $\hat{\mu}_1$, $\hat{\mu}_2$, $\hat{\mu}_3$, $\hat{\mu}_4$ and $\hat{\mu}$ for Example 2.3 over the 1000 replicates for the five methods

| | | IID | ILHD | SLHD | OU | SU |
|---|---|---|---|---|---|---|
| | mean | 77.7755 | 77.8042 | 77.6023 | 77.2948 | 77.6866 |
| $\hat{\mu}_1$ | sd | 11.5664 | **2.4418** | **2.4889** | 9.8889 | **2.1662** |
| | RMSE | 11.5621 | 2.4496 | 2.4876 | 9.8885 | 2.1671 |
| | mean | 80.2150 | 79.7500 | 79.4484 | 80.3261 | 79.6262 |
| $\hat{\mu}_2$ | sd | 12.1499 | **2.5507** | **2.5266** | 10.1182 | **2.2350** |
| | RMSE | 12.1605 | 2.5552 | 2.5287 | 10.1407 | 2.2344 |
| | mean | 75.5693 | 75.7247 | 75.7287 | 75.7332 | 75.7105 |
| $\hat{\mu}_3$ | sd | 11.1668 | **2.3312** | **2.3675** | 9.4562 | **2.0813** |
| | RMSE | 11.1634 | 2.3309 | 2.3671 | 9.4517 | 2.0817 |
| | mean | 74.5181 | 74.1663 | 74.0348 | 73.8067 | 74.0370 |
| $\hat{\mu}_4$ | sd | 10.3866 | **2.3689** | **2.3191** | 9.0644 | **2.0508** |
| | RMSE | 10.3913 | 2.3698 | 2.3182 | 9.0636 | 2.0500 |
| | mean | 77.0195 | 76.8613 | 76.7036 | 76.7902 | 76.7650 |
| $\hat{\mu}$ | sd | 5.8785 | 1.2139 | 1.1744 | **0.4501** | **0.4542** |
| | RMSE | 5.6483 | 1.2173 | 1.1366 | 0.4634 | 0.4576 |

Table 2.6: Comparison of the sample mean, standard deviation and RMSE of $\hat{\mu}_1$, $\hat{\mu}_2$, $\hat{\mu}_3$, $\hat{\mu}_4$ and $\hat{\mu}$ for Example 2.4 over the 1000 replicates for the five methods

| Run# | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | Run# | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 19 | 1 | 2 | 1 | 3 | 3 |
| 2 | 2 | 2 | 2 | 2 | 2 | 20 | 2 | 3 | 2 | 1 | 1 |
| 3 | 3 | 3 | 3 | 3 | 3 | 21 | 3 | 1 | 3 | 2 | 2 |
| 4 | 1 | 1 | 1 | 1 | 2 | 22 | 1 | 2 | 2 | 3 | 3 |
| 5 | 2 | 2 | 2 | 2 | 3 | 23 | 2 | 3 | 3 | 1 | 1 |
| 6 | 3 | 3 | 3 | 3 | 1 | 24 | 3 | 1 | 1 | 2 | 2 |
| 7 | 1 | 1 | 2 | 3 | 1 | 25 | 1 | 3 | 2 | 1 | 2 |
| 8 | 2 | 2 | 3 | 1 | 2 | 26 | 2 | 1 | 3 | 2 | 3 |
| 9 | 3 | 3 | 1 | 2 | 3 | 27 | 3 | 2 | 1 | 3 | 1 |
| 10 | 1 | 1 | 3 | 2 | 1 | 28 | 1 | 3 | 2 | 2 | 2 |
| 11 | 2 | 2 | 1 | 3 | 2 | 29 | 2 | 1 | 3 | 3 | 3 |
| 12 | 3 | 3 | 2 | 1 | 3 | 30 | 3 | 2 | 1 | 1 | 1 |
| 13 | 1 | 2 | 3 | 1 | 3 | 31 | 1 | 3 | 3 | 3 | 2 |
| 14 | 2 | 3 | 1 | 2 | 1 | 32 | 2 | 1 | 1 | 1 | 3 |
| 15 | 3 | 1 | 2 | 3 | 2 | 33 | 3 | 2 | 2 | 2 | 1 |
| 16 | 1 | 2 | 3 | 2 | 1 | 34 | 1 | 3 | 1 | 2 | 3 |
| 17 | 2 | 3 | 1 | 3 | 2 | 35 | 2 | 1 | 2 | 3 | 1 |
| 18 | 3 | 1 | 2 | 1 | 3 | 36 | 3 | 2 | 3 | 1 | 2 |

Table 2.7: An OA$(36, 3^5, 2)$ used in Example 2.5

| | | IID | ILHD | SLHD | OU | SU |
|---|---|---|---|---|---|---|
| | Mean | 2.1280 | 2.1417 | 2.1595 | 2.1373 | 2.1398 |
| $\hat{\mu}_1$ | sd | 0.2797 | **0.0250** | **0.0265** | 0.2314 | **0.0249** |
| | RMSE | 0.2799 | 0.0250 | 0.0328 | 0.2313 | 0.0249 |
| | Mean | 2.1511 | 2.1615 | 2.1614 | 2.1605 | 2.1597 |
| $\hat{\mu}_2$ | sd | 0.2801 | **0.0244** | **0.0266** | 0.2360 | **0.0246** |
| | RMSE | 0.2801 | 0.0244 | 0.0266 | 0.2359 | 0.0246 |
| | Mean | 2.1482 | 2.1509 | 2.1600 | 2.1548 | 2.1517 |
| $\hat{\mu}_3$ | sd | 0.2795 | **0.0264** | **0.0258** | 0.2292 | **0.0248** |
| | RMSE | 0.2794 | 0.0264 | 0.0270 | 0.2291 | 0.0248 |
| | Mean | 2.1424 | 2.1513 | 2.1603 | 2.1509 | 2.1509 |
| $\hat{\mu}$ | sd | 0.1614 | 0.0145 | 0.0076 | **0.0059** | **0.0056** |
| | RMSE | 0.1616 | 0.0145 | 0.0122 | 0.0059 | 0.0056 |

Table 2.8: Comparison of the sample mean, standard deviation and RMSE of $\hat{\mu}_1, \hat{\mu}_2$, $\hat{\mu}_3$ and $\hat{\mu}$ for Example 2.5 over the 1000 replicates for the five methods

| Run# | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | Run# | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | Run# | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 28 | 3 | 1 | 2 | 1 | 2 | 55 | 2 | 1 | 3 | 1 | 3 |
| 2 | 2 | 1 | 2 | 1 | 1 | 29 | 1 | 1 | 3 | 1 | 2 | 56 | 3 | 1 | 1 | 1 | 3 |
| 3 | 3 | 1 | 3 | 1 | 1 | 30 | 2 | 1 | 1 | 1 | 2 | 57 | 1 | 1 | 2 | 1 | 3 |
| 4 | 1 | 2 | 1 | 1 | 1 | 31 | 3 | 2 | 2 | 1 | 2 | 58 | 2 | 2 | 3 | 1 | 3 |
| 5 | 2 | 2 | 2 | 1 | 1 | 32 | 1 | 2 | 3 | 1 | 2 | 59 | 3 | 2 | 1 | 1 | 3 |
| 6 | 3 | 2 | 3 | 1 | 1 | 33 | 2 | 2 | 1 | 1 | 2 | 60 | 1 | 2 | 2 | 1 | 3 |
| 7 | 1 | 3 | 1 | 1 | 1 | 34 | 3 | 3 | 2 | 1 | 2 | 61 | 2 | 3 | 3 | 1 | 3 |
| 8 | 2 | 3 | 2 | 1 | 1 | 35 | 1 | 3 | 3 | 1 | 2 | 62 | 3 | 3 | 1 | 1 | 3 |
| 9 | 3 | 3 | 3 | 1 | 1 | 36 | 2 | 3 | 1 | 1 | 2 | 63 | 1 | 3 | 2 | 1 | 3 |
| 10 | 2 | 2 | 1 | 2 | 1 | 37 | 1 | 2 | 2 | 2 | 2 | 64 | 3 | 2 | 3 | 2 | 3 |
| 11 | 3 | 2 | 2 | 2 | 1 | 38 | 2 | 2 | 3 | 2 | 2 | 65 | 1 | 2 | 1 | 2 | 3 |
| 12 | 1 | 2 | 3 | 2 | 1 | 39 | 3 | 2 | 1 | 2 | 2 | 66 | 2 | 2 | 2 | 2 | 3 |
| 13 | 2 | 3 | 1 | 2 | 1 | 40 | 1 | 3 | 2 | 2 | 2 | 67 | 3 | 3 | 3 | 2 | 3 |
| 14 | 3 | 3 | 2 | 2 | 1 | 41 | 2 | 3 | 3 | 2 | 2 | 68 | 1 | 3 | 1 | 2 | 3 |
| 15 | 1 | 3 | 3 | 2 | 1 | 42 | 3 | 3 | 1 | 2 | 2 | 69 | 2 | 3 | 2 | 2 | 3 |
| 16 | 2 | 1 | 1 | 2 | 1 | 43 | 1 | 1 | 2 | 2 | 2 | 70 | 3 | 1 | 3 | 2 | 3 |
| 17 | 3 | 1 | 2 | 2 | 1 | 44 | 2 | 1 | 3 | 2 | 2 | 71 | 1 | 1 | 1 | 2 | 3 |
| 18 | 1 | 1 | 3 | 2 | 1 | 45 | 3 | 1 | 1 | 2 | 2 | 72 | 2 | 1 | 2 | 2 | 3 |
| 19 | 3 | 3 | 1 | 3 | 1 | 46 | 2 | 3 | 2 | 3 | 2 | 73 | 1 | 3 | 3 | 3 | 3 |
| 20 | 1 | 3 | 2 | 3 | 1 | 47 | 3 | 3 | 3 | 3 | 2 | 74 | 2 | 3 | 1 | 3 | 3 |
| 21 | 2 | 3 | 3 | 3 | 1 | 48 | 1 | 3 | 1 | 3 | 2 | 75 | 3 | 3 | 2 | 3 | 3 |
| 22 | 3 | 1 | 1 | 3 | 1 | 49 | 2 | 1 | 2 | 3 | 2 | 76 | 1 | 1 | 3 | 3 | 3 |
| 23 | 1 | 1 | 2 | 3 | 1 | 50 | 3 | 1 | 3 | 3 | 2 | 77 | 2 | 1 | 1 | 3 | 3 |
| 24 | 2 | 1 | 3 | 3 | 1 | 51 | 1 | 1 | 1 | 3 | 2 | 78 | 3 | 1 | 2 | 3 | 3 |
| 25 | 3 | 2 | 1 | 3 | 1 | 52 | 2 | 2 | 2 | 3 | 2 | 79 | 1 | 2 | 3 | 3 | 3 |
| 26 | 1 | 2 | 2 | 3 | 1 | 53 | 3 | 2 | 3 | 3 | 2 | 80 | 2 | 2 | 1 | 3 | 3 |
| 27 | 2 | 2 | 3 | 3 | 1 | 54 | 1 | 2 | 1 | 3 | 2 | 81 | 3 | 2 | 2 | 3 | 3 |

Table 2.9: An $\mathrm{OA}(81, 3^5, 2)$ used in Example 2.6

| | | IID | ILHD | SLHD | OU | SU |
|---|---|---|---|---|---|---|
| | Mean | 2.7978 | 2.7928 | 2.7852 | 2.8110 | 2.7961 |
| $\hat{\mu}_1$ | sd | 0.4986 | **0.2207** | **0.2321** | 0.4195 | **0.1315** |
| | RMSE | 0.4983 | 0.2206 | 0.2323 | 0.4195 | 0.1314 |
| | Mean | 2.5925 | 2.5909 | 2.5698 | 2.5739 | 2.5770 |
| $\hat{\mu}_2$ | sd | 0.5019 | **0.2536** | **0.2433** | 0.4116 | **0.1430** |
| | RMSE | 0.5019 | 0.2539 | 0.2432 | 0.4114 | 0.1429 |
| | Mean | 3.1020 | 3.0867 | 3.0855 | 3.0650 | 3.0924 |
| $\hat{\mu}_3$ | sd | 0.5113 | **0.2138** | **0.2168** | 0.4062 | **0.1211** |
| | RMSE | 0.5113 | 0.2137 | 0.2167 | 0.4066 | 0.1212 |
| | Mean | 2.8307 | 2.8235 | 2.8135 | 2.8166 | 2.8166 |
| $\hat{\mu}$ | sd | 0.2991 | 0.1328 | 0.1290 | **0.0696** | **0.0682** |
| | RMSE | 0.2991 | 0.1328 | 0.1291 | 0.0696 | 0.0682 |

Table 2.10: Comparison of the sample mean, standard deviation and RMSE of $\hat{\mu}_1, \hat{\mu}_2$, $\hat{\mu}_3$ and $\hat{\mu}$ for Example 2.6 over the 1000 replicates for the five methods

| | | IID | ILHD | SLHD | OU | SU |
|---|---|---|---|---|---|---|
| | mean | 77.9512 | 77.5718 | 77.7187 | 78.0804 | 77.6642 |
| $\hat{\mu}_1$ | sd | 11.5664 | **2.4042** | **2.4918** | 9.8875 | **2.2157** |
| | RMSE | 11.5642 | 2.4047 | 2.4912 | 9.8914 | 2.2146 |
| | mean | 80.1844 | 79.7101 | 79.5270 | 79.7121 | 79.5363 |
| $\hat{\mu}_2$ | sd | 11.7042 | **2.6151** | **2.5953** | 10.3270 | **2.1932** |
| | RMSE | 11.9707 | 3.3309 | 3.2046 | 10.5268 | 2.8950 |
| | mean | 75.9044 | 75.7767 | 75.7749 | 75.5833 | 75.8992 |
| $\hat{\mu}_3$ | sd | 10.6753 | **2.3695** | **2.3252** | 9.3122 | **2.0923** |
| | RMSE | 10.8210 | 3.0547 | 3.0216 | 9.5465 | 2.7637 |
| | mean | 74.1342 | 74.1875 | 74.0992 | 73.7175 | 73.9621 |
| $\hat{\mu}_4$ | sd | 10.7020 | **2.3368** | **2.3467** | 9.6971 | **2.0506** |
| | RMSE | 11.2566 | 4.1685 | 4.2474 | 10.4560 | 4.2106 |
| | mean | 77.0435 | 76.8115 | 76.7800 | 76.7733 | 76.7654 |
| $\hat{\mu}$ | sd | 5.5872 | 1.2288 | 1.1727 | **0.4264** | **0.4576** |
| | RMSE | 5.6187 | 1.4946 | 1.4678 | 0.9868 | 1.0077 |

Table 2.11: Comparison of the sample mean, standard deviation and RMSE of $\hat{\mu}_1, \hat{\mu}_2$, $\hat{\mu}_3$, $\hat{\mu}_4$ and $\hat{\mu}$ for Example 2.7 over the 1000 replicates for the five methods

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 3 | 2 | 1 | 1 |
| 2 | 3 | 1 | 1 |
| 3 | 3 | 2 | 1 |
| 2 | 1 | 2 | 1 |
| 1 | 2 | 2 | 1 |
| 2 | 2 | 3 | 1 |
| 1 | 3 | 3 | 1 |
| 3 | 1 | 3 | 1 |
| 2 | 2 | 3 | 2 |
| 1 | 3 | 3 | 2 |
| 3 | 1 | 3 | 2 |
| 1 | 1 | 1 | 2 |
| 3 | 2 | 1 | 2 |
| 2 | 3 | 1 | 2 |
| 3 | 3 | 2 | 2 |
| 2 | 1 | 2 | 2 |
| 1 | 2 | 2 | 2 |
| 3 | 3 | 2 | 3 |
| 2 | 1 | 2 | 3 |
| 1 | 2 | 2 | 3 |
| 2 | 2 | 3 | 3 |
| 1 | 3 | 3 | 3 |
| 3 | 1 | 3 | 3 |
| 1 | 1 | 1 | 3 |
| 3 | 2 | 1 | 3 |
| 2 | 3 | 1 | 3 |

(a)

| $x_1$ | $x_2$ |
|---|---|
| 3 | 4 |
| 19 | 12 |
| 18 | 25 |
| 27 | 19 |
| 14 | 1 |
| 5 | 18 |
| 11 | 13 |
| 8 | 22 |
| 24 | 8 |
| 15 | 10 |
| 6 | 23 |
| 25 | 9 |
| 2 | 5 |
| 22 | 16 |
| 12 | 27 |
| 20 | 20 |
| 17 | 2 |
| 9 | 15 |
| 23 | 21 |
| 10 | 3 |
| 4 | 17 |
| 16 | 11 |
| 1 | 24 |
| 21 | 6 |
| 7 | 7 |
| 26 | 14 |
| 13 | 26 |

(b)

| $x_1$ | $x_2$ |
|---|---|
| 1 | 2 |
| 7 | 4 |
| 6 | 9 |
| 9 | 7 |
| 5 | 1 |
| 2 | 6 |
| 4 | 5 |
| 3 | 8 |
| 8 | 3 |
| 5 | 4 |
| 2 | 8 |
| 9 | 3 |
| 1 | 2 |
| 8 | 6 |
| 4 | 9 |
| 7 | 7 |
| 6 | 1 |
| 3 | 5 |
| 8 | 7 |
| 4 | 1 |
| 2 | 6 |
| 6 | 4 |
| 1 | 8 |
| 7 | 2 |
| 3 | 3 |
| 9 | 5 |
| 5 | 9 |

(c)

Table 2.12: (a) An OA$(27, 3^4, 3)$ denoted by $\mathbf{A}$, which is divided into $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ and is further divided into $\mathbf{A}_{11}, \ldots, \mathbf{A}_{33}$, (b) a sliced U design $\mathbf{C}$ with three slices $\mathbf{C}_1, \mathbf{C}_2$ and $\mathbf{C}_3$ of nine runs (divided by the dashed lines), each of which becomes an OA$(9, 3^3, 2)$ associated with $Z_3$ after every element $a$ is collapsed by level-mapping $\lceil a/9 \rceil$, (c) each of $\mathbf{C}_1, \mathbf{C}_2$ and $\mathbf{C}_3$ is a Latin hypercube of nine runs taking values in $Z_9$ after every element $a$ is collapsed according to level-mapping $\lceil a/3 \rceil$

# Chapter 3

# Asymmetric Nested Lattice Samples

## 3.1 Motivation

Space-filling designs are widely used in computer experiments (Sacks et al., 1989a,b; Currin et al., 1991; Morris et al., 1993; Santner et al., 2003; Fang et al., 2005), numerical integration and other fields. Throughout, a design being space-filling means that when projected onto low dimensions, it achieves attractive uniformity. Popular classes of space-filling designs include ordinary Latin hypercube designs (McKay et al., 1979), orthogonal Latin hypercube designs (Ye, 1998; Steinberg and Lin, 2006; Bingham et al., 2009; Lin et al., 2009), orthogonal array-based sampling designs (Patterson, 1954; Tang, 1993) and nets (Niederreiter, 1992; Owen, 1995), among others.

