Biometrika (2006), **93**, 2, *pp*. 279–288
© 2006 Biometrika Trust
Printed in Great Britain

A construction method for orthogonal Latin hypercube designs

By DAVID M. STEINBERG

Department of Statistics and Operations Research, Tel-Aviv University, Tel-Aviv 69978, Israel

dms@post.tau.ac.il

AND DENNIS K. J. LIN

Department of Supply Chain & Information Systems, The Pennsylvania State University, University Park, Pennsylvania 16802, U.S.A.

dkl5@psu.edu

Summary

The Latin hypercube design is a popular choice of experimental design when computer simulation is used to study a physical process. These designs guarantee uniform samples for the marginal distribution of each single input. A number of methods have been proposed for extending the uniform sampling to higher dimensions. We show how to construct Latin hypercube designs in which all main effects are orthogonal. Our method can also be used to construct Latin hypercube designs with low correlation of first-order and second-order terms. Our method generates orthogonal Latin hypercube designs that can include many more factors than those proposed by Ye (1998).

Some key words: Computer experiment; Factorial design; Rotation design; U-design.

1. Introduction

Latin hypercube designs were introduced by McKay et al. (1979) and have proved to be a popular choice for experiments run on computer simulators (Santner et al., 2003, Ch. 5) and in global sensitivity analysis (Helton & Davis, 2000). Latin hypercube designs are geared for simultaneous study of p input factors. Whereas standard factorial designs limit each input factor to a small number of distinct values, Latin hypercube designs use different settings of each factor on each experimental run, with the settings spread out uniformly along each factor axis. Thus Latin hypercube designs achieve a very 'uniform' coverage of each individual factor.

There is no guarantee that Latin hypercube designs will have good multivariate properties. The original construction of McKay et al. (1979) was to mate the levels randomly for each of the p factors. Their proposal leads to designs in which most pairs of input factors have low correlations, but, with a large number of input factors, a common situation in computer experiments, there will usually be some pairs with correlations of 0·3 or more. The presence of correlated input factors can complicate the subsequent data analysis and make it more difficult to identify the most important input factors.

Several authors have proposed modifications to the original construction scheme that lead to Latin hypercube designs with low correlations between pairs of input factors.

Owen (1992) and Tang (1993) showed how orthogonal arrays could be used to generate Latin hypercube designs with better balance in low-dimensional projections. Owen (1994) presented an algorithm for generating Latin hypercube designs with low pairwise correlations between input factors. Tang (1998) extended this approach, considering correlations among the input factors and also with higher-order terms derived from the factors. Ye (1998) presented a method for constructing 'orthogonal' Latin hypercube designs in which all the input factors have zero correlation; some additional constructions can be found in a University of Michigan technical report by Ye. Butler (2001) showed how to construct Latin hypercube designs in which the terms in a class of trigonometric regression models are orthogonal to one another.

We present here a new construction method for orthogonal Latin hypercube designs. The construction is very simple and involves a combination of two ideas. The first, due to Beattie & Lin (1997, 2004, 2005), is that certain Latin hypercube designs can be constructed by rotating the points in a two-level factorial design, a technique that preserves the orthogonality of the original factorial. The second, due to Bursztyn & Steinberg (2002), is that rotations can be applied to groups of factors, thereby greatly increasing the number of factors in the resulting design. Our construction produces orthogonal Latin hypercube designs with n rows, where $n=2^k$ and $k=2^m$. The number of possible factors is almost as large as n. We use 16-run Latin hypercube designs to illustrate our ideas.

2. Latin hypercube designs

We will describe experimental designs for p factors in n runs using an $n \times p$ matrix D, where $D_{i,j}$ is the level of factor j on the ith experimental run. Throughout, we will assume that the input factor space is $[-1, 1]^p$ and that the factors can be varied independently in that region.

A design D_L is a Latin hypercube design if each column in the design matrix includes n uniformly spaced levels. The key question in constructing a Latin hypercube design is how to mate the levels for the different factors. In the original paper of McKay et al. (1979) the factors were mated by randomly permuting the entries in each column of the matrix. The papers cited earlier for improving Latin hypercube designs employ a number of alternative ideas for mating levels.

