Drawing a random number

Jørgen Bundgaard Wanscher

Majken Vildrik Sørensen

Kgs. Lyngby 2006

IMM-TECHNICAL REPORT-2006-2

Drawing a random number

Jørgen Bundgaard Wanscher,

Informatics and Mathematical Modelling, DTU, jbw@imm.dtu.dk

Building 321 st, DK-2800 Lyngby, Denmark

Majken Vildrik Sørensen,

Centre for Traffic and Transport, Technical University of Denmark, mvi@ctt.dtu.dk

Building 115 st, DK-2800 Lyngby, Denmark

February 24, 2006

Abstract: Random numbers are used for a great variety of applications in almost any field of computer and economic sciences today. Examples ranges from stock market forecasting in economics, through stochastic traffic modelling in operations research to photon and ray tracing in graphics.

The construction of a model or a solution method requires certain characteristics of the random numbers used. This is usually a distribution classification, which the sequence of random numbers must fulfill; of these some are very hard to fulfill and others are next to impossible. Today mathematics allows us to transform distributions into others with most of the required characteristics. In essence, a uniform sequence which is transformed into a new sequence with the required distribution. The subject of this article is to consider the well known highly uniform Halton sequence and modifications to it. The intent is to generate highly uniform multidimensional draws, which are highly relevant for todays traffic models.

This paper shows among others combined shuffling and scrambling seems needless, that scrambling gives the lowest correlation and that there are detectable differences between random numbers, dependent on their generation.

Keywords: Mixed Logit estimation, *a priori* distribution, random numbers, Halton numbers, Shuffled Halton, Scrambled Halton, Leaped Halton.

1 Introduction and background

A mixed logit model is a generalised member of the logit family, where one/ some of the coefficients are no longer scalars; but rather stochastically distributed variables. Over recent years increased focus has been put on how these variables could be described (which *a priori* distribution should be applied, could this be estimated, could this be tested, etc).

Mixed logit models, ie. models including distributed components are today at the frontier of application and development in transport modelling. The name Mixed Logit has for some years been used indiscriminately with Error Component Models, Random Parameters Logit, Models with Stochastic (Distributed) Preferences (Coefficients), Logit Kernel Models or Hybrid Choice models, for models where error components are added to the traditional (linear) utility function. However, the name conventions Mixed logit models seems to have taken the lead.

During the past 5 years the use of such models is growing rapidly, due to an increased access to software (both commercially by Hague Consulting Group (Alogit4ec), the self-contained BioGeme [Bierlaire, 2002] and shareware as Gauss code [Train, 2002]. In general, the method of Maximum Simulated Likelihood (MSL) is applied, although this only optimises within a given *a priori* distribution of the error components.

When mixed models are estimated or used for forecasting, emphasis must be put on how the stochastically distributed terms are obtained as this may significantly impact the model results. The interesting question of correlation between these error components, or the *a priori* assumption of shape of distribution has been dealt with in [Sørensen, 2003a, Sørensen, 2003b, Sørensen, 2004]. Yet an unresolved questions is that of how to generate the distributed terms.

This paper will focus on *how* random numbers can be generated, and the differences between these. First, the theoretical framework is defined in section 2, different methods for single dimensional draws will be covered in section 3, followed by multidimensional draws in section 4. Following this runtime analysis in section 6 will preceed comparison of the numbers generated by means of correlation in section 7. This paper is concluded in section 8.

2 Theoretical framework

In the following a general model is formulated, for which the empirical distribution of the stochastic coefficients in an error components model can be determined. Let the utility function be specified as

$$U_{ji} = V_{ji} + \nu_j \tag{1}$$

where V_{ji} respective ν_j represents the explained respective the unexplained variation in the utility function for choice of alternative i by individual j. The explained variation is often formulated as a linear relation, i.e. $\sum_k \beta_k X_{jik}$ where each attribute X_k is converted into an arbitrary measure *utility*. This framework is (usually) operationalised by assuming that the unexplained part of the variation is described by some well-known distribution (Extreme Value, type I (EVI) leads to the logit formulation). Estimation is performed by means of maximum likelihood, where the estimates of the coefficients are conditional on the specification of the utility function (which potentially transformed attributes are included) as well as the *a priori* (assumed) distribution of the ν.

