

Controlling Correlations in Latin Hypercube Samples

Art B. OWEN*

Monte Carlo integration is competitive for high-dimensional integrands. Latin hypercube sampling is a stratification technique that reduces the variance of the integral. Previous work has shown that the additive part of the integrand is integrated with error $O_p(n^{-1/2})$ under Latin hypercube sampling with n integrand evaluations. A bilinear part of the integrand is more accurately estimated if the sample correlations among input variables are negligible. Other authors have proposed an algorithm for controlling these correlations. We show that their method reduces the correlations by roughly a factor of 3 for $10 \leq n \leq 500$. We propose a method that, based on simulations, appears to produce correlations of order $O_p(n^{-3/2})$. An analysis of the algorithm indicates that it cannot be expected to do better than $n^{-3/2}$.

KEY WORDS: Computer experiment; Dependence induction; Lattice sampling; Monte Carlo integration.

1. INTRODUCTION

We begin by motivating Latin hypercube sampling (LHS) and describing how it removes terms from the error in Monte Carlo integrals. Then we show how controlling the correlations among the variables in the domain of the integrand removes further terms from the error. Finally, we sketch the rest of the article, which presents a method of controlling those correlations and gives empirical evidence that the correlations are successfully controlled. We use the standard Mann and Wald stochastic order symbols, O_p and o_p , throughout, referring the reader to appendix B of Pollard (1984) for a discussion.

Consider a deterministic function $Y = f(\mathbf{X})$, where $Y \in R$ and $\mathbf{X} \in R^p$ and $f(\cdot)$ is expensive to compute. The random vector \mathbf{X} has p statistically independent components X^1, \dots, X^p , which we refer to as input variables, and we want to estimate the mean of the response variable Y . We use superscripts for the components of \mathbf{X} , as there will be no need for powers of the vector \mathbf{X} . We write F_j for the distribution of X^j and F for $\prod_{j=1}^p F_j$, the distribution of \mathbf{X} .

One way to solve the problem is by Monte Carlo: Draw $\mathbf{X}_1, \dots, \mathbf{X}_n$ iid from F and use $\bar{Y} = n^{-1} \sum Y_i$ where $Y_i = f(\mathbf{X}_i)$ as the estimate of the mean. The components of \mathbf{X}_i will be denoted by X_i^j . For $p > 15$, Monte Carlo methods are among the best available (Davis and Rabinowitz 1984, chap. 5.10).

Because $f(\cdot)$ is expensive to compute it might pay to use a more complicated sampling scheme. McKay et al. (1979) introduce LHS for this purpose. A Latin hypercube sample is generated as follows:

$$X_i^j = F_j^{-1}((\pi_j(i) - U_{ij})/n) \quad i = 1, \dots, n, \\ j = 1, \dots, p, \quad (1)$$

where $\pi_j(1), \dots, \pi_j(n)$ is a random permutation of $1, \dots, n$ in which all $n!$ outcomes are equally probable, U_{ij} is a $U(0, 1)$ random variable, and the p permutations and np uniform variates are mutually independent. Now evaluate $Y_i = f(\mathbf{X}_i)$ and use the estimate \bar{Y} as before.

The lattice sampling technique of Patterson (1954) is a special case of LHS. In lattice sampling, the F_j are discrete

uniform distributions and n is a multiple of the number of atoms in each F_j . In this case the U_{ij} do not affect X_i^j , and they could all be taken to be .5.

The LHS method is seen to stratify on each of the p margins of the distribution of \mathbf{X} . The distribution of X_i^j is F_j . The rows \mathbf{X}_i are exchangeable but not independent. If F_j is continuous, then each of n equiprobable subintervals for X^j is represented by exactly one of the X_i^j . McKay, Conover, and Beckman (1979) showed that this reduces the sampling variance of \bar{Y} when f is monotone in each of the inputs.

Stein (1987) obtained a more informative result, using a decomposition of f into a mean, main effects, and a residual. A slight rearrangement of his decomposition is as follows: Let

$$\mu = \int f(\mathbf{X}) dF$$

be the desired integral. Define

$$\alpha_j(X^j) = \int (f(\mathbf{X}) - \mu) dF_{-j},$$

where $dF_{-j} = \prod_{k \neq j} dF_k$ integrates out all components but the j 'th, leaving a function of X^j alone. Now define the residual function $r(\mathbf{X})$ via

$$f(\mathbf{X}) = \mu + \alpha_1(X^1) + \dots + \alpha_p(X^p) + r(\mathbf{X}). \quad (2)$$

Stein showed that

$$\text{var}_{\text{LHS}}(\bar{Y}) = n^{-1} \int r(\mathbf{X})^2 dF + o(n^{-1}). \quad (3)$$

