Sliced Latin Hypercube Designs

Peter Z. G. Qian
Department of Statistics
University of Wisconsin-Madison, Madison, WI 53706

(October 1, 2011)

KEY WORDS: Computer experiment; Design of experiments; Latin hypercube design; Space-filling design; Ensembles of computer models.

Abstract

We propose a method for constructing a new type of space-filling design, called a sliced Latin hypercube design, intended for running computer experiments. Such a design is a special Latin hypercube design that can be partitioned into slices of smaller Latin hypercube designs. It is desirable to use the constructed designs for collective evaluations of computer models and ensembles of multiple computer models. The proposed construction method is easy to implement, capable of accommodating any number of factors and flexible in run size. Examples are given to illustrate the method. Sampling properties of the constructed designs are examined. Numerical illustration is provided to corroborate the derived theoretical results.

1 INTRODUCTION

Experiments with deterministic computer models, such as finite element analysis and computational fluid dynamics codes, are becoming ubiquitous in sciences, engineering and services for studying complex phenomena. Different from physical experiments, running a deterministic computer model twice at the same input value yields two identical output values (Santer, Williams and Notz 2003; Fang, Li and Sudjianto 2005). A main goal in many computer experiments is to estimate the expected output of a computer model given a distribution of inputs. To address this issue, McKay, Conover and Beckman (1979) introduced Latin hypercube designs, referred to as *ordinary Latin hypercube designs* hereinafter. Throughout, drawing a uniform permutation on a set of p integers means randomly taking a permutation on the set, with all $p!$ possible permutations being equally probable. Let $\mathbf{A} = (a_{ik})$ be an $n \times q$ Latin hypercube in which each column is a uniform permutation on $\{1, \ldots, n\}$ and all the columns are obtained independently. An ordinary Latin hypercube design $\mathbf{D}_0 = (d_{ik})$ of n runs in q factors is generated through

$$d_{ik} = (a_{ik} - u_{ik})/n, \text{ for } i = 1, \ldots, n, \; k = 1, \ldots, q, \tag{1}$$

where the u_{ik} are independent $U[0, 1)$ random variables, d_{ik} is the level of factor k on the ith run, and the u_{ik} and the a_{ik} are mutually independent. When $\mathbf{D}_0$ is projected onto any one dimension, precisely one point falls within one of the n equal spaced intervals of $(0, 1]$ given by $(0, 1/n], (1/n, 2/n], \ldots, ((n-1)/n, 1]$. Sampling properties of such designs were studied by McKay et al. (1979), Stein (1987), Owen (1992) and Loh (1996).

In this article, we propose a method for constructing a new type of space-filling design, called a *sliced Latin hypercube design* (SLHD). An SLHD is a special Latin hypercube design that can be partitioned into slices of smaller Latin hypercube designs. Such a design has two attractive features. (1) Each slice of the design achieves maximum uniformity in any one-dimensional projection. (2) When collapsed over all the slices, the whole design possesses maximum stratification in any one-dimensional projection. Section 3 will show that these properties lead to variance reduction in numerical integration.

It is desirable to use an SLHD to run a computer model in batches, with each batch of input values being one slice of the design. On the one hand, if it is feasible to borrow strength across all the batches, the combined design set forms a Latin hypercube design. On the other hand, when data from different batches must be analyzed separately, the set of input values for each batch is guaranteed to be a smaller Latin hypercube design. An SLHD is also useful for running multiple computer models based on similar mathematics (Williams, Morris and Santner 2009), where each model uses one slice of the design.

The remainder of the article is organized as follows. Section 2 presents a method for constructing SLHDs. Section 3 derives sampling properties of such designs. Section 4 gives numerical illustration of the derived theoretical results. Section 5 provides some discussion. All proofs are deferred to the Appendix.

2 CONSTRUCTION

In this section, we present a method for constructing SLHDs. This method is easy to implement, capable of accommodating an arbitrary number of factors and flexible in run size. We consider design construction for q continuous factors, each taking values in $(0, 1]$. Here are some useful definitions. For $a \in \mathbb{R}$, $\lceil a \rceil$ denotes the smallest integer no less than a and $\lfloor a \rfloor$ denotes the largest integer no greater than a. Similarly define $\lceil \mathbf{D} \rceil$ and $\lfloor \mathbf{D} \rfloor$ for a real matrix $\mathbf{D}$. For a matrix $\mathbf{A}$, let $\mathbf{A}(:, j)$ be its jth column, $\mathbf{A}(i, :)$ be its ith row and $A(i, j)$ be its (i, j)th element. Throughout, let m and t be strictly positive integers with $n = mt$. For an integer $b \geq 1$, let $\mathbf{Z}_b$ denote the set $\{1, \ldots, b\}$, We define a permutation matrix $\mathrm{PM}(m, t)$ on $\mathbf{Z}_n$ to be an $m \times t$ matrix in which each element of $\mathbf{Z}_n$ appears precisely once. Suppose that $\mathbf{A}$ is a $\mathrm{PM}(m, t)$ and each column of $\lceil \mathbf{A}/t \rceil$ forms a permutation on $\mathbf{Z}_m$. Then we call $\mathbf{A}$ an m by t *sliced permutation matrix*, denoted by $\mathrm{SPM}(m, t)$.

As a stepping stone for the construction of SLHDs, we introduce an algorithm for generating sliced permutation matrices. First divide the elements of $\mathbf{Z}_n$ into m blocks, $\mathbf{b}_1, \ldots, \mathbf{b}_m$, where

$$\mathbf{b}_i = \{a \in \mathbf{Z}_n | \lceil a/t \rceil = i\}, \quad \text{for } \ i = 1, \ldots, m. \tag{2}$$

Then an $\mathrm{SPM}(m, t)$ is generated in two steps.

