Statistica Sinica **19** (2009), 287-300

NESTED SPACE-FILLING DESIGNS FOR COMPUTER EXPERIMENTS WITH TWO LEVELS OF ACCURACY

Peter Z. G. Qian[1], Boxin Tang[2] and C. F. Jeff Wu[3]

[1]*University of Wisconsin,* [2]*Simon Fraser University and* [3]*Georgia Tech.*

Abstract: Computer experiments with different levels of accuracy have become prevalent in many engineering and scientific applications. Design construction for such computer experiments is a new issue because the existing methods deal almost exclusively with computer experiments with one level of accuracy. In this paper, we construct some *nested space-filling designs* for computer experiments with two levels of accuracy. Our construction makes use of Galois fields and orthogonal arrays.

Key words and phrases: Computer experiment, Galois field, Latin hypercube design, orthogonal array, Rao-Hamming construction.

1. Introduction

Experimentation to study complex real world systems in engineering and sciences can now be conducted at different levels of accuracy. Complex mathematical models, implemented in large computer codes, are widely used as a proxy to study the real systems. Conducting the corresponding physical experiments would be costly. For example, each physical run of the fluidized bed process in the food industry to coat certain food products with additives can take days or even weeks to finish (Reese, Wilson, Hamada, Martz and Ryan (2004)) while running the associated computer code only takes minutes per run. Because a large computer program can be run at different levels of sophistication with vastly varying computational times, multiple experiments with various levels of accuracy or fidelity have become prevalent in practice.

Study of such multiple experiments involves two aspects: experimental planning, and modeling and analysis of data. While some headway has been made to tackle the modeling issue (Kennedy and O'Hagan (2001), Reese et al. (2004), Qian, Seepersad, Joseph, Allen and Wu (2006) and Qian and Wu (2008)), no systematic study has hitherto been done to address the planning issue. This issue poses new challenges as the existing methods deal almost exclusively with computer experiments with one level of accuracy (Santner, Williams and Notz (2003) and Fang, Li and Sudjianto (2005)). The purpose of this article is to propose and construct some suitable designs in this new situation. For ease in presentation, we only consider the situation involving two experiments that are called the *low-accuracy experiment* (LE) and the *high-accuracy experiment* (HE), where HE is more accurate but more expensive than LE. The sets of design points for LE and HE are denoted by D_l and D_h, respectively. Throughout we assume that the design region for both D_l and D_h is the unit hypercube. Construction of D_l and D_h is guided by three principles:

Economy – the number n_2 of points in D_h is smaller than the number n_1 of points in D_l;

Nested relationship – D_h is nested within D_l, i.e., $D_h \subset D_l$;

Space-filling – both D_l and D_h achieve uniformity in low dimensions.

These principles were implicitly used in Qian et al. (2006) but not formally given therein. The principle of economy is concerned with different computing times of HE and LE; as LE is cheaper than HE, more LE runs can be afforded. The principle of nested relationship makes it easier to model data from HE and LE. It implies that, for every point in D_h, results from both LE and HE are available. This part of data can thus be used for modeling and calibrating the differences between these two experiments (Kennedy and O'Hagan (2001), Qian et al. (2006) and Qian and Wu (2008)). The principle of space-filling is based on the belief that interesting features of the true model are as likely to be in one part of the design space as in another. Designs that spread the points in a design space uniformly are often referred to as space-filling designs. Uniformity of design points can be achieved in several ways. Our focus in this paper is to consider designs that are space-filling in low dimensions (McKay, Beckman and Conover (1979), Owen (1992) and Tang (1993)). Other approaches include the use of distance criteria, as in Johnson, Moore and Ylvisaker (1990) and discrepancies as in Fang, Lin Winker and Zhang (2000), for design selection.

Considering the three principles, this paper constructs some nested space-filling designs. A nested space-filling design consists of two sets of design points, D_l and D_h, with a nested structure $D_h \subset D_l$ such that both D_h and D_l achieve uniformity in low dimensional projections. The basic idea is to construct an OA-based Latin hypercube (D_l) (Tang (1993)) that contains a sub-design (D_h) with an orthogonal array structure. The remainder of the article is organized as follows. Section 2 considers a motivating example. The main construction results are presented in Section 3. Section 4 concludes the paper with a discussion and some further results.

Before moving on to the next section, we make a note on the terminology used in this paper. When we say that a design is space-filling in low dimensions or achieves uniformity in low dimensions, we mean that when projected onto low dimensions, the design points are evenly scattered in the design region. The precise meanings of these phrases will be made clear when we present our concrete results, as in Theorem 2 and Lemma 1.