This is asymptotically at least as small as

$$\begin{aligned} \text{var}_{\text{IID}}(\bar{Y}) &= n^{-1} \int (f(\mathbf{X}) - \mu)^2 dF \\ &= n^{-1} \int r(\mathbf{X})^2 dF + n^{-1} \sum_{j=1}^p \int \alpha_j(X^j)^2 dF_j, \end{aligned}$$

and can be a lot smaller when the additive main effects α_j are large. As Stein (1987) put it, the main effects are filtered out. The α_j do not have to be monotone. Stein showed that if each α_j has a bounded first derivative, then the error term

* Art B. Owen is Associate Professor, Department of Statistics, Stanford University, Stanford, CA 94305. He is also an affiliated faculty member of Stanford's Center for Integrated Systems. This work was supported by the Semiconductor Research Corporation and by National Science Foundation Grant DMS-9011074.

in (3) is $O(n^{-2})$. Stein's result is a generalization to LHS of an anova decomposition due to Patterson (1954).

Substituting (2) into \bar{Y} , one gets

$$\bar{Y} = \mu + \sum_{j=1}^p n^{-1} \sum_{i=1}^n \alpha_j(X_i^j) + \bar{r}. \quad (4)$$

where $\bar{r} = n^{-1} \sum r(X_i)$. The right side of (4) is μ plus Riemann sums of the α_j plus the sample mean of the $r(X_i)$. If α_j has a derivative bounded in absolute value by C , then $|n^{-1} \sum_{i=1}^n \alpha_j(X_i^j)| \leq Cn^{-1}$. The first two theorems of Davis and Rabinowitz (1984, chap. 2.1) can be adapted to show this. Thus the Riemann sums are negligible compared to the Monte Carlo rate $O_p(n^{-1/2})$ for \bar{r} . Bounded differentiability of α_j is sufficient, but not necessary. Mere existence of $\int \alpha_j^2 dF$, which follows from that of $\int f^2 dF$, is sufficient according to Michael Stein (personal communication).

Equation (3) says that $\text{var}_{\text{LHS}}(\bar{Y}) = \text{var}_{\text{IID}}(\bar{r}) + o(n^{-1})$. Owen (1992) extended this result, showing that if f is bounded, then for any positive integer k ,

$$E_{\text{LHS}}((n^{1/2}\bar{r})^k) = E_{\text{IID}}((n^{1/2}\bar{r})^k) + O(n^{-1}). \quad (5)$$

(The multiple of n^{-1} implicit in $O(n^{-1})$ can depend on k .) So to the order given in (5), \bar{Y} under LHS behaves like μ plus the mean of an IID sample from r . This implies that $n^{1/2}(\bar{Y} - \mu)$ is asymptotically $N(0, \int r(\mathbf{X})^2 dF)$. Owen (1992) gave $n^{1/2}$ -consistent estimates of $\int r(\mathbf{X})^2 dF$ and $\int r(\mathbf{X})^3 dF$.

We assume from here on that each F_j is the uniform distribution on $[0, 1]$. There is no loss of generality in this, even for discrete F_j , as we can replace $f(\mathbf{X})$ by $f(F^{-1}(\mathbf{X}))$, where the j 'th component of $F^{-1}(\mathbf{X})$ is $F_j^{-1}(X^j) = \inf\{u | F_j(u) \geq X^j\}$.

The residual from additivity can be further decomposed as

$$r(\mathbf{X}) = \sum_{j < k} \gamma_{jk}(X^j - .5)(X^k - .5) + r'(\mathbf{X}), \quad (6)$$

where the bilinear coefficients are

$$\begin{aligned} \gamma_{jk} &= \frac{\int r(\mathbf{X})(X^j - .5)(X^k - .5) dF}{\int [(X^j - .5)(X^k - .5)]^2 dF} \\ &= 144 \int r(\mathbf{X})(X^j - .5)(X^k - .5) dF. \end{aligned}$$

The $p(p-1)/2 + 1$ terms in (6) are mutually orthogonal, in particular,

$$E(r'(\mathbf{X})(X^j - .5)(X^k - .5)) = 0$$

for all j, k . Substituting (6) into (4), we get

$$\bar{Y} = \mu + \sum_{j=1}^p n^{-1} \sum_{i=1}^n \alpha_j(X_i^j) + \sum_{j < k} \gamma_{jk} \rho_{jk} + n^{-1} \sum_{i=1}^n r'(\mathbf{X}_i) \quad (7)$$

where $\rho_{jk} = n^{-1} \sum_i (X_i^j - .5)(X_i^k - .5)$ is essentially the sample covariance between X^j and X^k . It differs from the sample covariance because the sample mean of X^j is $.5 + O_p(n^{-2})$, due to the U_{ij} .