Step 1: For $i = 1, \ldots, m$, fill the ith row of an $m \times t$ empty matrix $\mathbf{H}$ with a uniform permutation on the set $\mathbf{b}_i$, with the permutations carried out independently from one row to another.

Step 2: For $j = 1, \ldots, t$, randomly shuffle the entries in the jth column of $\mathbf{H}$, with the permutations carried out independently from one column to another.

It is easy to verify that $\mathbf{H}$ generated by this algorithm is an $\mathrm{SPM}(m, t)$.

Example 1. Let $m = 5$ and $t = 4$ with $n = 20$. The elements of $\mathbf{Z}_{20}$ are divided into five blocks as follows: $\mathbf{b}_1 = \{1, 2, 3, 4\}$, $\mathbf{b}_2 = \{5, 6, 7, 8\}$, $\mathbf{b}_3 = \{9, 10, 11, 12\}$, $\mathbf{b}_4 = \{13, 14, 15, 16\}$ and $\mathbf{b}_5 = \{17, 18, 19, 20\}$. A possible outcome from Step 1 of the foregoing algorithm is

$$\mathbf{H} = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 6 & 5 & 8 & 7 \\ 10 & 11 & 9 & 12 \\ 14 & 15 & 16 & 13 \\ 19 & 20 & 17 & 18 \end{pmatrix},$$

which, in Step 2 of the algorithm, may change to

$$\mathbf{H} = \begin{pmatrix} 2 & 5 & 3 & 13 \\ 6 & 1 & 16 & 18 \\ 14 & 15 & 17 & 12 \\ 19 & 20 & 9 & 4 \\ 10 & 11 & 8 & 7 \end{pmatrix},$$

where each column of $\lceil \mathbf{H}/4 \rceil$ is a permutation on $\mathbf{Z}_5$.

We now discuss how to use multiple sliced permutation matrices to obtain an SLHD. First, generate q independent SPM(m, t)s, $\mathbf{H}_1, \ldots, \mathbf{H}_q$. For $c = 1, \ldots, t$, obtain an $m \times q$ matrix $\mathbf{A}^{(c)}$ by letting its jth column be the cth column of $\mathbf{H}_j$, for $j = 1, \ldots, q$. Obtain a matrix $\mathbf{A}$ by combining $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(t)}$, row by row, given as

$$\mathbf{A} = \cup_{c=1}^{t} \mathbf{A}^{(c)}. \tag{3}$$

Note that $\mathbf{A}$ is a special Latin hypercube in which each column is a permutation on $\mathbf{Z}_n$, and, for $c = 1, \ldots, t$, $\lceil \mathbf{A}^{(c)}/t \rceil$ is a smaller Latin hypercube of m runs with each column being a permutation on $\mathbf{Z}_m$. We call $\mathbf{A}$ an n by q *sliced Latin hypercube* with t slices. Using

$\mathbf{A} = (a_{ik})$, an $n \times q$ design $\mathbf{D} = (d_{ik})$ is generated through

$$d_{ik} = (a_{ik} - u_{ik})/n, \text{ for}; i = 1, \ldots, n, \ k = 1, \ldots, q, \tag{4}$$

where the u_{ik} are independent $U[0, 1)$ random variables, d_{ik} is the level of factor k on the ith run, and the u_{ik} and the a_{ik} are mutually independent. For $c = 1, \ldots, t$, let $\mathbf{D}^{(c)}$ denote the subset of points in $\mathbf{D}$ corresponding to $\mathbf{A}^{(c)}$. The array $\mathbf{D}$ is an SLHD, as described by the following proposition.

Proposition 1. *Let m and t be strictly positive integers with $n = mt$. Consider $\mathbf{D}$ and the $\mathbf{D}^{(c)}$s constructed above. Then we have that*

(i) the design $\mathbf{D}$ is a Latin hypercube design with n levels;

(ii) the $\mathbf{D}^{(c)}$s form a partition of $\mathbf{D}$ and each of them is a Latin hypercube design with m levels.

This proposition indicates that when $\mathbf{D}$ is projected onto each of the q factors, precisely one point falls within each of the n intervals defined by $(0, 1/n], (1/n, 2/n], \ldots, ((n-1)/n, 1]$, and exactly one of the m design points of each $\mathbf{D}^{(c)}$ falls within each of the m intervals defined by $(0, 1/m], (1/m, 2/m], \ldots, ((m-1)/m, 1]$.

Example 2. Let $m = 3$, $t = 3$ and $q = 3$ with $n = 9$. A 9×3 sliced Latin hypercube $\mathbf{A}$ with three slices constructed by (3) is given by (in transpose)

$$\left(\begin{array}{ccc|ccc|ccc} 8 & 5 & 3 & 7 & 1 & 6 & 4 & 2 & 9 \\ 4 & 8 & 3 & 2 & 5 & 9 & 6 & 7 & 1 \\ 5 & 9 & 1 & 2 & 6 & 8 & 3 & 7 & 4 \end{array}\right),$$

where the solid lines divide $\mathbf{A}$ into three slices, $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$. For $c = 1, \ldots, 3$, $\lceil \mathbf{A}^{(c)}/3 \rceil$ is a Latin hypercube with three levels. Using $\mathbf{A}$, Proposition 1 gives an SLHD $\mathbf{D}$ of nine runs in three factors with three slices $\mathbf{D}^{(1)}, \mathbf{D}^{(2)}$ and $\mathbf{D}^{(3)}$. Figure 1 depicts the bivariate projections of $\mathbf{D}$, where, in one dimension, each of the nine equally spaced intervals of $(0, 1]$