The expansion to the mixed logit framework[1] corresponds to partitioning of the unexplained variation (the ν) into a *systematic* (describable) and an *unsystematic* part. Formulated as in (1)

$$U_{ji} = V_{ji}(X_{ij}) + \eta_j f(X_{ij}) + \epsilon_j, \tag{2}$$

where η_j is the systematic part of the unexplained variation, while ϵ_j is the unsystematic part. Both the deterministic part of the utility and the systematic part of the utility may be functions of the explanatory variable (X_{ij}), while the unsystematic part (ϵ) is (again) assumed to be EVI distributed; hence, estimation by means of the logit framework is facilitated.

The elements of the matrix ξ are either assumed to follow a stochastic distribution with mean zero and variance σ_j^2 or are identical 0 (fixed coefficients). Traditionally distributions suggested are the normal ([McFadden and Train, 2000]) and the lognormal ([Ben-Akiva et al., 1993], [Train, 1998]), though other distributions have been suggested which include the χ^2 ([Nunes et al., 2001]), the uniform ([Revelt and Train, 2000]) and the triangular ([Revelt and Train, 2000], [Train, 2001]). Covariance between the stochastic elements of ξ are allowed; though applications has been limited to the normal distribution. The un-systematic parts ϵ are IID.

Key issues are to discover and apply the correct shape of the included distributions - including application of a random number generator that does not bias the distribution.

[1] The 'Mixed Logit Framework' refers to the Mixed Logit Model ([McFadden and Train, 2000]), Error Components Logit ([Ben-Akiva et al., 1993]), Logit Kernel ([Ben-Akiva et al., 2001]), Random parameters Logit ([Ben-Akiva et al., 1993]) and the Hybrid Choice model [Ben, 2002].

3 Drawing a single number

The problem of drawing a single number or a sequence of numbers from a uniform distribution is usually solved by using the random number generator supplied by the computer[2]. Due to the nature of the implementation it is called a pseudo random number generator. The name "pseudo" is chosen because of the relatively high discrepancy, which is a measure of how close to uniform distribution the draws are.

High discrepancy means that the draw is far from uniform and low indicates the draw is close to uniform. This discrepancy can be estimated by selecting a representative subinterval in the range from which the sequence was drawn. The next step is to slide the subinterval through the draw range and for each position where the number of draws in the subinterval change note the count. From the produced set of numbers minimum and maximum is found. The discrepancy can then be described as the difference between these numbers - i.e. if the difference is high so is the discrepancy.

Several approaches to calculating the discrepancy exist. These are interesting if you wish to estimate the discrepancy of a known sequence. In this paper however we consider a special sequence generator, the Halton generator, which by construction yields low discrepancy sequences.

Following the coverage of the pseudo random number draw we cover a widely used quasi random number sequence called the Halton sequence and different variations hereof.

3.1 Pseudo random numbers

A *pseudo random number* is a general name for numbers generated to simulate uniform distribution draws fast. The crucial characteristic is the speed they are produced at, and not their discrepancy, the deviation from uniformity, as these usually are less important. Usually they are generated as

$$X_{t+1} = mod(aX_t + b, c) \tag{3}$$

with proper choices of a, b and c. Further description can be found in eg. [Ross, 1997]. In fact, the discrepancy is the reason why they are insufficient for model estimation.

Figure 1 shows how significant the discrepancy is. Even if the sequence is extended the discrepancy cannot be neglected, as evident from the figure.