2. Background and Motivating Example

2.1. Background material

We introduce Latin hypercubes, orthogonal arrays, and OA-based Latin hypercubes. An $n \times m$ matrix $D = (d_{ij})$ is called a Latin hypercube of n runs for m factors if each column of D is a permutation of $1, \ldots, n$. There are two natural ways of generating design points in the unit cube $[0, 1]^m$ based on a given Latin hypercube. The first is through $x_{ij} = (d_{ij} - 0.5)/n$, with the n points given by $(x_{i1}, \ldots, x_{im})$ with $i = 1, \ldots, n$. The other is through $x_{ij} = (d_{ij} - u_{ij})/n$, with the n points given by $(x_{i1}, \ldots, x_{im})$ with $i = 1, \ldots, n$, where u_{ij} are independent random variables with a common uniform distribution on $[0, 1]$. The difference between the two methods can be seen as follows. When projected onto each of the m variables, both methods have the property that one and only one of the n design points fall within each of the n small intervals defined by $[0, 1/n), [1/n, 2/n), \ldots, [(n - 1)/n, 1]$. The first method gives the mid-points of these intervals while the second gives points that are uniformly distributed in their corresponding intervals.

An orthogonal array of size n, m constraints, s levels, and strength $t \geq 2$ is an $n \times m$ matrix with entries from a set of s levels, usually taken as $1, \ldots, s$, such that for every $n \times t$ submatrix, each of the s^t level combinations occurs the same number of times. Such an array is denoted by $OA(n, m, s, t)$. Regular fractional factorial designs, as discussed in Wu and Hamada (2000), are the most familiar examples of orthogonal arrays. Let A be an $OA(n, m, s, t)$ with its s levels denoted by $1, \ldots, s$. Then in every column of A, each level occurs $q = n/s$ times. For each column of A, if we replace the q ones by a permutation of $1, \ldots, q$, replace the q twos by a permutation of $q + 1, \ldots, 2q$, and so on, we obtain an OA-based Latin hypercube (Tang (1993)). In addition to achieving maximum stratification in one dimensions, OA-based Latin hypercubes have attractive space-filling properties when projected onto t or lower dimensions.

2.2. A motivating example

We discuss an example taken from Qian et al. (2006). The example deals with designing a heat exchanger for a representative electronic cooling application. The response of interest is the heat transfer rate in the heat exchanger. Five

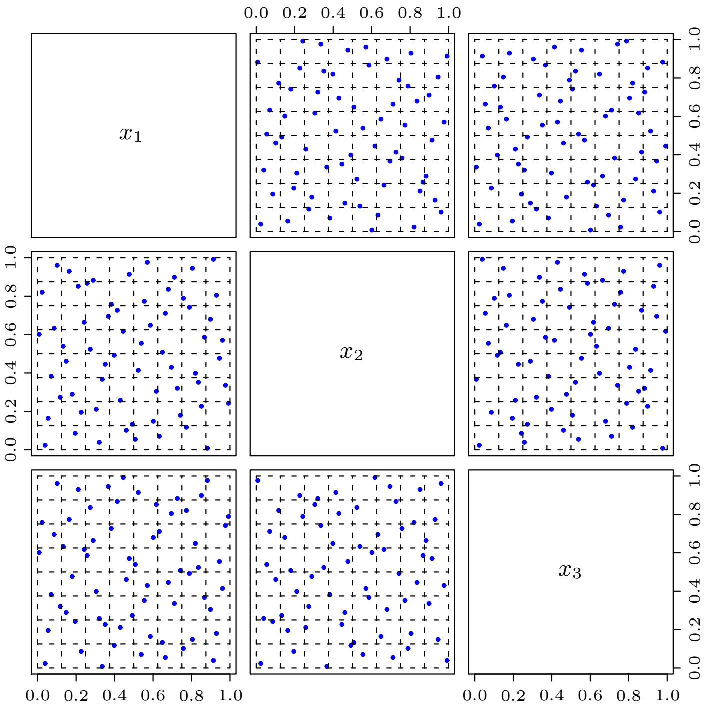

Figure 1. Bivariate projections among x_1, x_2, x_3 of a 64-run OA-based Latin hypercube D_l.

design variables x_1, x_2, x_3, x_4, x_5, including mass flow rate of entry air and temperature of entry air, can potentially affect the thermal process. These variables are assumed to take values in the unit hypercube $[0, 1]^5$. Two types of computer experiments – an HE based on finite element simulations (FLUENT (1998)) and an LE based on finite difference simulations (Incropera and DeWitt (1996) and Seepersad, Dempsey, Allen, Mistree and McDowell (2004)) – are used to analyze the impact of these factors on the heat transfer rate. The HE and LE have different levels of accuracy and computational times: each HE run requires two to three orders of magnitude more computing time than the corresponding LE run; the HE runs are generally more accurate than the LE runs by 10% to 15%. To construct D_h and D_l, Qian et al. (2006) used the following two-step procedure:

Step 1: Take D_l to be an OA-based Latin hypercube with n_1 runs;

Step 2: A subset D_h with n_2 runs is selected from D_l using the maximin distance criterion (Johnson, Moore and Ylvisaker (1990)).