Recent years have witnessed a surge of interest in using *nested space-filling designs* for a wide range of applications, including multi-fidelity computer experiments (Kennedy and O'Hagan, 2000; Qian et al., 2006; Qian, 2009), sequential evaluations (Wang, 2003; Tong, 2006; Sallaberry et al., 2008; Qian, 2009), multi-step functional fitting (Floater and Iske, 1996; Fasshauer, 2007) and linking parameters (Husslage et al.,

2003, 2005). Nested space-filling designs are two space-filling designs with the smaller design nested within the large design. Qian et al. (2009) use algebraic projections in Galois fields to construct several classes of nested space-filling designs. Another popular class is nested Latin hypercube designs (Husslage et al., 2005; Sallaberry et al., 2008; Qian, 2009) in which both the small design and the large design achieve uniformity in one-dimensional projections. By randomizing symmetric nested orthogonal arrays, Qian and Ai (2010) introduced *symmetric nested lattice samples* (SNLS's) to achieve better stratification than nested Latin hypercube designs. A pair of SNLS's associated with a symmetric nested orthogonal array of strength $t$ has a desirable nested structure and achieve uniformity in $t$ and lower dimensions. Due to the use of symmetric nested orthogonal arrays, a major limitation of SNLS's is that all axes are divided at the same scale of fineness.

To overcome this drawback, Qian et al. (2011) proposed approaches for constructing *asymmetric nested orthogonal arrays*, by using a level-collapsing approach and a replacement approach. We propose a new class of space-filling designs called *asymmetric nested lattice samples* (ANLS's), by randomizing asymmetric nested orthogonal arrays. Such designs can divide different axes at different scales of fineness. In the applications mentioned above, this flexibility is useful for situations where some factors are believed to be more important or deserve more attention than the other factors. For example, in running multi-fidelity computer models for designing a heat transfer device, the temperature of the heat source and the thermal conductivity of the solid material can have more significant effects on the heat transfer rate than the others and thus deserve more attention. In modeling the thermal dynamics of an information technology system, rack temperature rise and rack power can be of par-

ticular interest, given their significant effects on the energy efficiency of the system. In such situations, the asymmetric nature of ANLS's enables dividing the axes of the more important factors at finer scales. ANLS's are also useful for situations where different factors, by nature, require dividing their axes at different levels of finesse. For example, in prosthesis design, femoral head coverage in a low-accuracy computer code may be limited to take values from eight equally spaced intervals on some range while a high-accuracy computer code, due to more complicated mathematical structure, may be limited to take values from four equally spaced intervals on the same range. It is worth mentioning that space-filling designs with different numbers of levels for different columns were also considered in Tang (1993) and Owen (1994). The main contribution here is to randomize asymmetric nested orthogonal arrays to produce ANLS's with a desirable property.

The difference between ANLS's and SNLS's are two-fold. 1. Analogous to the well-known difference between ordinary orthogonal arrays with fixed levels (Rao, 1947; Bose and Bush, 1952; Rao, 1952) and ordinary orthogonal arrays with mixed levels (Wu, 1989; Wang and Wu, 1991; Hedayat et al., 1992; Wang, 1996), asymmetric nested orthogonal arrays used in ANLS's are constructed very differently from symmetric nested orthogonal arrays used in SNLS's. 2. The first randomization procedure for ANLS's in Section 3.3 comes from an entirely different angle from the randomization procedure for SNLS's in Qian and Ai (2010). The second randomization procedure in Section 3.3 uses nested permutations with different parameters to shuffle the levels of an asymmetric nested orthogonal array, whereas nested permutations with the same parameters are employed to randomize a symmetric nested orthogonal array in producing a pair of SNLS's.

The remainder of the chapter will unfold as follows. A formal definition of asymmetric nested orthogonal array is presented in Section 3.2. The two procedures for randomizing an asymmetric nested orthogonal arrays to generate a pair of ANLS's are presented in Section 3.3. We provide some discussion in Section 3.4.

## 3.2 Definitions and Notation

This section gives some useful definitions and notation. A symmetric orthogonal array (OA) of size $n$, $m$ constraints, $s$ levels and strength two, denoted by $OA(n, s^m, 2)$, is an $n \times m$ matrix with entries from a set of $s$ levels such that for every $n \times 2$ submatrix, all $s^2$ level combinations occurs equally often (Hedayat et al., 1999, referred to as HSS hereinafter). Throughout, we consider OA's of strength two and drop the strength parameter in $OA(n, s^m, 2)$. Mukerjee et al. (2008) introduced the concept of *nested orthogonal arrays* (NOA's), which is related to that of incomplete orthogonal arrays Hedayat et al. (1992). An NOA $OA((n_1, n_2), (s_1^d, s_2^d))$ is an $OA(n_1, s_1^d)$ containing an $OA(n_2, s_2^d)$ as a subarray, where $n_1 > n_2$ and $s_1 > s_2$. Several construction methods for symmetric NOA's were developed in Qian and Ai (2010). It is also possible to generate such arrays from ordered orthogonal arrays (Schürer and Schmid, 2010). For illustration, Table 3.1 presents an $OA((16, 4), (4^3, 2^3))$.

We now give a formal definition of asymmetric NOA's. Recall that an asymmetric orthogonal array $OA(n, s_1^{d_1} \cdots s_v^{d_v})$ of size $n$ and strength two with $d = \sum_{i=1}^{v} d_i$ is an $n \times d$ matrix where the first $d_1$ columns take values from a set of $s_1$ levels, the next $d_2$ columns take values from a set of $s_2$ levels and so on such that for every $n \times 2$ submatrix, all possible level combinations occur equally often. For $n_2 < n_1$ and $s_{i2} \leq s_{i1}$, $i = 1, \ldots, v$, suppose that $\mathbf{A}$ is an $OA(n_1, s_{11}^{d_1} \cdots s_{v1}^{d_v})$ containing a

$$\begin{pmatrix} 0101 & 001122223333 \\ 0011 & 232301230123 \\ 0110 & 233223013210 \end{pmatrix}$$

Table 3.1: An $OA((16,4),(4^3,2^3))$ (in transpose), where the first four rows form an $OA(4,2^3)$

submatrix, $\mathbf{B}$, that forms an $OA(n_2, s_{12}^{d_1} \cdots s_{v2}^{d_v})$. Then $\mathbf{A}$, or more precisely $\mathbf{B} \subset \mathbf{A}$, is called an *asymmetric NOA*, denoted by $OA((n_1,n_2),(s_{11}^{d_1} \cdots s_{v1}^{d_v}, s_{12}^{d_1} \cdots s_{v2}^{d_v}))$, where $s_{11},\ldots,s_{v1}$ are distinct but some of $s_{12},\ldots,s_{v2}$ could be identical. For $v = 1$, an $OA((n_1,n_2),(s_{11}^{d_1} \cdots s_{v1}^{d_v}, s_{12}^{d_1} \cdots s_{v2}^{d_v}))$ reduces to an $OA((n_1,n_2),(s_{11}^{d}, s_{12}^{d}))$, which is a symmetric NOA.

## 3.3 Randomization

In this section, we present two methods for randomizing an asymmetric NOA to produce a pair of ANLS's. It needs to be stressed here that these randomization procedures work for arbitrary asymmetric NOA's, not limited to those constructed in Qian et al. (2011). Here are additional definitions and notation. A uniform permutation on a set of $p$ integers is a permutation on the set, with all $p!$ possible permutations being equally probable. For $a \in R$, $\lceil a \rceil$ denotes the smallest integer no less than $a$. For an integer $n$, let $Z_n$ denote the set $\{1,\ldots,n\}$.

We first explain why in producing a pair of ANLS's, an asymmetric NOA should be randomized in a more elaborate manner than simply using uniform permutations. Hereinafter, let $\mathbf{V} \subset \mathbf{W}$ denote an $OA((n_1,n_2),(s_{11}^{d_1} \cdots s_{v1}^{d_v}, s_{12}^{d_1} \cdots s_{v2}^{d_v}))$ with

$d = \sum_{k=1}^{v} d_k$. Let $a_k$ denote the number of levels of column $k$ of $\mathbf{W}$ and let $b_k$ denote that of $\mathbf{V}$. For $k = 1, \ldots, d$, relabel the levels of column $k$ of $\mathbf{W}$, originally expressed as arbitrary symbols, so that the $a_k$ levels become $1, \ldots, a_k$, with the $b_k$ levels of column $k$ of $\mathbf{V}$ being $1, \ldots, b_k$. It is important to note that if the $a_k$ levels of column $k$ of $\mathbf{W}$ are randomized with a uniform permutation on $Z_{a_k}$, after the randomization $\mathbf{W}$ achieves uniformity in two dimensions, but its subset of points corresponding to $\mathbf{V}$ is not guaranteed to achieve attractive stratification.

The first randomization procedure is motivated by Lemma 3.1.

**Lemma 3.1.** *Let* $\mathbf{A}$ *be an* $OA(n, a_1^{d_1} \cdots a_v^{d_v})$ *of strength two with* $d = \sum_{k=1}^{v} d_k$.

*(a) Suppose that for* $k = 1, \ldots, d$, $b_k$ *divides* $a_k$. *Obtain a new array* $\mathbf{A}_1$ *by, for* $k = 1, \ldots, d$, *collapsing the* $a_k$ *symbols in column* $k$ *of* $\mathbf{A}$ *into* $b_k$ *new symbols, with each new symbol corresponding to* $a_k/b_k$ *old symbols. Then* $\mathbf{A}_1$ *is an* $OA(n, b_1^{d_1} \cdots b_v^{d_v})$ *of strength two.*

*(b) For* $k = 1, \ldots, d$, *permute the levels of column* $k$ *of* $\mathbf{A}$ *to obtain a new array* $\mathbf{A}_2$. *Then* $\mathbf{A}_2$ *is an* $OA(n, a_1^{d_1} \cdots a_v^{d_v})$ *of strength two.*

Lemma 3.1 (a) can be verified based on the definition of an asymmetric orthogonal array given in Section 3.2. Lemma 3.1 (b) is a well-known result on orthogonal arrays; see, for example, Chapter 1 of HSS.

For $k = 1, \ldots, d$, divide $Z_{a_k}$ into $b_k$ blocks, $\mathbf{c}_{k1}, \ldots, \mathbf{c}_{kb_k}$, each having $q_k = a_k/b_k$ entries, where

$$\mathbf{c}_{kl} \;=\; \{z \in Z_{a_k} : \lceil z/q_k \rceil = l\}, \quad \text{for } l = 1, \ldots, b_k. \tag{3.1}$$

The key here is to divide the $a_k$ levels of column $k$ of $\mathbf{V}$ into these $b_k$ blocks and randomize them in a block-by-block fashion. For $k = 1, \ldots, d$, the procedure is divided into two cases: Case I with $q_k > n_2/b_k$ and Case II with $q_k \leq n_2/b_k$.

For Case I, we have:

**Step 1**: Randomly assign $\mathbf{c}_{k1}, \ldots, \mathbf{c}_{kb_k}$ to form $b_k$ new groups $\mathbf{g}_{k1}, \ldots, \mathbf{g}_{kb_k}$.

**Step 2**: For $m = 1, \ldots, b_k$, replace all $n_2/b_k$ $m$'s in column $k$ of $\mathbf{V}$ with $n_2/b_k$ randomly selected entries from $\mathbf{g}_{km}$.

**Step 3**: For $m = 1, \ldots, b_k$, randomly replace all $(n_1/a_k - n_2/b_k)$ $m$'s in column $k$ of $\mathbf{W}\backslash\mathbf{V}$ by the $q_k - n_2/b_k$ entries of $\mathbf{g}_{km}$ not selected in Step 2 and $(n_1/a_k - q_k)/q_k$ copies of $\mathbf{g}_{km}$.

**Step 4**: Randomly divide the $a_k - b_k$ symbols that appear in column $k$ of $\mathbf{W}$ but not in column $k$ of $\mathbf{V}$ into $b_k$ sets, $\mathbf{h}_{k1}, \ldots, \mathbf{h}_{kb_k}$, each having $(a_k - b_k)/b_k$ entries. For $m = 1, \ldots, b_k$, replace the entries in column $k$ of $\mathbf{W}$ corresponding to each symbol in $\mathbf{h}_{km}$ with a uniform permutation on $n_1/(a_k q_k)$ copies of $\mathbf{g}_{km}$.

For Case II, we have:

**Step 1**: Randomly assign $\mathbf{c}_{k1}, \ldots, \mathbf{c}_{kb_k}$ to form $b_k$ new groups $\mathbf{g}_{k1}, \ldots, \mathbf{g}_{kb_k}$.

**Step 2**: For $m = 1, \ldots, b_k$, replace all $n_2/b_k$ $m$'s in column $k$ of $\mathbf{V}$ with a uniform permutation on $n_2/(b_k q_k)$ copies of $\mathbf{g}_{km}$.

**Step 3**: For $m = 1, \ldots, b_k$, randomly replace all $(n_1/a_k - n_2/b_k)$ $m$'s in column $k$ of $\mathbf{W}\backslash\mathbf{V}$ by $(n_1/a_k - n_2/b_k)/q_k$ copies of $\mathbf{g}_{km}$.

**Step 4**: Randomly divide the $a_k - b_k$ symbols that appear in column $k$ of $\mathbf{W}$ but not in column $k$ of $\mathbf{V}$ into $b_k$ sets, $\mathbf{h}_{k1}, \ldots, \mathbf{h}_{kb_k}$, each having $(a_k - b_k)/b_k$ entries. For $m = 1, \ldots, b_k$, replace the entries in column $k$ of $\mathbf{W}$ corresponding to each symbol in $\mathbf{h}_{km}$ with a uniform permutation on $n_1/(a_k q_k)$ copies of $\mathbf{g}_{km}$.

Let $\mathbf{V}^* \subset \mathbf{W}^*$ denote the pair of nested arrays after the foregoing algorithm. Though the algorithm looks somewhat abstract, the basic ideas behind the steps are clear: Step 1 randomizes the $b_k$ groups of the levels of column $k$ of $\mathbf{W}$; Step 2 randomizes the symbols in column $k$ of $\mathbf{V}$; Step 3 randomizes the symbols in column $k$ of $\mathbf{W}\backslash\mathbf{V}$ that also appear in column $k$ of $\mathbf{V}$; and Step 4 randomizes the symbols in column $k$ of $\mathbf{W}\backslash\mathbf{V}$ that do not appear in column $k$ of $\mathbf{V}$.

Using $\mathbf{V}^* \subset \mathbf{W}^*$, obtain an $n_1 \times d$ array $\mathbf{D}_1$ through

$$x_{ik} = a_k^{-1}[w_{ik}^* - u_{ik}],\ i = 1, \ldots, n_1,\ k = 1, \ldots, d, \tag{3.2}$$

where $w_{ik}^*$ is the $(i, k)$ th entry of $\mathbf{W}^*$, $x_{ik}$ is the $(i, k)$th entry of $\mathbf{D}_1$, the $u_{ik}$ are the independent $U[0, 1)$ random variables, and the $w_{ik}^*$ and the $u_{ik}$ are mutually independent. Let $\mathbf{D}_2$ be the subset of points in $\mathbf{D}_1$ corresponding to $\mathbf{V}^*$. Proposition 3.1 summarizes the space-filling properties of $\mathbf{D}_2 \subset \mathbf{D}_1$.

