Simulation Estimation of Mixed Discrete Choice Models Using Randomized Quasi-Monte Carlo Sequences: A Comparison of Alternative Sequences, Scrambling Methods, and Uniform-to-Normal Variate Transformation Techniques

Aruna Sivakumar, Chandra R. Bhat
The University of Texas at Austin
Department of Civil Engineering
1 University Station C1761
Austin, Texas 78712-0278.
Tel: 512-471-4535, Fax: 512-475-8744
Email: arunas@mail.utexas.edu, bhat@mail.utexas.edu

and Giray Ökten
Ball State University
Department of Mathematical Sciences
Muncie, IN 47306
Tel: 765-285-8677, Fax: 765-285-1721
Email: gokten@math.bsu.edu

Sivakumar, Bhat, and Ökten

ABSTRACT

This paper numerically compares the overall performance of the quasi-Monte Carlo (QMC) sequences proposed by Halton and Faure, and their scrambled versions, against each other and against the Latin Hypercube Sampling sequence in the context of the simulated likelihood estimation of a Mixed Multinomial Logit model of choice. In addition, the efficiency of the QMC sequences generated with and without scrambling across observations is compared, and the performance of the Box-Muller and Inverse Normal transform procedures is tested. The numerical experiments were performed in 5 dimensions with 25, 125 and 625 draws, and in 10 dimensions with 100 draws. The results of our analysis indicate that the Faure sequence consistently outperforms the Halton sequence, and the scrambled versions of the Faure sequence show the best performances overall.

1. INTRODUCTION
The incorporation of behavioral realism in econometric models helps establish the credibility of the models outside the modeling community, and can also lead to superior predictive and policy analysis capabilities. Behavioral realism is incorporated in econometric models of choice through the relaxation of restrictions that impose inappropriate behavioral assumptions regarding the underlying choice process. For example, the extensively used multinomial logit (MNL) model has a simple form that is achieved by the imposition of the restrictive assumption of independent and identically distributed error structures (IID). But this assumption also leads to the not-so-intuitive property of independence from irrelevant alternatives (IIA).

The relaxation of behavioral restrictions on choice model structures, in many cases, leads to analytically intractable choice probability expressions, which necessitate the use of numerical integration techniques to evaluate the multidimensional integrals in the probability expressions. The numerical evaluation of such integrals has been the focus of extensive research dating back to the late 1800s, when multidimensional polynomial-based cubature methods were developed as an extension of the one-dimensional numerical quadrature rules. These quadrature-based methods, however, suffered from the "curse of dimensionality"; and so Monte-Carlo (MC) and pseudo-Monte Carlo (PMC) simulation methods were proposed in the 1940s to overcome this problem. The MC simulation approach has an expected integration error of $N^{-0.5}$, which is independent of the number of dimensions 's' and thus provides a great improvement over the quadrature-based methods. Several variance reduction techniques (example, Latin Hypercube Sampling or LHS) have since been developed for the MC methods, which potentially lead to even more accurate integral evaluation with fewer draws. Despite the improvements achieved by these variance reduction techniques, the convergence rate of MC and PMC methods is generally slow for simulated likelihood estimation of choice models.

Extensive number theory research in the last few decades has led to the development of a more efficient simulation method, the quasi-Monte Carlo (QMC) method. This method uses the basic principle of the MC method in that it evaluates a multidimensional integral by replacing it with an average of the values of the integrand computed at N discrete points. However, rather than using random sequences, QMC methods use low-discrepancy, deterministic, quasi-Monte Carlo (or QMC) sequences that are designed to achieve a more even distribution of points in the integration space than the MC and PMC sequences.

Over the years, several different quasi-random sequences have been proposed for QMC simulation. Among these are the reverse radix-based sequences (such as the Halton sequence) and the (t,s)-sequences (such as the Sobol and Faure sequences). The even distribution of points provided by these low-discrepancy sequences leads to efficient convergence for the QMC method, generally at rates that are higher than the MC method. In particular, the theoretical upper bound for the integration error in the QMC method is of the order of N^{-1} for one-dimensional integration[1]. Despite these obvious advantages, the QMC method has two major limitations. First, the deterministic nature of the quasi-random sequences makes it difficult to estimate the error in the QMC simulation procedure (while there are theoretical results to estimate integration error via upper bounds with the QMC sequence, these are much too difficult to compute and are very conservative upper bounds). Second, a common problem with many low-discrepancy sequences is that they exhibit poor properties in higher dimensions. The Halton sequence, for example, suffers from significant correlations between the radical inverse functions for different

---

[1] In s-dimensions the upper bound for the integration error is $O[(\log N)^s / N]$.

dimensions, particularly in the larger dimensions. A growing field of research in QMC methods has resulted in the development, and continuous evolution, of efficient *randomization* strategies (to estimate the error in integral evaluation) and *scrambling* techniques (to break correlations in higher dimensions).

Research on the generation and application of randomized and scrambled QMC sequences clearly indicates the superior accuracy of QMC methods over PMC methods in the evaluation of multidimensional integrals [see Morokoff and Caflisch (*1,2*)]. In particular, the advantages of using QMC simulation for such applications in econometrics as simulated maximum likelihood inference, where parameter estimation entails the approximation of several multidimensional integrals at each iteration of the optimization procedure, should be obvious. However, the first introduction of the QMC method for the simulated maximum likelihood inference of econometric choice models occurred only in 1999, when Bhat tested Halton sequences for mixed logit estimation and found their use to be vastly superior to random draws. Since Bhat's initial effort, there have been several successful applications of QMC methods for the simulation estimation of flexible discrete choice models, though most of these applications have been based on the Halton sequence [see, for example, Revelt and Train (*3*); Bhat (*4*); Park et al. (*5*); Bhat and Gossen (*6*)]. Number theory, however, abounds in many other kinds of low-discrepancy sequences that have been proven to have better theoretical and empirical convergence properties than the Halton sequence in the estimation of a single multidimensional integral. For instance, Bratley and Fox (*7*) show that the Faure and Sobol sequences are superior to the Halton sequence in terms of accuracy and efficiency. There have also been several numerical studies on the simulation estimation of a single multidimensional integral that present significant improvements in the performance of QMC sequences through the use of scrambling techniques (*8,9*). It is, therefore, of interest to examine the performances of the different QMC sequences and their scrambled versions in the simulation estimation of flexible discrete choice models, which is the focus of the current paper.

The rest of this paper is organized as follows. Section 2 discusses the specific objectives of this study. Section 3 presents the background for the generation of alternative sequences. Section 4 describes the evaluation framework used in this study. Section 5 presents the computational results and Section 6 concludes the paper.

2. OBJECTIVES

As described in the previous section, the broad goal of this paper is to compare the performance of different kinds of low-discrepancy sequences, and their scrambled and randomized versions, in the simulated maximum likelihood estimation of the mixed logit class of discrete choice models. Specifically, we selected the extensively used Halton sequence and a special case of (*t,m,s*)-nets known as the Faure sequence. The choice of the Faure sequence was motivated by two reasons. First, the generation of the Faure sequence is a fairly straightforward and non-time consuming procedure. Second, it has been proved that the Faure sequence performs better than the Halton sequence in the evaluation of a single multidimensional integral (*8*).