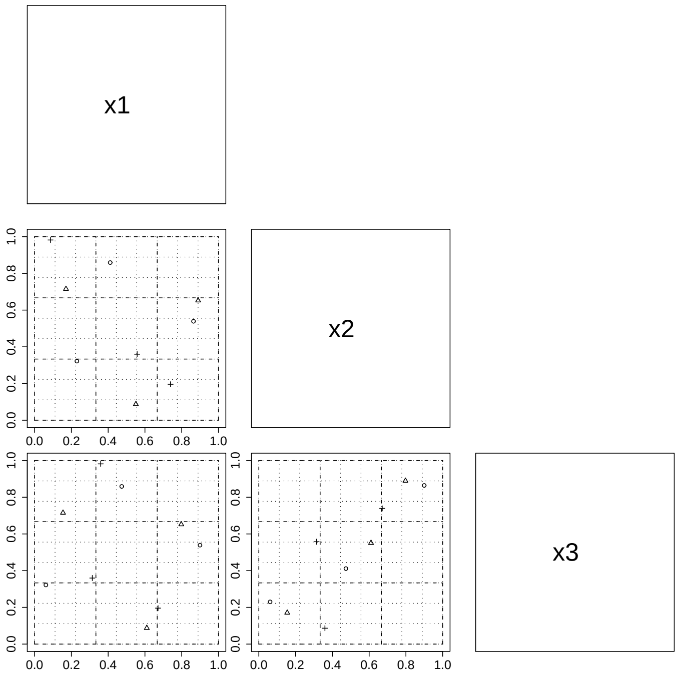

Figure 1: Bivariate projections of an SLHD $\mathbf{D}$ with three slices $\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \mathbf{D}^{(3)}$ ($\circ, +, \triangle$) in Example 2.

contains precisely one point and, for each of the three slices of $\mathbf{D}$, denoted by $\circ, +, \triangle$, each of the three equally spaced intervals of $(0, 1]$ contains exactly one point in every one dimension.

3 SAMPLING PROPERTIES

In this section, we derive sampling properties of SLHDs in the context of the two motivating problems described in Section 1. The first problem can be viewed as a special ensemble experiment using multiple computer models based on the same mathematics. As a result of this connection, we focus on the second problem and remark on how to simplify its results for the first problem.

Consider an ensemble experiment using t similar computer models, $f^{(1)}, \ldots, f^{(t)}$. Assume each $f^{(c)}$ has inputs $\mathbf{x} = (x_1, \ldots, x_q)$ whose distribution is the uniform measure on $(0, 1]^q$, denoted by F. For $c = 1, \ldots, t$, define $\mu_c = E[f^{(c)}(\mathbf{x})]$. For $c_1, c_2 = 1, \ldots, t$, define $\mathrm{cov}_{c_1 c_2} = \mathrm{cov}[f^{(c_1)}(\mathbf{x}), f^{(c_2)}(\mathbf{x})]$, which becomes $\sigma^2_{c_1} = \mathrm{var}[f^{(c_1)}(\mathbf{x})]$ if $c_1 = c_2$. The goal here is to run

$f^{(1)}, \ldots, f^{(t)}$ each at m selected input values in order to estimate $\mu_1, \ldots, \mu_t$. For $0 \leq \lambda_c \leq 1$, $c = 1, \ldots, t$, a linear combination of $\mu_1, \ldots, \mu_t$ given by

$$\eta = \sum_{c=1}^{t} \lambda_c \mu_c \tag{5}$$

can also be of interest in practice.

We consider three different schemes defined below to achieve this goal.

Definition 1. *Suppose that m and t are strictly positive integers with $n = mt$.*

(i) Let IID denote a scheme that takes an independent and identically distributed sample of m runs for each $f^{(c)}$, with the t samples generated independently.

(ii) Let LH denote a scheme that obtains t independent ordinary Latin hypercube designs of m runs from (1), each of which is associated with one $f^{(c)}$.

(iii) Let SLH denote a scheme that produces an $n \times q$ SLHD with t slices by using the method in Section 2, where each slice, assigned to one $f^{(c)}$, is a smaller Latin hypercube design of m levels.

Expectations, variances and covariances under these schemes (in the same order) are denoted by the subscripts IID, LH and SLH, respectively. The SLH scheme is referred to as *sliced Latin hypercube sampling.* For any of these schemes, let $\mathbf{D}^{(c)}$ denote the design set for $f^{(c)}$, $c = 1, \ldots, t$. Denote by $\mathbf{d}_i^c$ the ith row of $\mathbf{D}^{(c)}$ and d_{ik}^c the kth entry of $\mathbf{d}_i^c$. Obtain a design $\mathbf{D}$ by combining $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(t)}$ row by row. For $c = 1, \ldots, t$, μ_c is estimated by

$$\hat{\mu}_c = m^{-1} \sum_{i=1}^{m} f^{(c)}(\mathbf{d}_i^{(c)}) \tag{6}$$

and η by

$$\hat{\eta} = \sum_{c=1}^{t} \lambda_c \hat{\mu}_c. \tag{7}$$

Remark 1. *For a collective experiment that evaluates a computer model f with t batches of input values, we have that $f^{(c)} = f$ and $\lambda_c = t^{-1}$, for $c = 1, \ldots, t$, and $\eta = E[f(\mathbf{x})]$.*