The fallacy is that the numbers are based on a predictable stateful draw that sacrifices uniformity and randomness for speed. A newer approach to generating random numbers is based on entropy accummulation and is primarily available under Unix based operating systems. Any bit read from the entropy device is considered to be completely random and highly uniform, which can be translated into: given n-bits from

[2] More specifically by the operating system

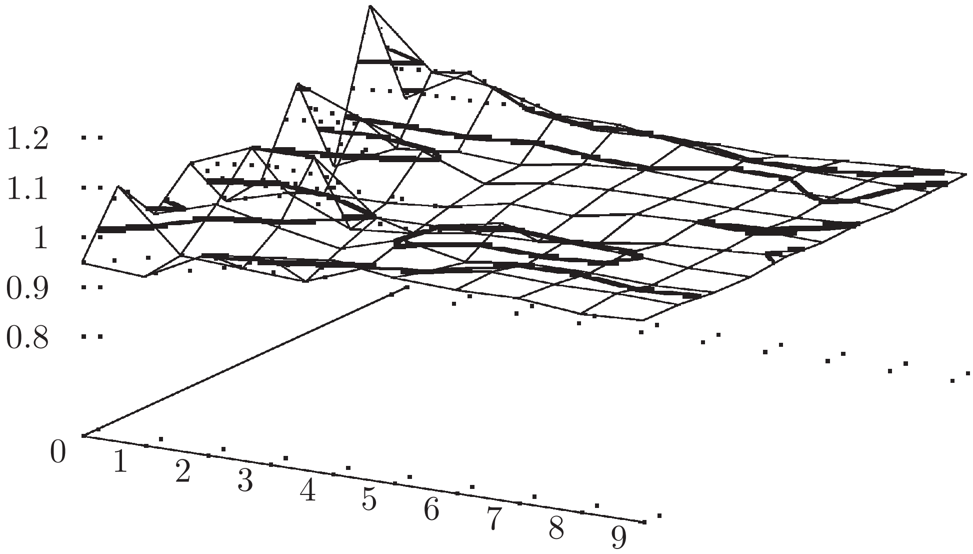


Figure 1: Visualization: discrepancy of the pseudo random draws

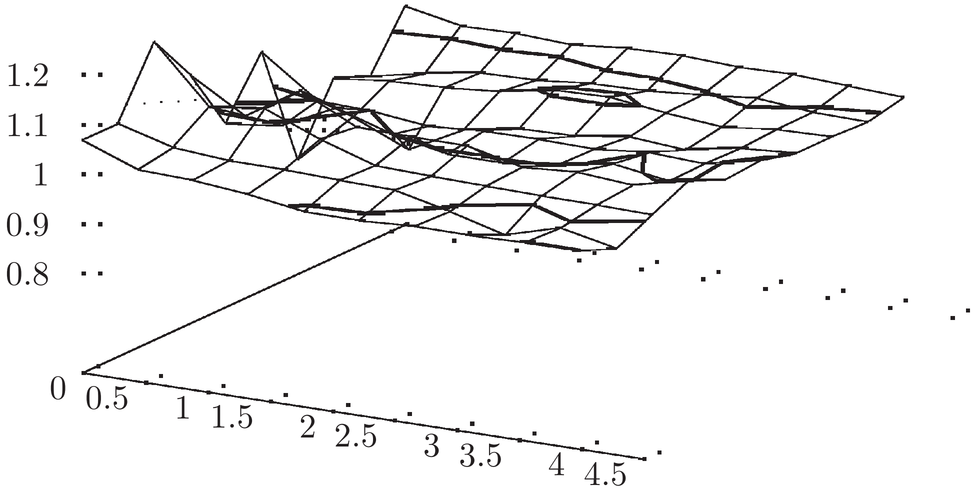


Figure 2: Visualization: Discrepancy of the entropy based random draw.

the stream, the number of zeros n_0 equals the number of ones n_1 as $n = n_0 + n_1$ goes to infinity. As with pseudo random numbers the entropy numbers are based on a stateful approach. The major difference is that the state is changed independently of the draw of a number. This, however, is not enough to guarantee sufficiently low discrepancy.

Figure 2 shows the discrepancy of a 32-bit precision entropy draw. Due to the nature of the entropy generated numbers performance cannot be directly measured, as the time it takes to generate a number is dependent on the entropy of the system. It is thus of no use trying to compare the run times of the two, as the entropy version uses events in the system and hardly any cpu-time as opposed the cpu intense calculations of the standard random draw.