There are several variations on how to space the levels 'uniformly' for each factor. The simplest scheme, and the one that we will employ in this paper, is to take $-1, -1+2/(n-1), \ldots, 1-2/(n-1), 1$ as the levels for each factor. Other lattices in $[-1, 1]$ with equally spaced levels could also be used. An alternative is to divide each factor axis into n bins of equal width and then to select a level at random from each bin. Yet another option, relevant for input factors that represent environmental variables with known probability distributions, is to choose the levels as quantiles of probabilities on an equally spaced grid.

3. Orthogonal Latin hypercube designs as rotated 2^k factorials

Beattie & Lin (1997, 2004, 2005) showed that a class of orthogonal Latin hypercube designs with $p=k$ factors can be generated as rotations of 2^k factorial designs. Their method can be applied when k is itself a power of 2, 2^m, and the number of runs is $n=2^k$. We briefly review here the construction of Beattie & Lin.

Denote by D_k the $n \times k$ design matrix for the 2^k factorial design, with -1 and 1 as the two levels for each factor. It is well known that this is an orthogonal design, i.e. that $D_k' D_k = nI$, where I is the identity matrix. A $k \times k$ real matrix R is a rotation matrix if $R'R = I$. Such a matrix can be used to construct a rotated version of the two-level factorial, $D_R = D_k R$. Rotation preserves the orthogonality of the two-level factorial, as $D_R' D_R = R' D_k' D_k R = R'(nI)R = nR'R = nI$.

Beattie & Lin defined a sequence of rotation matrices for which D_R is a Latin hypercube design. The rotation matrix for $k = 2^m$ factors is defined by the following recursive scheme. Let

$$V_0 = [1], \tag{1}$$

$$V_m = \begin{bmatrix} V_{m-1} & -(2^m)V_{m-1} \\ (2^m)V_{m-1} & V_{m-1} \end{bmatrix}. \tag{2}$$

It is easy to check that V_m is orthogonal. The entries of V_m are $1, \pm 2^1, \ldots, \pm 2^{k-1}$, where $k = 2^m$. Each of these integers, with either a plus or minus sign, appears exactly once in each column. The uth entry in the jth column of $D_k V_m$ thus has the form $\sum_{i=1}^{k} c_i 2^{i-1}$, where c_i is either -1 or 1, depending on $D_{k,u,j}$ and the sign of 2^{i-1} in column j of V_m. Running through all the rows of D_k generates all possible binary combinations of these powers of 2, regardless of the signs in column j of V_m. Thus $D_k V_m$ is an $n \times k$ matrix each of whose columns has the entries $-(n-1), -(n-3), \ldots, n-3, n-1$. Simple rescaling converts V_m into a rotation matrix,

$$R_m = (1/a_m)V_m, \tag{3}$$

with $a_0 = 1$ and $a_m = \{\prod_{j=1}^{m} (1 + 2^{2j})\}^{\frac{1}{2}}$ for $m = 1, 2, \ldots$. For example,

$$R_2 = 1/\surd(85) \begin{bmatrix} 1 & -2 & -4 & 8 \\ 2 & 1 & -8 & -4 \\ 4 & -8 & 1 & -2 \\ 8 & 4 & 2 & 1 \end{bmatrix}. \tag{4}$$

Note that the design $D_{R(2)} = D_4 R_2$ has factor levels that are no longer in the design space $[-1, 1]^k$. Further scaling produces an orthogonal Latin hypercube design for k factors in n runs with the desired levels $-1, -1 + 2/(n-1), \ldots, 1 - 2/(n-1), 1$ for each factor.

4. Rotating in groups

Bursztyn & Steinberg (2002) proposed the idea of independently rotating groups of factors in two-level designs. Let D_F be a 2^{f-q} fractional factorial design with $n = 2^k$ runs and let R be a $t \times t$ rotation matrix. Suppose we can decompose the f factors in D_F into B sets of t factors each, with $f - Bt$ factors left over. Let $D_{S1}, \ldots, D_{SB}$ be the design matrices obtained from projecting D_F on to each of the B sets of t factors. We now obtain an orthogonal rotation design by rotating each of the sets. Let R_B be a $Bt \times Bt$ block diagonal matrix with B copies of R on the diagonal. Then R_B is a rotation matrix and the rotation design is

$$D_{R(B)} = [D_{S1} \vdots \ldots \vdots D_{SB}] R_B \tag{5}$$

$$= [D_{S1} R \vdots \ldots \vdots D_{SB} R]. \tag{6}$$

The orthogonality follows from the fact that $D'_{Si} D_{Si} = nI$ for all i and $D'_{Si} D_{Sj} = 0$ whenever $i \neq j$.