For later development, we describe the ANOVA decomposition of integrable functions on $(0,1]^q$ (Owen 1992; Loh 1996). Recall that F denotes the uniform distribution on $(0,1]^q$ for $\mathbf{x} = (x_1, \ldots, x_q)$. Let $dF = \prod_{k=1}^{q} dF_k$ and $dF_{-k} = \prod_{l \neq k} dF_l$. If $f : R^q \to R$ is a measurable function of $\mathbf{x}$ and $E\{[f(\mathbf{x})]^2\}$ is well defined and finite, then f can be decomposed as

$$f(\mathbf{x}) = \mu + \sum_{k=1}^{q} f_{-k}(x_k) + r(\mathbf{x}), \tag{8}$$

where $\mu = \int f(\mathbf{x}) dF$ is the grand mean and the functional main effect of x_k is

$$f_{-k}(x_k) = \int [f(\mathbf{x}) - \mu] dF_{-k}, \text{ for } k = 1, \ldots, q. \tag{9}$$

For $k = 1, \ldots, q$, note that $\int f_{-k} dF_k = 0$ and

$$\int r(\mathbf{x}) dF_{-k} = 0. \tag{10}$$

Lemma 1 presents some joint probability mass functions of sliced permutation matrices constructed in Section 2.

Lemma 1. *For strictly positive integers m and t with $n = mt$, let $\mathbf{H}$ be an* $\mathrm{SPM(m,t)}$ *on $\mathbf{Z}_n$ generated by using the algorithm in Section 2. Denote by h_{ij} the (i,j)th entry of $\mathbf{H}$. Consider $u, v \in \mathbf{Z}_n$. Then we have that*

(i) for $i = 1 \ldots, m$ and $j = 1 \ldots, t$, the probability mass function for h_{ij} is

$$\Pr(h_{ij} = u) = 1/n;$$

(ii) for $i_1, i_2 = 1, \ldots, m$, $i_1 \neq i_2$ and $j = 1, \ldots, t$, the joint probability mass function for $h_{i_1 j}$ and $h_{i_2 j}$ is

$$\Pr(h_{i_1 j} = u, h_{i_2 j} = v) = \begin{cases} [n(n-t)]^{-1}, & \lceil u/t \rceil \neq \lceil v/t \rceil \\ 0, & o.w.; \end{cases}$$

(iii) for $i_1, i_2 = 1, \ldots, m$ and $j_1, j_2 = 1, \ldots, t, j_1 \neq j_2$, the joint probability mass function for $h_{i_1 j_1}$ and $h_{i_2 j_2}$ is

$$\Pr(h_{i_1j_1} = u, h_{i_2j_2} = v) = \begin{cases} n^{-2}, & \lceil u/t \rceil \neq \lceil v/t \rceil, \\ n^{-1}(n-m)^{-1}, & \lceil u/t \rceil = \lceil v/t \rceil \text{ and } u \neq v, \\ 0 & o.w. \end{cases}$$

These probability mass functions are more complicated than those of a uniform permutation on $\mathbf{Z}_n$ associated with an ordinary Latin hypercube design in (1).

Now, let $\mathbf{D}$ be an $n \times q$ SLHD with t slices, $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(t)}$, constructed by Proposition 1 in Section 2, where each slice is a Latin hypercube design of m runs. Lemma 2 derives the distribution of each $\mathbf{D}^{(c)}$.

Lemma 2. *For strictly positive integers m and t with $n = mt$, consider $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(t)}$ constructed above. Then we have that each $\mathbf{D}^{(c)}$ is statistically equivalent to an $m \times q$ ordinary Latin hypercube design as constructed in (1).*

Next, we present results on $\hat{\mu}_c$ in (6) and $\hat{\eta}$ in (7) under sliced Latin hypercube sampling. Lemma 1 implies that $\hat{\mu}_c$ and $\hat{\eta}$ under this scheme are unbiased estimators for μ_c and η, respectively. These estimators under the other two schemes in Definition 1 are unbiased as well. We now move on to derive variance formulas for sliced Latin hypercube sampling. The following theorem provides a non-asymptotic result under some monotonicity assumptions of $f^{(1)}, \ldots, f^{(t)}$.

Theorem 1. *Suppose that, for $c = 1, \ldots, t$, $f^{(c)}(\mathbf{x})$ is monotonic in each argument x_i of $\mathbf{x} = (x_1, \ldots, x_q)$ and any pair of functions $f^{(c_1)}$ and $f^{(c_2)}$, $c_1 \neq c_2$, are either both increasing or both decreasing in each argument x_i of $\mathbf{x}$. Consider the three schemes described by Definition 1. Then we have that*

(i) for $c = 1, \ldots, t$ and $\hat{\mu}$ defined in (6),

$$\mathrm{var}_{\mathrm{SLH}}(\hat{\mu}_c) \leq \mathrm{var}_{\mathrm{IID}}(\hat{\mu}_c);$$

(ii) for $\hat{\eta}$ defined in (7),

$$\mathrm{var}_{\mathrm{SLH}}(\hat{\eta}) \leq \mathrm{var}_{\mathrm{LH}}(\hat{\eta}) \leq \mathrm{var}_{\mathrm{IID}}(\hat{\eta}). \tag{11}$$

This result indicates that an SLHD not only possesses an attractive slicing structure but also achieves efficient variance reduction in a fashion similar to an ordinary Latin hypercube design (McKay et al. 1979).

By dropping the monotonicity assumptions in Theorem 1, below we give a more general result for $\hat{\mu}_c$ and $\hat{\eta}$ under sliced Latin hypercube sampling.