For a Latin hypercube sample in which the sample correlations among the inputs are $O_p(n^{-1/2})$, the mean of the $r'(\mathbf{X}_i)$ is the dominant error in the integral. In addition to the additive component of f , a sum of bilinear components is filtered out.

Iman and Conover (1982) described an algorithm for selecting a Latin hypercube sample with rank correlations among the input variables near some target values. Taking the target correlation matrix to be the identity should yield improved accuracy in the integral. Another motivation is that some applications require dependent input variables. When such dependence can be captured by inducing rank correlations among the inputs, a more relevant integral may be estimated.

The method that we propose in Section 2 is seen (empirically) in Section 3 to produce correlations that are $O_p(n^{-1/2})$. We conjecture that the resulting sample means, \bar{Y} , are asymptotically normal with mean μ and variance $n^{-1} \int r'(\mathbf{X})^2 dF$ as $n \rightarrow \infty$. Owen (1992) showed that such a normal limit arises for unmodified LHS.

The plan of the article is as follows. Section 2 describes Iman and Conover's dependence induction algorithm and the ranked Gram-Schmidt (RGS) algorithm proposed here. Section 3 assesses the effectiveness of these two algorithms for reducing off-diagonal correlations. The magnitude of these correlations plotted versus $10 \leq n \leq 500$ is surprisingly linear on a log-log scale, and the variance about the line is quite small. The RGS method appears to produce Latin hypercube samples with correlations of order $O_p(n^{-3/2})$ among input variables. Section 4 suggests a reason for the empirical findings of Section 3 and speculates on lower bounds for the magnitude of the correlations. Section 5 contains a discussion, including a description of an integrand arising in computer experiments in which one would expect large bilinear terms.

2. ALGORITHMS

We consider the problem of generating $P \geq p$ columns of a Latin hypercube sample, of which p will be used to estimate $\int f dF$. We may let P grow with n , but we always take $P \leq n - 1$. After excluding multiples of $(1, \dots, 1)'$, there can be at most $n - 1$ uncorrelated vectors of n components.

Iman and Conover (1982) described their algorithm, which I will refer to as ranked Cholesky (RC) herein. Dr. Iman kindly sent me a computer package that produces Latin hypercube samples. The package allows the specification of a rank correlation matrix and provides many options, including a large selection of marginal distributions.

The following description is based on the computer program LHS by R. Iman and M. Shortencarier of Sandia National Laboratories, because it is later than the article by Iman and Conover (1982). The description assumes that the margins are to be $U(0, 1)$ and that the desired rank covariance matrix is the identity. First, an n -by- P matrix \mathbf{U} of independent uniform random variables is generated. Then an n by P matrix \mathbf{Z} is generated in which each column $\mathbf{Z}^j, j = 1, \dots, P$ is a random permutation of $\Phi^{-1}(i/(n+1)), i = 1, \dots, n$, where Φ is the standard normal cdf. All $(n!)^P$ -possible matrices \mathbf{Z} are equally probable. Let \mathbf{C} be the observed co-

variance matrix of \mathbf{Z} . If \mathbf{C} is rank-deficient, then another matrix \mathbf{Z} is generated; except for very small n this is quite unlikely to be necessary. A lower triangular matrix \mathbf{Q} , such that $\mathbf{Q}\mathbf{Q}' = \mathbf{C}$, is computed using the Cholesky decomposition. Then $\mathbf{Z}^* = \mathbf{Z}\mathbf{Q}^{-1}$ is computed. The n rows of \mathbf{Z}^* , regarded as P by 1 column vectors, have sample covariance equal to the identity. Finally, the values in the column \mathbf{U}^j are permuted so that the vector of ranks of \mathbf{U}^j matches the vector of ranks of \mathbf{Z}^{*j} . Any given \mathbf{U}_i^j has a uniform distribution, but the vector \mathbf{U}^j is stratified across $(0, 1)$ and the correlations among the \mathbf{U}^j tend to be smaller than before the permutation. Experimental results in the next section show that the correlations are roughly one-third as large this way, for n between 10 and 500. It does not seem to matter whether P increases with n .

Iman and Shortencarier's program allows a choice of many other marginal distributions for \mathbf{U} and allows target covariances other than the identity. If the target covariance is \mathbf{V} , and $\mathbf{SS}' = \mathbf{V}$ is a Cholesky decomposition, then they take $\mathbf{Z}^* = \mathbf{Z}\mathbf{Q}^{-1}\mathbf{S}$.