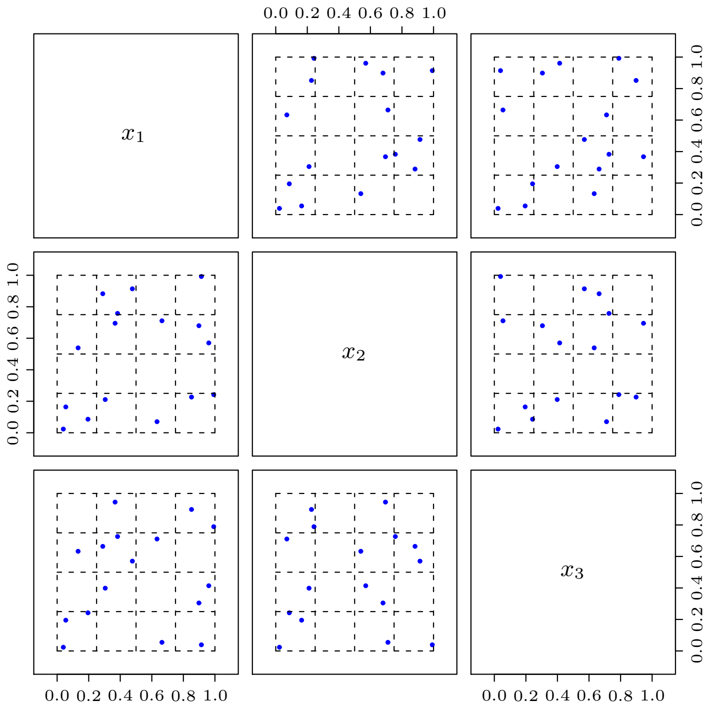

Figure 2. Bivariate projections among x_1, x_2, x_3 of the 16-run design D_h selected from D_l in Figure 1 using the maximin distance criterion.

In this example, n_1 and n_2 are 64 and 16, respectively. An $OA(64, 5, 8, 2)$ from Neil Sloane's webpage (*http://www.research/att.com/˜njas*) is used to construct an OA-based Latin hypercube. This is D_l in step 1. Figure 1 depicts the bivariate projections of D_l and, for brevity, only those among the variables x_1, x_2, x_3 are presented. Computing a 16-run design D_h in step 2 is carried out by using a simulated annealing algorithm (Belisle (1992)) with 2,000 iterations. Figure 2 presents the bivariate projections of D_h among x_1, x_2, x_3, showing that D_h is far from being space-filling in two dimensions.

Table 1 gives a special version of $OA(64, 5, 8, 2)$, constructed using a general method in Section 3. We now use this orthogonal array to obtain an OA-based Latin hypercube, which serves as our new D_l. The bivariate projections of this design are similar to those in Figure 1.

Now let D_h be obtained by selecting the 16 points from D_l that correspond to runs 1-4, 9-12, 17-20, and 25-28 of the array in Table 1. The bivariate projections are given in Figure 3. We see that this design D_h has an underlying orthogonal array structure. In fact, the matrix given by collecting runs 1-4, 9-12, 17-20,

Table 1. An $OA(64,5,8,2)$ that contains a nesting $OA(16,5,4,2)$. More precisely, the matrix consisting of runs 1-4, 9-12, 17-20, and 25-28 becomes an $OA(16,5,4,2)$ if the eight levels are collapsed into four levels according to the scheme: $(1,2)\rightarrow 1$, $(3,4)\rightarrow 2$, $(5,6)\rightarrow 3$, $(7,8)\rightarrow 4$.

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

and 25-28 of the $OA(64,5,8,2)$ in Table 1 becomes an $OA(16,5,4,2)$ if the eight levels are collapsed into four levels according to the following scheme:

$$(1,2)\rightarrow 1;\quad (3,4)\rightarrow 2;\quad (5,6)\rightarrow 3;\quad (7,8)\rightarrow 4.$$

The $OA(64,5,8,2)$ in Table 1 is a special case of the general results to come in Section 3.