Theorem 2. *Suppose that $E\{[f^{(1)}(\mathbf{x})]^2\}, \ldots, E\{[f^{(t)}(\mathbf{x})]^2\}$ are all well defined and finite. For $c = 1, \ldots, t$, let $f_{-k}^{(c)}$ be the functional main effect for the input x_k of $\mathbf{x} = (x_1, \ldots, x_q)$ in the ANOVA decomposition of $f^{(c)}$ in (9). Let m and t be strictly positive integers with $n = mt$ and let the SLH scheme be as defined in Definition 1. Then as $n \to \infty$ with t fixed, we have that*

(i) for $c = 1, \ldots, t$ and $\hat{\mu}_c$ defined in (6) based on the slice $\mathbf{D}^{(c)}$,

$$\mathrm{var}_{\mathrm{SLH}}(\hat{\mu}_c) = \sigma_c^2 \frac{t}{n} - \frac{t}{n} \sum_{k=1}^{q} \int_0^1 [f_{-k}^{(c)}(x_k)]^2 dx_k + o(m^{-1});$$

(ii) for $\hat{\eta}$ defined in (7) associated with all slices $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(t)}$,

$$\mathrm{var}_{\mathrm{SLH}}(\hat{\eta}) = \frac{t}{n} \sum_{c=1}^{t} \lambda_c^2 \sigma_c^2 - \frac{t}{n} \sum_{c=1}^{t} \left\{ \lambda_c^2 \sum_{k=1}^{q} \int_0^1 [f_{-k}^{(c)}(x_k)]^2 dx_k \right\} + o(n^{-1}).$$

Some observations based on Theorem 2 are worth mentioning. First, in Theorem 2 (i) and (ii), the main effects of $f_{-k}^{(c)}(x_k)$ are filtered out, thus achieving the degree of variance reduction similar to an ordinary Latin hypercube design. Second, for a collective experiment evaluating a computer model f in t batches defined in Remark 1 with $\lambda_c = t^{-1}$ and $f^{(c)} = f$, we have that $\sigma_c^2 = \sigma^2 = \mathrm{var}[f(\mathbf{x})]$ and Theorem 2 (ii) becomes

$$\mathrm{var}_{\mathrm{SLH}}(\hat{\eta}) = \frac{1}{n}\sigma^2 - \frac{1}{n} \sum_{k=1}^{q} \int_0^1 [f_{-k}(x_k)]^2 dx_k + o(n^{-1}),$$

which appears similar to that of an ordinary Latin hypercube design of n runs derived in Stein (1987) and Loh (1996).

4 Numerical Illustration

In this section, we provide numerical illustration to corroborate some theoretical results derived in the previous section.

Example 3. This example uses the five-dimensional function (Drew and Homem-de Mello 2005)

$$f(\mathbf{x}) = \log(x_1 x_2 x_3 x_4 x_5) \tag{12}$$

to act as a computer model. Consider using four machines to estimate $\eta = E[f(\mathbf{x})]$, where the distribution of $\mathbf{x}$ is the uniform measure on $(0, 1]^5$ and $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(4)}$ denote the sets of input values for the four machines, respectively. We estimate η by $\hat{\eta}$ based on $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(4)}$, defined in (7), with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$, and estimate μ_1 by $\hat{\mu}_1$ based on $\mathbf{D}^{(1)}$ for machine 1, defined in (6). Compare four different methods to produce $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(4)}$. The first method generates four independent and identically distributed samples of m runs. The second method randomly splits an ordinary Latin hypercube design of $4m$ runs into four subdesigns of equal size, corresponding to $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(4)}$, respectively. The third method combines four ordinary Latin hypercube designs of m runs. The fourth method generates an SLHD of $4m$ runs, where each of the four slices is a smaller Latin hypercube design of m runs. These methods are denoted by IID, SPLIT, COMBINE and SLHD, respectively,

For each method, we computed $\hat{\eta}$ and $\hat{\mu}_1$ 2000 times for $m = 5, 10, 20, 40$. Table 1 presents the RMSE (root mean square error) of $\hat{\eta}$ over the 2000 replicates for every method. The RMSEs for $\hat{\mu}_1$ are given in Table 2. These tables clearly indicate that for every value of m, the SLHD method, corresponding to column 4, not only achieves the same degree of variance reduction as the SPLIT method for $\hat{\eta}$ but also achieves the same degree of variance reduction as the COMBINE method for $\hat{\mu}_1$.

Table 3 compares the RMSEs of $\hat{\eta}$ over 2000 replicates for the four methods with $\lambda_1 = 1/2$

and $\lambda_2 = \lambda_3 = \lambda_4 = 1/6$, where the SLHD method performs the best.

	IID	SPLIT	COMBINE	SLHD
$m = 5$	0.5040	0.1225	0.2272	0.1127
$m = 10$	0.3446	0.0601	0.1135	0.0593
$m = 20$	0.2524	0.0293	0.0576	0.0294
$m = 40$	0.1731	0.0146	0.0280	0.0143

Table 1: RMSEs of $\hat{\eta}$ of the Four Methods for Example 3 with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$

	IID	SPLIT	COMBINE	SLHD
$m = 5$	1.0011	0.8821	0.4581	0.4685
$m = 10$	0.7006	0.6151	0.2198	0.2321
$m = 20$	0.4983	0.4216	0.1130	0.1165
$m = 40$	0.3520	0.3163	0.0574	0.0573

Table 2: RMSEs of $\hat{\mu}_1$ of the Four Methods for Example 3

	IID	SPLIT	COMBINE	SLHD
$m = 5$	0.5721	0.3265	0.2640	0.1914
$m = 10$	0.4172	0.2090	0.1357	0.0965
$m = 20$	0.2861	0.1518	0.0662	0.0468
$m = 40$	0.2070	0.1030	0.0334	0.0239

Table 3: RMSEs of $\hat{\eta}$ of the Four Methods for Example 3 with $\lambda_1 = 1/2$ and $\lambda_2 = \lambda_3 = \lambda_4 = 1/6$