5. Orthogonal 16-run Latin hypercube designs from group rotations

In this section we use our construction method to generate orthogonal Latin hypercube designs with 16 runs. We rotate factors in the saturated 2^{15-11} design in groups of four, using the matrix V_2 defined earlier, and take care of all scaling in a single step. Throughout we denote the columns in the saturated design by A, B, C, D and their products.

Example 1. Our method can be used to derive an orthogonal 16-run Latin hypercube design with 12 factors. Such a design is shown in Table 1. This design was generated by grouping the columns A, B, C and D in one set, AB, AC, ABC and AD in a second set and BC, BD, ABD and BCD in the third set. As we will see in the next section, the key property of the sets for achieving a Latin hypercube design is that each one gives a full 2^4 factorial design. Examination of the signs of the entries in Table 1 shows that it has the orthogonal array structure of a U-design (Tang, 1993), corresponding to the 2^{12-8} base design that was rotated.

The design in Table 1 has a very high factor-to-run ratio and can be useful in factor screening. Typically, this would proceed by fitting a first-order model. If only a small number of factors were found to be important, a next step might be to look for some higher-order effects involving those factors, as, for example, in Hamada & Wu (1992).

A useful way of characterising the properties of the design is to use the alias matrix for fitting a first-order model when second-order effects may be present. Let X denote the

Table 1: *Example* 1. *An orthogonal Latin hypercube design for* 12 *input factors in* 16 *runs. The numbers in the table should be divided by* 15 *to scale the design to the unit hypercube*

1	2	3	4	5	6	7	8	9	10	11	12
−15	5	9	−3	7	11	−11	7	−9	3	−15	5
−13	1	1	13	−7	−11	11	−7	−1	−13	−13	1
−11	7	−7	−11	13	−1	−1	−13	9	−3	15	−5
−9	3	−15	5	−13	1	1	13	1	13	13	−1
−7	−11	11	−7	11	−7	7	11	5	15	−3	−9
−5	−15	3	9	−11	7	−7	−11	13	−1	−1	−13
−3	−9	−5	−15	1	13	13	−1	−5	−15	3	9
−1	−13	−13	1	−1	−13	−13	1	−13	1	1	13
1	13	13	−1	−9	3	−15	5	11	−7	7	11
3	9	5	15	9	−3	15	−5	3	9	5	15
5	15	−3	−9	−3	−9	−5	−15	−11	7	−7	−11
7	11	−11	7	3	9	5	15	−3	−9	−5	−15
9	−3	15	−5	−5	−15	3	9	−7	−11	11	−7
11	−7	7	11	5	15	−3	−9	−15	5	9	−3
13	−1	−1	−13	−15	5	9	−3	7	11	−11	7
15	−5	−9	3	15	−5	−9	3	15	−5	−9	3

regression matrix for the first-order model, including a column of ones and the 12 factors in the design, scaled to the unit hypercube. Let X_{int} denote the 16×66 matrix with all the possible two-factor interactions and let X_{quad} denote the 16×12 matrix with all the pure quadratic terms. The alias matrices for the first-order model associated with the two-factor interactions and the pure quadratic terms are then given by

$$A_{\text{int}} = (X'X)^{-1} X' X_{\text{int}}, \qquad (7)$$
$$A_{\text{quad}} = (X'X)^{-1} X' X_{\text{quad}},$$

respectively.

A good design for factor screening should maintain relatively small terms in these bias matrices. The orthogonal Latin hypercube design in Table 1 is very successful in terms of minimising bias due to second-order coefficients. Table 2 summarises the two-factor interaction and pure quadratic alias terms for the 12 rows of the alias matrices that relate to the factor main effects. Only 3 of 936 terms, made up of 792 terms in A_{int} plus 144 terms in A_{quad}, exceed 0·2 in absolute value, and they all equal 0·202. Table 2 also compares the alias matrices for the design in Table 1 with alias matrices for Latin hypercube designs created with random ordering of the columns and for Latin hypercube designs that are U-designs (Tang, 1993), in which the plus and minus signs of the Latin hypercube design are a 2^{12-8} fractional factorial design. For both of these competing classes of designs, we examined the alias matrices for 100 randomly generated designs and chose the design with the best performance. Table 2 shows that even these best selections are clearly inferior to our 12-factor design.