In the RGS algorithm proposed here, a further simplification is made. The quantities \mathbf{U}_i^j in (1) are all taken to be .5 instead of $\mathbf{U}(0, 1)$, so our sample is of the lattice type. This would be undesirable if there were periodicities in f with a wavelength of nearly $1/n$, but in many applications this may be safely neglected. The RGS algorithm starts with $\mathbf{X}_i^j = (\pi_j(i) - .5)/n$ where the π_j are independent permutations as before. Then Gram-Schmidt orthogonalization is applied to the variables $\mathbf{X}^j - .5$, $j = 1, \dots, k$ with a modification to preserve their stratification. Some notation helps here. Use the function *takeout* (\mathbf{x}, \mathbf{y}) to denote the residuals from a linear regression (including an intercept) of the vector \mathbf{y} on the vector \mathbf{x} . The vector \mathbf{x} is taken out of the vector \mathbf{y} . Let $\text{rank}(\mathbf{x})$ denote the vector of ranks of \mathbf{x} . The forward step of RGS is

```

for  $j = 2, \dots, P$ 
  for  $k = 1, \dots, j-1$ 
     $\mathbf{X}^k \leftarrow \text{takeout}(\mathbf{X}^j, \mathbf{X}^k)$ 
     $\mathbf{X}^k \leftarrow (\text{rank}(\mathbf{X}^k) - .5)/n,$ 
```

where \leftarrow denotes assignment, and the backward step of RGS is

```

for  $j = P-1, \dots, 1$ 
  for  $k = P, \dots, j+1$ 
     $\mathbf{X}^k \leftarrow \text{takeout}(\mathbf{X}^j, \mathbf{X}^k)$ 
     $\mathbf{X}^k \leftarrow (\text{rank}(\mathbf{X}^k) - .5)/n.$ 
```

The full algorithm proceeds by alternating forward and backward steps. Convergence is discussed in Section 3.

As discussed in the next section, the RGS algorithm is more successful at reducing correlations than the RC algorithm. The difference is too large to be accounted for by the setting of the uniform random variables to .5 in RGS.

The computational load in both of these algorithms is of order nP^2 , if we ignore the usually smaller load of order $nP \log(n)$ taken up by sorting. This assumes that the number

of iterations needed in RGS does not grow with n . The \mathbf{x} vector in each takeout (\mathbf{x}, \mathbf{y}) is always a permutation of $(i - .5)/n$, where $i = 1, \dots, n$, and this can be exploited to simplify the computations. When speed is a major issue, much of the RGS computation can be done in integer arithmetic.

3. PERFORMANCE

In this section we explore and compare the performance of the RGS and RC algorithms. First, we analyze unmodified LHS, using the lattice version. Let the columns $\mathbf{X}^1, \dots, \mathbf{X}^P$ be independent permutations of $(i - .5)/n$ for $i = 1, \dots, n$ with all $(n!)^P$ possibilities equally probable. The sample correlation between columns \mathbf{X}^j and \mathbf{X}^k is denoted by ρ_{jk} . It is easy to show that for $j \neq k$, ρ_{jk} has mean zero and variance $1/(n-1)$. This calculation is made easier by the fact that the denominator in the sample correlation is not random. The ρ_{jk} have a limiting normal distribution as $n \rightarrow \infty$. The $P(P-1)/2$ off-diagonal correlations are pairwise independent, though some triples of them are dependent. For $j \neq k$, ρ_{jk}^2 has variance $2n^{-2} + O(n^{-3})$. To show this, one must compute $E_{\text{LHS}}(\rho_{jk}^4)$. Equation (5) helps here. We can take $r(\mathbf{X}) = (\mathbf{X}^j - .5)(\mathbf{X}^k - .5)$, so that \bar{r} differs from ρ_{jk} by a normalizing constant. The result follows after some arithmetic.

This makes it convenient to adopt the root mean square correlation among columns of \mathbf{X} as a performance measure; that is,

$$\rho_{\text{rms}}^2(\mathbf{X}) = \frac{\sum_{j=2}^P \sum_{k=1}^{j-1} \rho_{jk}^2}{(P-1)P/2},$$

where ρ_{jk} is the sample correlation between vectors \mathbf{X}^j and \mathbf{X}^k . Clearly, $E_{\text{LHS}}(\rho_{\text{rms}}^2) = (n-1)^{-1}$, and because the $P(P-1)/2$ off-diagonal sample correlations are uncorrelated under LHS, it follows that $\text{var}_{\text{LHS}}(\rho_{\text{rms}}^2) = 4[n^2P(P-1)]^{-1}(1 + O(n^{-1}))$. If $P \rightarrow \infty$ as $n \rightarrow \infty$, then it follows from a Taylor expansion that $E_{\text{LHS}}(\rho_{\text{rms}}^2) \doteq n^{-1/2}$ and $\text{var}_{\text{LHS}}(\rho_{\text{rms}}^2) \doteq (nP^2)^{-1}$.