In this paper, we compare the performance of the Halton and Faure sequences against the performance of a stratified random sampling PMC sequence (the LHS sequence) by constructing numerical experiments within a simulated maximum likelihood inference framework. Further, the numerical experiments also include a comparison of scrambled versions of the QMC sequences against their standard versions to examine potential improvements in performance through scrambling. The performances of the various non-scrambled and scrambled sequences

are evaluated based on their ability to efficiently and accurately recover the true model parameters.

The total number of draws of a QMC sequence required for the estimation of a Mixed Multinomial Logit (MMNL) model can be generated either *with* or *without* scrambling across observations (see section 3.5 for a description of these methods), and both these approaches are compared in this paper. Another important point to note is that the standard and scrambled versions of the QMC and the LHS sequences are all generated as uniformly-distributed sequences of points. In this study we test and compare the Box-Muller and the Inverse Normal transformation procedures to convert the uniformly-distributed sequences to normally-distributed sequences that are required for the estimation of the random coefficients MMNL model.

To summarize, the objectives of this paper are three-fold. The first objective is to experimentally compare the overall performance of the Halton and Faure sequences (and their scrambled versions) against each other and against the LHS sequence[2]. The second objective is to compare the efficiency of the QMC sequences with and without scrambling across observations. The third objective is to compare the Box-Muller and the Inverse Normal transform procedures for translating uniformly-distributed sequences to normally-distributed sequences.

3. BACKGROUND FOR GENERATION OF ALTERNATIVE SEQUENCES

This section describes the various procedures to generate PMC and QMC sequences. Specifically, the following sections discuss the generation of PMC sequences using the LHS procedure (Section 3.1), and the generation of the QMC sequences proposed by Halton and Faure (Section 3.2); the scrambling strategies (Section 3.3) and randomization techniques (Section 3.4) applied in this study; the generation of sequences *with* and *without* scrambling across observations (Section 3.5); and basic descriptions of the Box-Muller and Inverse Normal transforms (Section 3.6).

3.1. PMC Sequences

A typical PMC simulation uses a simple random sampling (SRS) procedure to generate a uniformly-distributed PMC sequence over the integration space. An alternate approach known as Latin Hypercube sampling (LHS), that yields asymptotically lower variance than SRS, is described in the following section.

3.1.1. Latin Hypercube Sampling

The LHS method was first proposed as a variance reduction technique within the context of PMC sequence-based simulation (*11*). The basis of LHS is a full stratification of the integration space, with a random selection inside each stratum. This method of stratified random sampling in multiple dimensions can be easily applied to generate a well-distributed sequence. The LHS technique involves drawing a sample of size *N* from multiple dimensions such that for each

---

[2] Sandor and Train (*10*) perform a comparison of four different kinds of (t,m,s)-nets, the standard Halton, and random-start Halton sequences against simple random draws. They estimate a 5-dimensional mixed logit model using 64 QMC draws per observation, and compare the bias, standard deviation and RMSE associated with the estimated parameters. In this study we have conducted numerical experiments both in 5 and 10 dimensions in order that the comparisons may capture the effects of dimensionality. For the 5-dimensional mixed logit estimation problem, we also examined the impact of varying number of draws (25, 125 and 625). Finally, we examine the performance of the Faure sequence and LHS method, along with the Halton sequence, and consider different scrambling variants of these sequences.

individual dimension the sample is maximally stratified. A sample is said to be maximally stratified when the number of strata equals the sample size N, and when the probability of falling in each of the strata equals N^1.

An LHS sequence of size N in K dimensions is given by

$$\psi_{lhs}^{(N)} = ((p - \xi) / N), \tag{1}$$

where, $\psi_{lhs}^{(N)}$ is an NxK matrix consisting of N draws of a K-dimensional LHS sequence; p is an NxK matrix consisting of K different random permutations of the numbers $1,\ldots,N$; ξ is an NxK matrix of uniformly-distributed random numbers between 0 and 1; and the K permutations in p and the NK uniform variates ξ_{ij} are mutually independent.

In essence, the LHS sequence is obtained by slightly shifting the elements of an SRS sequence, while preserving the ranks (and rank correlations) of these elements, to achieve maximal stratification. For instance, in a 2-dimensional LHS sequence of 6 (N) points, each of the six equal strata in either dimension will contain exactly one point [see Sivakumar et al. for an illustration (*12*)].

3.2. QMC Sequences

QMC sequences are essentially deterministic sets of low-discrepancy points that are generated to be evenly distributed over the integration space. Many of the low-discrepancy sequences in use today are linked to the van der Corput sequence, which was originally introduced for dimension $s = 1$ and base $b = 2$. Sequences based on the van der Corput sequence are also referred to as the reverse radix-based sequences. To find the nth term, x_n, of a van der Corput sequence, we first write the unique digit expansion of n in base b as:

$$n = \sum_{j=0}^{\infty} a_j(n)b^j, \text{ where } 0 \le a_j(n) \le b-1 \text{ and } b^J \le n \le b^{J+1}. \tag{2}$$

This is a unique expansion of n that has only finitely many non-zero coefficients $a_j(n)$. The next step is to evaluate the radical inverse function in base b, which is defined as

$$\phi_b(n) = \sum_{j=0}^{\infty} a_j(n)b^{-j-1}. \tag{3}$$

The van der Corput sequence in base b is then given by $x_n = \phi_b(n)$, for all $n \ge 0$. This idea that the coefficients of the digit expansion of an increasing integer n in base b can be used to define a one-dimensional low-discrepancy sequence inspired Halton to create an s-dimensional low-discrepancy Halton sequence by using s van der Corput sequences with relatively prime bases for the different dimensions.

An alternative approach to the generation of low-discrepancy sequences is to start with points placed into certain equally sized volumes of the unit cube. These fixed length sequences are referred to as (*t,m,s*)-nets, and related sequences of indefinite lengths are called (*t,s*)-sequences. Sobol suggested a multidimensional (*t,s*)-sequence using base 2, which was further developed by Faure who suggested alternate multidimensional (*t,s*)-sequences with base $b \ge s$. For a detailed description of the various QMC sequences see Niederreiter (*13*).

The following sub-sections describe the procedures used in this paper to generate the standard Halton and Faure sequences.

3.2.1. Halton Sequences

The standard Halton sequence in s dimensions is obtained by pairing s one-dimensional van der Corput sequences based on s pairwise relatively prime integers, $b_1, b_2, ..., b_s$ (usually the first s primes) as discussed earlier. The Halton sequence is based on prime numbers, since the sequence based on a non-prime number will partition the unit space in the same way as each of the primes that contribute to the non-prime number. Thus, the nth multidimensional point of the sequence is as follows:

$$\phi(n) = (\phi_{b_1}(n), \phi_{b_2}(n), ..., \phi_{b_s}(n)). \tag{4}$$

The standard Halton sequence of length N is finally obtained as

$$\psi_h^{(N)} = [\phi(1)', \phi(2)', ..., \phi(n)']'. \tag{5}$$

The Halton sequence is generated number-theoretically as described above rather than randomly and so successive points of the sequence "know" how to fill in the gaps left by earlier points, leading to a more even distribution within the domain of integration than the randomly generated LHS sequence [see Sivakumar et al. for an illustration (12)].