The key point in a sequence generator is not how uniform a single sequence is, but instead the worst discrepancy of all sequences generated by it. The problem in the pseudo draws is that this worst case discrepancy is completely unacceptable. This makes them theoretically useless eventhough practical examples show that many of the generated sequences have a low discrepancy.

The need for a theoretical guarantee has lead to the invention of sequence generators ensuring a very good worst discrepancy. One, among many of the low discrepancy sequence generators, makes the Halton sequences and we thus call it the Halton generator.

Some of the following have already been covered in great detail in [Bhat, 2001] and [Bhat, 2003], but we include it here as we consider a broader selection of Halton based sequences.

3.2 Standard Halton

Rather than ensuring that each number is equally random, the idea is that a sequence of numbers covers the interval uniformingly.

This is where the Halton sequences comes into play. In technical terms is it called a reverse radix-based sequence, where a radical inverse function is used to obtain the point in the interval corresponding to a specific number [Kocis and Whiten, 1997].

Mathematically the n^{th} number in the Halton sequence Φ_b is in [Hess and Polak, 2003] section 3.1 stated as:

$$\Phi_b(n) = \sum_i a_i(n) b^{-i-1}, \text{ where } \sum_{i=0} a_i(n) b^i = n \tag{4}$$

Where b is the number generating the sequence and i is the level in the sequence. The problem is the determination of the $a_i(n)$ variables, and the sequences is thus much better described by an example.

The sequence is constructed by continously subdividing the open ended unit interval $[0, 1[$. The first b numbers – the first level – are simply nb^{-1}, dividing the interval into b equally sized subintervals. The next $b^2 - b$ numbers – the second level – divide each subinterval into b new equal sized intervals. After drawing b^2 numbers the interval is divided into b^2 equal sized intervals, which during the next $b^3 - b^2$ (third level) draws is further subdivided into b^3 intervals, etc.

The sequence based on base number 2, is

$$\left\{0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}, \frac{1}{16} \cdots \right\}$$

The sequence of the numbers is such that it is always the first of the widest subintervals that is divided into two by specific proportions based on b.

A standard Halton sequence is far from random, but it is highly uniform as long as the number of draws completes an entire level. In other words the sequence must be of length b^n for $n \in \mathbb{N}$ to satisfy uniformity. If this is not fulfilled the mean of the sequence is be biased towards zero. This is due to the division strategy, where intervals closest to zero are divided first, according to the definition.

This is further strengthened by the first draw of 0, which is never balanced by a draw of 1. This is easily circumvented by simply discarding the 0 and thus using the sequence from the second element, whereby the mean of the sequence $E(\cdot) = \frac{1}{2}$.

3.3 Scrambled Halton

The scrambled Halton draw was invented to introduce some randomness into the sequences and, as we will show a little later, alleviate correlation in multidimensional draws.

The idea is simply to permute the interval subdivision such that it is not the first of the widest intervals but instead any one of the widest intervals still to be divided in this level, that is split.

The mathematics are given in [Hess and Polak, 2003] section 3.2 as follows:

$$\Phi_b(n) = \sum_i \sigma_b(a_i(n)) b_j^{-i-1}, \text{ where } \sum_{i=0} a_i(n) b^i = n \tag{5}$$

The only difference from equation (4) is the σ_b function which performs a permutation of all integers from 0 to $b - 1$

It is defined above that the permutation function is the same for all levels and given a σ function as $[012] \rightarrow [102]$ the sequence based on 3 becomes:

$$\left\{\frac{1}{3}, 0, \frac{2}{3}, \frac{4}{9}, \frac{1}{9}, \frac{7}{9}, \frac{5}{9}, \frac{2}{9}, \frac{8}{9}, \frac{13}{27} \cdots\right\}$$

Again the sequences are not random, but it is less obvious what the next number is due to the scrambling function σ. It is still highly uniform as the numbers drawn are the same as for the standard Halton, but if a sequence does not complete a level there is no guarantee that the mean $E(\cdot) = \frac{1}{2}$.