Example 4. This example uses the following functions

$$
\begin{aligned}
f_1(\mathbf{x}) &= \log\left(\frac{1}{\sqrt{x_1}} + \frac{1}{\sqrt{x_2}}\right) \\
f_2(\mathbf{x}) &= \log\left(\frac{0.98}{\sqrt{x_1}} + \frac{0.95}{\sqrt{x_2}}\right) \\
f_3(\mathbf{x}) &= \log\left(\frac{1.02}{\sqrt{x_1}} + \frac{1.02}{\sqrt{x_2}}\right) \\
f_4(\mathbf{x}) &= \log\left(\frac{1}{\sqrt{x_1}} + \frac{1.03}{\sqrt{x_2}}\right)
\end{aligned}
$$

to act as four similar computer models, where the distribution of $\mathbf{x}$ is the uniform measure on $(0, 1]^2$ and f_1 was taken from Drew and Homem-de Mello (2005). For each of the four methods in Example 3, we computed $\hat{\eta}$ with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$ and $\hat{\mu}_1$ 2000 times for $m = 5, 10, 20, 40$. Tables 4 and 5 present the RMSEs of $\hat{\eta}$ and $\hat{\mu}_1$ over the 2000 replicates. Once more, for every value of m, the SLHD method simultaneously achieves the same variance reduction as the SPLIT method for $\hat{\eta}$ and the same variance reduction as the COMBINE method for $\hat{\mu}_1$.

	IID	SPLIT	COMBINE	SLHD
$m = 5$	0.0944	0.0335	0.0581	0.0336
$m = 10$	0.0685	0.0181	0.0301	0.0186
$m = 20$	0.0501	0.0103	0.0165	0.0103
$m = 40$	0.0356	0.0059	0.0093	0.0059

Table 4: RMSEs of $\hat{\eta}$ of the Four Methods for Example 4

	IID	SPLIT	COMBINE	SLHD
$m = 5$	0.1936	0.1800	0.1069	0.1121
$m = 10$	0.1389	0.1230	0.0633	0.0615
$m = 20$	0.1004	0.0869	0.0345	0.0332
$m = 40$	0.0686	0.0617	0.0186	0.0177

Table 5: RMSEs of $\hat{\mu}_1$ of the Four Methods for Example 4

5 DISCUSSION

We have proposed a new type of design, called a sliced Latin hypercube design, intended for computer experiments. In addition to the motivating problems described in Section 1, other potential applications of the proposed designs include computer models with qualitative and quantitative factors (Qian, Wu and Wu 2008; Han et al. 2009; Qian and Wu 2009), cross-validation and stochastic optimization. MATLAB and R programs for constructing SLHDs are available from the author.

Existing variants of ordinary Latin hypercube designs (McKay et al. 1979) include orthogonal Latin hypercube designs with (nearly) orthogonal columns (Ye 1998; Steinberg and Lin 2006; Bingham, Sitter and Tang 2009; Lin, Mukerjee and Tang 2009; Sun, Liu and Lin 2009) and orthogonal array-based Latin hypercube designs (Tang 1993) with stratification in more than one dimensions. Compared with these designs, a unique feature of SLHDs is that such a design can be divided into slices, each of which is guaranteed to be a smaller Latin hypercube design. The proposed designs are easy to construct and exist when the sample size of the whole design is a multiple of that of each slice. By using more sophisticated methods, sliced designs with better stratification but more restrictive run size can be generated by exploiting *slicing* in orthogonal arrays (Qian and Wu 2009) and scrambled nets (Owen 1995; Keller 2010).

Section 2 uses sliced random permutations to generate SLHDs. In a subsequent project, we will develop methods for constructing optimal SLHDs based on the correlation (Ye 1998) and distance criteria (Johnson, Moore and Ylvisaker 1990; Morris and Mitchell 1995).

ACKNOWLEDGEMENTS

The author wishes to thank the Editor, the Associate Editor and three referees for their helpful comments that have led to improvements in the article. He thanks Youngdeok Hwang for assistance in computing and Xu He for useful comments. This work is supported by an IBM Faculty Award and an NSF Faculty Early Career Development Award DMS 1055214.

Appendix

Proof of Lemma 1

Proof. Consider $u, v \in \mathbf{Z}_n$. Let a_{ij} be the (i, j)th entry of $\mathbf{H}$ after Step 1 of the algorithm in Section 2 is carried out. For $j = 1, \ldots, t$, let π_j be the permutation used for manipulating the entries of the jth column of $\mathbf{H}$ in Step 2 of the algorithm, which is a vector-valued function from $\mathbf{Z}_m$ to $\mathbf{Z}_m$. Let h_{ij} denote the (i,j)th entry of $\mathbf{H}$ after Step 2 of the algorithm is completed. Clearly, the rows of $\mathbf{H}$ are exchangeable and so are the columns.

To prove (i), it suffices to consider h_{11} because of the exchangeability of $\mathbf{H}$. By symmetry, $\Pr(h_{11} = u)$ takes the same value for all $u \in \mathbf{Z}_n$, which is n^{-1} as $\mathbf{Z}_n$ has n elements. Thus, the result in (i) holds.

To prove (ii), it suffices to consider h_{11} and h_{21} because of the exchangeability of $\mathbf{H}$. Clearly, for u, v in the same block, i.e., $\lceil u/t \rceil = \lceil v/t \rceil$, $\Pr(h_{11} = u, h_{21} = v) = 0$. Because there are $(m^2 - m)t^2$ pairs of (u,v) satisfying $\lceil u/t \rceil \neq \lceil v/t \rceil$, by symmetry, $\Pr(h_{11} = u, h_{21} = v) = [n(n-t)]^{-1}$ for any such (u,v). Thus, the conclusion in (ii) holds.