3.2.2. Faure Sequences

The standard Faure sequence is a (t,s)-sequence designed to span the domain of the s-dimensional cube uniformly and efficiently. In one dimension, the generation of the Faure sequence is identical to that of the Halton sequence. In s dimensions, while the Halton sequence simply pairs s one-dimensional sequences generated by the first s primes, the higher dimensions of the Faure sequence are generated recursively from the elements of the lower dimensions. So if b is the smallest prime number such that $b \geq s$ and $b \geq 2$, then the first dimension of the s-dimensional Faure sequence corresponding to n can be obtained by taking the radical inverse of n to the base b:

$$\phi_b^1(n) = \sum_{j=0}^{J} a_j^1(n) b^{-j-1} \tag{6}$$

The remaining dimensions are found recursively. Assuming we know the coefficients $a_j(n)$ corresponding to the first $(k-1)$ dimensions, the coefficients for the kth dimension are generated as follows:

$$a_j^k(n) = \sum_{i \geq j}^{J} {}^iC_j a_i^{k-1}(n) \bmod b, \tag{7}$$

where ${}^iC_j = i! / j!(i-j)!$ is the combinatorial function. Thus the next level of coefficients required for the kth element in the s-dimensional sequence is obtained by multiplying the coefficients of the $(k-1)^{th}$ element by an upper triangular matrix C with the following elements.

$$C = \begin{bmatrix} {}^0C_0 & {}^1C_0 & {}^2C_0 & {}^3C_0 & \cdots \\ 0 & {}^1C_1 & {}^2C_1 & {}^3C_1 \\ 0 & 0 & {}^2C_2 & {}^3C_2 \\ 0 & 0 & 0 & {}^3C_3 \\ \vdots \end{bmatrix}$$

These new coefficients $a_j^k(n)$ are then reflected about the decimal point to obtain the kth element as follows:

$$\phi_b^k(n) = \sum_{j=0}^{J} a_j^k(n)b^{-j-1}, \ 2 \le k \le s \tag{8}$$

This recursive procedure generates the s points corresponding to the integer n in the Faure sequence based on $b \ (\ge s)$. Thus the nth multidimensional point in the sequence is

$$\phi(n) = (\phi_b^1(n), \phi_b^2(n), ..., \phi_b^S(n))$$

The standard Faure sequence of length N is then obtained in the same manner as the standard Halton sequence:

$$\psi_f^{(N)} = [\phi(1)', \phi(2)', ..., \phi(n)']' \tag{9}$$

Faure sequences are essentially (t,m,s)-nets in any prime b with $b \ge s$ and $t = 0$. A Faure sequence of b^m points is generated to be evenly distributed over the integration space, such that if we plot the sequence in the integration space together with the elementary intervals of area b^{-m}, exactly one point will fall in each elementary interval [see Ökten and Eastman (*14*)]. It must be noted, however, that in order to obtain an even distribution of points over an s-dimensional integration space a Faure sequence of b^m (b prime, $b \ge s$, $m = 1, 2, 3\ldots$) points is required.

Earlier studies have shown that for higher dimensions, the properties of the Faure sequence are poor for small values of n in Equation 9 (*15*). We overcome this in our study by dropping the first 100,000 multidimensional points for all the standard and scrambled Faure sequences generated.

3.3. Scrambling Techniques Used With QMC Sequences

Research has shown that finite parts (for moderate sizes) in higher dimensions of many QMC sequences have poor properties, which can be alleviated using suitable scrambling techniques. The standard Halton sequence, for instance, suffers from significant correlations between the radical inverse functions at higher dimensions. For example, the fourteenth dimension (corresponding to the prime number 43) and the fifteenth dimension (corresponding to the prime number 47) consist of 43 and 47 increasing numbers, respectively. This generates a correlation between the fourteenth and fifteenth coordinates of the Halton sequence as illustrated in Figure 1a. The standard Faure sequence, on the other hand, forms distinct patterns in higher dimensions that also cover the unit integration space in diagonal strips, thus showing significantly higher discrepancies in the higher dimensions. Figure 1b illustrates this in a plot of the fifteenth and sixteenth coordinates of the Faure sequence.

Several scrambling techniques have been suggested to redistribute the points and thus improve the uniformity of the QMC sequences in higher dimensions. In this study, we have implemented the Braaten-Weller scrambling for Halton sequences, and the Random Digit and Random Linear scrambling for Faure sequences. Each of these methods is described in greater detail in the following sections.

3.3.1. Braaten-Weller Scrambling

Braaten and Weller (*16*), describe a permutation of the coefficients $a_j(n)$ in Equation 3 that minimizes the discrepancy of the resulting scrambled Halton sequence. Their method suggests different permutations for different prime numbers, thus effectively breaking the correlation across dimensions. Braaten and Weller have also proved that their scrambled sequence retains the theoretically appealing N^{-1} order of integration error of the standard Halton sequence.

Figure 2a presents the Braaten-Weller scrambled Halton sequence in the fourteenth and fifteenth dimensions. The effectiveness of this scrambling technique in breaking correlations is evident from a comparison of Figures 1a and 2a.

To illustrate the Braaten-Weller scrambling procedure, take the 5th number in base 3 of the Halton sequence, which in the digitized form is 0.21 (or $\frac{7}{9}$). The suggested permutation for the coefficients $(0, 2, 1)$ for the prime 3 is $(0, 1, 2)$, which when expanded in base 3 translates to $1 \times 3^{-1} + 2 \times 3^{-2} = \frac{5}{9}$. The first 8 numbers in the standard Halton sequence corresponding to base 3 are $\left(\frac{1}{3}, \frac{2}{3}, \frac{1}{9}, \frac{4}{9}, \frac{7}{9}, \frac{2}{9}, \frac{5}{9}, \frac{8}{9} \right)$. The Braaten-Weller scrambling procedure yields the following scrambled sequence: $\left(\frac{2}{3}, \frac{1}{3}, \frac{2}{9}, \frac{8}{9}, \frac{5}{9}, \frac{1}{9}, \frac{7}{9}, \frac{4}{9} \right)$.

3.3.2. Random Digit Scrambling

The Random Digit scrambling approach for Faure sequences is conceptually similar to the Braaten-Weller method, and suggests random permutations of the coefficients $a_j^k(n)$ to scramble the standard Faure sequence [see Matoušek for a description (17)].

Figure 2b presents the Random Digit scrambled Faure sequence in the fifteenth and sixteenth dimensions. A comparison of Figures 1b and 2b indicates that the Random Digit scrambling technique is very effective in breaking the patterns in higher dimensions and generating a more even distribution of points.