Table 2: *Example* 1. *A comparison of the alias properties of* 16-*run*, 12-*factor Latin hypercube designs. For rows associated with the* 12 *main effects, the table shows the percentage of entries in the two-factor interaction alias matrix* A_{int} *and the pure quadratic alias matrix* A_{quad} *with absolute values that are greater than the listed cut-off values. For the standard Latin hypercube designs and U-designs results correspond to the best performers from* 100 *randomly generated designs*

	Two-factor interactions				Pure quadratics			
Design	⩾0·1	⩾0·2	⩾0·4	⩾0·6	⩾0·1	⩾0·2	⩾0·4	⩾0·6
Orthogonal LHD	12·0	0·4	0·0	0·0	7·6	0·0	0·0	0·0
Standard LHD	69·3	40·9	9·0	0·4	49·3	35·4	6·3	1·4
U-design	47·3	18·8	5·1	0·5	54·2	26·4	0·7	0·0

LHD, Latin hypercube design

Other groupings of the factors into three sets of 4 could be used to generate orthogonal Latin hypercube designs with our algorithm. We examined all possible different groupings and found that the resulting designs, although not isomorphic to one another, had nearly identical properties in terms of the alias matrix.

Example 2. We can also produce a 16-run design with 8 factors in which the main effects are orthogonal to each other and to all second-order terms. This design is shown in Table 3. The grouping scheme for this design has A, B, C and D in the first group and the four three-factor interaction columns in the second group. The 16-run design in Ye (1998) has the same orthogonality properties, but with only six factors. Inspection of the

Table 3: *Example* 2. *An orthogonal Latin hypercube design for* 8 *input factors in* 16 *runs, in which the main effects are also orthogonal to all second-order terms. The numbers in the table should be divided by* 15 *to scale the design to the unit hypercube*

1	2	3	4	5	6	7	8
−15	5	9	−3	−15	5	9	−3
−13	1	1	13	−1	−13	−13	1
−11	7	−7	−11	7	11	−11	7
−9	3	−15	5	9	−3	15	−5
−7	−11	11	−7	11	−7	7	11
−5	−15	3	9	5	15	−3	−9
−3	−9	−5	−15	−3	−9	−5	−15
−1	−13	−13	1	−13	1	1	13
1	13	13	−1	13	−1	−1	−13
3	9	5	15	3	9	5	15
5	15	−3	−9	−5	−15	3	9
7	11	−11	7	−11	7	−7	−11
9	−3	15	−5	−9	3	−15	5
11	−7	7	11	−7	−11	11	−7
13	−1	−1	−13	1	13	13	−1
15	−5	−9	3	15	−5	−9	3

designs shows that Ye's design is not isomorphic to some set of 6 columns in our design. In particular, we have 4 design points in which all factors have absolute value of $\frac{1}{15}$ or $\frac{13}{15}$, a property that is not shared by any of the rows in Ye's design. We note that an additional orthogonal Latin hypercube with 8 factors, with the same orthogonality of main effects to all second-order terms, can be found in the technical report by Ye.

6. General construction algorithm

In this section we describe a general method for producing orthogonal Latin hypercube designs with many factors. The number of runs will be n, where $n = 2^k$ and $k = 2^m$. The rotation matrix that we apply is R_m from § 3. This matrix rotates separate groups of k factors. The subdesign $D_{Si} R_m$ for the ith group, rescaled to the unit hypercube, will be a Latin hypercube design if D_{Si} is a full factorial design. The entire design will be a Latin hypercube design provided each subdesign is a Latin hypercube design. The fractional factorial design that we will rotate will be the largest fractional factorial in n runs that can be constructed from sets of k columns, each of which is a full factorial design. We will use the term 'maximal projection set' to describe such a division of the factors in a 2^{p-q} design and will denote by B_k the maximal number of sets achieved.