To prove (iii), it suffices to consider h_{11} and h_{12} because of the exchangeability of $\mathbf{H}$. Divide all (u,v), $u \neq v \in \mathbf{Z}_n$, into two sets, B_1 and B_2, where B_1 consists of $m(m-1)t^2$ pairs of (u,v) with $\lceil u/t \rceil \neq \lceil v/t \rceil$ and B_2 consists of $mt(t-1)$ pairs of (u,v) with $\lceil u/t \rceil = \lceil v/t \rceil$ and $u \neq v$. Now compute $\Pr(h_{11} = 1, h_{12} = t+1)$. Note that $1 \in \mathbf{b}_1$ and $t+1 \in \mathbf{b}_2$, where $\mathbf{b}_1$ and $\mathbf{b}_2$ are defined in (2). By the construction of $\mathbf{H}$, we have that

$$\begin{aligned} \{h_{11} = 1\} &= \{a_{11} = 1\} \cap \{\pi_1(1) = 1\}, \\ \{h_{12} = t+1\} &= \{a_{22} = t+1\} \cap \{\pi_2(2) = 1\}. \end{aligned} \tag{A1}$$

Because the permutations in Step 1 of the construction of $\mathbf{H}$ are taken independently from one row to another, $\{a_{11} = 1\}$ and $\{a_{22} = t+1\}$ are independent. By the independence of π_1 and π_2, $\{\pi_1(1) = 1\}$ and $\{\pi_2(2) = 1\}$ are independent. Combining these two results implies that the four events $\{a_{11} = 1\}$, $\{\pi_1(1) = 1\}$, $\{a_{22} = t+1\}$ and $\{\pi_2(2) = 1\}$ in (A1) are all independent. Therefore, $\Pr(h_{11} = 1, h_{12} = t+1) = \Pr(h_{11} = 1)\Pr(h_{12} = t+1)$, which, by the result in (i), equals n^{-2}. Since $(1, t+1) \in B_1$, by symmetry of the pairs in B_1,

$$\Pr(h_{11} = u, h_{12} = v) = \Pr(h_{11} = 1, h_{12} = t+1) = n^{-2}$$

for all $(u,v) \in B_1$. Then by symmetry of the pairs of B_2, for all $(u,v) \in B_2$, $\Pr(h_{11} =$

$u, h_{12} = v)$ takes the same value, which, by the definitions of B_1 and B_2, equals

$$[1 - m(m-1)t^2 n^{-2}]/[mt(t-1)] = [n(n-m)]^{-1}.$$

$\square$

Proof of Lemma 2

Proof. For $c = 1, \ldots, t$, express the (i,k)th entry $d_{ik} = d_{ik}^{(c)}$ of $\mathbf{D}^{(c)}$ as

$$d_{ik} = \frac{b_{ik}t - e_{ik} - u_{ik}}{n}, \text{ for } i = 1, \ldots, m, k = 1, \ldots q, \tag{A2}$$

Here, $b_{1k}, \ldots, b_{mk}$ constitute a uniform permutation on $\mathbf{Z}_m$; each e_{ik} is a discrete random variable with probability mass function

$$\Pr(e_{ik} = a) = t^{-1}, \text{ for } a = 0, 1, \ldots, t-1;$$

the u_{ik} are independent $U[0,1)$ random variables; and the b_{ik}, the e_{ik} and the u_{ik} are mutually independent.

Letting $w_{ik} = e_{ik} + u_{ik}$, d_{ik} in (A2) becomes

$$\frac{b_{ik}}{m} - \frac{w_{ik}}{n}. \tag{A3}$$

Since $b_{1k}, \ldots, b_{mk}$ form a uniform permutation on $\mathbf{Z}_m$ and the b_{ik} and the w_{ik} are mutually independent, it remains to verify that w_{ik}/n is a $U[0, \frac{1}{m})$ random variable, which is shown as follows. For $x \in [0, 1/m)$, let $x_0 = \lfloor nx \rfloor$ and note that

$$\begin{aligned}\Pr\left(\frac{w_{ik}}{n} \le x\right) &= \frac{1}{t}\sum_{a=0}^{t-1}\Pr\left(\frac{a+u_{ik}}{n} \le x\right), \\ &= \frac{1}{t}\left[\sum_{a=0}^{x_0-1}\Pr\left(\frac{a+u_{ik}}{n} \le x\right) + \Pr\left(\frac{x_0+u_{ik}}{n} \le x\right) + \sum_{a=x_0+1}^{t-1}\Pr\left(\frac{a+u_{ik}}{n} \le x\right)\right].\end{aligned} \tag{A4}$$

Note that $\Pr\left(\frac{a+u_{ik}}{n} \le x\right) = 1$, for $a = 0, 1, \ldots, x_0 - 1$; $\Pr\left(\frac{x_0+u_{ik}}{n} \le x\right) = nx - x_0$; and $\Pr\left(\frac{a+u_{ik}}{n} \le x\right) = 0$, for $a = x_0 + 1, \ldots, t - 1$, which simplifies (A4) to mx. Thus, $\frac{w_{ik}}{n}$ is a $U[0, \frac{1}{m})$ random variable, which completes the proof. □

Proof of Theorem 1

Proof. (i) follows readily from Lemma 2 and the theorem in McKay et al. (1979).