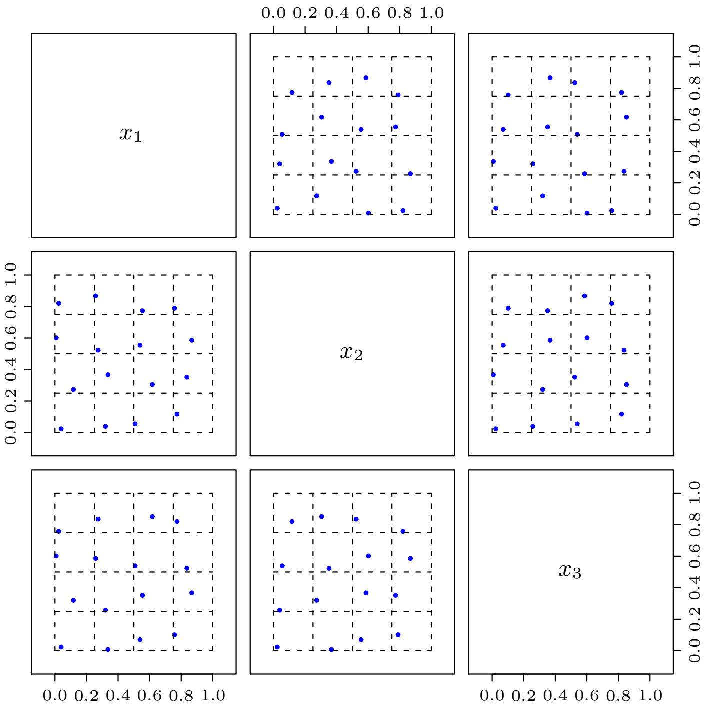

Figure 3. The bivariate projections among x_1, x_2, x_3 of D_h obtained from the Latin hypercube D_l constructed using the orthogonal array in Table 1.

3. General Results

3.1. Galois fields and Rao-Hamming construction

We give a brief account of Galois fields and the Rao-Hamming construction for orthogonal arrays. Interested readers can refer to Hedayat, Sloane and Stufken (1999) for more detailed discussion. A field F is a nonempty set equipped with two binary operations $+$ and $*$ on F such that the following properties hold:

1. $a + b = b + a$ for all $a, b \in F$;
2. $(a + b) + c = a + (b + c)$ for all $a, b, c \in F$;
3. there exists a unique element $0 \in F$ such that $a + 0 = a$ all $a \in F$;
4. for any $a \in F$, there exists a unique element $-a \in F$ such that $a + (-a) = 0$;
5. $a * b = b * a$ for all $a, b \in F$;
6. $(a * b) * c = a * (b * c)$ for all $a, b, c \in F$;
7. there exists a unique element $1 \in F$ such that $a * 1 = a$ all $a \in F$;

8. for any $a \in F, a \neq 0$, there exists a unique element $a^{-1} \in F$ such that $a * a^{-1} = 1$;
9. $a * (b + c) = a * b + a * c$ for $a, b, c \in F$.

All rational numbers form a field with respect to the usual addition and multiplication; so do all real numbers. A field with a finite number of elements is called a finite field or Galois field, and we use $GF(s)$ to denote a Galois field with s elements. Let p be a prime number. Then the set of residues $\{0, 1, \ldots, p-1\}$ modulo p forms a Galois field $GF(p)$ of order p under addition and multiplication modulo p. Let $g(x) = b_0 + b_1 x + \cdots + b_u x^u$, where $b_j \in GF(p)$ and $b_u = 1$ be an irreducible polynomial of degree u. Then the set of all polynomials of degree $u-1$ or lower $\{a_0 + a_1 x + \cdots + a_{u-1} x^{u-1} | a_j \in GF(p)\}$ is a Galois field $GF(p^u)$ of order p^u under addition and multiplication of polynomials modulo $g(x)$. For any polynomial $f(x)$ with coefficients from $GF(p)$, there exist unique polynomials $q(x)$ and $r(x)$ such that $f(x) = q(x)g(x) + r(x)$ where the degree of $r(x)$ is smaller than u. This $r(x)$ is the residue of $f(x)$ modulo $g(x)$, which is usually written as $f(x) = r(x) \pmod{g(x)}$. For every prime p and every integer $u \geq 1$, there exists a $GF(p^u)$. In fact, all Galois fields have this form. Another important result is that the multiplicative group $GF(p^u) \setminus \{0\}$ is cyclic, allowing easy calculations under multiplication.