The procedures RC and RGS were applied with $n \in \{10, 20, 30, 50, 100, 150, 250, 500\}$ and $P = n-1$. For each value of $n < 500$, there were four replicates. Only one simulation was done for $n = 500$. Figure 1 shows a plot of $\rho_{\text{rms}}^{\text{RC}}$ and $\rho_{\text{rms}}^{\text{RGS}}$ versus n on a log-log scale. The RGS points are plotted as octagons, and the RC points are plotted as crosses. There are three reference lines. The top line gives the $n^{-1/2}$ expected for unmodified LHS; the other two lines are regression lines of $\log(\rho_{\text{rms}}^{\text{RC}})$ and $\log(\rho_{\text{rms}}^{\text{RGS}})$ versus $\log(n)$.

From Figure 1, we see that there is very little variation in ρ_{rms}^2 for a given method and value of n . In this case ρ_{rms}^2 is the mean of nearly $n^2/2$ squared correlations. We have $\text{var}_{\text{LHS}}(\rho_{\text{rms}}^2) \doteq n^{-3}$, and RC or RGS may reduce this variance further. Very little lack of fit is evident in the regressions in Figure 1. From Figure 1, it is clear that the RC algorithm really does reduce the magnitude of the correlations among input variables and that the RGS algorithm reduces it further. More interestingly, the regression line for RC is almost parallel to that for uncorrected LHS, while the line for RGS has a steeper slope.

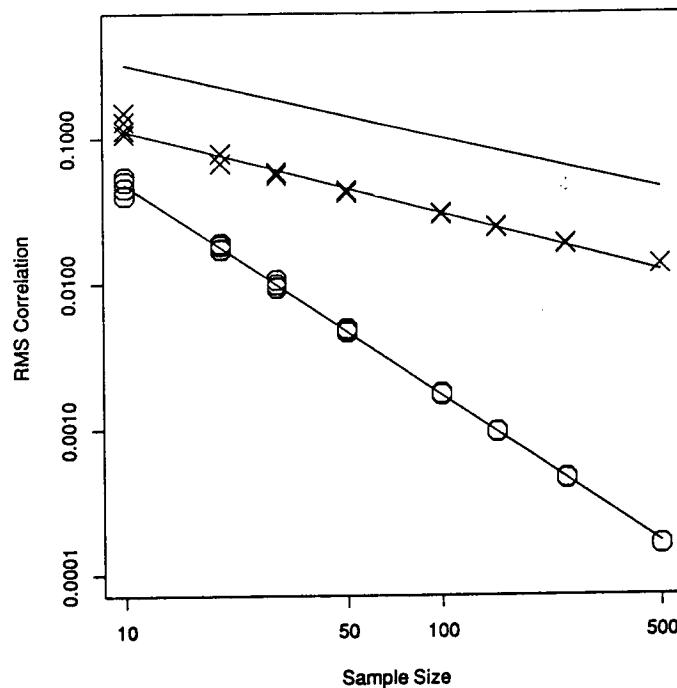


Figure 1. Root Mean Square Correlation ρ_{rms} Versus Sample Size n for $P = n - 1$ Dimensional Latin Hypercube Samples. Crosses denote ρ_{rms}^{RC} for the ranked Cholesky algorithm, and octagons denote ρ_{rms}^{RGS} for the ranked Gram-Schmidt algorithm. The reference lines, from top to bottom, are $n^{-1/2}$, least squares regression of $\log(\rho_{rms}^{RC})$ on $\log(n)$, and least squares regression of $\log(\rho_{rms}^{RGS})$ on $\log(n)$.

These regression lines do not prove anything about the asymptotic behavior of ρ_{rms} under RC or RGS. But for some purposes, these empirical results are more useful than typical asymptotic results. The linearity in Figure 1 is evident for samples as small as $n = 10$ or 20 and continues at least to $n = 500$. Moreover, we can extract constants from the regressions, whereas many asymptotic results give only the power of n .