**Proposition 3.1.** *Consider $\mathbf{D}_2 \subset \mathbf{D}_1$ constructed above. Then we have that*

*(a) the array $\mathbf{D}_1$ achieves two-dimensional stratification on the $b_{k_1} \times b_{k_2}$ grids when projected onto factors $k_1$ and $k_2$, and achieves one-dimensional stratification with respect to the $a_k$ equally spaced intervals on $(0, 1]$ when projected onto factor $k$;*

*(b) the array $\mathbf{D}_2$ achieves two-dimensional stratification on the $b_{k_1} \times b_{k_2}$ grids when projected onto two factors $k_1$ and $k_2$, and achieves one-dimensional stratification*

*with respect to the $b_k$ equally spaced intervals on $(0, 1]$ when projected onto factor $k$.*

$$\begin{pmatrix} 1122 & 1111112222223333333344444444555555556666666677777777888888888 \\ 1212 & 3456783456781234567812345678123456781234567812345678123456781234567812345678 \\ 1221 & 3412344321433412341243214321123412342143214334123412432143214321 \end{pmatrix}$$

Table 3.2: An $OA((64, 4), (8^2 4^1, 2^3))$ (in transpose) for Example 3.1, where $a_1 = 8, a_2 = 8, a_3 = 4$ and $b_1 = 2, b_2 = 2, b_3 = 2$

$$\begin{pmatrix} 8632 & 85577612414313421423142332147586685765877685432131427586675 8 \\ 2815 & 25526117617236467453473883844537516438256373171872822646845 1 \\ 3214 & 31244224311431243113234224312442133242143113412431132442233 1 \end{pmatrix}$$

Table 3.3: The array (in transpose) obtained by randomizing the NOA in Table 3.2 for Example 3.1 using the first procedure

**Example 3.1.** *Let $\mathbf{V} \subset \mathbf{W}$ be the $OA((64, 4), (8^2 4^1, 2^3))$ given in Table 3.2. Using the foregoing procedure to randomize $\mathbf{V} \subset \mathbf{W}$ produces a pair of nested arrays $\mathbf{V}^* \subset \mathbf{W}^*$ given in Table 3.3. The step-to-step randomization of column 1 is described in detail. Since $q_1 = 4$ is greater than $n_2/b_1 = 4/2 = 2$, Case I applies.*

***Step 1**: Assign $\mathbf{c}_{11} = (1, 2, 3, 4)$ and $\mathbf{c}_{11} = (5, 6, 7, 8)$ to obtain $\mathbf{g}_{11} = (5, 6, 7, 8)$ and $\mathbf{g}_{12} = (1, 2, 3, 4)$.*

***Step 2**: The two 1's in column 1 of $\mathbf{V}$ are replaced with 8 and 6, respectively, randomly taken from $\mathbf{g}_{11}$, and the two 2's in the column are replaced with 3 and 2, respectively, randomly taken from $\mathbf{g}_{12}$.*

***Step 3****: Since the two 1's of* $\mathbf{V}$ *are replaced with 8 and 6 from* $\mathbf{g}_{11}$*, the remaining six 1's are replaced with* $8, 5, 5, 7, 7, 6$*, respectively, which form a permutation of 5, 7 and* $\mathbf{g}_{11}$*. The six 2's in the column are replaced by* $1, 2, 4, 1, 4, 3$*, respectively, which form a permutation on 1, 4 and* $\mathbf{g}_{12}$*.*

***Step 4****: Six symbols,* $3, 4, 5, 6, 7, 8$*, in column 1 of* $\mathbf{V}\backslash\mathbf{W}$ *do not appear in column 1 of* $\mathbf{V}$*. These symbols are randomized in this step. Divide them into two sets,* $\mathbf{h}_{11} = (5, 6, 8)$ *and* $\mathbf{h}_{12} = (3, 4, 7)$*. The entries in column 1 of* $\mathbf{V}\backslash\mathbf{W}$ *corresponding to the symbols in* $\mathbf{h}_{11}$ *are replaced with a uniform permutation on two copies of* $\mathbf{g}_{11}$*. Specifically, the eight 5's in the column are replaced with* $7, 5, 8, 6, 6, 8, 5, 7$*, respectively, the eight 6's in the column are replaced with* $6, 5, 8, 7, 7, 6, 8, 5$*, respectively, and the eight 8's in the column are replaced with* $7, 5, 8, 6, 6, 7, 5, 8$*, respectively. The entries in the column corresponding to the symbols in* $\mathbf{h}_{12}$ *are replaced with a uniform permutation on two copies of* $\mathbf{g}_{12}$*. Specifically, the eight 3's are replaced with* $1, 3, 4, 2, 1, 4, 2, 3$*, respectively, the eight 4's are replaced with* $1, 4, 2, 3, 3, 2, 1, 4$*, respectively, and the eight 7's are replaced with* $4, 3, 2, 1, 3, 1, 4, 2$*, respectively.*

*Figures 3.1 and 3.2 depict the bivariate projections of* $\mathbf{D}_2 \subset \mathbf{D}_1$ *from Proposition 3.1 associated with* $\mathbf{V}^* \subset \mathbf{W}^*$*. In Figure 3.1,* $\mathbf{D}_1$ *achieves two-dimensional stratification on the* $2 \times 2$ *grids for any two factors, columns 1 and 2 of the design achieve one-dimensional stratification with respect to the eight equally spaced intervals on* $(0, 1]$ *and column 3 of the design achieves one-dimensional stratification with respect to the four equally spaced intervals on* $(0, 1]$*. In Figure 3.2,* $\mathbf{D}_2$ *achieves two-dimensional stratification on the* $2\times2$ *grids and each column of the design achieves one-dimensional*

*stratification with respect to the two equally spaced intervals on* $(0, 1]$.

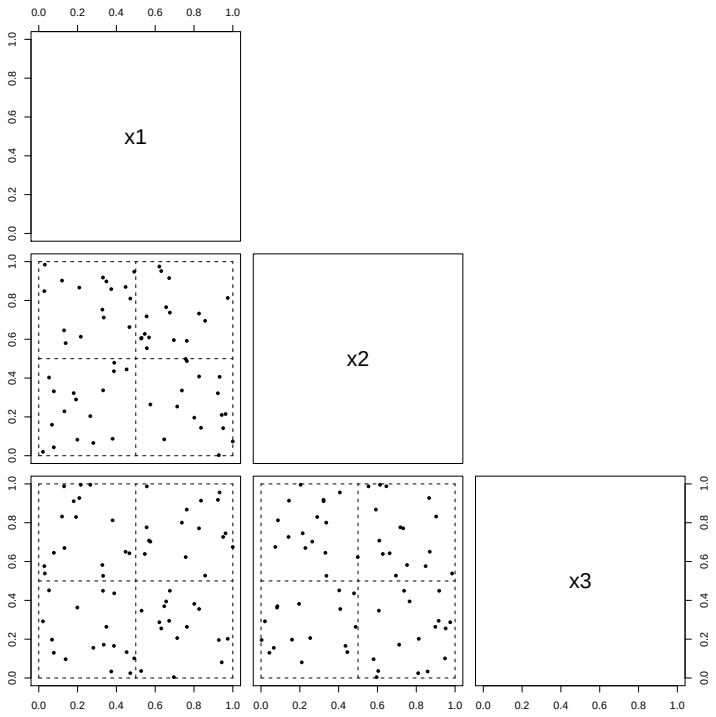


Fig. 3.1.— Bivariate projections of $\mathbf{D}_1$ in Example 3.1.

**Example 3.2.** *Let* $l(\mathbf{x})$ *and* $h(\mathbf{x})$ *denote a low-accuracy computer experiment and a high-accuracy computer experiment, respectively. Assume* $l(\mathbf{x}) = \sum_{i=1}^{3} e^{x_i+x_1}[c_i + x_i - ln(\sum_{j=1}^{3} e^{x_j})]$ *with* $c_1 = -5.914$, $c_2 = -24.721$ *and* $c_3 = -14.986$ *(Jin et al., 2001). Assume* $h(\mathbf{x}) = \sum_{i=1}^{3} e^{x_i+x_1}[d_i + x_i - ln(\sum_{j=1}^{3} e^{x_j})]$ *with* $d_1 = -8$, $d_2 = -26$ *and*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 4 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 4 & 2 & 3 & 1 & 4 & 2 & 3 & 1 & 4 & 2 & 3 & 1 & 4 & 2 & 3 \\ 1 & 4 & 2 & 3 & 4 & 1 & 3 & 2 & 2 & 3 & 1 & 4 & 3 & 2 & 4 & 1 \end{pmatrix}$$

Table 3.4: An $OA(16, 4^3)$ (in transpose) with $b_1 = 4, b_2 = 4, b_3 = 4$ embedded in an $OA(256, 16^2 8^1)$ with $a_1 = 16, a_2 = 16, a_3 = 8$ for Example 3.2

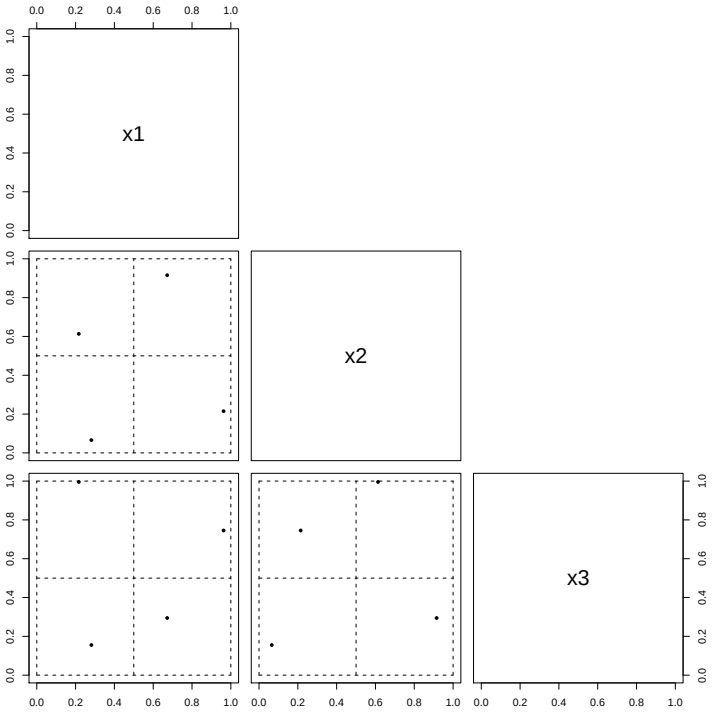


Fig. 3.2.— Bivariate projections of $\mathbf{D}_2$ in Example 3.1.

$$\begin{pmatrix} 16 & 15 & 14 & 13 & 11 & 12 & 10 & 9 & 1 & 4 & 3 & 2 & 6 & 7 & 8 & 5 \\ 9 & 3 & 6 & 15 & 12 & 1 & 7 & 14 & 11 & 4 & 5 & 16 & 10 & 2 & 8 & 13 \\ 4 & 6 & 8 & 1 & 6 & 3 & 2 & 8 & 7 & 1 & 3 & 5 & 2 & 7 & 5 & 4 \end{pmatrix}$$

Table 3.5: The array $\mathbf{V}^*$ (in transpose) in Example 3.2

$d_3 = -16$. *For* $l$ *and* $h$, *assume the distribution of* $\mathbf{x}$ *is the uniform measure on* $(0,1]^3$. *By using a very large Latin hypercube design,* $\mu_l = E[l(\mathbf{x})]$ *and* $\mu_h = E[h(\mathbf{x})]$ *are found to be -145.85 and -159.38, respectively.*

*Let* $\mathbf{V} \subset \mathbf{W}$ *be the* $OA((256,16),(16^2 8^1, 4^3))$ *generated by collapsing the third column of an* $OA((256,16),(16^3,4^3))$, *obtained by using the subfield method (Qian and Ai, 2010), into eight levels. Table 3.4 presents* $\mathbf{V}$. *Table 3.5 presents* $\mathbf{V}^*$ *obtained by applying the foregoing randomization procedure to* $\mathbf{V}$. *The randomization of column 1 is described below. Since* $q_1 = 4$ *equals* $n_2/b_1 = 16/4 = 4$, *Case II applies.*

***Step 1****: Assign* $\mathbf{c}_{11}, \mathbf{c}_{12}, \mathbf{c}_{13}$ *and* $\mathbf{c}_{14}$ *to obtain* $\mathbf{g}_{11} = (13,14,15,16)$, $\mathbf{g}_{12} = (1,2,3,4)$, $\mathbf{g}_{13} = (5,6,7,8)$ *and* $\mathbf{g}_{14} = (1,2,3,4)$.

***Step 2****: The 1's, 2's, 3's and 4's in column 1 of* $\mathbf{V}$ *are randomized in this step. The four 1's in column 1 of* $\mathbf{V}$ *are replaced with* $16,15,14,13$, *respectively, which form a permutation on* $\mathbf{g}_{11}$, *the four 2's in the column are replaced with* $1,4,3,2$, *respectively, which form a permutation on* $\mathbf{g}_{12}$, *the four 3's in the column are replaced with* $6,7,8,5$, *respectively, which form a permutation on* $\mathbf{g}_{13}$, *and the four 4's are replaced with* $11,12,10,9$, *respectively, which form a permutation on* $\mathbf{g}_{14}$.

***Step 3****: Since the four 1's are replaced with* $\mathbf{g}_{11}$ *in Step 2, the remaining 12 1's are replaced with a uniform permutation on three copies of* $\mathbf{g}_{11}$. *Similarly, the remaining 12 2's are replaced with a uniform permutation on three copies of* $\mathbf{g}_{12}$, *the remaining 12 3's are replaced with a uniform permutation on three copies of* $\mathbf{g}_{13}$, *and the remaining 12 4's are replaced with a uniform permutation on three copies of* $\mathbf{g}_{14}$.

***Step 4***: *Note that 12 symbols,* $5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16$*, in column 1 of* $\mathbf{V}\backslash\mathbf{W}$ *do not appear in column 1 of* $\mathbf{V}$*. These symbols are randomized in this step. Divide them into four sets,* $\mathbf{h}_{11} = (9, 15, 16)$*,* $\mathbf{h}_{12} = (7, 10, 12)$*,* $\mathbf{h}_{13} = (8, 11, 13)$ *and* $\mathbf{h}_{14} = (5, 6, 14)$*. The entries in column 1 of* $\mathbf{V}\backslash\mathbf{W}$ *corresponding to the symbols in* $\mathbf{h}_{11}$ *are replaced with a uniform permutation on four copies of* $\mathbf{g}_{11}$*, those corresponding to the symbols in* $\mathbf{h}_{12}$ *are replaced with a uniform permutation on four copies of* $\mathbf{g}_{12}$*, those corresponding to the symbols in* $\mathbf{h}_{13}$ *are replaced with a uniform permutation on four copies of* $\mathbf{g}_{13}$ *and those corresponding to the symbols in* $\mathbf{h}_{14}$ *are replaced with a uniform permutation on four copies of* $\mathbf{g}_{14}$*.*

*Figure 3.3 and 3.4 present the bivariate projections of* $\mathbf{D}_2 \subset \mathbf{D}_1$ *from Proposition 3.1 associated with* $\mathbf{V}^* \subset \mathbf{W}^*$*. In Figure 3.3,* $\mathbf{D}_1$ *achieves two-dimensional stratification on the* $4 \times 4$ *grids for any two factors, columns 1 and 2 of the design achieve one-dimensional stratification with respect to the sixteen equally spaced levels on* $[0, 1)$ *and column 3 of the design achieves one-dimensional stratification with the respect to the eight evenly spaced intervals on* $[0, 1)$*. In Figure 3.3,* $\mathbf{D}_2$ *achieves two-dimensional stratification on the* $4 \times 4$ *grids for any two factors and each column achieves one-dimensional stratification with respect to the four equally spaced intervals on* $[0, 1)$*.*

*We compare the ANLS method with three other methods of the same run sizes. (i) IID: take* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *to be two IID samples; (ii) LHD: take* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *to be two ordinary Latin hypercube designs (McKay et al., 1979); and (iii) NLHD: take* $\mathbf{D}_2 \subset \mathbf{D}_1$ *to be a pair of nested Latin hypercube designs (Qian, 2009). Each method*

*is replicated 200 times to compute* $\hat{\mu}_h = 16^{-1}\sum_{i=1}^{16} h(\mathbf{x}_i)$ *and* $\hat{\mu}_l = 256^{-1}\sum_{i=1}^{256} l(\mathbf{x}_i)$, *where* $\mathbf{x}_1, \ldots, \mathbf{x}_{256}$ *are the runs of* $\mathbf{D}_1$ *with the first 16 runs corresponding to* $\mathbf{D}_2$. *Table 3.6 presents the sample means, sample standard deviations and RMSE's of* $\hat{\mu}_l$ *and* $\hat{\mu}_h$ *over the 200 replicates for the four methods. This table indicates that the ANLS method performs the best in terms of the RMSE. Compared with the LHD and NLHD methods, the ANLS method achieves some 40 % reduction in the RMSE for both* $l$ *and* $h$. *The LHD and NLHD methods give very similar RMSEs, empirically validating a theoretical result derived in (Qian, 2009).*

| | | IID | LHD | NLHD | ANLS |
|---|---|---|---|---|---|
| | Mean | -146.0189 | -145.9049 | -145.8745 | -145.8531 |
| $l$ | Sd | 3.4160 | 0.4847 | 0.4767 | 0.2823 |
| | RMSE | 3.4016 | 0.4862 | 0.4745 | 0.2825 |
| | Mean | -158.8190 | -159.5671 | -159.4366 | -159.3980 |
| $h$ | Sd | 14.3294 | 2.4234 | 2.4254 | 1.4119 |
| | RMSE | 14.2663 | 2.4179 | 2.4240 | 1.4071 |

Table 3.6: Comparison of the sample means, sample standard deviations and RMSEs of $\hat{\mu}_l$ and $\hat{\mu}_h$ over the 200 replicates computed by four different methods for Example 3.2

We now turn to the second randomization procedure. Recall that for an integer $m \geq 1$, $Z_m$ denotes the set $\{1, \ldots, m\}$. Let $a > b \geq 1$ be two integers where $b$ divides $a$. Define $c = a/b$. Following (Qian, 2009), a nested permutation $\boldsymbol{\pi}_{\text{np}}^{a,b} = \boldsymbol{\pi}_{\text{np}} = (\pi_{\text{np}}(1), \ldots, \pi_{\text{np}}(a))$ is generated as follows.

**Step 1:** Draw a uniform permutation $\boldsymbol{\lambda} = (\lambda(1), \ldots, \lambda(b))$ on $Z_b$.

**Step 2:** For $i = 1, \ldots, b$, draw $\pi_{\text{np}}(i)$ from the discrete uniform distribution with support $\{(\lambda(i) - 1)c + 1, \cdots, \lambda(i)c\}$.

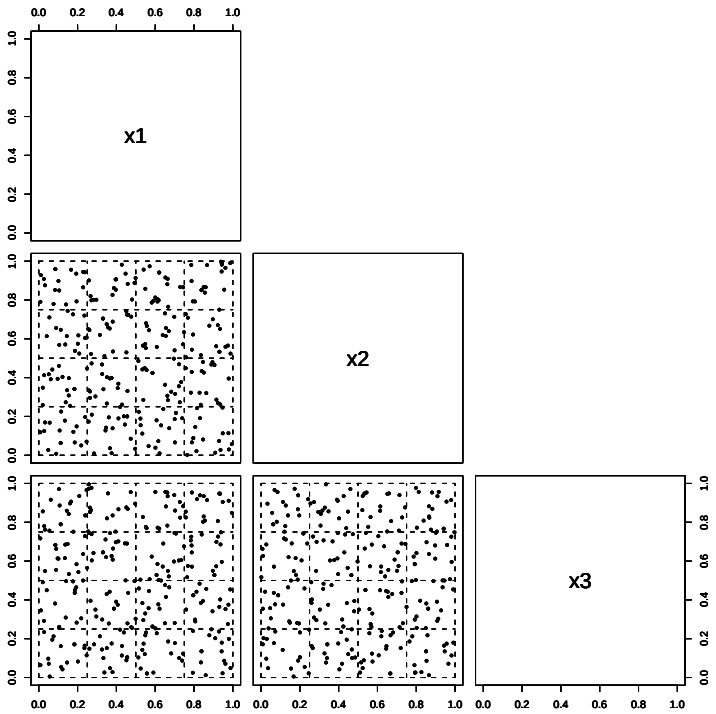


Fig. 3.3.— Bivariate projections of $\mathbf{D}_1$ in Example 3.2.

**Step 3:** Obtain $(\pi_{\rm np}(b+1), \ldots, \pi_{\rm np}(a))$ as a uniform permutation on the intersection of $Z_a$ and the set theoretic complement of $\{\pi_{\rm np}(1), \ldots, \pi_{\rm np}(b)\}$.

The term "nested permutation" suggests that, concerning the first $b$ elements of a $\boldsymbol{\pi}_{\rm np}^{a,b}$, $(\lceil \pi_{\rm np}(1)/c \rceil, \ldots, \lceil \pi_{\rm np}(b)/c \rceil)$ constitute a permutation on $Z_b$. Such a permutation is essentially Owen's randomization method for nets (Owen, 1995).