Let $s = p^u$. The Rao-Hamming construction gives an $OA(n, m, s, 2)$ where $n = s^k$ and $m = (s^k - 1)/(s - 1)$ for any integer $k \geq 2$. This is done as follows. Let z_j be a column vector of length k with the jth component equal to one and all the others equal to zero for $j = 1, \ldots, k$. We then obtain a $k \times m$ matrix Z with $m = (s^k - 1)/(s - 1)$ by collecting all the column vectors given by

$$z = c_1 z_1 + \cdots + c_k z_k, \quad \text{where } c_j \in GF(s)$$

and the first nonzero entry in $(c_1, \ldots, c_k)$ is one. Taking all linear combinations of the row vectors of Z with coefficients from $GF(s)$, we obtain an $OA(n, m, s, 2)$ with $n = s^k$ and $m = (s^k - 1)/(s - 1)$.

3.2 Construction of nested orthogonal arrays

Let $s_1 = p^{u_1}$ and $s_2 = p^{u_2}$ be two powers of the same prime p where $u_1 > u_2 \geq 1$. If a polynomial with coefficients from $GF(p)$ has degree $u_2 - 1$ or lower, it belongs to both $GF(s_1)$ and $GF(s_2)$. So $GF(s_2)$ is a subset of $GF(s_1)$, although it is not necessarily true that $GF(s_2)$ is a subfield of $GF(s_1)$. For $GF(s_1)$ to have a subfield of order s_2, we must have that u_2 divides u_1.

All polynomials considered in this paper have their coefficients from $GF(p)$ and we make no further mention of this. We use A to denote the $OA(n_1, m_1, s_1, 2)$, where $n_1 = s_1^k$ and $m_1 = (s_1^k - 1)/(s_1 - 1)$, given by the Rao-Hamming construction as discussed in Section 3.1. Recall that the rows of this array A are all the linear combinations of the row vectors of Z with coefficients from $GF(s_1)$, where Z consists of all column vectors $z = c_1z_1 + \cdots + c_kz_k$, where $c_j \in GF(s_1)$ and the first nonzero entry in $(c_1, \ldots, c_k)$ is one. Now consider a subarray A_1 of A, obtained by taking all linear combinations of the row vectors of Z_1 with coefficients from $GF(s_1)$, where Z_1 is a submatrix of Z given by collecting all the column vectors $z = c_1z_1 + \cdots + c_kz_k$, with $c_j \in GF(s_2) \subset GF(s_1)$ and the first nonzero entry in $(c_1, \ldots, c_k)$ is one. Clearly, A_1 is an $OA(n_1, m_2, s_1, 2)$ where $n_1 = s_1^k$ and $m_2 = (s_2^k - 1)/(s_2 - 1)$. It should be stressed that all the calculations in the construction of A_1, as well as that of A, are performed in $GF(s_1)$, notwithstanding that the subset $GF(s_2)$ of $GF(s_1)$ is used for selecting the columns of A_1.

Let us focus on A_1. Let $g_1(x)$ be the chosen irreducible polynomial that defines $GF(s_1)$. Now consider how the entries of A_1 are obtained during the construction of A_1. Calculations for an entry of A_1 in $GF(s_1)$ can be carried out in two stages. In the first stage, we only conduct polynomial calculations without being modulo $g_1(x)$ and let the resulting polynomial be $f(x)$. In the second stage, the residue of $f(x)$ modulo $g_1(x)$ is found and it is this residue that becomes the entry of A_1. For convenience in presentation, we use $f_{g_1}(x)$ to denote the residue of $f(x)$ modulo $g_1(x)$. Using this notation, all entries of A_1 have the form of $f_{g_1}(x)$.

Now consider the submatrix A_2 of A_1 given by those linear combinations of the row vectors of Z_1 with coefficients from $GF(s_2)$, a subset of $GF(s_1)$. The matrix A_2 may not be an orthogonal array in itself, but becomes an $OA(n_2, m_2, s_2, 2)$ if its s_1 levels are suitably collapsed into s_2 levels, where $n_2 = s_2^k$ and $m_2 = (s_2^k - 1)/(s_2 - 1)$. Level collapsing is done modulo $g_2(x)$, the irreducible polynomial that defines $GF(s_2)$. We are ready to present the results.