The major difference is that the bias caused by incomplete levels is completely dependent on the scrambling function. Choosing a function like $\sigma_5 : [01234] \rightarrow [21304]$ will ensure that the bias is minimized as the interval divisions are performed symmetrically around 0.5.

3.4 Leaped Halton

To increase the degree of randomness the leaped Halton draw was invented. It again is based on the standard Halton but instead of using all numbers only every l^{th} number is used. As long as the number l is a prime different from the prime base b it can be shown that the resulting draw is equally uniform. Refer to section 2.4 of [Kocis and Whiten, 1997] for further elaboration on this.

Furthermore it can be shown that the leaping can be considered as a scrambling and pertubation function that rather effectively removes the apparent predictability of the standard and scrambled Halton draws.

The drawback is the possible computational overhead of drawing $n \times l$ elements, instead of just n elements.

The first 10 numbers of the leaped Halton sequence based on $b = 29$ and $l = 409$ are;

0.08561237 0.20570749 0.45782935 0.94415515 0.43052196
0.91684776 0.40321456 0.88954037 0.37590717 0.86223297

with mean ≈ 0.557.

The problem is here that the leap length is constant, which means that while no numbers from the first level are selected in the example above $(29^7 - 29^6)/409 = 40721400$ numbers from level 7 are selected in strictly circular manner. The permutation and pertubation is thus most influencial when the number of draws is relatively small.

4 Multidimensional draws

When using multidimensional draws it is important that the different dimensions are uncorrelated - or rather, controlled given that correlation is desired. If and only if they are uncorrelated and uniform, then the draw will be uniform in the s-dimensional unit space $[0, 1[^s$.

4.1 Linear draws

The straight forward approach to this is to draw anew for each dimension. As we focus on Halton draws, any of the above can be used. For a 3 dimensional draw we can use three standard Halton sequences each based on a different number b. However, this is not sufficient. Selection of these base numbers are highly important and must be done with great care. The standard Halton sequence is cyclic and inappropriate base numbers introduces high correlation between the dimensions; thus resulting in a non-uniform draw.

Figure 3 shows correlation between 2 standard Halton sequences.

Even choosing the numbers optimally will not result in a uniform multidimensional draw. The cycles of the individual standard Halton sequences will combine into a supercycle, which will result in correlation.

The first or smallest step to overcome the correlation is to use the *scrambled Halton* instead. This disrupts the cycles of the standard version, but again, dependent on the scrambling function, the cycles may reappear in longer sequences. The correlation is significantly reduced, but has not been removed.

Similarly, using the *leaped Halton* might again reduce the correlation, but the computational overhead can become a drawback.

To define the linear draw more strictly we can define the linear s-dimensional draw $\mathbb{L}^s$, constructed by:

$$\mathbb{L}^s(n) = \{\Sigma_1(n), \Sigma_2(n), \ldots, \Sigma_s(n)\}$$

where $\Sigma_i(n)$ is the n^{th} number in the i^{th} sequence, which can be any sequence from a uniform draw.