The Random Digit scrambling technique uses independent random permutations for each coefficient in each dimension of the sequence. For example, consider the following 5-dimensional Faure sequence,

$$\{\{(2,1,0), (2,3,1), (2,4,2), (4,2,3), (1,0,4)\}, \{(1,0,0), (3,2,1), (0,2,4), (0,4,4), (4,4,0)\}\}.$$

In each of the 5 dimensions, the vector's base 5 expansion has 3 digits, which implies that we need 15 independent random permutations $\pi = (\pi_1, \ldots, \pi_{15})$. π_1, for example, could be the following permutation

$$\pi_1(0) = 4; \pi_1(2) = 0; \pi_1(3) = 1; \pi_1(4) = 3.$$

So when all 15 permutations are applied to the sequence, we obtain the scrambled Faure sequence as follows:

$$\{\{(\pi_1(2), \pi_2(1), \pi_3(0)), (\pi_4(2), \pi_5(3), \pi_6(1)), (\pi_7(2), \pi_8(4), \pi_9(2)),$$
$$(\pi_{10}(4), \pi_{11}(2), \pi_{12}(3)), (\pi_{13}(1), \pi_{14}(0), \pi_{15}(4))\},$$
$$\{(\pi_1(1), \pi_2(0), \pi_3(0)), (\pi_4(3), \pi_5(2), \pi_6(1)), (\pi_7(0), \pi_8(2), \pi_9(4)),$$
$$(\pi_{10}(0), \pi_{11}(4), \pi_{12}(4)), (\pi_{13}(4), \pi_{14}(4), \pi_{15}(0))\}\}.$$

3.3.3. Random Linear Scrambling

The Random Linear Scrambling technique for Faure sequences proposed by Matoušek is based on the concept of cleverly introducing randomness in the recursive procedure of generating the coefficients for each successive dimension (17).

Figure 2c presents the Random Linear scrambled Faure sequence in the fifteenth and sixteenth dimensions. A comparison of Figures 1b and 2c indicates that the Random Linear

scrambling method results in a much more even distribution of points in the fifteenth and sixteenth coordinates than the Random Digit scrambling method (Figure 2b)[3].

In the Random Linear scrambling approach of Matoušek is easily implemented by modifying the upper triangular combinatorial matrix *C* used in generating Faure sequences (see Section 3.2.2). A linear combination *AC+B* is used in the place of the matrix C, where *A* is a randomly generated matrix and B is a random vector, both consisting of uniform random variates *U*[*0, b*–1].

3.4. Randomization of QMC Sequences

QMC sequences, such as the standard Halton sequence described in Section 3.2, are fundamentally deterministic and do not permit the practical estimation of integration error. Since a comparison of the performance of these sequences necessitates the computation of simulation variances and errors, it is necessary to randomize these QMC sequences. Randomization of QMC sequences is a technique that introduces randomness into a deterministic QMC sequence while preserving the equidistribution property of the sequence [see Shaw (*18*); Tuffin (*19*)].

In the numerical experiments in this paper, we use Tuffin's randomization procedure [see Bhat (*20*) for a detailed explanation of the randomization procedure] to perform 20 estimation runs for each test scenario. The results of these 20 estimation runs are used to compute the relevant statistical measures.

3.5. Generation of Draws With and Without Scrambling Across Observations

The simulated maximum likelihood estimation of an MMNL with a *K*-dimensional mixing distribution involves generating a *K*-dimensional PMC or QMC sequence for a specified number of draws '*N*' for each individual in the dataset. Therefore estimating an MMNL model on a dataset with *Q* individuals will require an *N×Q* *K*-dimensional PMC or QMC sequence, where each set of *N* *K*-dimensional points computes the contribution of an individual to the log-likelihood function. A PMC or QMC sequence of length *N×Q* can be generated either as one continuous sequence of length *N×Q* or as *Q* independent sets of length *N* each. In the case of PMC sequences, both these approaches amount to the same since a PMC sequence is identical to a random sequence with each point of the sequence being independent of all the previous points. In the case of QMC sequences, *Q* independent sets of length *N* can be generated by first constructing a sequence of length *N* and then scrambling it *Q* times, which is known as generation *with scrambling across observations*. The other alternative of generating a continuous QMC sequence of length *N×Q* is known as generation *without scrambling across observations*. QMC sequences generated with and without scrambling across observations exhibit different properties [see Train (*21*); Bhat (*20*); Sivakumar et al. (*12*)].

In this study we examine the performance of the various scrambled and standard QMC sequences generated both with and without scrambling across observations.

3.6. Box-Muller and Inverse Normal Transforms

The standard and scrambled versions of the Halton and Faure sequences, and the LHS sequence are generated to be uniformly distributed over the multidimensional unit cube. Simulation

---

[3] The behavior of the Random Linear scrambling technique seemed to not always be predictable in terms of uniformity of coverage. In particular, the results of the Random Linear scrambling method for the nineteenth and twentieth dimensions of the Faure sequence were observed to be rather poor as the redistribution of points occurs in a fixed pattern.

applications, however, may require these sequences to take on other distributional forms. For example, the estimation of the MMNL model described in Section 4 of this paper requires normally-distributed multivariate sequences that span the multidimensional domain of integration. The transformation of the uniformly-distributed LHS and QMC sequences to normally-distributed sequences can be achieved using either the inverse standard normal distribution or one of the many approximation procedures discussed in the literature, such as the Box-Muller Transform, Moro's method and Ramberg and Schmeiser approximation. In this paper we compare the performances of the inverse normal and the Box-Muller transforms.

If Y is a K-dimensional matrix of length $N*Q$ containing the uniformly-distributed LHS or QMC sequence, the inverse normal transformation yields $X = \Phi^{-1}(Y)$, where X is a normally-distributed sequence of points in K-dimensions. The Box-Muller method approximates this transformation as follows. The uniformly-distributed sequence of points in Y are transformed to the normally-distributed sequence X using the equations

$$X_{ij} = \cos(2\pi Y_{i(j+1)})\sqrt{-2\log Y_{ij}} \text{ and } X_{i(j+1)} = \sin(2\pi Y_{i(j+1)})\sqrt{-2\log Y_{ij}} , \tag{10}$$

for all $i = 1, 2, \ldots N*Q$, and $j = 1, 3, 5, \ldots K-1$, assuming that K is even. If K is odd, then we simply generate an extra column of the sequence and perform the Box-Muller transform with the $K+1$ even columns. The $(K+1)^{\text{th}}$ column of the transformed matrix X can then be dropped.

4. EVALUATION FRAMEWORK

We evaluate the performance of the sequences discussed earlier in the context of the simulated maximum likelihood estimation of an MMNL model using simulated datasets. This section describes in detail the evaluation framework used in our numerical experiments. All the numerical experiments in this study were implemented using the GAUSS matrix programming language.