In general, with $n = 2^k$ runs, the saturated two-level design has $n - 1$ factors, bounding the maximal number of full factorial sets by $B_k \leqslant \lfloor (n-1)/k \rfloor$, where $\lfloor c \rfloor$ is the greatest integer less than or equal to c. For example, with 16 runs, a saturated 2^{p-q} design has 15 factors, so a maximal projection set cannot exceed three sets of 4 factors each. Similarly, with 64 runs a maximal projection set includes at most 10 sets of 6 factors each and with 256 runs a maximal set includes at most 31 sets of 8 factors each.

The following algorithm shows how to construct maximal projection sets that achieve the above bound, $B_k = \lfloor (n-1)/k \rfloor$. The main idea of the algorithm is to associate the columns of the saturated n-run design with elements of the Galois field $\text{GF}(2^k)$; see Appendix A of Hedayat et al. (1999) for a concise summary of relevant results on Galois fields. The elements of $\text{GF}(2^k)$ are polynomials of degree $k-1$ or less, all of whose coefficients are 0 or 1. Each polynomial can be identified by its vector $(a_0, \ldots, a_{k-1})$ of coefficients, and is associated with the generalised interaction of all factors j for which $a_{j-1} = 1$. For example, the polynomial x^2 corresponds to the main effect of factor 3 and the polynomial $1 + x + x^4$ to the interaction of factors 1, 2 and 5. The zero element of $\text{GF}(2^k)$ is given by a vector of k zeros and corresponds to the column of ones in the regression matrix of the factorial design. It is known that there exists a primitive polynomial $f(x)$ of degree k such that the powers of x, modulo $f(x)$, cycle through all $2^k - 1$ nonzero elements of $\text{GF}(2^k)$. The coefficients of $f(x)$ are all 0 or 1 and the coefficients of the powers of x are computed modulo 2.

The following theorems show that our algorithm provides a maximal projection set and that the resulting orthogonal Latin hypercube design is also a U-design (Tang, 1993).

Theorem 1. *The powers of* x *in* $\text{GF}(2^k)$, $x^0, x, x^2, \ldots, x^{2^k-2}$, *provide an ordering of the effect columns into maximal projection sets with* $B_k = \lfloor (n-1)/k \rfloor$. *In fact, every consecutive set of* k *columns in this ordering is a full factorial design.*

Proof. The powers of x in $\text{GF}(2^k)$, $x^0, x, x^2, \ldots, x^{2^k-2}$, generate all the nonzero elements of $\text{GF}(2^k)$ and thus, by the correspondence described above, all the columns in the saturated, regular two-level fractional factorial with $2^k - 1$ factors in 2^k runs. The first k terms in this sequence, $x^0, x, \ldots, x^{k-1}$, correspond to the main-effects columns and clearly are a full 2^k factorial. Now consider any set of k successive terms in the sequence, beginning with x^t, say. The columns corresponding to these terms will be a full factorial unless there is a linear dependency among the corresponding elements of $\text{GF}(2^k)$, that is unless $\sum_{j=0}^{k-1} \varepsilon_j x^{t+j} \equiv 0$ for a set of ε_j which are not all equal to 0. Dividing the last relation by x^t, we would then have $\sum_{j=0}^{k-1} \varepsilon_j x^j \equiv 0$, stating that there is a linear dependency among the first k columns. This contradicts the observation that the first k columns in the ordering do provide a 2^k factorial. Thus the ordering satisfies the property that each set of k successive columns is a full 2^k factorial design. In particular, dividing the ordered columns into blocks of k leads to each such block being a full 2^k factorial design. Thus we obtain $B_k = \lfloor (n-1)/k \rfloor$ blocks. □

Theorem 2. *The Latin hypercube designs generated from the above algorithm are* U*-designs with respect to an orthogonal array with two symbols in each column. The signs of the columns in the rotated Latin hypercube design exactly match the signs of the columns in the two-level fractional factorial that was rotated.*

Proof. Each column in the rotated design is a linear combination of k columns in the two-level fractional factorial design that was rotated. The weights assigned to these k columns are $\pm 1, \pm 2, \ldots, \pm 2^{k-1}$. The signs of the entries in the rotated column will then correspond exactly to the signs of the column in the original design that received weight 2^{k-1}. The signs will match exactly if the weight is positive and will be multiplied by -1 if the weight is negative. □