For $k = 1, \ldots, d$, this procedure uses a nested permutation $\boldsymbol{\pi}_{\rm np}^{a_k,b_k}$ to permute the $a_k$ levels of column $k$ of $\mathbf{W}$ such that, after the randomization, one and only one of the $b_k$ levels of column $k$ of $\mathbf{V}$ falls within each of the $b_k$ blocks defined by $1, \ldots, q_k; q_k + 1, \ldots, 2q_k; \ldots; (b_k - 1)q_k + 1, \ldots, b_k q_k$, where $q_k = a_k/b_k$. Precisely, a pair of ANLS's based on $\mathbf{V} \subset \mathbf{W}$ is generated as follows. Obtain an $n_1 \times d$ array $\mathbf{D}_1$

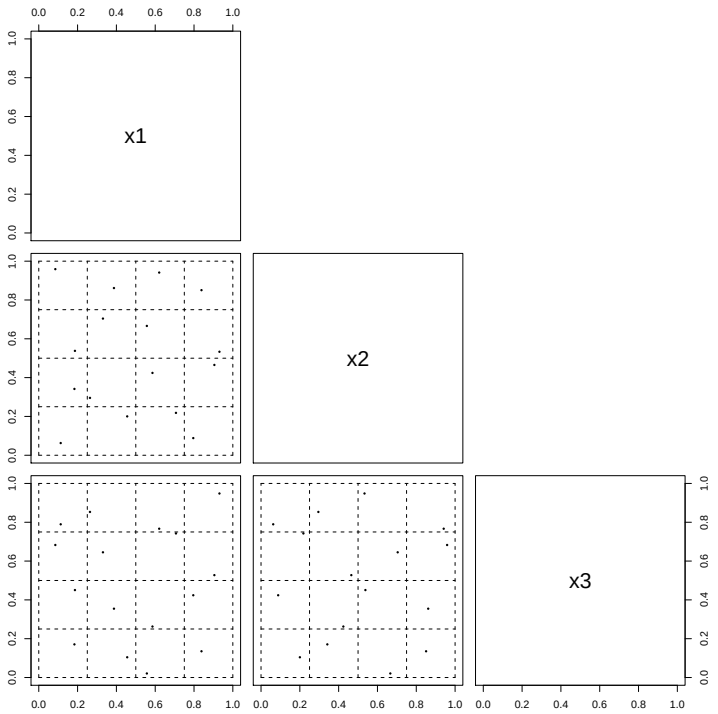


Fig. 3.4.— Bivariate projections of $\mathbf{D}_2$ in Example 3.2.

through

$$x_{ik} = a_k^{-1}[\boldsymbol{\eta}_k(w_{ik}) - u_{ik}],\ i = 1, \ldots, n_1,\ k = 1, \ldots, d, \tag{3.3}$$

where $w_{ik}$ is the $(i,k)$th entry of $\mathbf{W}$, $x_{ik}$ is the $(i,k)$th entry of $\mathbf{D}_1$, $\boldsymbol{\eta}_k$ is a nested permutation $\boldsymbol{\pi}_{\rm np}^{a_k,b_k}$, the $\boldsymbol{\eta}_k$ are obtained independently, the $u_{ik}$ are independent $U[0,1)$ random variables, and the $\boldsymbol{\eta}_k$ and the $u_{ik}$ are mutually independent. In Qian and Ai (2010), a symmetric NOA is randomized by nested permutations $\boldsymbol{\pi}_{\rm np}$ with the same values of $a$ and $b$, whereas in (3.3) the columns of the asymmetric NOA are randomized by nested permutations with different parameters. This difference is due to the asymmetric nature of $\mathbf{V} \subset \mathbf{W}$. Let $\mathbf{D}_2$ be the subset of points in $\mathbf{D}_1$ corresponding to $\mathbf{V}$. Then $\mathbf{D}_2 \subset \mathbf{D}_1$ provide a pair of ANLS's with attractive stratification in which

different axes can be divided at different levels of fineness. We make this precise in Proposition 3.2.

**Proposition 3.2.** *Consider* $\mathbf{D}_2 \subset \mathbf{D}_1$ *constructed above. Then we have that*

*(a) the array* $\mathbf{D}_1$ *achieves stratification on the* $a_{k_1} \times a_{k_2}$ *grids when projected onto factors* $k_1, k_2$*;*

*(b) the array* $\mathbf{D}_2$ *achieves stratification on the* $b_{k_1} \times b_{k_2}$ *grids when projected onto factors* $k_1, k_2$*.*

*Proof.* First, observe that the number of occurrence of each level combination of factors $k_1, k_2$ in $\mathbf{W}$ equals the number of points of $\mathbf{D}_1$ located in each reference square on the corresponding $a_{k_1} \times a_{k_2}$ grids. Second, because $\mathbf{W}$ is an orthogonal array with strength two, all level combinations in any two columns appear equally often. Combining these two facts together establishes part (a). Similarly, part (b) follows by noting that $\mathbf{V}$ is an orthogonal array with strength two and the points in $\mathbf{D}_2$ correspond to $\mathbf{V}$. □

$$\begin{pmatrix} 0011 & 00000011111122222222333333334444444455555555666666667777777 \\ 0101 & 23456723456701234567012345670123456701234567012345670123456701234567 \\ 0110 & 23012332103223012301321032100123012310321032230123013210321 0 \end{pmatrix}$$

Table 3.7: An $OA((64, 4), (8^2 4^1, 2^3))$ (in transpose) in Example 3.3, where the first four rows form an $OA(4, 2^3)$

**Example 3.3.** *Let* $\mathbf{V} \subset \mathbf{W}$ *be the* $OA((64, 4), (8^2 4^1, 2^3))$ *from Table 3.7. Relabel the levels* $0, 1, 2, \ldots, 7$ *of* $\mathbf{W}$ *as* $1, 2, 3, \ldots, 8$*, respectively, where the levels of* $\mathbf{V}$ *become 1 and 2. A pair of ANLS's,* $\mathbf{D}_2 \subset \mathbf{D}_1$*, is generated in (3.3) by using two independent nested permutation* $\boldsymbol{\pi}_{\mathrm{np}}^{8,2}$ *to randomize the levels of the first two columns of* $\mathbf{W}$ *and a nested permutation* $\boldsymbol{\pi}_{\mathrm{np}}^{4,2}$ *to randomize the levels of the third column of* $\mathbf{W}$*. In the bivariate projections of* $\mathbf{D}_1$*, the points are evenly distributed on the* $8 \times 8$ *grids in the dimensions of factors 1 and 2, on the* $8 \times 4$ *grids in those of factors 1 and 3, and on the* $8 \times 4$ *grids in those of factors 2 and 3. In the bivariate projections of* $\mathbf{D}_2$*, the points are evenly scattered on the* $2 \times 2$ *grids in any two dimensions.*

The two randomization procedures naturally complement each other. The first one is more sophisticated and can provide ANLS's with better variance reduction properties, whereas the second one is much easier to implement. Analogous to middle-point Latin hypercube designs (Morris and Mitchell, 1995; Ye, 1998; Steinberg and Lin, 2006; Bingham et al., 2009; Lin et al., 2009) and middle-point randomized orthogonal arrays (Owen, 1992a), middle-point ANLS's can be constructed by replacing $u_{ik}$ in (3.2) and (3.3) with 1/2.

## 3.4 Discussion

We have constructed a new class of space-filling designs, called asymmetric nested lattice samples. Such designs are generated by randomizing asymmetric nested orthogonal arrays with two sophisticated procedures. In addition to the applications described in Section 3.1, the constructed designs are useful for calibration and validation of computer models (Reese et al., 2004; Higdon et al., 2008) and sequential integration in stochastic optimization like the Monte Carlo EM algorithm.

We now remark on directions for future work. First, because the proposed randomization methods in Section 3.3 accommodate the asymmetric NOA's with strength two, their corresponding asymmetric nested lattice samples are guaranteed to achieve uniformity in two dimensions only. To obtain asymmetric nested lattice samples with better space-filling properties, one can develop methods to randomize asymmetric NOA's with strength three or higher. Second, one may be interested in constructing *optimal asymmetric nested lattice samples* guided by either the minimax or maximin distance criterion (Johnson et al., 1990; Tang, 1994; Leary et al., 2003). This extension poses significant challenges because of the four-fold requirements of optimality, nesting, low-dimensional stratification and asymmetry.

# Chapter 4

# Statistical Emulation of Multi-fidelity Simulations of Mechanical Dynamics Systems

## 4.1 Motivation

Over the last two decades, multibody dynamics has emerged as a critical tool in various engineering fields such as automotive and aeronautics. The interest of this work is to understand the system-level behavior of a complex multibody model. Consider, for example, a crawler model in Fig. 4.1, which is a subcomponent of a low-mobility hydraulic mining excavator. Extreme operating conditions can cause high mechanical stresses on crawler tracks, especially in the case of hydraulic excavators of 1,000 tons and higher. Long haulage distances, frequent place changes, and 90% machine availabilities are standard requirements in the industry. The ability to replace costly hardware prototypes in the early design phase with simulations can provide tremendous productivity gains and quality improvement. However, it is computation-

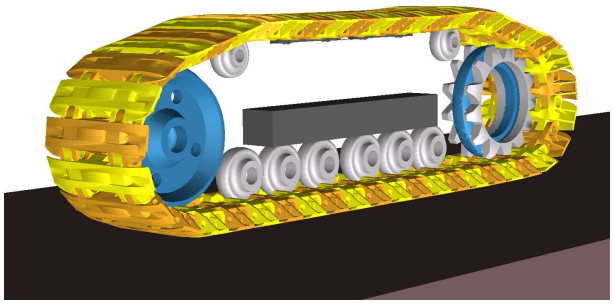

Fig. 4.1.— A track model consisting of more than 550 contacts between model components.

ally impossible to use a design process that requires a large number of simulation runs to choose a winning design and then to pass it through a rigorous validation process spanning a range of operating regimes. For example, a simulation run for modeling the subsystem dynamic behavior over 10 seconds of the crawler model in Fig. 4.1 requires 12 hours of CPU time.

To cut down computational cost, a large mechanical dynamics code is often run at two levels of accuracy, thus resulting in an accurate but time-consuming version and a less accurate but faster version. For example, the two computer experiments in Section 4.6 for studying the motion of the same slider-crank system differ in terms of computational methods and cost. In general, metamodels built with a reduced number of runs are less accurate than those with a larger number of runs. If sophisticated, computationally expensive runs are replaced with approximate ones, more data can be available. However, a metamodel built solely on approximate runs may produce inaccurate results. An effective strategy is to run a large number of approximate simulation runs and a smaller number of detailed simulation runs and then combine the two sets of results to build an accurate metamodel or emulator. This strategy has been

developed in Kennedy and O'Hagan (2000), Huang and Allen (2005) and Qian et al. (2006), among others, for integrating multi-fidelity computer experiments with scalar responses. As an extension of these methods, here we develop a statistical approach to modeling time-evolution, functional data from two computer experiments with different accuracy for studying the same mechanical dynamics system. The corresponding experimental design issue is also addressed.

The remainder of the chapter is organized as follows. Section 4.2 introduces some basics of multi-fidelity simulations of mechanical dynamics systems. Section 4.3 presents an efficient experimental design strategy for running such simulations. The issue on modeling data from such simulations is discussed in Section 4.4. Section 4.5 considers estimation and prediction of the proposed model. The proposed methodology is illustrated with an example for studying a slider-crank dynamics system in Section 4.6. Section 4.7 concludes the chapter with some discussion.

## 4.2 Basics of multi-fidelity simulations for mechanical dynamics systems

This section gives a brief description of multi-fidelity simulations of mechanical dynamics systems. Consider a mechanical dynamics system for which the state of the system at the position level is represented by a vector of generalized coordinates given by $\mathbf{q} = [q_1, \ldots, q_n]^{\mathrm{T}}$. The velocity of the system is described by the vector of generalized velocities $\dot{\mathbf{q}} = [\dot{q}_1, \ldots, \dot{q}_n]^{\mathrm{T}}$.

In the system, joints connecting bodies restrict their relative motion and impose constraints on the generalized coordinates. Kinematic constraint equations are

formulated as algebraic expressions involving generalized coordinates given by

$$\Phi(\mathbf{q}, t) = \left[ \begin{array}{ccc} \Phi_1(\mathbf{q}, t) & \ldots & \Phi_m(\mathbf{q}, t) \end{array} \right]^{\mathrm{T}} = \mathbf{0} \tag{4.1}$$

where $m$ is the total number of constraint equations to be satisfied by the generalized coordinates. Assume the $m$ constraint equations are independent and the constraints are holonomic.

The state of the mechanical system changes in time under the effect of applied forces. Its time evolution is governed by the following Lagrange multiplier form of the constrained equations of motion:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\lambda = \mathbf{Q}\left(\dot{\mathbf{q}}, \mathbf{q}, t\right). \tag{4.2}$$

Here, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{p \times p}$ is the generalized mass, and $\mathbf{Q}\left(\dot{\mathbf{q}}, \mathbf{q}, t\right) \in \mathbb{R}^{p}$ is the action force acting on the generalized coordinates $\mathbf{q} \in \mathbb{R}^{p}$, which differs from the reaction $\Phi_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\lambda$ (Haug, 1989; Shabana, 2005), and for the reaction force $\Phi_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\lambda$, $\lambda \in \mathbb{R}^{m}$ is the Lagrange multiplier associated with the kinematic constraint equations.

Equations (4.1) and (4.2) form a set of *Index 3 Differential-Algebraic Equations* (DAEs) (Brenan et al., 1989), which are not ordinary differential equations (Petzold, 1982). In general, obtaining a numerical solution of a DAE is substantially more difficult and more prone to intense numerical computation than solving an ordinary differential equation (ODE). For a detailed description of numerical integration methods on DAEs of mltibody dynamics, the reader is referred to Brenan et al. (1989), Ascher and Petzold (1998), Hairer and Wanner (1991), Potra (1993), Lubich et al. (1995) and Eich-Sollner and Fuhrer (1998).

Throughout, we will use a two-dimensional *slider-crank system* given in Fig. 4.2 to illustrate the proposed methodology. For this system, it suffices to use a set of three

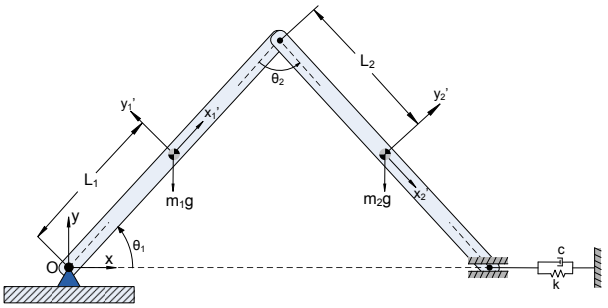


Fig. 4.2.— A Slider-crank system.

generalized coordinates $\mathbf{r}_i = [x_i, y_i]'$ and $\theta_i$ to uniquely position and orient each body $i$ in space. Using the notation in Fig. 4.2, (4.1) implies that the generalized coordinates $\mathbf{r}_1, \theta_1, \mathbf{r}_2, \theta_2$ must satisfy a set of kinematic constraint equations given by

$$\Phi(\mathbf{r}_1, \theta_1, \mathbf{r}_2, \theta_2) = \begin{bmatrix} x_1 - L_1 \cos\theta_1 \\ y_1 - L_1 \sin\theta_1 \\ x_2 - (2L_1 \cos\theta_1 + L_2 \cos\theta_2) \\ y_2 - (2L_1 \sin\theta_1 + L_2 \sin\theta_2) \\ y_2 + L_2 \sin\theta_2 \end{bmatrix} = \mathbf{0}. \tag{4.3}$$

Then the equations of motion for the slider-crank system can be computed from (4.2).

### 4.2.1 A high-accuracy computer experiment for the slider-crank system

Since DAEs are differential equations defined on sub-manifolds of $\mathbb{R}^n$, they can be reduced to an ODE problem with a smaller number of independent generalized coordinates. The number of independent generalized coordinates equals the number of degrees of freedom associated with the multibody system of interest. For example, the slider-crank system in Fig. 4.2 has one degree of freedom, and a good choice of independent generalized coordinate is $\theta_1$. The equation of motion in $\theta_1$ has a highly nonlinear form given by

$$\ddot{\theta}_1 = -\frac{\dot{\theta}_1^2 k_1 + k_2}{k_3}, \tag{4.4}$$

where $S_i = \sin\theta_i$, $C_i = \cos\theta_i$, $f = k\left(2L_1C_1 + 2L_2C_2 - 2L_2\right) + c\left(-2L_1\dot{\theta}_1S_1 - 2L_2\dot{\theta}_2S_2\right)$, and $k_i$ is a function of $L_i$, $C_i$ and $m_i$. Since $L_1\sin\theta_1 + L_2\sin\theta_2 = 0$, there is a functional relationship between $\theta_1$ and $\theta_2$ and (4.4) is a nonlinear second-order ODE in $\theta_1$. After reducing this equation to a set of first-order ODEs, a fourth-order Runge-Kutta method (Hairer and Wanner, 1991), implemented in MATLAB (MATLAB, 2010), can be used to accurately compute the time evolution of the slider-crank system. The high-accuracy numerical solution is consistent in that it satisfies the equation of motion and all the kinematic constraint equations at the position, velocity, and acceleration levels.