Theorem 1. *Consider A_1 and A_2 as constructed above. Then we have that*

(i) *the matrix A_1 is an $OA(n_1, m_2, s_1, 2)$, and*

(ii) *provided that $2u_2 \le u_1 + 1$, the submatrix A_2 of A_1 becomes an $OA(n_2, m_2, s_2, 2)$ when the s_1 levels are collapsed into s_2 levels according to the scheme: $f_{g_1}(x) \to (f_{g_1})_{g_2}(x)$.*

Proof. Only part (ii) of Theorem 1 requires a proof. Let $f_{g_1}(x)$ be an entry of A_2, where $f(x)$ denotes the polynomial for this entry before being modulo $g_1(x)$. Then the matrix A_2^* obtained by replacing every entry $f_{g_1}(x)$ of A_2 by $f_{g_2}(x)$ is an $OA(n_2, m_2, s_2, 2)$, as A_2^* is simply the Rao-Hamming construction based on $GF(s_2)$. Part (ii) of Theorem 1 is established if we can show that

$$(f_{g_1})_{g_2}(x) = f_{g_2}(x) \tag{3.1}$$

for every $f(x)$ that may result from calculating the entries of A_2. Note that $f(x)$ would be the entry of A_2 if calculations modulo $g_1(x)$ were not performed. Since the degree of polynomial $f(x)$ is at most $2(u_2-1)$, which is less than or equal to u_1-1, we must have that $f_{g_1}(x)=f(x)$. Thus (3.1) holds.

Equation (3.1) does not hold in general. So the condition $2u_2 \le u_1+1$ is needed to ensure the validity of part (ii) of Theorem 1. The level collapsing scheme in Theorem 1 may look a bit abstract but the idea is simple. Once A_2 and A_1 have been constructed, the s_1 levels are all the polynomials of degree u_1-1 or lower, and the irreducible polynomial $g_1(x)$ plays no further role in level collapsing. A polynomial of degree u_1-1 or lower, as one of the s_1 levels, is simply mapped to its residue modulo $g_2(x)$.

Example 1. Let $p=2$, $u_1=3$, $u_2=2$, $s_1=p^{u_1}=8$ and $s_2=p^{u_2}=4$. The condition $2u_2 \le u_1+1$ is satisfied. We use $g_1(x)=x^3+x+1$ and $g_2(x)=x^2+x+1$, both irreducible, to define $GF(8)$ and $GF(4)$. We obtain A_1 and A_2 using the construction method described earlier for $k=2$. From Theorem 1, A_1 is an $OA(64,5,8,2)$, and A_2 becomes an $OA(16,5,4,2)$ when the eight levels, $0,1,x,x+1,x^2,x^2+1,x^2+x,x^2+x+1$ are collapsed into four levels $0,1,x,x+1$ according to $(0,x^2+x+1)\to 0$, $(1,x^2+x)\to 1$, $(x,x^2+1)\to x$, $(x+1,x^2)\to x+1$. For example, x^2 is mapped to $x+1$ because the residue of x^2 modulo $g_2(x)=x^2+x+1$ is $x+1$.

3.3. Construction of nested space-filling designs

The pair of nested orthogonal arrays $A_2 \subset A_1$ does not automatically generate nested space-filling designs if the s_1 levels are arbitrarily labeled. In constructing OA-based Latin hypercube using A_1, the s_1 levels of A_1, originally represented by the polynomials of degree u_1-1 or lower, have to be first labeled as $1,\ldots,s_1$. Although any such labeling of the s_1 levels will give an OA-based Latin hypercube which is space-filling in two dimensions, the subset of points corresponding to A_2 may not have good space-filling properties. Care should be taken in labeling the levels to ensure that this subset of points also achieves stratification in two dimensions.

The key idea here is that the s_1 levels of A_1 must be labeled in such a way that the group of levels that are mapped to the same level should form a consecutive subset of $\{1,\ldots,s_1\}$. We now give a precise description of how the levels should be labeled. The level collapsing scheme in Theorem 1 through $g_2(x)$ divides the s_1 levels into s_2 groups, each of size $e=s_1/s_2$. Two levels $f_1(x)$ and $f_2(x)$ belong to the same group if $f_1(x)-f_2(x)=0 \pmod{g_2(x)}$. These s_2 groups can be arbitrarily, or randomly if one wishes, labeled as groups $1,\ldots,s_2$. Then the e levels within the ith group can be arbitrarily, or randomly if one wishes, labeled as $(i-1)e+1,\ldots,(i-1)e+e$ for $i=1,\ldots,s_2$.