4.1. Simulated Maximum Likelihood Estimation of the MMNL Model

In the numerical experiments in this paper, we use a random-coefficients interpretation of the MMNL model structure. However, the results from these experiments can be generalized to any model structure with a mixed logit form. The random-coefficients structure essentially allows heterogeneity in the sensitivity of individuals to exogenous attributes. The utility that an individual q associates with alternative i is written as:

$$U_{qi} = \beta_q' x_{qi} + \varepsilon_{qi} \tag{11}$$

where, x_{qi} is a vector of exogenous attributes, β_q is a vector of coefficients that varies across individuals with density $f(\beta)$, and ε_{qi} is assumed to be an independently and identically distributed (across alternatives) type I extreme value error term. With this specification, the unconditional choice probability of alternative i for individual q, P_{qi}, is given by the following mixed logit formula:

$$P_{qi}(\theta) = \int_{-\infty}^{\infty} L_{qi}(\beta) f(\beta \mid \theta) d(\beta), \quad L_{qi}(\beta) = \frac{e^{\beta' x_{qi}}}{\sum_j e^{\beta' x_{qj}}} , \tag{12}$$

where, β represents parameters which are random realizations from a density function $f(.)$ called the mixing distribution, and θ is a vector of underlying moment parameters characterizing $f(.)$. While several density functions may be used for $f(.)$, the most commonly used is the normal distribution with θ representing the mean and variance.

The objective of simulated maximum likelihood inference is to estimate the parameters 'θ' by numerical evaluation of the choice probabilities for all the individuals using simulation. Using 'N' draws from the mixing distribution $f(.)$, each labeled β^n, $n = 1,…,N$, the simulated probability for an individual can be calculated as

$$SP_{qi}(\theta) = (1/N) \sum_{n=1,…,N} L_{qi}(\beta^n). \qquad (13)$$

$SP_{qi}(\theta)$ has been proved to be an unbiased estimate of $P_{qi}(\theta)$ whose variance decreases as the number of draws 'N' increases. The simulated log-likelihood function is then computed as

$$SLL(\theta) = \sum_{q=1,…,Q} \ln(SP_{qi}(\theta)), \qquad (14)$$

where i is the chosen alternative for individual q. The parameters 'θ' that maximize the simulated log-likelihood function are then calculated. Properties of this estimator have been studied, among others, by Lee (*22*) and Hajivassiliou and Ruud (*23*).

4.2. Experimental Design

The data for the numerical experiments conducted in this study were generated by simulation. Two sample data sets were generated containing 2000 observations (or individuals q in Equation 11) and four alternatives per observation. The first data set was generated with 5 independent variables to test the performance of the sequences in 5 dimensions. The values for each of the 5 independent variables for the first two alternatives were drawn from a univariate normal distribution with mean 1 and standard deviation of 1, while the corresponding values for each independent variable for the third and fourth alternatives were drawn from a univariate normal distribution with mean 0.5 and standard deviation of 1. The coefficient to be applied to each independent variable for each observation was also drawn from a univariate normal distribution with mean 1 and standard deviation of 1 (*i.e.*, $\beta_{qi} \sim N(1,1)$, $q = 1,2,…,2000$ and $i = 1,…,4$). The values of the error term, ε_{qi}, in Equation 11 were generated from a type I extreme value (or Gumbel) distribution, and the utility of each alternative was computed. The alternative with the highest utility for each observation was then identified as the chosen alternative. The second data set was generated similarly but with 10 independent variables to test the performance of the sequences in 10 dimensions.

4.3. Test Scenarios

This study uses the simulated datasets described above to numerically evaluate the performance of the LHS sequence, and the standard and scrambled versions of the Halton and Faure sequences within the MMNL framework. We first estimated random-coefficients mixed logit models, in 5 and 10 dimensions, using a simulated estimation procedure with 20,000 random draws (N = 20,000 in Equation 13). The resulting estimates were declared to be the "true" parameter values. We then evaluated the various sequences by computing their abilities to recover the "true" model parameters. This technique has been used in several simulation-related studies in the past [see Bhat (*4*); Hajivassiliou et al. (*24*)].

We tested the performance of the standard Halton, Braaten-Weller scrambled Halton, standard Faure, Random Digit Scrambled Faure, Random Linear Scrambled Faure, and LHS sequences. For each of these six sequences we tested cases with 25, 125 and 625 draws (N in Equation 13) for 5 dimensions and with 100 draws for 10 dimensions. The number of draws in the test cases was limited by the requirement that Faure sequences must contain b^m points

(b prime, $b \geq s$, s = number of dimensions, m = 1,2,3…). So in 5 dimensions we can generate Faure sequences in base b = 5 containing 25, 125, 625, 3125 points and so on.

5. COMPUTATIONAL RESULTS

The estimation of the 'true' parameter values served as the benchmark to compare the performances of the different sequences. The performance evaluation of the various sequences was based on their ability to recover the true model parameters accurately. Specifically, the evaluation of the proximity of estimated and true values was based on two performance measures: (a) root mean square error (RMSE), and (b) mean absolute percentage error (MAPE). Further, for each performance measure we computed two properties: (a) bias, or the difference between the mean of the relevant values across the 20 runs and the true values, and (b) total error, or the difference between the estimated and true values across all runs[4].

One general note before we proceed to present and discuss the results. The Box-Muller transform method to translate uniformly-distributed sequences to normally-distributed sequences resulted in higher bias and total error than the inverse normal transform method almost universally for all the scenarios we tested [this is consistent with the finding of Tan and Boyle, (*25*)]. In this paper we therefore present only the results of the inverse transform procedure to save on space (the Box-Muller results are available from the authors).

The computational results are divided into four tables (Tables 1a-1d), one each for 25, 125, 625 (5 dimensions) and 100 draws (10 dimensions). In each table, the first column specifies the type of sequence used. The second column indicates whether the sequence is generated with or without scrambling across observations. The remaining columns list the RMSE and MAPE performance measures for the estimators in each case.

In the following sections we first examine and interpret the results separately for each of the 25 draws, 125 draws, 625 draws and 100 draws cases; and then finally examine the overall trends in the results.

5.1. 5 Dimensions and 25 Draws

Table 1a indicates that the standard and scrambled Halton sequences generated with scrambling across observations yield lower RMSE and MAPE bias and total error than the corresponding sequences generated without scrambling across observations. A similar result holds for the standard Faure sequence. However, for the scrambled Faure sequences, the sequences without scrambling across observations yield about equal or lower RMSE and MAPE total error than the sequences with scrambling across observations.

Overall, we can make the following inferences regarding the performance of the sequences in 5 dimensions and with 25 draws:

(a) The standard Halton sequence yields lower RMSE and MAPE bias and total errors than the Braaten-Weller scrambled Halton sequence.

(b) The standard Faure sequence also yields lower RMSE and MAPE bias and total errors than the corresponding scrambled versions.

(c) The standard Faure sequence performs better than the corresponding standard Halton sequence on all counts. The LHS sequence performs at about the same level as all other sequences except the standard Faure.

---

[4] We also computed the simulation variance, *i.e.*; the variance in relevant values across the 20 runs and the true values. However, we chose not to discuss the results of those computations here in order to simplify presentation and also because the total error captures simulation variance.

(d) The standard Faure sequence with scrambling across observations provides the best results in the overall.

5.2. 5 Dimensions and 125 Draws
Table 1b indicates that, for the standard Halton sequence, the case without scrambling across observations provides lower bias but slightly higher total error. For the scrambled Halton, the case without scrambling across observations dominates (this latter result is the reverse of what we found in the 25 draws case). For the Faure sequences, no scrambling across observations provides better results than scrambling across observations for the standard and Random Digit scrambled Faure versions. However, the reverse is the case for the Random Linear Faure sequence.