Ordinary least squares regression gives

$$\log \rho_{rms}^{RGS} \doteq .30 - 1.45 \log(n)$$

and

$$\log \rho_{rms}^{RC} \doteq -.88 - .57 \log(n),$$

compared to the theoretical rate

$$\log \rho_{rms} \doteq -.5 \log(n)$$

for uncorrected LHS. Ordinary least squares does not bias the slope estimates beyond the bias due to lack of fit. A standard error for these slopes must account for the heteroscedasticity. In this article standard errors are based on formula (2.6) of Wu (1986) without the factor $1 - k/n$ (in Wu's notation). Using this method, one gets a standard error of .011 for the slope estimate in the RGS case and a standard error of .018 for the slope estimate in the RC case. So ρ_{rms} differs by somewhat more than sampling error from the rates $n^{-3/2}$ for RGS and $n^{-1/2}$ for RC.

If only the cases with $n \geq 100$ are included, the coefficient for RGS becomes -1.48 with a standard error of .0064, and

the coefficient for RC becomes $-.517$ with a standard error of .0098. One might expect this behavior if the asymptotes really were like $n^{-3/2}$ and $n^{-1/2}$.

Based on the data presented in Figure 1, $\rho_{rms} \doteq 1.35n^{-1.45}$ for RGS and $\rho_{rms} \doteq .42n^{-.57}$ for RC when n is between 10 and 500. For RC, this approximation indicates that the correlations are smaller by a factor ranging from 2.8 (for $n = 10$) to 3.68 (for $n = 500$) than those obtained by LHS.

The sample variance of the four RGS readings at $n = 250$ is $8.9 * 10^{-12}$. A simple regression of the log variances on $\log n$ indicates that the variance decreases roughly as $n^{-4.95}$ (with a standard error of .23 for the exponent). This is not surprising; one might have guessed that reducing ρ_{rms} by a multiple of n would reduce its variance by a multiple of n^2 . Due to this behavior, we might take the value for the single run at $n = 500$ as quite representative of what happens for $n = 500$. If the asymptote is of order $n^{-3/2}$ for RGS, then we can estimate that $\rho_{rms} \doteq 1.77 * n^{-3/2}$, where the coefficient is based solely on the final, most accurate, observation. There is very little sampling variability in this coefficient. Though systematic errors are hard to assess, they should also be smallest for the $n = 500$ observation.

A more conservative measure of the behavior of the off-diagonal correlations is based on the largest off-diagonal correlation. An analysis like the foregoing for RGS leads to $.52 - 1.17 \log(n)$ for the regression of the log-maximum absolute correlation on $\log n$. The standard error for the exponent is .025.

Figure 2 shows data analogous to that in Figure 1, for the case in which $P = 9$. For $n = 500$, four realizations were computed in this case, because the cost was not very high.

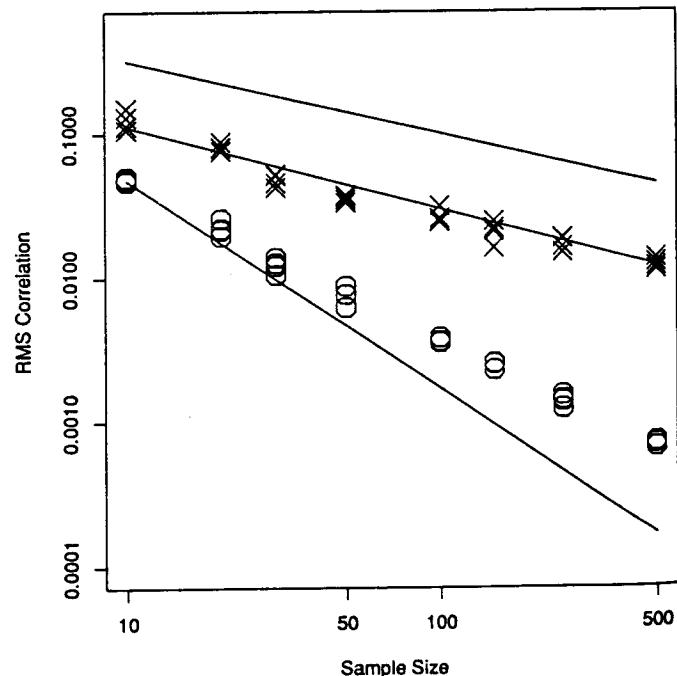


Figure 2. Root Mean Square Correlation ρ_{rms} Versus Sample Size n for $P = 9$ Dimensional Latin Hypercubes. Crosses denote ρ_{rms}^{RC} for the ranked Cholesky algorithm, and octagons denote ρ_{rms}^{RGS} for the ranked Gram-Schmidt algorithm. The reference lines are identical to the ones in Figure 1, which are least squares fits to the Figure 1 data.