In Table 3 we presented a 16-run, 8-factor design in which all the main effects were orthogonal to all second-order terms. This construction can also be easily extended to the general case where $n = 2^k$ and $k = 2^m$, to generate a design with 2^{k-1} factors. We take as our initial design the standard resolution IV fractional factorial design with 2^k runs and 2^{k-1} factors, including from the saturated design all columns that correspond to main effects and to interactions of an odd number of factors 1 to k, inclusive. We use the matrix R_m to rotate the factors in the initial design in groups of k. As the group size, $k = 2^m$, divides the total number of columns in the initial design, 2^{k-1}, all the columns can be rotated. To order the columns, we apply a variation of the procedure in Theorem 1. The columns of our initial design correspond to the vectors in GF(2^k) that have an odd number of ones and so are fully determined by their first $k-1$ entries. Moreover, the columns include all possible binary vectors of length $k-1$. Ignoring the kth coordinate, apply the procedure of Theorem 1 to the $2^{k-1}-1$ nonzero coefficient vectors, ordering them as elements in GF(2^{k-1}). The only remaining vector is $(0, \ldots, 0, 1)$, which corresponds to the main effect of factor k. Make this the first column in the ordering.

THEOREM 3. *The procedure described above generates an orthogonal Latin hypercube design with $n = 2^k$ runs, where $k = 2^m$, and 2^{k-1} factors, in which all main effects are orthogonal to all second-order terms.*

Proof. The standard two-level fractional factorial of resolution IV has fold-over structure: the reflection of each design point about the origin is also in the design. The fold-over property is not affected by reordering the columns. Bursztyn & Steinberg (2001) showed that rotation designs preserve fold-over structure. The orthogonality of the main effects to all second-order terms then follows from the results in Box & Wilson (1951).

We must also show that the ordering algorithm described above produces blocks of k columns, each of which is a full 2^k factorial design. In fact, as in Theorem 1, we show that each set of k successive columns is a 2^k design. The proof requires one additional fact from Galois field theory. The primitive polynomial $f(x)$ for GF(2^{k-1}) can be written $f(x) = x^{k-1} + \sum_{j=0}^{k-2} d_j x^j$, where each d_j is 0 or 1, and the number of d_j that equal 1 is even. The parity of the number of coefficients follows from the fact that a primitive polynomial cannot be divisible by any lower degree polynomial. Were the number of d_j that equal 1 odd, it would then hold that $f(1) \equiv 0 \bmod 2$ and $f(x)$ would be divisible by $x+1$.

Now consider k consecutive columns in our ordering. The first k columns are the main-effect columns and are a full factorial. Any other set of k columns is generated by successive powers of x in GF(2^{k-1}), $x^t, x^{t+1}, \ldots, x^{t+k-1}$, say. Denote the coefficient vectors of these columns in GF(2^k) by $c_t, \ldots, c_{t+k-1}$. The first $k-1$ elements of c_t give the point in GF(2^{k-1}) corresponding to x^t and the kth element is defined by the requirement that each vector have an odd number of nonzero entries. The set of k columns is a 2^k factorial unless it satisfies a linear dependency, $\sum_{j=0}^{k-1} \varepsilon_j c_{t+j} = 0$, where 0 here is a k-vector of zeros. By Theorem 1, there cannot be a linear dependency that is limited to $k-1$ consecutive columns, as that would give, in particular, a linear dependency in the first $k-1$ coordinates. We can use the structure of the ordering to find exactly the linear dependency among all k columns that holds for the first $k-1$ coordinates of the c_{t+j} and then will show that it does not carry over to the kth coordinate. We can write the linear dependency for the first $k-1$ coordinates only as $\sum_{j=0}^{k-1} \varepsilon_j x^{t+j} \equiv 0$. Dividing by x^t gives the equivalent condition $\sum_{j=0}^{k-1} \varepsilon_j x^j \equiv 0$. For $j = 0, \ldots, k-2$, $x^j \equiv x^j \bmod f(x)$ and $x^{k-1} \equiv \sum_{j=0}^{k-2} d_j x^j \bmod f(x)$. Thus the linear dependency for the first $k-1$ coordinates is given precisely by the primitive polynomial, i.e. by setting $\varepsilon_j = d_j$, for $j = 1, \ldots, k-2$, and $\varepsilon_{k-1} = 1$. The total number of nonzero coefficients in the primitive polynomial is odd and the vector c_i for each column in our design has an odd number of ones. Thus the sum of all the coordinates in the vector $\sum_{j=0}^{k-1} \varepsilon_j c_{t+j}$ must be an odd number. It follows that the vector cannot be equivalent to 0 mod 2. □