Example 2. In Example 1, the four groups of levels are $(0, x^2+x+1)$, $(1, x^2+x)$, (x, x^2+1), and $(x+1, x^2)$. One choice of level labeling according to the just described method is to label $(0, x^2+x+1)$ as levels 1 and 2, $(1, x^2+x)$ as levels 3 and 4, (x, x^2+1) as levels 5 and 6, and $(x+1, x^2)$ as levels 7 and 8. The $OA(64, 5, 8, 2)$ in Table 1 is obtained in precisely this way. This particular choice of level labeling does not lose any generality. If one wishes to randomize, one could randomly permute the four groups of levels and then randomize the two levels within each group to obtain a permutation of the eight levels $1, \ldots, 8$; the levels in this permutation could then be sequentially relabeled as levels $1, \ldots, 8$. For example, randomly permuting the four groups of levels might give $(3, 4), (1, 2), (5, 6), (7, 8)$ and further randomizing the two levels within each group might give $(4, 3), (1, 2), (6, 5), (7, 8)$. Then levels 4, 3, 1, 2, 6, 5, 7, 8 are relabled as 1, 2, 3, 4, 5, 6, 7, 8.

Suppose that the levels of A_1 in Theorem 1 have been appropriately labeled as $1, \ldots, s_1$ in accordance with the above method. Naturally, the entries of A_2 also come from this set of levels. We now use A_1 to obtain an OA-based Latin hypercube design as discussed in Section 2.1 and let D_l denote the set of points. Let D_h be the subset of points of D_l that correspond to A_2. Then D_h and D_l provide two space-filling designs with D_h nested within D_l. We make this precise in the following theorem.

Theorem 2. *Let $D_h \subset D_l$ be as constructed above. Then we have that*

(i) *in addition to achieving maximum stratification in one dimensions, design D_l achieves stratification on $s_1 \times s_1$ grids in two dimensions, and*

(ii) *D_h achieves stratification on $s_2 \times s_2$ grids in two dimensions.*

4. Discussion and Further Results

The nested space-filling designs $D_h \subset D_l$ constructed in Section 3 achieve more than just what the three principles require. Both D_h and D_l achieve stratification in two dimensions, but design D_l does that on finer grids and therefore provides a better coverage of the design space in two dimensions. This is fairly natural as D_l has more design points and thus more can be expected from it in terms of filling the design space.

The above discussion leads to an alternative approach. One might wish to consider nested designs $D_h \subset D_l$ with the property that, while both D_h and D_l still achieve stratification in two dimensions, D_l also achieves stratification in three dimensions. This approach is especially attractive when one feels that the response variable depends on the input variables in such a complex fashion that three-way interactions could play a significant role in predicting the response. Construction of such nested space-filling designs is considerably simpler. Let

A be an $OA(n_1, m, s, 3)$, an orthogonal array of strength three with its s levels denoted by $1, \ldots, s$. Consider the submatrix B of A obtained by selecting those rows of A with the entries in the first column being level 1. We now obtain two matrices A_1 and A_2 with A_1 given by deleting the first column of A and A_2 being the submatrix of A_1 with its rows corresponding to B. Then we have that A_1 is an $OA(n_1, m-1, s, 3)$ and A_2 is an $OA(n_2, m-1, s, 2)$, where $n_2 = n_1/s$. Let us use A_1 to construct an OA-based Latin hypercube. Denote the resulting design by D_l and the subset of points corresponding to A_2 by D_h. Then D_h and D_l give a pair of nested designs with varying degrees of space-filling properties.

Lemma 1. *Let $D_h \subset D_l$ be as constructed above. We have that*
(i) *design D_l achieves stratification on $s \times s \times s$ grids in three dimensions, and*
(ii) *design D_h achieves stratification on $s \times s$ grids in two dimensions.*

Two useful results for orthogonal arrays of strength three are that an $OA(s^3, s+1, s, 3)$ can be constructed if s is an odd prime power and an $OA(s^3, s+2, s, 3)$ can be constructed if s is an even prime power. These results are due to Bush (1952), and are also available from Section 3.2 in Hedayat, Sloane and Stufken (1999).

Example 3. Let $s = 2^2 = 4$. We can construct an $OA(64, 6, 4, 3)$ using the second result just mentioned. From this array, we obtain A_1 and A_2 where A_1 is an $OA(64, 5, 4, 3)$ and A_2, nested within A_1, is an $OA(16, 5, 4, 2)$. Let D_l be a Latin hypercube based on A_1 and D_h be the subset of points corresponding to A_2. Then design D_h achieves stratification on 4×4 grids in two dimensions whereas D_l achieves stratification on $4 \times 4 \times 4$ grids in three dimensions. It is interesting to compare this pair of nested space-filling designs with that discussed in Section 2.2, although the two D_h's are similar, the two D_l's are quite different. The D_l in this example achieves stratification in three dimensions while the D_l in Section 2.2 achieves stratification in two dimensions, but on finer 8×8 grids.