### 4.2.2 A low-accuracy computer experiment for the slider-crank system

The high accuracy solution in Section 4.2.1 requires long simulation times compared with a direct method that only considers the equations of motion along with the position kinematic constraint equations (Orlandea et al., 1977). The simplified method can be implemented in the commercial simulation package MSC.ADAMS (MSCsoftware, 2005) using implicit integration formulas to compute the time evolution of the multibody system. The involved integration formula for generating the low-accuracy solution is a variant of the Hilber-Hughes-Taylor (HHT) method (Hilber et al., 1977). Using MSC.ADAMS can lead to significantly shorter simulation times at the price of a somewhat less accurate solution. Most of the differences in the solution for the velocities and accelerations of the bodies is a consequence of the lack of enforcement of the kinematic constraint equations at the velocity and acceleration levels. Specifically, the low-accuracy method uses the following integration formulas

$$\begin{aligned} \mathbf{q}_{n+1} &= \mathbf{q}_n + h\dot{\mathbf{q}}_n + \tfrac{h^2}{2}\left[(1-2\beta)\,\mathbf{a}_n + 2\beta\mathbf{a}_{n+1}\right], \\ \dot{\mathbf{q}}_{n+1} &= \dot{\mathbf{q}}_n + h\left[(1-\gamma)\,\mathbf{a}_n + \gamma\mathbf{a}_{n+1}\right] \end{aligned} \tag{4.5}$$

to model the solution from $t_n$ to $t_{n+1}$ by a step size of $h$. As discussed in Hughes (1987), for second order convergence and A-stability, the HHT method requires that the parameters associated with the method be defined as $\gamma = \frac{1-2\alpha}{2}$ and $\beta = \frac{(1-\alpha)^2}{4}$ for an arbitrary $\alpha \in \left[-\frac{1}{3}, 0\right]$. The parameter $\alpha$ controls the amount of numerical damping associated with the method, ranging from maximum damping for $\alpha = -\frac{1}{3}$ to no damping $\alpha = 0$ (trapezoidal formula). From (4.5), the discretization nonlinear system to be solved at each time point is obtained based on an equation of motion and an position kinematic constraint equation given by

$$\begin{aligned} \mathbf{M}(\mathbf{q}_{n+1})\mathbf{a}_{n+1} + (1+\alpha)(\Phi_{\mathbf{q}}^{\mathrm{T}}\lambda - \mathbf{Q})_{n+1} - \alpha(\Phi_{\mathbf{q}}^{\mathrm{T}}\lambda - \mathbf{Q})_n &= \mathbf{0}, \\ \Phi(\mathbf{q}_{n+1}, t_{n+1}) &= \mathbf{0}. \end{aligned} \tag{4.6}$$

This set of nonlinear equations is solved at each time step $t_{n+1}$ for the unknowns $\mathbf{a}_{n+1}$ and $\lambda_{n+1}$. Given $\mathbf{a}_{n+1}$, the integration formulas in (4.5) are used to compute the generalized positions and velocities $\mathbf{q}_{n+1}$ and $\dot{\mathbf{q}}_{n+1}$. More details on the HHT method can be found in Cardona and Geradin (1989) and Negrut et al. (2007) applied to the context of multibody dynamics analysis.

## 4.3 Design of experiments

This section discusses the issue of how to efficiently take observations from a pair of *high-accuracy computer experiment* (HE) and *low-accuracy computer experiment* (LE) for the same mechanical dynamics system. An attractive solution to this problem is to run the HE and LE with a pair of *nested space-filling designs* that are two space-filling designs with one nested within the other. Such designs can be generated by exploiting *nesting* in orthogonal arrays. Recall that an orthogonal array $OA(n, s^d, 2)$ with $s$ levels, strength 2 is an $n \times d$ matrix with entries from $\{1, \ldots, s\}$ such that,

in every $n \times 2$ submatrix, all $s^2$ possible combinations occur equally often (Hedayat et al., 1999). A *nested orthogonal array* is a special orthogonal array that contains a subarray that becomes a smaller orthogonal array after some suitable level-collapsing. Such arrays can be constructed by using algebraic methods developed in Qian et al. (2009). A pair of nested space-filling designs $\mathcal{D}_2 \subset \mathcal{D}_1$ has three desirable properties:

**Economy:** The number of points in $\mathcal{D}_2$, $n_2$, is smaller than the number of points in $\mathcal{D}_1$, $n_1$.

**Nesting:** The design $\mathcal{D}_2$ is nested within $\mathcal{D}_1$, i.e., $\mathcal{D}_2 \subset \mathcal{D}_1$.

**Space-Filling:** Both $\mathcal{D}_1$ and $\mathcal{D}_2$ achieve uniformity in low dimensions.

These properties make a pair of nested space-filling design $\mathcal{D}_2 \subset \mathcal{D}_1$ appealing for conducting the HE and LE. First, the property of economy indicates that more runs are available for $\mathcal{D}_1$ than $\mathcal{D}_2$; more LE runs are available as the LE is less expensive. Second, the nesting property ensures that the LE result is always available at every point of $\mathcal{D}_2$. This part of data can be used for modeling and calibrating the differences between the two sources. Third, the space-filling property of $\mathcal{D}_1$ and $\mathcal{D}_2$ ensures a uniform exploration of the design space for the HE and LE. For illustration, Table 4.1 presents an $OA(64, 8^6, 2)$ with eight levels in which the first 16 runs form an $OA(16, 4^6, 2)$ after the eight levels are collapsed into four levels as follows: (1, 2) $\rightarrow$ 1, (3, 4) $\rightarrow$ 2, (5, 6) $\rightarrow$ 3, (7, 8) $\rightarrow$ 4. In addition to nested designs constructed in Qian et al. (2009) based on projections in Galois fields, other families of nested space-filling designs include *nested Latin hypercube designs* (Qian, 2009) and *nested lattice samples* (Qian and Ai, 2010).

| Run # | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 3 | 1 | 3 | 5 | 8 | 6 |
| 3 | 5 | 1 | 5 | 8 | 7 | 3 |
| 4 | 7 | 1 | 7 | 4 | 2 | 8 |
| 5 | 1 | 3 | 3 | 3 | 3 | 3 |
| 6 | 3 | 3 | 1 | 7 | 6 | 8 |
| 7 | 5 | 3 | 7 | 6 | 5 | 1 |
| 8 | 7 | 3 | 5 | 2 | 4 | 6 |
| 9 | 1 | 5 | 5 | 5 | 5 | 5 |
| 10 | 3 | 5 | 7 | 1 | 4 | 2 |
| 11 | 5 | 5 | 1 | 4 | 3 | 7 |
| 12 | 7 | 5 | 3 | 8 | 6 | 4 |
| 13 | 1 | 7 | 7 | 7 | 7 | 7 |
| 14 | 3 | 7 | 5 | 3 | 2 | 4 |
| 15 | 5 | 7 | 3 | 2 | 1 | 5 |
| 16 | 7 | 7 | 1 | 6 | 8 | 2 |
| 17 | 8 | 1 | 8 | 7 | 4 | 5 |
| 18 | 6 | 1 | 6 | 3 | 5 | 2 |
| 19 | 4 | 1 | 4 | 2 | 6 | 7 |
| 20 | 2 | 1 | 2 | 6 | 3 | 4 |
| 21 | 8 | 3 | 6 | 5 | 2 | 7 |
| 22 | 6 | 3 | 8 | 1 | 7 | 4 |
| 23 | 4 | 3 | 2 | 4 | 8 | 5 |
| 24 | 2 | 3 | 4 | 8 | 1 | 2 |
| 25 | 8 | 5 | 4 | 3 | 8 | 1 |
| 26 | 6 | 5 | 2 | 7 | 1 | 6 |
| 27 | 4 | 5 | 8 | 6 | 2 | 3 |
| 28 | 2 | 5 | 6 | 2 | 7 | 8 |
| 29 | 8 | 7 | 2 | 1 | 6 | 3 |
| 30 | 6 | 7 | 4 | 5 | 3 | 8 |
| 31 | 4 | 7 | 6 | 8 | 4 | 1 |
| 32 | 2 | 7 | 8 | 4 | 5 | 6 |

| Run # | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| 33 | 1 | 8 | 8 | 8 | 8 | 8 |
| 34 | 3 | 8 | 6 | 4 | 1 | 3 |
| 35 | 5 | 8 | 4 | 1 | 2 | 6 |
| 36 | 7 | 8 | 2 | 5 | 7 | 1 |
| 37 | 8 | 8 | 1 | 2 | 5 | 4 |
| 38 | 6 | 8 | 3 | 6 | 4 | 7 |
| 39 | 4 | 8 | 5 | 7 | 3 | 2 |
| 40 | 2 | 8 | 7 | 3 | 6 | 5 |
| 41 | 1 | 6 | 6 | 6 | 6 | 6 |
| 42 | 3 | 6 | 8 | 2 | 3 | 1 |
| 43 | 5 | 6 | 2 | 3 | 4 | 8 |
| 44 | 7 | 6 | 4 | 7 | 5 | 3 |
| 45 | 8 | 6 | 3 | 4 | 7 | 2 |
| 46 | 6 | 6 | 1 | 8 | 2 | 5 |
| 47 | 4 | 6 | 7 | 5 | 1 | 4 |
| 48 | 2 | 6 | 5 | 1 | 8 | 7 |
| 49 | 1 | 4 | 4 | 4 | 4 | 4 |
| 50 | 3 | 4 | 2 | 8 | 5 | 7 |
| 51 | 5 | 4 | 8 | 5 | 6 | 2 |
| 52 | 7 | 4 | 6 | 1 | 3 | 5 |
| 53 | 8 | 4 | 5 | 6 | 1 | 8 |
| 54 | 6 | 4 | 7 | 2 | 8 | 3 |
| 55 | 4 | 4 | 1 | 3 | 7 | 6 |
| 56 | 2 | 4 | 3 | 7 | 2 | 1 |
| 57 | 1 | 2 | 2 | 2 | 2 | 2 |
| 58 | 3 | 2 | 4 | 6 | 7 | 5 |
| 59 | 5 | 2 | 6 | 7 | 8 | 4 |
| 60 | 7 | 2 | 8 | 3 | 1 | 7 |
| 61 | 8 | 2 | 7 | 8 | 3 | 6 |
| 62 | 6 | 2 | 5 | 4 | 6 | 1 |
| 63 | 4 | 2 | 3 | 1 | 5 | 8 |
| 64 | 2 | 2 | 1 | 5 | 4 | 3 |

Table 4.1: An $OA(64, 8^6, 2)$ of eight levels $1, \ldots, 8$, where the subarray consisting of the first sixteen runs becomes an $OA(16, 4^6, 2)$ after the eight levels are collapsed into four levels as follows: (1, 2) $\rightarrow$ 1, (3, 4) $\rightarrow$ 2, (5, 6) $\rightarrow$ 3, (7, 8) $\rightarrow$ 4.

## 4.4 Modeling

Integration of data from a pair of HE and LE for the same mechanical dynamics system is not a straightforward task because the two sets of results are based on different computational methods and have different levels of accuracy. Suppose the HE and LE are conducted by using a pair of nested space-filling designs $\mathcal{D}_2 \subset \mathcal{D}_1$, with $n_2$ and $n_1$ runs, respectively. Assume $\mathcal{D}_2$ consists of the first $n_2$ runs of $\mathcal{D}_1$. Denote by $\mathbf{x}_i$ the $i$th run of $\mathcal{D}_1$. Suppose the HE and LE are measured at $u$ time points for any chosen input value. The HE output for the $i$th run of $\mathcal{D}_2$ is $\mathbf{y}_i = (y_{i1}, \ldots, y_{iu})$, and the LE output of the $i$th run of $\mathcal{D}_1$ is $\mathbf{z}_i = (z_{i1}, \ldots, z_{iu})$. The subscripts in these two vectors correspond to the $u$ time points used for measuring the evolution of the system.

Since the HE is more accurate than the LE, the objective here is to create a metamodel to produce predictions close to the HE results. To achieve this goal, we propose an approach for building a metamodel by exploiting the accuracy of the HE data and the abundance of the LE data. It uses Gaussian process (GP) models (Santner et al., 2003), and functional data analysis techniques to accommodate the fact that multi-fidelity dynamics computer experiments produce time-evolution, functional responses. The proposed approach is an extension of the methods in Kennedy and O'Hagan (2000) and Qian et al. (2006) for emulating multi-fidelity computer codes with scalar responses. The basic idea of our approach is simple: first creating a metamodel using the LE data and then refining the model by incorporating more accurate HE data. Specifically, it consists of three steps. In Step 1, based on the nested relationship $\mathcal{D}_2 \subset \mathcal{D}_1$, obtain the discrepancy between the HE and LE results

for the $i$th run of $\mathcal{D}_2$ as

$$\boldsymbol{\delta}_i = \mathbf{y}_i - \mathbf{z}_i, \tag{4.7}$$

which has $u$ entries $\delta_{i1}, \ldots, \delta_{iu}$, with $\delta_{it}$ corresponding to time point $t$. In Step 2, for $i = 1, \ldots, n_1$, express the vector $\mathbf{z}_i$ over a spline basis $\boldsymbol{h}_j$ as

$$\mathbf{z}_i \;=\; \sum_{j=1}^{q} c_{ij}\boldsymbol{h}_j. \tag{4.8}$$

Since $\mathbf{y}_i$ and $\mathbf{z}_i$ are smooth functions of time, we choose $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_q$ to be natural cubic spline basis functions (Ramsay and Silverman, 2005). Note that $\boldsymbol{h}_j$ has $u$ components since $\mathbf{z}_i$ is measured at $u$ different time points. The degree of freedom and the location of internal knots for $\boldsymbol{h}_j$ can be determined by using various techniques like leave-one-out cross-validation. For $i = 1, \ldots, n_2$, express $\boldsymbol{\delta}_i$ as

$$\boldsymbol{\delta}_i \;=\; \sum_{j=1}^{q} d_{ij}\boldsymbol{h}_j, \tag{4.9}$$

where $d_{ij}$ is the spline coefficient for the basis $\boldsymbol{h}_j$. In Step 3, for $i = 1, \ldots, n_1$ and $j = 1, \ldots, q$, model the spline coefficient $c_{ij}$ in (4.8) as

$$c_{ij} = \mathbf{f}(\mathbf{x}_i)\boldsymbol{\eta}_j + \epsilon_j, \tag{4.10}$$

where $\epsilon_j(\cdot)$ is a realization of a stationary Gaussian process with zero mean and covariance

$$\operatorname{cov}\{\epsilon_j(\mathbf{x}_i), \epsilon_j(\mathbf{x}_{i'})\} \;=\; \sigma_{j1}^2 R_{j1}(\mathbf{x}_i, \mathbf{x}_{i'}), \tag{4.11}$$

for $i \neq i'$, and $\mathbf{f}$ is a pre-specified $1 \times p$ regressor and $\boldsymbol{\eta}_j = (\eta_{1j}, \ldots, \eta_{pj})^{\mathrm{T}}$ is a vector of unknown regression parameters. For $i = 1, \ldots, n_2$ and $j = 1, \ldots, q$, model the spline coefficient $d_{ij}$ in (4.9) as

$$d_{ij} = \mathbf{f}(\mathbf{x}_i)\boldsymbol{\beta}_j + \xi_j, \tag{4.12}$$

where $\xi_j(\cdot)$ is a realization of a stationary Gaussian process with zero mean and covariance

$$\text{cov}\left\{\xi_j(\mathbf{x}_i), \xi_j(\mathbf{x}_{i'})\right\} = \sigma^2_{j2} R_{j2}(\mathbf{x}_i, \mathbf{x}_{i'}), \tag{4.13}$$

and $\boldsymbol{\beta}_j = (\beta_{1j}, \ldots, \beta_{pj})^{\mathrm{T}}$ is a vector of unknown regression parameters. In (4.11) and (4.13), the correlation function $R_{jk}$ $(k = 1, 2)$ determines the correlation of the GP at any two input values. A popular choice for $R_{jk}$ is the *Gaussian correlation function*:

$$R_{jk}(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp\left(-\sum_{m=1}^{d} \theta_{jkm} |x_{im} - x_{i'm}|^2\right). \tag{4.14}$$

The value of the function in (4.14) is determined by a weighted distance between two input values $\mathbf{x}_i$ and $\mathbf{x}_{i'}$, and the vector of correlation parameters $\boldsymbol{\theta}_{km} = (\theta_{km1}, \ldots, \theta_{kmd})^{\mathrm{T}}$ controls the smoothness of the underlying process. The Gaussian process associated with this correlation function is infinitely differentiable in the mean square sense (Santner et al., 2003), which is a reasonable assumption for many practical applications.

## 4.5 Estimation, prediction and ANOVA decomposition

Parameters in the GP models in (4.10) and (4.12) to be estimated are $\boldsymbol{\eta}_j, \boldsymbol{\beta}_j, \sigma^2_{jk}$ and $\boldsymbol{\theta}_{jk}$, $j = 1, \ldots, q$ and $k = 1, 2$. These parameters can be estimated by using the maximum likelihood method (Santner et al., 2003). For $j = 1, \ldots, q$, collect the spline coefficients $c_{1j}, \ldots, c_{n_1 j}$ in (4.8) into a vector $\boldsymbol{c}_j$. Given $\boldsymbol{\theta}_{j1}$, the *maximum likelihood estimators* (MLEs) for $\boldsymbol{\eta}_j$ and $\sigma^2_{j1}$ are

$$\begin{aligned}
\widehat{\boldsymbol{\eta}}_j &= (\mathbf{F}_1^{\mathrm{T}} \mathbf{R}_{j1}^{-1} \mathbf{F}_1)^{-1} \mathbf{F}_1^{\mathrm{T}} \mathbf{R}_{j1}^{-1} \boldsymbol{c}_j, \\
\widehat{\sigma}^2_{j1} &= \frac{(\boldsymbol{c}_j - \mathbf{F}_1 \widehat{\boldsymbol{\eta}}_j)^{\mathrm{T}} \mathbf{R}_{j1}^{-1} (\boldsymbol{c}_j - \mathbf{F}_1 \widehat{\boldsymbol{\eta}}_j)}{n}, \text{ for } j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{R}_{j1}$ is the $n_1 \times n_1$ correlation matrix with entries $R_{j1}(\mathbf{x}_i, \mathbf{x}_{i'})$, $i, i' = 1, \ldots, n_1$, defined in (4.14). Here, $\mathbf{F}_1$ is an $n_1 \times p$ regressor matrix, given as $\left[\mathbf{f}^{\mathrm{T}}(\mathbf{x}_1), \ldots, \mathbf{f}^{\mathrm{T}}(\mathbf{x}_{n_1})\right]^{\mathrm{T}}$, related to $\mathbf{f}$ defined in (4.10). The MLE $\widehat{\boldsymbol{\theta}}_{j1}$ for $\boldsymbol{\theta}_{j1}$ can be obtained by solving an optimization problem:

$$\widehat{\boldsymbol{\theta}}_{j1} = \arg\min_{\boldsymbol{\theta}} \left( n \ln \widehat{\sigma}^2_{j1} + \ln |\mathbf{R}_{j1}| \right).$$