Overall, we can make the following inferences regarding the performance of the sequences in 5 dimensions and with 125 draws:
   (a) The Braaten-Weller scrambled Halton sequence, in general, does better than the standard Halton, a reversal from the case with 25 draws.
   (b) The Braaten-Weller scrambled Halton sequence without scrambling across observations is the "winner" across all standard and scrambled Halton sequences.
   (c) The scrambled versions of the Faure sequence perform better than the standard Halton, the scrambled Halton, and the standard Faure sequences.
   (d) The Random Linear scrambled Faure sequence with scrambling across observations performs the best in terms of total error. In terms of bias, the Random Digit scrambled Faure sequence without scrambling across observations performs the best, although the Random Linear scrambled sequence with scrambling across observations comes a close second.
   (e) The LHS yields the highest bias and total error across all the sequences

5.3. 5 Dimensions and 625 Draws
From Table 1c, we observe that the standard and scrambled Halton sequences yield lower bias and total error when they are generated with scrambling across observations. The same result also extends to the standard Faure and Random Linear scrambled Faure sequences, but the case without scrambling across observations does better than with scrambling across observations for the Random Digit scrambled Faure.

The following inferences can be made regarding the overall performance of the sequences in 5 dimensions and with 625 draws:
   (a) The Braaten-Weller scrambled Halton does better than the standard Halton in terms of bias. But in terms of total error, the Braaten-Weller scrambled Halton is better than the standard Halton only for the case without scrambling across observations.
   (b) Curiously, the standard Halton with scrambling across observations does the best among the many Halton sequences in terms of total error. However, the Braaten-Weller scrambled Halton with scrambling across observations does almost as well.
   (c) The standard Faure performs better than the scrambled versions in terms of total error. However, the bias associated with the standard Faure is generally higher than the best alternatives among the scrambled Faure sequences. Among the scrambled Faure sequences, the Random Digit scrambled Faure without scrambling across observations has the lowest bias and total error values.

(d) All the Faure sequences clearly perform better than the Halton sequences in terms of yielding lower bias and total error.

(e) The LHS shows the worst performance across all test scenarios, with the highest bias and total error.

(f) The standard and scrambled Faure sequences exhibit the best performance. While it is not possible to clearly pick a "winner" among the many Faure sequences, we do note that the Random Digit scrambled Faure without scrambling across observations has the lowest bias among all the sequences. The standard Faure yields the lowest total error across all the alternatives, but also yields rather high bias values.

5.4. 10 Dimensions and 100 Draws

The results in Table 1d indicate that the standard Halton sequence exhibits a better performance when generated with scrambling across observations, whereas the scrambled Halton sequence performs better when generated without scrambling across observations. The standard and scrambled Faure sequences generally exhibit better performances when they are generated without scrambling across observations.

We draw the following conclusions regarding the overall performance of the sequences from Table 1d:

(a) The standard Halton sequence with scrambling across observations performs better than the standard Halton without scrambling across observations; however, the reverse is the case for the Braaten-Weller scrambled Halton sequence. Overall, the Braaten-Weller scrambled Halton without scrambling across observations appears to do best.

(b) Among the standard and scrambled Faure sequences, the Random Linear scrambled Faure sequence performs better than the Random Digit scrambled Faure sequence, which in turn performs better than the standard Faure sequence.

(c) Interestingly, in 10 dimensions, the LHS sequence performs comparably with the standard Halton sequence.

(d) There is no clear winner in this case. In terms of total error, the Random Linear scrambled Faure sequence without scrambling across observations clearly performs the best. In terms of bias, on the other hand, the Braaten-Weller scrambled Halton without scrambling across observations performs the best. The Random Linear scrambled Faure without scrambling across observations is, however, close on its heels.

5.5. General Trends

The different test scenarios of the QMC sequences in 5 dimensions clearly indicate that a larger number of draws results in lower bias, and total error. However, the margin of improvement decreases as the number of draws increases. The following are other key observations from our analysis.

1. At very low draws, the standard versions of the Halton and Faure sequences perform better than the scrambled versions. However, the bias and total error of the estimates is very high and we strongly recommend against the use of 25 or less draws in simulation estimation.

2. The scrambled versions of both the Halton and Faure sequences perform better than their standard versions at 125 draws (for 5 dimensions) and 100 draws (for 10 dimensions). At 625 draws for 5 dimensions, the standard versions of both the Halton and Faure sequences perform marginally better than their scrambled versions in terms of total error

but yield much higher bias. Overall, using about 100-125 draws with scrambled versions of QMC sequences seems appropriate (though one would always gain by using a higher number of draws at the expense of more computational cost).

3. The Faure sequence generally performs better than the Halton sequence across both 5 and 10 dimensions. The only exception is the case of 100 draws for 10 dimensions, which indicates that, in terms of bias, the Braaten-Weller scrambled Halton sequence without scrambling across observations performs slightly better than the corresponding Random Linear scrambled Faure. However, this difference is marginal and the Random Linear scrambled Faure clearly yields the lowest total error.

4. Among the Faure sequences, the Random Linear and Random Digit scrambled Faure sequences perform better than the standard Faure (except the case with 25 draws for 5 dimensions, which we anyway do not recommend; see point 1 above). However, between the two scrambled Faure versions there is no clear winner.

5. The Random Linear scrambled Faure with scrambling across observations performs better than without scrambling across observations for 5 dimensions (for 125 and 625 draws). For 10 dimensions, the Random Linear scrambled Faure with scrambling across observations performs slightly less well than without scrambling across observations. However, this difference is rather marginal.

6. The Random Digit scrambled Faure performs better when generated without scrambling across observations in all the cases.

7. Overall, our analysis concludes that the Random Linear and Random Digit scrambled Faure sequences are among the most effective QMC sequences for simulated maximum likelihood estimation of the MMNL model. While both the scrambled versions of the Faure sequence perform well in 5 dimensions, the Random Digit scrambled Faure without scrambling across observations performs marginally better. In 10 dimensions, on the other hand, the Random Linear scrambled Faure without scrambling across observations yields the best performance both in terms of bias and total error.

8. Our study also strongly recommends the use of the inverse transform to convert uniform QMC sequences to normally-distributed sequences.

6. SUMMARY AND FUTURE WORK

Simulation techniques have evolved over the years, and the use of quasi-Monte Carlo (QMC) sequences for simulation is slowly beginning to replace pseudo-Monte Carlo (PMC) methods, as the efficiency and faster convergence rates of the low-discrepancy QMC sequences makes them more desirable. There have been several studies comparing the performance of different QMC sequences in the evaluation of a single multidimensional integral. The use of QMC sequences in the simulated maximum likelihood estimation of flexible discrete choice models, which entails the estimation of parameters by the approximation of several multidimensional integrals at each iteration of the optimization procedure, is, however, relatively recent.

In this paper, we have experimentally compared the overall performance of the Halton and Faure sequences, against each other and against the LHS sequence in the context of the simulated likelihood estimation of an MMNL model of choice. We have also compared different scrambled versions of QMC sequences, and observed the effects of scrambling on the accuracy and efficiency of these sequences. In addition, we have compared the efficiency of the QMC sequences generated with and without scrambling across observations. The numerical

experiments were performed in 5 dimensions with 25, 125 and 625 draws, and in 10 dimensions with 100 draws.