The reference lines in Figure 2 are the same as the ones in Figure 1. The most interesting feature is that the values of ρ_{rms}^{RGS} are above the reference line from $P = n - 1$. So generating $P = n - 1$ columns and then selecting $p = 9$ of them at random will give smaller covariances than generating just the desired nine columns. Of course, one could do still better by generating $P = n - 1$ columns, and selecting nine of them among which the correlations are especially small. Unlike the RGS case, ρ_{rms}^{RC} does not appear to be smaller for $P = n - 1$ than for $P = 9$. For this data, $\log \rho_{rms}^{RGS} \doteq -0.62 - 1.080 \log(n)$ with a standard error of .012 for the slope. When only the runs for $n \geq 100$ are used, $\log \rho_{rms}^{RGS} \doteq -0.83 - 1.038 \log(n)$ with standard error .015 for the slope.

We conclude this section with remarks on the convergence behavior observed for the RGS algorithm. In the example with $P = 9$, the RGS algorithm converges in two to five complete passes, in every case but one. In that case n was 10 and the design matrix alternated between two designs as the algorithm alternated between forward and backward steps. The alternation started in the fourth pass. The number of iterations required did not appear to grow with n .

For $P = n - 1$, there were eight complete passes in each of the runs. Convergence was not recorded for these runs, but subsequent investigation revealed that nonconvergence by eight passes is common for $P = n - 1$ and large n . Only relatively small changes in ρ_{rms}^{RGS} occur after four or five passes.

Another series of runs had P taken as the closest integer to $3n^{1/2}$. (This experiment is discussed in Section 4.) Four runs each were done at $n \in \{20, 30, 50, 100, 150, 250, 500\}$. The number of passes to convergence was always between three and seven. In two cases the algorithm alternated between two designs. In one of these n was 100; in the other n was 250. In both cases the alternation started on the fourth pass. Only one case took seven passes (n was 30), and only one took six passes (n was 20).

4. BOUNDS AND EXPLANATIONS

This section considers how small ρ_{rms} might reasonably be and considers why $n^{-3/2}$ might be the best that could be attained via RGS. Consider a single "takeout" step in the RGS algorithm,

$$\mathbf{X}^k \leftarrow \text{takeout}(\mathbf{X}^j, \mathbf{X}^k).$$

This amounts to

$$\mathbf{X}^k \leftarrow \mathbf{X}^k - (\mathbf{X}^j - .5) \rho_{jk} \sigma_k / \sigma_j, \quad (8)$$

where $\mathbf{X}^j - .5$ means the vector \mathbf{X}^j minus .5 times a vector of 1s, ρ_{jk} is the sample correlation between \mathbf{X}^j and \mathbf{X}^k before the update, and σ_j and σ_k are the standard deviations of the two column vectors. For this step to result in a change to \mathbf{X}^k , the rank vector of \mathbf{X}^k must change. That requires modifying the components of \mathbf{X}^k by $O(n^{-1})$. The change is $O(\rho_{jk})$, however. So if ρ_{jk} is small compared to n^{-1} , then this step is unlikely to change \mathbf{X}^k by enough to change its rank vector.

Suppose that P vectors are to be taken out of \mathbf{X}^k before it is reranked. Then the sum of P updates like the one in (8)

is subtracted from \mathbf{X}^k . We might expect these updates to be of both positive and negative signs and to have small correlations among themselves. In that case \mathbf{X}^k changes by $O(P^{1/2})$ times the change from a typical \mathbf{X}^j . Therefore, if the ρ_{jk} become small compared to $n^{-1}P^{-1/2}$, then the RGS algorithm will stop changing \mathbf{X} . This argument supports the empirical appearance of $\rho_{rms}^{RGS} = O_p(n^{-3/2})$ when $P = n - 1$ and $\rho_{rms}^{RGS} = O_p(n^{-1})$ when $P = 9$. A repeat of the experiment with P the closest integer to $3n^{1/2}$ yielded a slope estimate of -1.30. This is close to the -1.25 that might be predicted based on the preceding, but it differs by more than sampling error.

The foregoing seems to explain why the algorithm does as well as it does, and why the results are better when more columns are used. How close does the method come to the best possible correlation matrix? A very crude heuristic argument runs as follows. There are $(n!)^P \doteq n^{nP}$ equally probable Latin hypercube samples. There are $P(P - 1)/2$ correlations of interest, which have a standard deviation of essentially $n^{-1/2}$. If we suppose that their distribution is roughly multivariate normal, then we can approximate the probability that all of them are smaller than $a_n n^{-1/2}$. For $P = n - 1$, the critical rate for a_n is n^{-2} in that the expected number of Latin hypercube designs with all correlations smaller than $a_n n^{-1/2}$ diverges to ∞ if $a_n n^2 \rightarrow \infty$ and converges to 0 if $a_n n^2 \rightarrow 0$. So for $P = n - 1$, this crude argument suggests that $\rho_{rms} = O(n^{-5/2})$ is the critical rate. So it may be possible to get much better control over the correlations than RGS gets.