The MLE $\widehat{\boldsymbol{\beta}}_j$ of $\boldsymbol{\beta}_j$ and the MLE $\widehat{\sigma}^2_{j2}$ of $\sigma^2_{j2}$ in (4.12) can be computed similarly. Then at any untried point $\mathbf{x}_0$, $c_{0j}$ in (4.10) can be predicted by using the *empirical best linear unbiased predictor* (EBLUP):

$$\widehat{c}_{0j} = \mathbf{f}_0 \widehat{\boldsymbol{\eta}}_j + \widehat{\mathbf{r}}^{\mathrm{T}}_{j1} \widehat{\mathbf{R}}^{-1}_{j1} (\boldsymbol{c}_j - \mathbf{F}_1 \widehat{\boldsymbol{\eta}}_j), \text{ for } j = 1, \ldots, q,$$

where $\widehat{\mathbf{r}}_{j1} = [\widehat{R}_{j1}(\mathbf{x}_0, \mathbf{x}_1), \ldots, \widehat{R}_{j1}(\mathbf{x}_0, \mathbf{x}_{n_1})]^{\mathrm{T}}$, $\mathbf{f}_0 = \mathbf{f}(\mathbf{x}_0)$, and both $\widehat{\mathbf{r}}_{j1}$ and $\widehat{\mathbf{R}}_{j1}$ are functions of $\widehat{\sigma}^2_{jk}$ and $\widehat{\boldsymbol{\theta}}_{jk}$. Similarly, at any untried point $\mathbf{x}_0$, $d_{0j}$ can be predicted by

$$\widehat{d}_{0j} = \mathbf{f}_0 \widehat{\boldsymbol{\beta}}_j + \widehat{\mathbf{r}}^{\mathrm{T}}_{j2} \widehat{\mathbf{R}}^{-1}_{j2} (\boldsymbol{d}_j - \mathbf{F}_2 \widehat{\boldsymbol{\beta}}_j), \text{ for } j = 1, \ldots, q,$$

where $\boldsymbol{d}_j = (d_{1j}, \ldots, d_{n_2 j})$ and $\mathbf{F}_2 = \left[\mathbf{f}^{\mathrm{T}}(\mathbf{x}_1), \ldots, \mathbf{f}^{\mathrm{T}}(\mathbf{x}_{n_2})\right]^{\mathrm{T}}$. Now, at any input value $\mathbf{x}_0$, the HE response $\mathbf{z}_0$ and the discrepancy $\boldsymbol{\delta}_0$ can be, respectively, predicted by

$$\begin{array}{rcl} \widehat{\mathbf{z}}_0 & = & \sum_{j=1}^{q} \widehat{c}_{0j} \boldsymbol{h}_j, \\ \widehat{\boldsymbol{\delta}}_0 & = & \sum_{j=1}^{q} \widehat{d}_{0j} \boldsymbol{h}_j, \end{array} \tag{4.15}$$

from which the final predictor of the HE response is $\widehat{\mathbf{y}}_0 = \widehat{\mathbf{z}}_0 + \widehat{\boldsymbol{\delta}}_0$. Here, $\widehat{\mathbf{y}}_0$ is a vector with entries $\widehat{y}_{01}, \ldots, \widehat{y}_{0u}$ and $\widehat{y}_{0t}$ corresponds to time point $t$. It needs to be stressed here that $\widehat{\mathbf{z}}_0$ does not predict the HE accurately since it is based solely on the LE data. In contrast, $\widehat{\mathbf{y}}_0$ integrates the data from the two sources and incorporates elaborate adjustment between the LE and HE to produce predictions closer to the HE.

Once the metamodel in (4.15) is fitted, the functional ANOVA decomposition technique can be used to visualize the effects of different factors on the response. This technique was originally proposed for Gaussian process models (Santner et al., 2003) with scalar responses. Extending it to the present situation with functional responses is straightforward. Here, the components of such a decomposition will be functions of both time $t$ and the factors. For $t = 1, \ldots, u$, the basic idea of this technique is to express $\widehat{y}_{0t}$ as the sum of the grand mean, the functional main effect for each factor, functional two-factor interaction effects and higher-order terms. For easier presentation, assume the factors $x_1, \ldots, x_d$ are scaled to $(0, 1]^d$. Then for $t = 1, \ldots, u$, the decomposition of $\widehat{y}_{0t}$ is done as follows. The grand mean over the experimental space is

$$\mu_{0,t} = \int \widehat{y}_{0t} \prod_{m=1}^{d} dx_m.$$

For $v = 1, \ldots, d$, the functional main effect of $x_j$ is

$$\mu_{v,t}(x_v) = \int \widehat{y}_{0t} \prod_{m \neq v}^{d} dx_m - \mu_{0,t}. \tag{4.16}$$

For $v, w = 1, \ldots, d, v \neq w$, the functional interaction effect of $x_v$ and $x_w$ is

$$\mu_{vw,t}(x_v, x_w) = \int \widehat{y}_{0t} \prod_{m \neq v,w}^{d} dx_m - \mu_{v,t}(x_j) - \mu_{w,t}(x_k) - \mu_{0,t}. \tag{4.17}$$

Higher-order terms can be defined similarly.

## 4.6 Case study

In this section the proposed methodology is illustrated with an example for studying the time-evolution of the slider-crank system in Fig. 4.2. The HE is based on a fourth-order Runge-Kutta integration method, implemented in MATLAB (MATLAB,

2010). Each HE run takes approximately 260 seconds to simulate the two-second evolution of the system. The dynamics of the slider-crank system is determined by a set of ODEs in (4.4). The LE is implemented in ADAMS (MSCsoftware, 2005) using an HHT integration method (Hilber et al., 1977) described in Section 4.2.2. Each LE run requires 0.4 seconds to complete the evolution over a period of two seconds of the system.

This system has six factors: mass of crank ($x_1$), half length of crank ($x_2$), mass of the connecting rod ($x_3$), half length of the connecting ($x_4$), the spring stiffness coefficient ($x_5$) and the damping coefficient ($x_6$). From a mechanical engineering point of view, these factors determine the set of conditions for the underlying ODE problem of the system. By using the algebraic method in Qian et al. (2009), a pair of nested space-filling designs $\mathcal{D}_2 \subset \mathcal{D}_1$ of 16 and 64 runs, respectively, is generated for the HE and LE. The design $\mathcal{D}_1$ is based on the $OA(64, 8^6, 2)$ given in Table 4.1, and $\mathcal{D}_2$ is associated with the first 16 runs of this orthogonal array, which form an $OA(16, 4^6, 2)$ after the eight levels are collapsed into four levels as follows: (1, 2) $\to$ 1, (3, 4) $\to$ 2, (5, 6) $\to$ 3, (7, 8) $\to$ 4. Figs. 4.3 and 4.4 present the bivariate projections of $\mathcal{D}_2$ and $\mathcal{D}_1$, where the two designs achieve maximum stratification on $4 \times 4$ and $8 \times 8$ grids, respectively. Since both the HE and LE are available for every run in $\mathcal{D}_2$, this part of data make it easier to model the discrepancy between the two sources. Table 4.2 presents the six factors of the slider-crank system. The design points are first generated on the unit hypercube $(0, 1]^6$, as given in Table 4.2, and then scaled back according to the original ranges of the inputs. For each input value, the LE or HE produces the evolution of the system every 0.02 seconds within a two-second period. Therefore, both $\mathbf{y}_i$ and $\mathbf{z}_i$ are 101 dimensional vectors. The profiles of the time evolution of the

system for the 16 HE runs are displayed in Fig. 4.5.

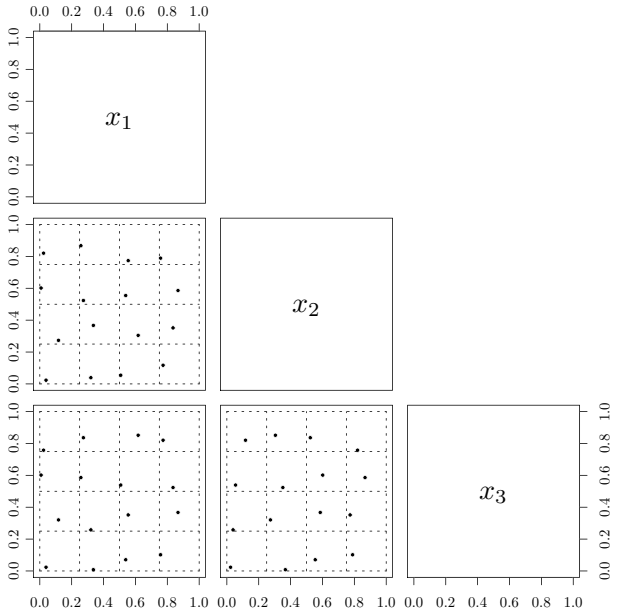


Fig. 4.3.— Some bivariate projections of $\mathcal{D}_2$ of 16 runs for the HE of the slider-crank system, where the points achieve maximum stratification on $4 \times 4$ grids.

By using cross-validation, nine equally spaced internal knots located at every 1/10 length of the time interval are chosen for the spline basis functions $\boldsymbol{h}_j$ in (4.8) and (4.9). Because $\mathbf{y}_i$ and $\mathbf{z}_i$ are smooth functions of time, the degrees of freedom of $\boldsymbol{h}_j$ are fixed at three in each interval, and thus the number of the basis functions is 11. The metamodel in (4.15) is fitted to the data of this example by following the three steps described in Section 4.4, where $\mathbf{f}$ in (4.12) is chosen to be a constant.

To validate the fitted metamodel, test data are generated by using a Latin hypercube design (McKay et al., 1979) with 512 runs. The accuracy of the model is

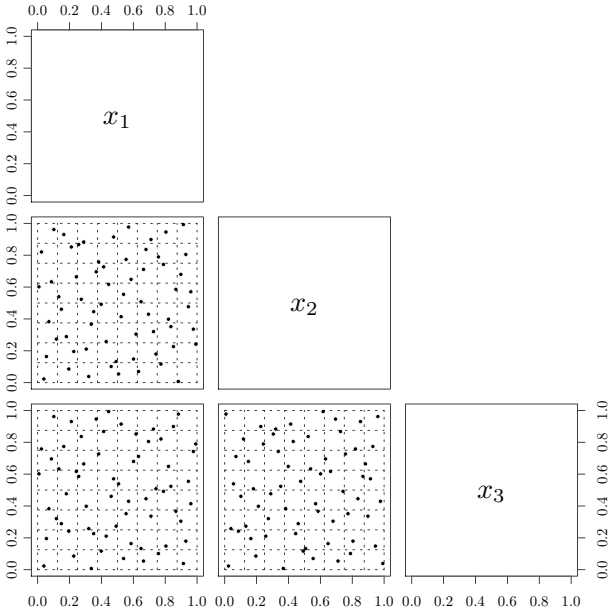


Fig. 4.4.— Some bivariate projections of $\mathcal{D}_1$ of 64 runs for the LE of the slider-crank system, where the points achieve maximum stratification on $8 \times 8$ grids.

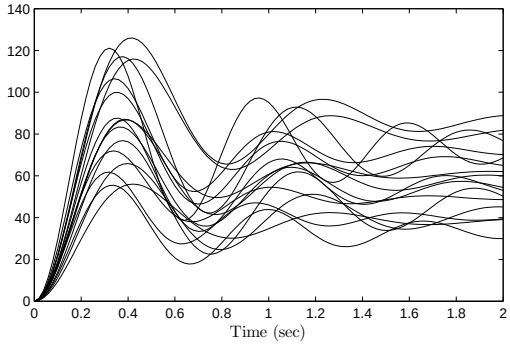


Fig. 4.5.— The time evolution of the slider-crank system modeled by the HE in Section 4.2.1 under 16 different design configurations. Each curve corresponds to one design point.

assessed by the *root mean square error* (RMSE) on the testing set:

$$\sqrt{\frac{1}{512}\sum_{i=1}^{512}\|\mathbf{y}_i^0 - \widehat{\mathbf{y}}_i^0\|^2}, \tag{4.18}$$

where $\|\cdot\|$ denotes Euclidean distance, and $\mathbf{y}_i^0$ and $\widehat{\mathbf{y}}_i^0$ denote the true and fitted HE values of the $i$th run of the testing data. The RMSE of the proposed method is 7.44, which suggests decent fit in reference to the magnitude of the HE values in Fig. 4.5. In comparison, the RMSE of the metamodel in (4.8), built solely on the LE runs from ADAMS, is 20.14, which is much larger. These results indicate that the proposed method works well for this example.

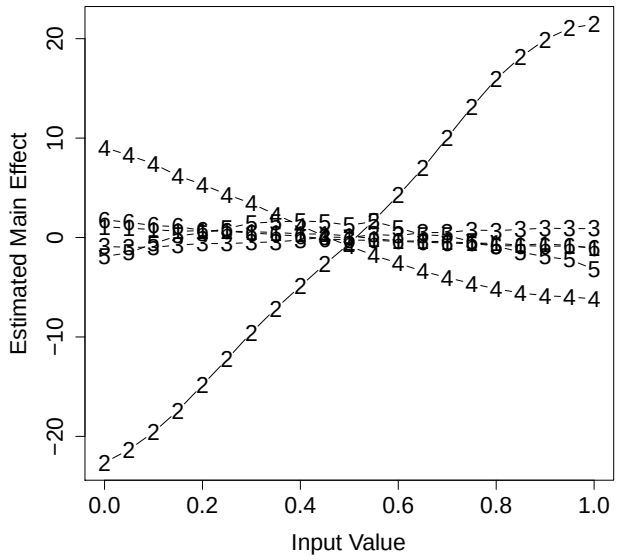


Fig. 4.6.— Functional main effects plots for six variables of the slider-crank system, where the displayed value for each factor is averaged over the two-second time interval.

Fig. 4.6 depicts the functional main effects of the six factors, computed by using the ANOVA decomposition technique described in Section 4.5, where the average of each mean effect function over the two-second time interval is displayed. This plot indicates that the main effect of $x_2$ is the most significant. This suggests that $x_2$, half length of crank, is most critical for determining the initial condition of the underlying ODE of the system of interest. Fig. 4.7 plots the change of the main effects of $x_2$

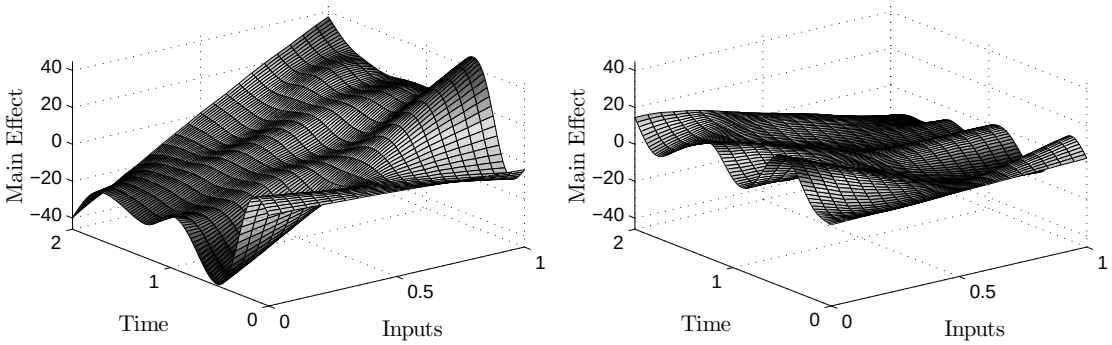


Fig. 4.7.— (Left) the main effect plot of $x_2$ of the fitted meta-model for the slider-crank system; (right) the main effect plot of $x_4$ of the fitted meta-model for the slider-crank system.

and $x_4$ over the time evolution, where the main effect of $x_2$ changes abruptly at the beginning of the evolution and then stabilizes.

## 4.7 Conclusions

Complex multibody codes are typically time-consuming to run. This adversely impacts the potential of virtual prototyping in engineering design. To mitigate this difficulty, a popular strategy in engineering is to replace computer-intensive components of the original model with simplified, less expensive representations. We have proposed a statistical approach for integrating computer experiments with different levels of accuracies for studying the same dynamic system. The effectiveness of the proposed method has been successfully illustrated with a multi-body dynamics system.

A pair of nested space-filling designs is attractive for running two variable-fidelity computer experiments for the same dynamics system. The nested relationship between the designs allows for directly obtaining the discrepancy between the two sources, and their space-filling properties guarantee that the design points are evenly spread in the

design space.