The results of our analysis indicate that the Faure sequence consistently outperforms the Halton sequence. The Random Linear and Random Digit scrambled Faure sequences, in particular, are among the most effective QMC sequences for simulated maximum likelihood estimation of the MMNL model.

REFERENCES

1. Morokoff, W. J., and R.E. Caflisch. Quasi-Random Sequences and their Discrepancies. *SIAM Journal of Scientific Computation*, Vol. 15, No.6, 1994, pp. 1251-1279.

2. Morokoff, W. J., and R.E. Caflisch. Quasi-Monte Carlo Integration. *Journal of Computational Physics*, Vol. 122, 1995, pp. 218-230.

3. Revelt, D., and K. Train. Customer-Specific Taste Parameters and Mixed Logit: Household's Choice of Electricity Supplier. *Economics Working Papers E00-274*, Department of Economics, University of California, Berkeley, 2000.

4. Bhat, C. R. Quasi-Random Maximum Simulated Likelihood Estimation of the Mixed Multinomial Logit Model. *Transportation Research Part B*, Vol. 35, 2001, pp. 677-693.

5. Park, Y. H., S. B. Rhee, and E. T. Bradlow. An Integrated Model for Who, When, and How Much in Internet Auctions. *Working Paper*, Department of Marketing, Wharton, 2003.

6. Bhat, C. R., and R. Gossen. A Mixed Multinomial Logit Model Analysis of Weekend Recreational Episode Type Choice. *Transportation Research Part B*, Vol. 38, No. 9, 2004, pp. 767-787.

7. Bratley, P., and B. L. Fox. Implementing Sobol's Quasi-random Sequence Generator. *ACM Transactions on Mathematical Software*, Vol. 14, 1988, pp. 88-100.

8. Kocis, L., and W. J. Whiten. Computational Investigations of Low-Discrepancy Sequences. *ACM Transactions on Mathematical Software*, Vol. 23, No. 2, 1997, pp. 266-294.

9. Wang, X., and F. J. Hickernell. Randomized Halton Sequences. *Mathematical and Computer Modelling*, Vol. 32, 2000, pp. 887-899.

10. Sandor, Z., and K. Train. Quasi-random Simulation of Discrete Choice Models. *Transportation Research Part B*, Vol. 38, 2004, pp. 313-327.

11. McKay, M. D., W. J. Conover, and R. J. Beckman. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, Vol. 21, 1979, pp. 239-245.

12. Sivakumar, A., C. R. Bhat, and G. Okten. Simulation Estimation of Mixed Discrete Choice Models Using Randomized Quasi-Monte Carlo Sequence: A Comparison of Alternative Sequences, Scrambling Methods, and Uniform-to-Normal Variate Transformations. Technical Paper, Department of Civil Engineering, The University of Texas at Austin, 2004.

13. Niederreiter, H. Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia, 1992.

14. Ökten, G., and W. Eastman. Randomized Quasi-Monte Carlo Methods in Pricing Securities. *Journal of Economic Dynamics & Control*, Vol. 28, No. 12, 2004, pp. 2399-2426.

15. Fox, B. L. Implementation and Relative Efficiency of Quasi-random Sequence Generators. *ACM Transactions on Mathematical Software*, Vol. 12, 1986, pp. 362-376.

16. Braaten, E., and G. Weller. An Improved Low-Discrepancy Sequence for Multidimensional Quasi-Monte Carlo Integration. *Journal of Computational Physics*, Vol. 33, 1979, pp. 249-258.

17. Matoušek, J. On the L_2-Discrepancy for Anchored Boxes. *Journal of Complexity*, Vol. 14, 1998, pp. 527-556.

18. Shaw, J. E. H. A Quasirandom Approach to Integration in Bayesian Statistics. *The Annals of Statistics*, Vol. 16, No. 2, 1988, pp. 895-914.

19. Tuffin, B. On the use of Low-Discrepancy Sequences in Monte Carlo Methods. *Monte Carlo Methods and Applications*, Vol. 2, 1996, pp. 295-320.

20. Bhat, C. R. Simulation Estimation of Mixed Discrete Choice Models using Randomized and Scrambled Halton Sequences. *Transportation Research Part B*, Vol. 37, 2003, pp. 837-855.

21. Train, K. Halton Sequences for Mixed Logit. *Working Paper No. E00-278*, Department of Economics, University of California, Berkeley, 1999.

22. Lee, L-F. On Efficiency of Methods of Simulated Moments and Maximum Simulated Likelihood Estimation of Discrete Choice Models. *Econometric Theory*, Vol. 8, 1992, pp. 518-552.

23. Hajivassiliou, V. A., and P. A. Ruud. Classical Estimation Methods for LDV Models using Simulation. In: Engle, R.F. and D.L. McFadden (Eds.) *Handbook of Econometrics*, IV. Elsevier, New York, 1994, pp. 2383-2441.

24. Hajivassiliou, V. A., D. L. McFadden, and P. A. Ruud. Simulation of Multivariate Normal Rectangle Probabilities and their Derivatives: Theoretical and Computational Results. *Journal of Econometrics*, Vol. 72, 1996, pp. 85-134.

25. Tan, K. S., and P. P. Boyle. Applications of Randomized Low Discrepancy Sequences to the Valuation of Complex Securities. *Journal of Economic Dynamics & Control*, Vol. 24, 2000, pp. 1747-1782.

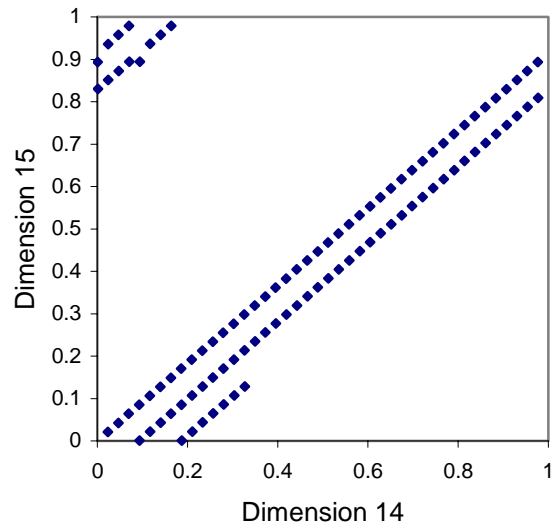**LIST OF FIGURES**

LIST OF TABLES

FIGURE 1a Standard Halton sequence: first 100 points [Source: Bhat (*20*)].
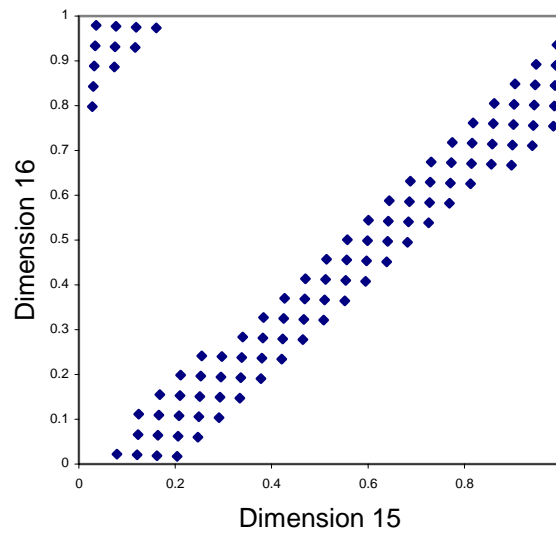