Another bound may be found by a parity argument. The correlation of columns \mathbf{X}^1 and \mathbf{X}^2 is not changed by adding .5 to each and multiplying them by n . This makes both columns into permutations of $1, \dots, n$. Without loss of generality, take $X_1^i = i$ and $X_2^i = \pi_i$, where (π_1, \dots, π_n) is a permutation of $1, \dots, n$. Then

$$\rho_{12} = (\sum i\pi_i - n(n+1)^2/4)/D, \quad (9)$$

where $D = \sum i^2 - n(n+1)^2/4 = n(n^2 - 1)/12$. If $n \equiv 2 \pmod{4}$, it is impossible for ρ_{12} to be zero. In (9), $\sum i\pi_i$ is an integer but $n(n+1)^2/4$ is not. So $|\rho_{12}|$ has a lower bound of about $6n^{-3}$ when $n \equiv 2 \pmod{4}$.

5. DISCUSSION

The decomposition (7) shows that an integral estimated on a Latin hypercube sample selected by the RGS method will have an error dominated by the mean of the $r'(\mathbf{X}_i)$. This follows because all of the other error terms are $O_p(n^{-1/2})$. We conjecture that the sampling variance of such an estimate is asymptotic to $n^{-1} \int r'(\mathbf{X})^2 dF$. This seems reasonable, because the steps of the RGS algorithm are based on sample means of $(\mathbf{X}^j - .5)(\mathbf{X}^k - .5)$ and $(\mathbf{X}^j - .5)(\mathbf{X}^k - .5)$ is uncorrelated with $r'(\mathbf{X})$.

Here is a case in which filtering out the bilinear terms may greatly reduce variance. Suppose that we are interested in fitting a linear model,

$$f(\mathbf{X}) \doteq \mu + \sum_{j=1}^k \beta_j (X^j - .5). \quad (10)$$

The minimum integrated squared error model of the form (10) has

$$\begin{aligned}\beta_j &= \left(\int (X^j - .5)^2 dF \right)^{-1} \int f(\mathbf{X})(X^j - .5) dF \\ &= 12 \int f(\mathbf{X})(X^j - .5) dF.\end{aligned}$$

We can estimate β_j by $\hat{\beta}_j = 12n^{-1} \sum_i f(X_i)(X_i^j - .5)$. Suppose that much of the variation in f under sampling $\mathbf{X} \sim F$ is explained by the linear model (10). Then it follows that $f(\mathbf{X})(X^j - .5)$, $j = 1, \dots, p$ will have large bilinear (and quadratic) terms that will be filtered out by the design.

It may be possible to reduce the correlations among the input variables by much more than RGS does. But because the Monte Carlo error is already dominated by the mean of the $r'(\mathbf{X}_i)$, there would be very little improvement to be gained this way. There may also be something to lose; if a design is found that has very small correlations but some sharply nonuniform features, then we might expect it to be a poor choice for some functions f . We take instead the advice of those experimenters who "control what they can and randomize the rest."

If it is desired to induce some nonzero correlation structure among the columns of \mathbf{X} , one can do this by modifying the takeout stages to replace \mathbf{X}^k by a linear combination of \mathbf{X}^j and takeout $(\mathbf{X}^j, \mathbf{X}^k)$.

[Received February 1990. Revised November 1993.]

REFERENCES

- Davis, P. J., and Rabinowitz, P. (1984), *Methods of Numerical Integration* (2nd ed.), Orlando, FL: Academic Press.
- Iman, R. L., and Conover, W. J. (1982), "A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables," *Communications in Statistics, Part B—Simulation and Computation*, 11, 311–334.
- 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.
- Owen, A. B. (1992), "A Central Limit Theorem for Latin Hypercube Sampling," *Journal of the Royal Statistical Society, Ser. B*, 54, 541–551.
- Patterson, H. D. (1954), "The Errors of Lattice Sampling," *Journal of the Royal Statistical Society, Ser. B*, 16, 140–149.
- Pollard, D. (1984), *Convergence of Stochastic Processes*, New York: Springer.
- Stein, M. L. (1987), "Large Sample Properties of Simulations Using Latin Hypercube Sampling," *Technometrics*, 29, 143–151.
- Wu, C. F. J. (1986), "Jackknife, Bootstrap, and Other Resampling Methods in Regression Analysis" (with discussion), *The Annals of Statistics*, 14, 1261–1350.