The proposed approach applies generally to variable-fidelity simulations of complex systems. The method is also useful for situations where field data and simulation results for the same dynamics system are available for producing a time-dependent bias function to correct the simulation results. Technical effort will be focused in the future on further developing this method in model calibration, global optimization, and model reduction studies associated with simulations of dynamics systems. The method can be extended to deal with more complicated complex systems such as those with qualitative and quantitative factors (Qian et al., 2008; Han et al., 2009b), with both tuning and calibration parameters (Han et al., 2009a), or with branching and nested factors (Hung et al., 2009).

| Run # | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | Run # | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.039 | 0.023 | 0.023 | 0.102 | 0.102 | 0.039 | 33 | 0.102 | 0.961 | 0.961 | 0.961 | 0.930 | 0.961 |
| 2 | 0.320 | 0.039 | 0.258 | 0.523 | 0.773 | 0.930 | 34 | 0.289 | 0.883 | 0.664 | 0.461 | 0.164 | 0.008 |
| 3 | 0.508 | 0.055 | 0.539 | 0.977 | 0.383 | 0.867 | 35 | 0.570 | 0.977 | 0.430 | 0.070 | 0.539 | 0.211 |
| 4 | 0.773 | 0.117 | 0.820 | 0.398 | 0.633 | 0.242 | 36 | 0.805 | 0.945 | 0.148 | 0.586 | 0.320 | 0.789 |
| 5 | 0.117 | 0.273 | 0.320 | 0.258 | 0.336 | 0.367 | 37 | 0.914 | 0.992 | 0.039 | 0.195 | 0.820 | 0.602 |
| 6 | 0.336 | 0.367 | 0.008 | 0.773 | 0.617 | 0.664 | 38 | 0.711 | 0.898 | 0.336 | 0.742 | 0.055 | 0.492 |
| 7 | 0.617 | 0.305 | 0.852 | 0.664 | 0.133 | 0.539 | 39 | 0.477 | 0.914 | 0.570 | 0.758 | 0.727 | 0.305 |
| 8 | 0.836 | 0.352 | 0.523 | 0.180 | 0.977 | 0.414 | 40 | 0.164 | 0.930 | 0.773 | 0.367 | 0.477 | 0.680 |
| 9 | 0.008 | 0.602 | 0.602 | 0.570 | 0.508 | 0.586 | 41 | 0.086 | 0.633 | 0.695 | 0.711 | 0.664 | 0.648 |
| 10 | 0.273 | 0.523 | 0.836 | 0.117 | 0.273 | 0.445 | 42 | 0.367 | 0.695 | 0.945 | 0.133 | 0.445 | 0.336 |
| 11 | 0.539 | 0.555 | 0.070 | 0.477 | 0.945 | 0.289 | 43 | 0.586 | 0.648 | 0.164 | 0.305 | 0.852 | 0.383 |
| 12 | 0.867 | 0.586 | 0.367 | 0.945 | 0.148 | 0.742 | 44 | 0.789 | 0.742 | 0.492 | 0.836 | 0.070 | 0.523 |
| 13 | 0.023 | 0.820 | 0.758 | 0.867 | 0.758 | 0.773 | 45 | 0.898 | 0.680 | 0.305 | 0.445 | 0.555 | 0.805 |
| 14 | 0.258 | 0.867 | 0.586 | 0.289 | 0.039 | 0.180 | 46 | 0.664 | 0.711 | 0.055 | 0.914 | 0.289 | 0.227 |
| 15 | 0.555 | 0.773 | 0.352 | 0.211 | 0.648 | 0.117 | 47 | 0.414 | 0.727 | 0.867 | 0.602 | 0.914 | 0.086 |
| 16 | 0.758 | 0.789 | 0.102 | 0.695 | 0.414 | 0.945 | 48 | 0.242 | 0.664 | 0.617 | 0.039 | 0.242 | 0.914 |
| 17 | 0.883 | 0.008 | 0.977 | 0.820 | 0.180 | 0.398 | 49 | 0.070 | 0.383 | 0.383 | 0.492 | 0.461 | 0.477 |
| 18 | 0.633 | 0.070 | 0.711 | 0.273 | 0.961 | 0.570 | 50 | 0.352 | 0.445 | 0.227 | 0.930 | 0.711 | 0.555 |
| 19 | 0.461 | 0.102 | 0.461 | 0.227 | 0.352 | 0.711 | 51 | 0.523 | 0.414 | 0.914 | 0.508 | 0.117 | 0.633 |
| 20 | 0.195 | 0.086 | 0.242 | 0.648 | 0.586 | 0.352 | 52 | 0.820 | 0.398 | 0.648 | 0.023 | 0.836 | 0.273 |
| 21 | 0.977 | 0.336 | 0.742 | 0.555 | 0.492 | 0.148 | 53 | 0.945 | 0.477 | 0.555 | 0.727 | 0.258 | 0.023 |
| 22 | 0.727 | 0.320 | 0.883 | 0.008 | 0.695 | 0.852 | 54 | 0.695 | 0.430 | 0.805 | 0.148 | 0.602 | 0.898 |
| 23 | 0.430 | 0.258 | 0.211 | 0.383 | 0.023 | 0.992 | 55 | 0.398 | 0.492 | 0.117 | 0.336 | 0.195 | 0.758 |
| 24 | 0.180 | 0.289 | 0.477 | 0.992 | 0.805 | 0.102 | 56 | 0.148 | 0.461 | 0.289 | 0.852 | 0.898 | 0.195 |
| 25 | 0.961 | 0.570 | 0.414 | 0.352 | 0.680 | 0.977 | 57 | 0.055 | 0.164 | 0.195 | 0.164 | 0.227 | 0.133 |
| 26 | 0.648 | 0.508 | 0.133 | 0.789 | 0.398 | 0.055 | 58 | 0.305 | 0.211 | 0.398 | 0.680 | 0.883 | 0.820 |
| 27 | 0.445 | 0.617 | 0.992 | 0.633 | 0.867 | 0.164 | 59 | 0.602 | 0.148 | 0.680 | 0.805 | 0.367 | 0.883 |
| 28 | 0.133 | 0.539 | 0.633 | 0.242 | 0.086 | 0.836 | 60 | 0.852 | 0.227 | 0.898 | 0.320 | 0.523 | 0.070 |
| 29 | 0.930 | 0.805 | 0.180 | 0.055 | 0.992 | 0.695 | 61 | 0.992 | 0.242 | 0.789 | 0.898 | 0.008 | 0.320 |
| 30 | 0.680 | 0.836 | 0.445 | 0.539 | 0.211 | 0.258 | 62 | 0.742 | 0.180 | 0.508 | 0.430 | 0.789 | 0.727 |
| 31 | 0.383 | 0.758 | 0.727 | 0.883 | 0.570 | 0.461 | 63 | 0.492 | 0.133 | 0.273 | 0.086 | 0.430 | 0.508 |
| 32 | 0.211 | 0.852 | 0.930 | 0.414 | 0.305 | 0.617 | 64 | 0.227 | 0.195 | 0.086 | 0.617 | 0.742 | 0.430 |

Table 4.2: A pair of nested space-filling designs $\mathcal{D}_2 \subset \mathcal{D}_1$ for the HE and LE used for studying a slider-crank system with six factors. The design $\mathcal{D}_1$ is an OA-based Latin hypercube design with 64 runs based on the $OA(64, 8^6, 2)$ from Table 4.1, where $\mathcal{D}_2$ is based on the first 16 runs the orthogonal array, which form an $OA(16, 4^6, 2)$ after the eight levels are collapsed into four levels as follows: $(1, 2) \to 1$, $(3, 4) \to 2$, $(5, 6) \to 3$, $(7, 8) \to 4$.

# Appendix A

# Proofs

## A.1 Proof of Proposition 2.2

*Proof.* Assume $x, y \in Z_{n_2}$. To show (i), by Lemma 2.2 and the symmetry of $\mathbf{C}_1, \ldots, \mathbf{C}_s$, $P(c_{m,ik} = x)$ is the same for all $x$ and hence equals $n_2^{-1}$. To show (ii), note that for any $x \neq y \in Z_{n_2}$,

$$P(c_{m,ik} = x, c_{m,jk} = y) = P(c_{m,jk} = y | c_{m,ik} = x) P(c_{m,ik} = x),$$

where $P(c_{m,ik} = x) = n_2^{-1}$ from (i) and $P(c_{m,jk} = y | c_{m,ik} = x)$ is $(n_2 - s)^{-1}$, because if $c_{m,ik} = x$, the remaining $s - 1$ elements of the same $\mathbf{g}_{n_2}$ of (2.2) cannot be in $\mathbf{C}_m$. To show (iii), divide all $x, y \in Z_{n_2}$ into three groups:

$$g_1 = \{(x,y) | \lceil x/s \rceil \neq \lceil y/s \rceil\},\ g_2 = \{(x,y) | \lceil x/s \rceil = \lceil y/s \rceil,\ x \neq y\},\ g_3 = \{(x,y) | x = y\}.$$

These groups have $n_2(n_2 - s), n_2(s-1)$ and $n_2$ pairs, respectively. Thus, $P[(c_{m_1,ik}, c_{m_2,jk}) \in g_1] = (n_1 - 1)n_1^{-1}$, $P[(c_{m_1,ik}, c_{m_2,jk}) \in g_2] = n_1^{-1}$ and $P[(c_{m_1,ik}, c_{m_2,jk}) \in g_3] = 0$. The result now follows by the symmetry of $g_1, g_2$ and $g_3$. □

## A.2 Proof of Proposition 2.3

*Proof.* Part (i) is from Remark 2.1 and Proposition 5 of He and Qian (2011). To show (ii), let $P[(i,j) \in H_h]$ denote the probability that the rows $(i,j)$ are in $H_h$ in Table 2.3, for $h = 1, \ldots, 8$. Now use Lemma 2.2 to derive $P[(i,j) \in H_h]$, for $h = 2, \ldots, 8$. Note that

$$\begin{array}{rcl} P[(i,j) \in H_2] & = & \frac{\lambda-1-\tau}{\lambda^3 s(s-1)} \\ P[(i,j) \in H_3] & = & \frac{2(\lambda-1)(\lambda-1-\tau)}{\lambda^3 s(s-1)} \\ P[(i,j) \in H_4] & = & \frac{(\lambda-1)^2(\lambda-1-\tau)}{\lambda^3 s(s-1)} \\ P[(i,j) \in H_6] & = & \frac{2(\lambda s-2\lambda+1+\tau)}{\lambda^2 s(s-1)} \\ P[(i,j) \in H_7] & = & \frac{2(\lambda-1)(\lambda s-2\lambda+1+\tau)}{\lambda^2 s(s-1)} \\ P[(i,j) \in H_8] & = & \frac{\lambda s^2-3\lambda s+3\lambda-1-\tau}{\lambda s(s-1)}. \end{array} \tag{A.1}$$

Then the result in (ii) follows by the symmetry of each of $H_2, \ldots, H_8$ and column 3 of Table 2.3. □

## A.3 Proof of Theorem 2.1

*Proof.* Result (i) is from Remark 2.1 and Theorem 1 of He and Qian (2011). We now show (ii) by using tools developed in Owen (1994) and He and Qian (2011) coupled with the sliced structure of a sliced U design. For $i = 1, \ldots, n_2$, let $\rho(i)$ indicate which function is evaluated by the $i$th run $\mathbf{x}_i$ of $\mathbf{D}$. Observe that

$$\begin{aligned} \mathrm{var}(\hat{\mu}) &= E\left[ n_2^{-2} \sum_{|u|>0} \sum_{|v|>0} \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} f_{\rho(i),u}(\mathbf{x}_i) f_{\rho(j),v}(\mathbf{x}_j) \right] \\ &= n_2^{-2} \sum_{|u|>0} \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} E\left[ f_{\rho(i),u}(\mathbf{x}_i) f_{\rho(j),u}(\mathbf{x}_j) \right]. \end{aligned} \tag{A.2}$$

For $|u| = 1$,

$$
\begin{aligned}
& n_2^{-2} \sum_{|u|=1} \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} E\left[f_{\rho(i),u}(\mathbf{x}_i) f_{\rho(j),u}(\mathbf{x}_j)\right] \\
& = n_2^{-2} n_1 \sum_{m=1}^{s} \mathrm{var}[f_{m,u}(\mathbf{x})] - n_2^{-2} n_1 (n_1 - 1)(n_1 - 1)^{-1} \left( \sum_{m=1}^{s} \mathrm{var}[f_{m,u}(\mathbf{x})] + o(s) \right) \\
& + n_2^{-1}(n_2 - n_1) o(n_2^{-1}) \\
& = o(n_2^{-1}). \qquad \text{(A.3)}
\end{aligned}
$$

For $|u| = 2$,

$$
\begin{aligned}
& n_2^{-2} \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} E\left[f_{\rho(i),u}(\mathbf{x}_i) f_{\rho(j),u}(\mathbf{x}_j)\right] \\
& = n_2^{-2} \sum_{m=1}^{s} n_1 \mathrm{var}[f_{m,u}(\mathbf{x})] + n_2^{-2} \sum_{m=1}^{s} n_1 (n_1 - 1) \left( \tau n_1^{-1} \mathrm{var}[f_{m,u}(\mathbf{x})] + o(n_1^{-1}) \right) \\
& - n_2^{-2} \sum_{m_1 \neq m_2} n_1^2 [(1 + \tau) n_2^{-1} \mathrm{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})] + o(n_2^{-1})] \\
& = (1 + \tau) n_2^{-1} s^{-1} \left( \sum_{m=1}^{s} \mathrm{var}[f_{m,u}(\mathbf{x})] \right) - (s - 1)^{-1} \sum_{m_1 \neq m_2} \mathrm{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})] \\
& + o(n_2^{-1}). \qquad \text{(A.4)}
\end{aligned}
$$

For $|u| > 2$, let

$$
w = w_{ij}(u) = \{k \in u | \alpha_{ik} = \alpha_{jk}\} \qquad \text{(A.5)}
$$

and consider three different cases for $(i, j)$.

Case I: $i = j$, then $\rho(i) = \rho(j)$ and hence $E\left[f_{\rho(i),u}(\mathbf{x}_i) f_{\rho(j),u}(\mathbf{x}_j)\right] = \mathrm{var}[f_{\rho(i),u}(\mathbf{x})]$,

Case II: $i \neq j$ and $\rho(i) = \rho(j)$, we have

$$
E\left[f_{\rho(i),u}(\mathbf{x}_i) f_{\rho(j),u}(\mathbf{x}_j)\right] = (s - 1)^{-(|u| - |w|)} (-1)^{|u| - |w|} \mathrm{var}[f_{\rho(i),u}(\mathbf{x})] + o(s^{-(|u| - |w|)}),
$$

Case III: $\rho(i) \neq \rho(j)$, we have

$$E\left[f_{\rho(i),u}(\mathbf{x}_i)f_{\rho(j),u}(\mathbf{x}_j)\right] = (s-1)^{-(|u|-|w|)}(-1)^{|u|-|w|}\text{cov}[f_{\rho(i),u}(\mathbf{x}), f_{\rho(j),u}(\mathbf{x})]+o(s^{-(|u|-|w|)}).$$

Combining these three cases yields

$$\begin{aligned}
&n_2^{-2}\sum_{|u|>2}\sum_{i=1}^{n_2}\sum_{j=1}^{n_2}E\left[f_{\rho(i),u}(\mathbf{x}_i)f_{\rho(j),u}(\mathbf{x}_j)\right] \\
&= \sum_{|u|>2}\left(\sum_{m=1}^{s}\sum_{r=0}^{|u|}n_2^{-2}M_m(u,r)(s-1)^{-(|u|-r)}(-1)^{|u|-r}\text{var}[f_{m,u}(\mathbf{x})]\right. \\
&\left.+\sum_{m_1\neq m_2}\sum_{r=0}^{|u|}n_2^{-2}M_{m_1,m_2}(u,r)(s-1)^{-(|u|-r)}(-1)^{|u|-r}\text{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})]\right)+o(n_2^{-1}) \\
&= \sum_{|u|>2}\left(\sum_{m=1}^{s}n_2^{-2}M_m(u,|u|)\text{var}[f_{m,u}(\mathbf{x})]\right. \\
&\left.+\sum_{m_1\neq m_2}n_2^{-2}M_{m_1,m_2}(u,|u|)\text{cov}[f_{m_1,u}(\mathbf{x}), f_{m_2,u}(\mathbf{x})]\right)+o(n_2^{-1}). \qquad \text{(A.6)}
\end{aligned}$$

Plugging (A.3), (A.4) and (A.6) into (A.2) gives (ii). □

## A.4 Proof of Proposition 2.4

*Proof.* Only (iv) needs a proof. Let $\mathbf{D}$ denote an ordinary U design based on an OA$(n_2, s^{q+1}, 2)$. For an arbitrary element $m$ of $Z_s$, randomly divide $\mathbf{D}$ into $s$ slices of $n_1$ runs each, and let $\mathbf{D}_m$ be the $m$th slice. For $\hat{\mu}_m$ in (2.19) based on $\mathbf{D}_m$,

$$\text{var}(\hat{\mu}_m) \;=\; n_1^{-2}\sum_{|u|>0}\sum_{i=1}^{n_1}\sum_{j=1}^{n_1}E\left[f_{m,u}(\mathbf{x}_i)f_{m,u}(\mathbf{x}_j)\right]. \qquad \text{(A.7)}$$

For $|u| = 1$, consider two different cases for $(i, j)$. For $i = j$, $E\left[f_{m,u}(\mathbf{x}_i)f_{m,u}(\mathbf{x}_j)\right] = \text{var}[f_{m,u}(\mathbf{x})]$, and for $i \neq j$, by the continuity of $f_m$, $E\left[f_{m,u}(\mathbf{x}_i)f_{m,u}(\mathbf{x}_j)\right] = -n_2^{-1}\text{var}[f_{m,u}(\mathbf{x})]+ o(n_2^{-1})$. Combining these two cases yields

$$n_1^{-2}\sum_{|u|=1}\sum_{i=1}^{n_1}\sum_{j=1}^{n_1}E\left[f_{m,u}(\mathbf{x}_i)f_{m,u}(\mathbf{x}_j)\right] = \sum_{|u|=1}n_1^{-1}\text{var}[f_{m,u}(\mathbf{x})]+o(n_1^{-1}). \qquad \text{(A.8)}$$

For $|u| = 2$, observe that for two different runs $\mathbf{x}_i = (x_{i1}, \ldots, x_{iq})$ and $\mathbf{x}_j = (x_{j1}, \ldots, x_{jq})$ of $\mathbf{D}_m$, the joint probability density function of $(x_{ik}, x_{il})$ and $(x_{jk}, x_{jl})$ is

$$\begin{cases} 0 & (x_{ik}, x_{il}, x_{jk}, x_{jl}) \in H_1, H_5, \\ \frac{\lambda-1}{n_2-1} / \left(\frac{\lambda s-1}{n_2}\right)^2 & (x_{ik}, x_{il}, x_{jk}, x_{jl}) \in H_2, H_3, H_4, \\ \frac{\lambda s-\lambda}{n_2-1} / \left[\frac{\lambda s(s-1)(\lambda s-1)}{n_2^2}\right] & (x_{ik}, x_{il}, x_{jk}, x_{jl}) \in H_6, H_7, \\ \frac{n_2-2\lambda s+\lambda}{n_2-1} / \left[\frac{\lambda s(s-1)}{n_2}\right]^2 & (x_{ik}, x_{il}, x_{jk}, x_{jl}) \in H_8, \end{cases} \tag{A.9}$$

where $H_1, \ldots, H_8$ are defined in Table 2.3. Now consider two different cases for $(i, j)$. First, for $i = j$, $E\left[f_{m,u}(\mathbf{x}_i) f_{m,u}(\mathbf{x}_j)\right] = \text{var}[f_{m,u}(\mathbf{x})]$. Second, for $i \neq j$ and $H_h$, $h = 1, \ldots, 8$, in Table 2.3, let $\int_{H_h}$ denote integration over the region defined by (2.11) for $(x_{ik}, x_{il}, x_{jk}, x_{jl})$ associated with all $i, j$ in $H_h$. By (A.9) and the continuity of $f_m$,

$$\begin{aligned} E\left[f_{m,u}(\mathbf{x}_i) f_{m,u}(\mathbf{x}_j)\right] &= [-1 + o(1)] \int_{H_1, H_5} + [-1/\lambda + o(1)] \int_{H_2, H_3, H_4} + [o(1)] \int_{H_6, H_7} \\ &= -n_2^{-1} \text{var}[f_{m,u}(\mathbf{x})] + o(n_2^{-1}). \end{aligned}$$

Combining these two cases of $(i, j)$ yields

$$n_1^{-2} \sum_{|u|=2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} E\left[f_{m,u}(\mathbf{x}_i) f_{m,u}(\mathbf{x}_j)\right] = \sum_{|u|=2} n_1^{-1} \text{var}[f_{m,u}(\mathbf{x})] + o(n_1^{-1}). \tag{A.10}$$