We now show (ii). Under the LH scheme in Definition 1, $\hat{\mu}^{(1)}, \ldots, \hat{\mu}^{(t)}$ are based on t independent ordinary Latin hypercube designs of m runs and by the theorem in McKay et al. (1979), $\mathrm{var}_{\mathrm{LH}}(\hat{\mu}_c) \le \mathrm{var}_{\mathrm{IID}}(\hat{\mu}_c)$, for $c = 1, \ldots, t$. Thus, the second inequality in (11) holds. To establish the first inequality in (11), observe that

$$\begin{aligned}\mathrm{var}_{\mathrm{SLH}}(\hat{\eta}) &= \sum_{c=1}^{t}\lambda_c^2\mathrm{var}_{\mathrm{SLH}}(\hat{\mu}_c) + \sum_{c_1=1}^{t}\sum_{c_2=1, c_2\neq c_1}^{t}\lambda_{c_1}\lambda_{c_2}\mathrm{cov}_{\mathrm{SLH}}(\hat{\mu}_{c_1}, \hat{\mu}_{c_2}), \\ &= \mathrm{var}_{\mathrm{LH}}(\hat{\eta}) + \sum_{c_1=1}^{t}\sum_{c_2=1, c_2\neq c_1}^{t}\lambda_{c_1}\lambda_{c_2}\mathrm{cov}_{\mathrm{SLH}}[f^{(c_1)}(\mathbf{d}_1^{(c_1)}), f^{(c_2)}(\mathbf{d}_1^{(c_2)})],\end{aligned} \tag{A5}$$

where the second equality follows from the row exchangeability of each slice $\mathbf{D}^{(c)}$ of $\mathbf{D}$ under the SLH scheme. Lemma 2 and the proof of Theorem 1 in Qian (2009) on nested Latin hypercube designs imply that, for $k = 1, \ldots, q$, $d_{1k}^{(c_1)}$ and $d_{2k}^{(c_2)}$, $c_1 \neq c_2$, are *negatively quadrant dependent* (Lehmann, 1966). That is, for $0 \le u, v \le 1$ and $c_1 \neq c_2$,

$$\Pr(d_{1k}^{(c_1)} \le u, d_{1k}^{(c_2)} \le v) \le \Pr(d_{1k}^{(c_1)} \le u)\Pr(d_{2k}^{(c_2)} \le v). \tag{A6}$$

By Theorem 1 of Lehmann (1966), this result and the assumption on $f^{(c_1)}$ and $f^{(c_2)}$ together yield that

$$\mathrm{cov}_{\mathrm{SLH}}[f^{(c_1)}(\mathbf{d}_1^{(c_1)}), f^{(c_2)}(\mathbf{d}_1^{(c_2)})] \leq 0. \tag{A7}$$

Since λ_{c_1} and λ_{c_2} are non-negative, plugging (A7) into (A5) proves the first inequality in (11). This completes the proof. □

Proof of Theorem 2

Proof. For fixed t, the condition $n \to \infty$ implies that $m = nt^{-1} \to \infty$. By Lemma 2, each $\mathbf{D}^{(c)}$ is an ordinary Latin hypercube design with m levels. Then by Theorem 1 in Stein (1987) or Theorem 1 in Loh (1996), the result in (i) holds.

As $n \to \infty$ with fixed t, Lemma 2 and the proof of Lemma 1 in Qian (2009) for nested Latin hypercube designs give that for $c = 1, \ldots, t$ and $j_1, j_2 = 1, \ldots, m$, $j_1 \neq j_2$,

$$\mathrm{cov}[f^{(c)}(\mathbf{d}_{j_1}^{(c)}), f^{(c)}(\mathbf{d}_{j_2}^{(c)})] = -\frac{t}{n}\sum_{k=1}^{q}\int_0^1 [f_{-k}^{(c)}(x_k)]^2 dx_k + o(n^{-1}); \tag{A8}$$

and, for $c_1, c_2 = 1, \ldots, t$, $c_1 \neq c_2$, and $j_1, j_2 = 1, \ldots, m$,

$$\mathrm{cov}[f^{(c_1)}(\mathbf{d}_{j_1}^{(c_1)}), f^{(c_2)}(\mathbf{d}_{j_2}^{(c_2)})] = o(n^{-1}). \tag{A9}$$

Now (ii) follows by using an exchangeability argument along with (A8) and (A9). □

References

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

[2] Drew, S. and T. Homem-de Mello (2005), "Some Large Deviation Results for Latin Hypercube Sampling,"In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines (Eds.), Proceedings of the 2005 Winter Simulation Conference, pp. 673-681.

[3] Fang, K. F., Li, R. Z. and Sudjianto, A. (2005), *Design and Modeling for Computer Experiments*, New York: Chapman & Hall/CRC Press.

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

[5] Keller, A. (2010), "Partitioning Low Discrepancy Sequences," Presenation, the 9th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Warsaw, August, 2010.

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

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

[8] McKay, M. D., Conover, W. J. 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.

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

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

[11] Owen, A. B. (1992), "A Central Limit Theory for Latin Hypercube Sampling," *Journal of the Royal Statistical Society Series B*, **54**, 541–551.

[12] Owen, A. B. (1995), "Randomly Permuted (t, m, s)-nets and (t, s)-sequences," in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. Lecture Notes in Statistics*, **106**, 299–317, New York: Springer.

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

[14] 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.

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

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

[17] Steinberg, D. M. and Lin, D. K. J. (2006), "A Construction Method for Orthogonal Latin Hypercube Designs," *Biometrika*, **93**, 279–88.

[18] Sun, F., Liu, M., and Lin, D. K. J. (2009), "Construction of Orthogonal Latin Hypercube Designes," *Biometrika*, **96**, 971–974.

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

[20] 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.

[21] Ye, K. Q. (1998), "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments," *Journal of the American Statistical Association*, **88**, 1392–1397.