7. Discussion

We note that Ye's (1998) Latin hypercube designs cannot be obtained directly as rotations of two-level factorials. All points in a 2^k design, or any of its fractions, are equidistant from the origin. As rotations are isometric, the points in the rotated design must also be equidistant from the origin. The points in Ye's designs do not share this property. Subsets of columns in a rotated design will not have the equidistance property. We do not know if Ye's 16-run designs can be obtained by rotating a larger design, with 8 or 12 factors, say, and then selecting a subset of 6 columns. In the orthogonal Latin hypercube design in the technical report by Ye, all points are equidistant from the origin. However, we have not found a way to generate this design as a rotation of a 16-run two-level fractional factorial.

The construction method that we present in this paper leads to much larger orthogonal Latin hypercube designs than were previously known. The primary limitation to our method is the severe sample size constraint; we require the sample size to be $n = 2^k$, where k is also a power of 2, $k = 2^m$. Extension of the proposed algorithm for accommodating various run sizes is currently under study.

Acknowledgement

We are grateful to Noga Alon for suggesting the proof of Theorem 1 and to Ronit Steinberg for help in enumerating all possible divisions of the 15 columns in a saturated 16-run two-level factorial into three sets of 4 independent columns.

References

Beattie, S. D. & Lin, D. K. J. (1997). Rotated factorial design for computer experiments. In *Proc. Phys. Eng. Sci. Sect. Am. Statist. Assoc.* Washington DC: American Statistical Association.

Beattie, S. D. & Lin, D. K. J. (2004). Rotated factorial designs for computer experiments. *J. Chin. Statist. Assoc.* **42**, 289–308.

Beattie, S. D. & Lin, D. K. J. (2005). A new class of Latin hypercube for computer experiments. In *Contemporary Multivariate Analysis and Experimental Designs In Celebration of Professor Kai-Tai Fang's 65th Birthday*, Ed. J. Fan and G. Li, pp. 206–66. Singapore: World Scientific.

Box, G. E. P. & Wilson, K. B. (1951). On the experimental attainment of optimum conditions (with Discussion). *J. R. Statist. Soc.* B **13**, 1–34.

Bursztyn, D. & Steinberg, D. M. (2001). Rotation designs for experiments in high bias situations. *J. Statist. Plan. Infer.* **97**, 399–414.

Bursztyn, D. & Steinberg, D. M. (2002). Rotation designs: Orthogonal first-order designs with higher order projectivity. *J. Appl. Stoch. Models Bus. Indust.* **18**, 197–206.

Butler, N. A. (2001). Optimal and orthogonal Latin hypercube designs for computer experiments. *Biometrika* **88**, 847–57.

Hamada, M. & Wu, C. F. J. (1992). Analysis of designed experiments with complex aliasing. *J. Qual. Technol.* **24**, 130–7.

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

Helton, J. C. & Davis, F. J. (2000). Sampling-based methods. In *Sensitivity Analysis*, Ed. A. Saltelli, K. Chan and E. M. Scott, pp. 101–53. Chichester: John Wiley & Sons.

McKay, M. D., Beckman, R. J. & 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**, 239–45.

OWEN, A. B. (1992). Orthogonal arrays for computer experiments, integration and visualization. *Statist. Sinica* **2**, 439–52.

OWEN, A. B. (1994). Controlling correlations in Latin hypercube samples. *J. Am. Statist. Assoc.* **89**, 1517–22.

SANTNER, T. J., WILLIAMS, B. J. & NOTZ, W. I. (2003). *The Design and Analysis of Computer Experiments*. New York: Springer.

TANG, B. (1993). Orthogonal array-based Latin hypercubes. *J. Am. Statist. Assoc.* **88**, 1392–7.

TANG, B. (1998). Selecting Latin hypercubes using correlation criteria. *Statist. Sinica* **8**, 965–77.

YE, K. Q. (1998). Orthogonal column Latin hypercubes and their application in computer experiments. *J. Am. Statist. Assoc.* **93**, 1430–9.

[*Received April* 2005. *Revised October* 2005]