FIGURE 1b Standard Faure sequence: first 100 points.
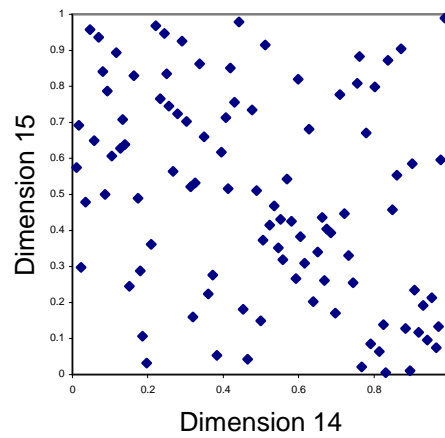
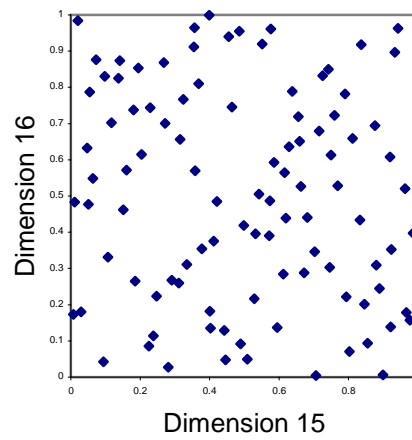**FIGURE 2a Braaten-Weller Scrambled Halton sequence: first 100 points.**

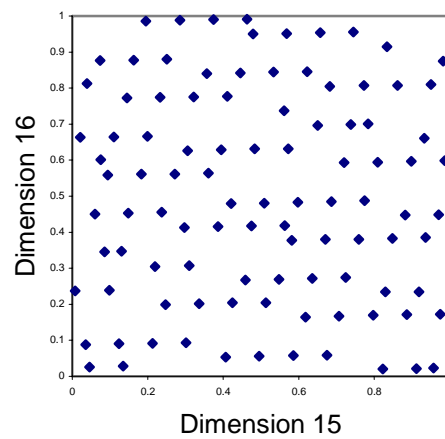**FIGURE 2b Random Digit Scrambled Faure sequence: first 100 points.**

FIGURE 2c Random Linear Scrambled Faure sequence: first 100 points.

TABLE 1a Evaluation of Ability to Recover Model Parameters: 5 Dimensions, 25 Draws

Sequence Type	Scrambling across observations	RMSE		MAPE	
		Bias	Total error	Bias	Total error
Standard Halton	No Scrambling	0.2987	0.3275	30.6976	30.6976
	Scrambling	0.2817	0.2997	29.7409	29.7409
Braaten-Weller Scram. Halton	No Scrambling	0.3157	0.3515	32.5745	32.5745
	Scrambling	0.2948	0.3259	30.4528	30.4544
Standard Faure	No Scrambling	0.2586	0.2869	27.2551	27.2551
	Scrambling	0.2374	0.2887	24.0570	24.0937
Random Digit Scram. Faure	No Scrambling	0.2955	0.3332	28.8420	28.8420
	Scrambling	0.2947	0.3541	29.8144	29.8144
Random Linear Scram Faure	No Scrambling	0.2677	0.2978	27.9082	27.9082
	Scrambling	0.2848	0.3209	29.4035	29.4035
LHS	N/A	0.2650	0.3059	27.7668	27.7668

TABLE 1b Evaluation of Ability to Recover Model Parameters: 5 Dimensions, 125 Draws

Sequence Type	Scrambling across observations	RMSE		MAPE	
		Bias	Total error	Bias	Total error
Standard Halton	No Scrambling	0.0538	0.0672	5.6565	6.0881
	Scrambling	0.0560	0.0627	5.9892	6.0709
Braaten-Weller Scram. Halton	No Scrambling	0.0383	0.0560	4.0664	5.1062
	Scrambling	0.0445	0.0646	4.7313	5.9334
Standard Faure	No Scrambling	0.0393	0.0553	4.1668	4.5773
	Scrambling	0.0455	0.0630	4.8227	5.3210
Random Digit Scram. Faure	No Scrambling	0.0298	0.0489	3.1551	4.2517
	Scrambling	0.0432	0.0563	4.5803	5.0752
Random Linear Scram Faure	No Scrambling	0.0364	0.0534	3.9041	4.4663
	Scrambling	0.0310	0.0450	3.2947	4.1762
LHS	N/A	0.0715	0.0789	7.5294	7.6367

TABLE 1c Evaluation of Ability to Recover Model Parameters: 5 Dimensions, 625 Draws

Sequence Type	Scrambling across observations	RMSE		MAPE	
		Bias	Total error	Bias	Total error
Standard Halton	No Scrambling	0.0088	0.0189	0.8701	1.6096
	Scrambling	0.0065	0.0161	0.6021	1.3830
Braaten-Weller Scram. Halton	No Scrambling	0.0069	0.0177	0.7053	1.5221
	Scrambling	0.0060	0.0170	0.6013	1.4086
Standard Faure	No Scrambling	0.0070	0.0131	0.7148	1.1309
	Scrambling	0.0047	0.0129	0.3596	1.0538
Random Digit Scram. Faure	No Scrambling	0.0025	0.0138	0.2354	1.1987
	Scrambling	0.0059	0.0174	0.5914	1.4629
Random Linear Scram Faure	No Scrambling	0.0049	0.0161	0.4702	1.4698
	Scrambling	0.0035	0.0152	0.3423	1.2542
LHS	N/A	0.0152	0.0311	1.5890	2.7455

TABLE 1d Evaluation of Ability to Recover Model Parameters: 10 Dimensions, 100 Draws

Sequence Type	Scrambling across observations	RMSE		MAPE	
		Bias	Total error	Bias	Total error
Standard Halton	No Scrambling	0.2224	0.2692	26.6145	26.8211
	Scrambling	0.1953	0.2489	23.5067	23.9490
Braaten-Weller Scram. Halton	No Scrambling	0.1681	0.2500	19.8661	21.4625
	Scrambling	0.3297	0.3666	30.2559	30.5939
Standard Faure	No Scrambling	0.1969	0.3114	22.1754	26.5580
	Scrambling	0.2337	0.3068	27.7484	29.8256
Random Digit Scram. Faure	No Scrambling	0.1844	0.2577	21.8181	22.4525
	Scrambling	0.1998	0.2585	24.5396	24.7051
Random Linear Scram Faure	No Scrambling	0.1740	0.2266	20.9043	21.2949
	Scrambling	0.1802	0.2679	20.7861	22.5148
LHS	N/A	0.2213	0.3013	25.6583	26.5579