The idea of using orthogonal arrays of strength three naturally generalizes. If an orthogonal array of strength t is used, we can construct D_l and D_h such that D_l achieves stratification in t dimensions and D_h achieves stratification in $t-1$ dimensions.

The two methods for constructing nested space-filling designs both produce a much larger n_1 than n_2, a desirable feature as the LE is much cheaper than the HE. In situations where the costs of running the LE and HE are not so drastically different, as in the case of the motivating example in Section 2.2, one can follow a simple strategy which we briefly describe. Let A_2 be an $OA(n_2, m, s, t)$ and A_1 be an $OA(n_1, m, s, t)$ obtained by juxtaposing A_2 several times. If D_l is an OA-based Latin hypercube constructed from A_1 and D_h the subset of points corresponding to A_2, then D_h and D_l provide a pair of nested designs that both achieve stratification on s^t grids in t dimensions. One can also consider independently randomizing the levels of each A_2 within A_1 - this has no effect on the space-filling properties of D_h and D_l in t dimensions, but can possibly improve their space-filling properties in higher dimensions. This method, though simple, is very flexible and deserves consideration in practical applications.

We conclude the paper with a brief discussion on the modeling and analysis of computer experiments with two levels of accuracy. Gaussian process models are popular for computer experiments and can be used for the data from both the LE and the HE. Integration of the two sets of results from analyzing LE and HE data is not straightforward but the basic idea is simple. Since the HE is more accurate than the LE, the objective is to build a prediction model that is capable of producing results close to the HE data. We can achieve this by first fitting a Gaussian process model to the LE data and then adjusting the fitted model using the HE data so that the resulting model can better predict the HE data. For details on this analysis method, we refer to Kennedy and O'Hagan (2001), Reese et al. (2004), Qian et al. (2006) and Qian and Wu (2008).

Acknowledgement

The authors would like to thank an associate editor and two referees for helpful comments. The research of Peter Qian is supported by NSF grant DMS-0705206 and that of Jeff Wu is supported by NSF grant DMI 0620259. Boxin Tang's research is supported by the Natural Sciences and Engineering Research Council of Canada.

References

Belisle, C. J. P. (1992). Convergence theorems for a class of simulated annealing algorithms on R^d. *J. Appl. Probab.* **29**, 885-895.

Bush, K. A. (1952). Orthogonal arrays of index unity. *Ann. Math. Statist.* **23**, 426-434.

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

Fang, K. T., Lin, D. K., Winker, P. and Zhang, Y. (2000). Uniform design: theory and application. *Technometrics* **42**, 237-248.

FLUENT (1998). Fluent Inc., Release 5.5.14 (3d, segregated, laminar).

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

Incropera, F. P. and DeWitt, D. P. (1996). *Fundamentals of Heat and Mass Transfer.* 3rd Edition. Wiley, New York.

Johnson, M. E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *J. Statist. Plann. Inference* **26**, 131-148.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *J. Roy. Statist. Soc. Ser. B* **63**, 425-464.

McKay, D., Beckman, R. J. and Conover, W. 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.

Owen, A. (1992). Orthogonal arrays for computer experiments, integration and visualization. *Statist. Sinica* **2**, 439-452.

Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K. and Wu, C. F. J. (2006). Building surrogate models based on detailed and approximate simulations. *ASME J. Mechanical Design* **128**, 668-677.

Qian, Z. and Wu, C. F. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics* **50**, 192-204.

Reese, C. S., Wilson, A. G, Hamada, M., Martz, H. F. and Ryan, K. J. (2004). Intergated analysis of computer and physical experiments. *Technometrics* **46**, 153-164.

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

Seepersad, C. C., Dempsey, B. M., Allen, J. K., Mistree, F. and McDowell, D. L. (2004). Design of multifunctional honeycomb materials. *AIAA J.* **42**, 1025-1033.

Tang, B. (1993). Orthogonal array-based Latin hypercubes. *J. Amer. Statist. Assoc.* **88**, 1392-1397.

Wu, C. F. J. and Hamada, M. (2000). *Experiments: Planning, Analysis and Parameter Design Optimization*. Wiley, New York.

Department of Statistics, University of Wisconsin, Madison, WI 53706, U.S.A.

E-mail: zhiguang@stat.wisc.edu

Department of Statistics and Actuarial Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.

E-mail: boxint@stat.sfu.ca

School of Industrial and Systems Engineering, Georgia Tech., Atlanta, GA 30332, U.S.A.

E-mail: jeffwu@isye.gatech.edu

(Received December 2006; accepted August 2007)