For $|u| > 2$, consider two different cases of $(i, j)$. First, for $i = j$, $E\left[f_{m,u}(\mathbf{x}_i) f_{m,u}(\mathbf{x}_j)\right] = \text{var}[f_{m,u}(\mathbf{x})]$. Second, for $i \neq j$,

$$E\left[f_{m,u}(\mathbf{x}_i) f_{m,u}(\mathbf{x}_j)\right] = (s-1)^{-(|u|-|w|)} (-1)^{|u|-|w|} \text{var}[f_{m,u}(\mathbf{x})] + o(s^{-(|u|-|w|)}).$$

Now, for $r = 0, \ldots, q$, let $\xi_r$ denote the number of runs in $\mathbf{D}_m$ satisfying $|w|$, defined in (A.5), equals $r$. As $\mathbf{D}_m$ is obtained by randomly dividing $\mathbf{D}$, $\xi_r$ is a random variable. Define $e_r = E(\xi_r)$. For $|u| > 2$, combining two cases, with $i = j$ and $i \neq j$, respectively,

yields

$$
\begin{aligned}
& n_1^{-2} \sum_{|u|>2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} E\left[f_{m,u}(\mathbf{x}_i) f_{m,u}(\mathbf{x}_j)\right) \\
= \; & n_1^{-1} \sum_{|u|>2} \left( \mathrm{var}[f_{m,u}(\mathbf{x})] + \sum_{r=0}^{|u|} \left[ e_r s^{-(|u|-r)} (-1)^{|u|-r} \mathrm{var}[f_{m,u}(\mathbf{x})] + o(s^{-(|u|-r)}) \right] \right).
\end{aligned}
$$

Next, we calculate

$$
e_r s^{-(|u|-r)} (-1)^{|u|-r} \mathrm{var}[f_{m,u}(\mathbf{x})] + o(s^{-(|u|-r)}) \tag{A.11}
$$

for different cases of $r$ and $|u|$. Note that $e_r = n_2^{-1} s^{-1}[M(u,r) - n_2]$ for $r = |u|$ and $e_r = n_2^{-1} s^{-1} M(u,r)$ for $r < |u|$. For $r = |u|$, $e_r \leq (\lambda - 1)/s$, and hence (A.11) is $o(1)$. For $r = |u| - 1$, $e_r \leq (\lambda - 1)/s$ because $|u| > 2$ and $r \geq 2$, and hence (A.11) is $o(n_1^{-1})$. For $r = |u| - 2$, $e_r \leq (\lambda s - 1)/s$ because $|u| > 2$ and $r \geq 1$, and hence (A.11) is $o(n_1^{-1})$. For $r < |u| - 2$, $e_r \leq (n_2 - 1)/s$, and hence (A.11) is $o(n_1^{-1})$. Combining these four cases of $r$ and $|u|$ gives

$$
n_1^{-2} \sum_{|u|>2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} E\left[f_{m,u}(\mathbf{x}_i) f_{m,u}(\mathbf{x}_j)\right] = \sum_{|u|>2} n_1^{-1} \mathrm{var}[f_{m,u}(\mathbf{x})] + o(n_1^{-1}). \tag{A.12}
$$

Plugging (A.8), (A.10) and (A.12) into (A.7) gives the result in (i). Part (ii) can be shown by using a similar argument. □

## A.5 Proof of Proposition 3.1

*Proof.* To show (a), note that $w_{ik}^*$ denotes the $(i,k)$ th entry of $\mathbf{W}^*$. Observe that the level mapping $\rho : w_{ik}^* \to \lceil w_{ik}^*/q_k \rceil$ collapses the $a_k$ levels, $1, \ldots, a_k$, of column $k$ of $\mathbf{W}^*$ into $b_k$ levels, $1, \ldots, b_k$, for $k = 1, \ldots, d$. In this mapping, each new symbol

corresponds to $q_k = a_k/b_k$ old symbols. Thus, by Lemma 3.1 (a), $\mathbf{W}^*$ becomes an $OA(n_1, b_1^{d_1} \cdots b_v^{d_v})$ of strength two after level-collapsing according to $\rho$. Notice that in some steps of the foregoing algorithm, the symbols in each column of $\mathbf{W}^*$ are permuted. By Lemma 3.1 (b), such permutations does not alter this orthogonal array structure of $\mathbf{W}^*$. Now, when projected onto factors $k_1$ and $k_2$, two-dimensional uniformity of $\mathbf{D}_1$ on the corresponding $b_{k_1} \times b_{k_2}$ grids follows by noting the connection between $\mathbf{D}_1$ and $\mathbf{W}^*$ in (3.2). When projected onto factor $k$, stratification of $\mathbf{D}_1$ with respect to the $a_k$ equally spaced intervals on $(0, 1]$ follows by noting that the steps in the foregoing algorithm are designed in such a way that the symbols of each block in (3.1) appears equally often in column $k$ of $\mathbf{W}^*$, for $k = 1, \ldots, d$.

To show (b), let $v_{ik}^*$ denote the $(i, k)$ entry of $\mathbf{V}^*$. Because of the link between $\mathbf{V}^*$ and $\mathbf{V}$, $\mathbf{V}^*$ becomes an $OA(n_2, b_1^{d_1} \cdots b_v^{d_v})$ after mapping $v_{ik}^*$ to $\lceil v_{ik}^*/q_k \rceil$. By Lemma 3.1 (b), this underlying orthogonal array structure of $\mathbf{V}^*$ stays intact after randomly permuting the symbols in each column of $\mathbf{V}^*$ in Steps 1 and 2 of the foregoing algorithm. The result in (b) now follows by noting the connection between $\mathbf{D}_2$ and $\mathbf{V}^*$ in (3.2). □

# Bibliography

Ascher, U. M. and Petzold, L. R. (1998), *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Philadelphia, PA: SIAM.

Bingham, D., Sitter, R. R., and Tang, B. (2009), "Orthogonal and Nearly Orthogonal Designs for Computer Experiments," *Biometrika*, 96, 51–65.

Bose, R. C. and Bush, K. A. (1952), "Orthogonal Arrays of Strength Two and Three," *The Annals of Mathematical Statistics*, 23, 508–524.

Brenan, K. E., Campbell, S. L., and Petzold, L. R. (1989), *Numerical Solution of Initial-value Problems in Differential-Algebraic Equations*, New York: North-Holland.

Cardona, A. and Geradin, M. (1989), "Time Integration of the Equation of Motion in Mechanical Analysis," *Computer and Structures*, 33, 801–820.

Cox, D. D., Park, J. S., and Singer, C. E. (2001), "A Statistical Method for Tuning a Computer Code to a Data Base," *Computational Statistics and Data Analysis*, 37, 77 – 92.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian Prediction of

Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, 86, 953–963.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005), "Scalable Test Problems for Evolutionary Multiobjective Optimization," in *Evolutionary Multiobjective Optimization*, eds. Jain, L., Wu, X., Abraham, A., Jain, L., and Goldberg, R., Berlin Heidelberg: Springer, Advanced Information and Knowledge Processing, pp. 105–145.

Eich-Sollner, E. and Fuhrer, C. (1998), *Numerical Methods in Multibody Dynamics*, Stuttgart: Teubner-Verlag.

Fang, K.-T., Li, R., and Sudjianto, A. (2005), *Design and Modeling for Computer Experiments*, New York: Chapman & Hall/CRC.

Fasshauer, G. E. (2007), *Meshfree Approximation Methods With MATLAB*, Hackensack: World Scientific Publishing.

Floater, M. S. and Iske, A. (1996), "Multistep Scattered Data Interpolation Using Compactly Supported Radial Basis Functions," *Journal of Computational and Applied Mathematics*, 73, 65–78.

Hairer, E. and Wanner, G. (1991), *Solving Ordinary Differential Equations*, vol. II of *Computational Mathematics*, Berlin: Springer-Verlag.

Han, G., Santner, T. J., Notz, W. I., and Bartel, D. L. (2009a), "Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables," *Technometrics*, 51, 278–288.

Han, G., Santner, T. J., and Rawlinson, J. J. (2009b), "Simultaneous Determination of Tuning and Calibration Parameters for Computer Experiments," *Technometrics*, 51, 464–474.

Haug, E. J. (1989), *Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume-I*, New Jersey: Prentice-Hall.

He, X. and Qian, P. Z. G. (2011), "Nested Orthogonal Array Based Latin Hypercube Designs," *Biometrika*, 98, 721–731.

Hedayat, A. S., Pu, K., and Stufken, J. (1992), "On the Construction of Asymmetrical Orthogonal Arrays," *The Annals of Statistics*, 20, 2142–2152.

Hedayat, A. S., Sloane, N. J. A., and Stufken, J. (1999), *Orthogonal Arrays: Theory and Applications*, New York: Springer.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, 103, 570–583.

Hilber, H. M., Hughes, T. J. R., and Taylor, R. L. (1977), "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," *Earthquake Engineering & Structural Dynamics*, 5, 283–292.

Huang, D. and Allen, T. T. (2005), "Design and Analysis of Variable Fidelity Experimentation Applied to Engine Valve Heat Treatment Process Design," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 54, 443–463.

Hughes, T. J. R. (1987), *Finite Element Method - Linear Static and Dynamic Finite Element Analysis*, Englewood Cliffs, New Jersey: Prentice-Hall.

Hung, Y., Joseph, V. R., and Melkote, S. N. (2009), "Design and Analysis of Computer Experiments With Branching and Nested Factors," *Technometrics*, 51, 354–365.

Husslage, B., Dam, E. V., Hertog, D. D., Stehouwer, P., and Stinstra, E. (2003), "Collaborative Metamodeling: Coordinating Simulation-Based Product Design," *Concurrent Engineering*, 11, 267–278.

Husslage, B. G. M., Dam, E. R. V., and Hertog, D. D. (2005), "Nested Maximin Latin Hypercube Designs in Two Dimensions," CentER Discussion Paper 2005-79. Tilburg University, Tilburg, The Netherlands.

Jin, R., Chen, W., and Simpson, T. (2001), "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria," *Journal of Structural and Multidisciplinary Optimization*, 23, 1–13.

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990), "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26, 131–148.

Kennedy, M. and O'Hagan, A. (2000), "Predicting the Output from a Complex Computer Code When Fast Approximations are Available," *Biometrika*, 87, 1–13.

Leary, S., Bhaskar, A., and Keane, A. (2003), "Optimal Orthogonal Array-Based Latin Hypercubes," *Journal of Applied Statistics*, 30, 585–598.

Lin, C. D., Bingham, D., Sitter, R. R., and Tang, B. (2010), "A New and Flexi-

ble Method for Constructing Designs for Computer Experiments," *The Annals of Statistics*, 38, 1460–1477.

Lin, C. D., Mukerjee, R., and Tang, B. (2009), "Construction of Orthogonal and Nearly Orthogonal Latin Hypercubes," *Biometrika*, 96, 243–247.

Lin, D. K. J. (1993), "A New Class of Supersaturated Design," *Technometrics*, 35, 28–31.

Loh, W.-L. (1996), "On Latin Hypercube Sampling," *The Annals of Statistics*, 24, 2058–2080.

Lubich, C., Nowak, U., Pohle, U., and Engstler, C. (1995), "MEXX - Numerical Software for the Integration of Constrained Mechanical Multibody Systems," *Mechanics of Structures and Machines*, 23, 473–495.

MATLAB (2010), *Version 7.10.0 (R2010a)*, Massachusetts: The MathWorks Inc.

McKay, M., Conover, W., and Beckman, R. J. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 21, 239–245.

Morris, M. D. and Mitchell, T. J. (1995), "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, 43, 381–402.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993), "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, 35, 243–255.

MSCsoftware (2005), "ADAMS User Manual," Available at http://www.mscsoftware.com.

Mukerjee, R., Qian, P. Z. G., and Wu, C. F. J. (2008), "On the Existence of Nested Orthogonal Arrays," *Discrete Mathematics*, 308, 4635–4642.

Negrut, D., Rampalli, R., Ottarsson, G., and Sajdak, A. (2007), "On an Implementation of the Hilber-Hughes-Taylor Method in the Context of Index 3 Differential Algebraic Equations of Multibody Dynamics," *Journal of Computational and Nonlinear Dynamics*, 2, 73–85.

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia: Society for Industrial Mathematics.

Orlandea, N., Chace, M. A., and Calahan, D. A. (1977), "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems – Part I and Part II," *Transactions of the ASME Journal of Engineering for Industry*.

Owen, A. B. (1992a), "A Central Limit Theorem for Latin Hypercube Sampling," *Journal of the Royal Statistical Society. Series B (Methodological)*, 54, 541–551.

— (1992b), "Orthogonal Arrays for Computer Experiments, Integration and Visualization," *Statistica Sinica*, 2, 439–452.

— (1994), "Lattice Sampling Revisited: Monte Carlo Variance of Means over Randomized Orthogonal Arrays," *The Annals of Statistics*, 22, 930–945.

— (1995), "Randomly Permuted $(t, m, s)$-nets and $(t, s)$-sequences," *Monte Carlo and*

*Quasi-Monte Carlo Methods in Scientific Computing, Lecture Notes in Statistics*, 106, 299–317.

Patterson, H. D. (1954), "The Errors of Lattice Sampling," *Journal of the Royal Statistical Society. Series B (Methodological)*, 16, 140–149.

Petzold, L. R. (1982), "Differential algebraic equations are not ODE's," *SIAM J. Numer. Anal.*, 3, 367–384.

Potra, F. A. (1993), "Implementation of linear multistep methods for solving constrained equations of motion," *SIAM Journal on Numerical Analysis*, 30, 474–489.

Qian, P. Z. G. (2009), "Nested Latin Hypercube Designs," *Biometrika*, 96, 957–970.

— (2012), "Sliced Latin Hypercube Designs," *Journal of the American Statistical Association*, 107, 393–399.

Qian, P. Z. G., Ai, M., Hwang, Y., and Su, H. (2011), "Asymmetric Nested Lattice Samples," *Technical Report*.

Qian, P. Z. G. and Ai, M. Y. (2010), "Nested Lattice Sampling: A New Sampling Scheme Derived by Randomizing Nested Orthogonal Arrays," *Journal of the American Statistical Association*, 105, 1147–1155.

Qian, P. Z. G., Ai, M. Y., and Wu, C. F. J. (2009), "Construction of Nested Space-Filling Designs," *The Annals of Statistics*, 37, 3616–3643.

Qian, P. Z. G. and Wu, C. F. J. (2009), "Sliced Space-filling Designs," *Biometrika*, 96, 945–956.

Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008), "Gaussian Process Models for Computer Experiments with Qualitative and Quantitative Factors," *Technometrics*, 50, 383–396.

Qian, Z., Seepersad, C. C., Roshan, V. R., Allen, J. K., and Wu, C. F. J. (2006), "Building Surrogate Models Based on Detailed and Approximate Simulations," *ASME Transactions, Journal of Mechanical Design*, 128, 668–677.

Ramsay, J. and Silverman, B. W. (2005), *Functional Data Analysis*, New York, NY: Springer.

Rao, C. R. (1947), "Factorial Experiments Derivable from Combinatorial Arrangements of Arrays," *Supplement to the Journal of the Royal Statistical Society*, 9, 128–139.

— (1952), "Orthogonal Arrays of Index Unity," *The Annals of Mathematical Statistics*, 23, 426–434.

Reese, C. S., Wilson, A. G., Hamada, M., Martz, H. F., and Ryan, K. J. (2004), "Integrated Analysis of Computer and Physical Experiments," *Technometrics*, 46, 153–164.

Sacks, J., Schiller, S. B., and Welch, W. J. (1989a), "Designs for Computer Experiments," *Technometrics*, 31, pp. 41–47.

Sacks, J., Welch, W., Mitchell, T. J., and Wynn, H. P. (1989b), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–423.

Sallaberry, C. J., Helton, J. C., and Hora, S. C. (2008), "Extension of Latin Hypercube Samples With Correlated Variables," *Reliability Engineering and System Safety*, 93, 1047–1059.

Santner, T. J., Williams, B. J., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer, 1st ed.

Schürer, R. and Schmid, W. C. (2010), "MinT–Architecture and Applications of the $(t, m, s)$-net and OOA Database," *Mathematics and Computers in Simulation*, 80, 1124–1132.

Shabana, A. A. (2005), *Dynamics of Multibody Systems*, New York: Cambridge University Press.

Steinberg, D. M. and Lin, D. K. J. (2006), "A Construction Method for Orthogonal Latin Hypercube Designs," *Biometrika*, 93, 279–288.

Storlie, C. and Reich, B. (2011), "Calibration and Prediction Using Multiple Computer Models," Presenation, The 2011 INFORMS Annual Conference, Charlotte, NC.

Tang, B. (1993), "Orthogonal Array-Based Latin Hypercubes," *Journal of the American Statistical Assocation*, 88, 1392–1397.

— (1994), "A Theorem for Selecting OA-based Latin Hypercubes Using a Distance Criterion," *Communications in Statistics–Theory and Methods*, 23, 2047–2058.

Tong, C. (2006), "Refinement Strategies for Stratified Sampling Methods," *Reliability Engineering and System Safety*, 91, 1257–1265.

Wang, G. G. (2003), "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points," *Journal of Mechanical Design*, 125, 210–220.

Wang, J. C. (1996), "Mixed Difference Matrices and the Construction of Orthogonal Arrays," *Statistics and Probability Letters*, 28, 121–126.

Wang, J. C. and Wu, C. F. J. (1991), "An Approach to the Construction of Asymmetric Orthogonal Arrays," *Journal of the American Statistical Association*, 86, 450–456.

Williams, B., Morris, M., and Santner, T. (2009), "Using Multiple Computer Models/Multiple Data Sources Simultaneously to Infer Calibration Parameters," Presenation, The 2009 INFORMS Annual Conference, San Diego, CA.

Wu, C. F. J. (1989), "Construction of $2^m4^n$ Designs via a Grouping Scheme," *The Annals of Statistics*, 1880–1885.

Xu, H. (2005), "Some Nonregular Designs from the Nordstrom-Robinson Code and Their Statistical Properties," *Biometrika*, 92, 385–397.

Xu, H. and Wu, C. F. J. (2005), "Construction of Optimal Multi-level Supersaturated Designs," *The Annals of Statistics*, 33, 2811–2836.

Xu, X., Haaland, B., and Qian, P. Z. G. (2011), "Sudoku-based Space-filling Designs," *Biometrika*, 98, 711–720.

Ye, K. Q. (1998), "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments," *Journal of the American Statistical Association*, 93, 1430–1439.