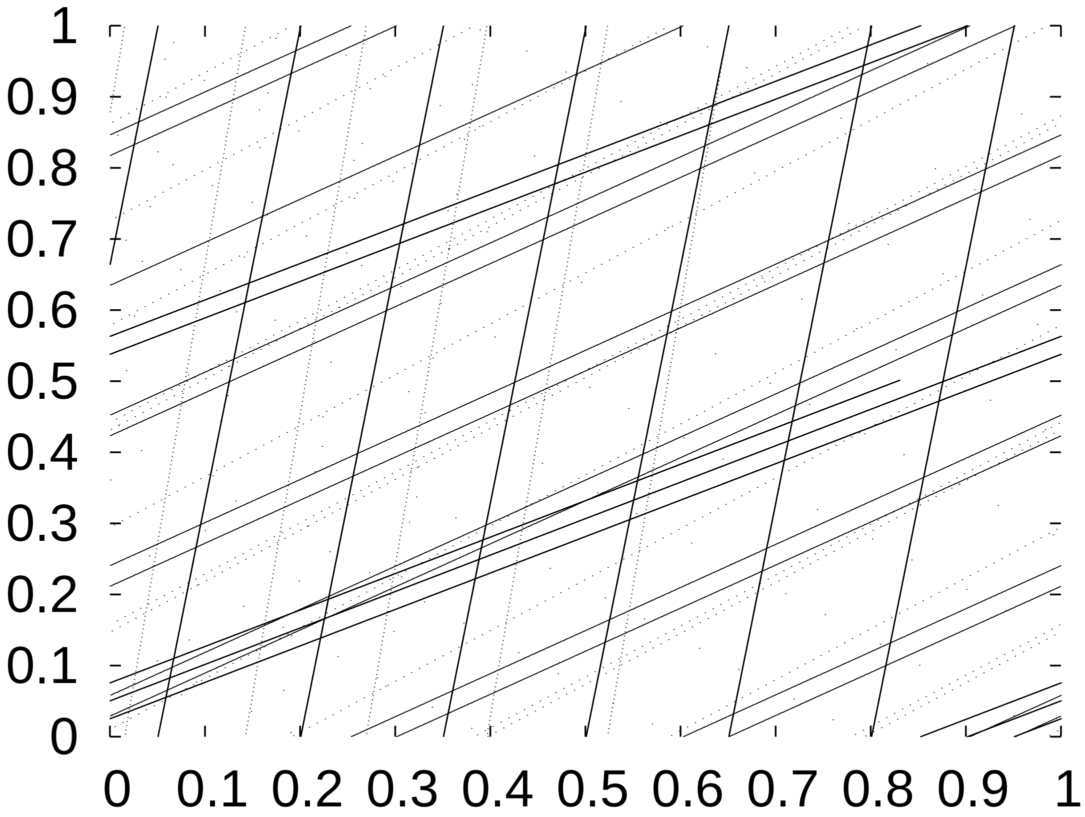


Figure 3: Visualization: The correlation for generator $b_1 = 11$ and $b_2 = 13$

4.2 Shuffled draws

The problem above is that the number drawn for each dimension is drawn from the same sequence in every draw. The idea in shuffled Halton draws is to permute or shuffle the sequences among the dimensions for each draw.

The basic shuffled version proposed lately in [Hess and Polak, 2003] is based on standard Halton sequences shuffled randomly among the dimensions at every draw. This disruption has impact on the correlation as it no longer is exactly two dimensions that are strictly correlated. Instead the correlation might become more or less as the correlated sequences are no longer bound to specific dimensions.

As above, the shuffled draw $\mathbb{S}$ can be described by:

$$\mathbb{S}^s(n) = \sigma_{sn}\left(\Sigma_1(n), \Sigma_2(n), \ldots, \Sigma_s(n)\right)$$

Where σ_{sn} is a function, that dependent on n, permutes the s elements.

A sequence of shufflings could be:

$$\begin{aligned}\sigma_{50}(x_0,x_1,x_2,x_3,x_4) &= (x_2,x_1,x_3,x_0,x_4)\\ \sigma_{51}(x_0,x_1,x_2,x_3,x_4) &= (x_4,x_3,x_1,x_0,x_2)\\ \sigma_{52}(x_0,x_1,x_2,x_3,x_4) &= (x_1,x_4,x_2,x_3,x_0)\end{aligned}$$

5 Obtaining the results

The generation time is trivially acquired by examining the per process timing information upon inaugeration and completion of the actual generation.

The correlation between two sequences are found by:

$$r(X,Y) = \frac{Cov(X,Y)}{S(X) \cdot S(Y)}$$

Where $Cov(\cdot,\cdot)$ is the covariance and $S(\cdot)$ is the standard deviation:

$$Cov(X,Y) = E\left[(X - E(X))(Y - E(Y))\right] \text{ and } S(X) = \sqrt{Cov(X,X)}$$

Two one dimensional sequences were defined as correlated if the numerical value of the correlation exceeded $2N^{-1/2}$, that is:

$$|r(X,Y)| > \frac{2}{\sqrt{N}} \Rightarrow \text{Sequences X and Y are correlated}$$

For the multidimensional draws the results are found by first generating a large set of sequences – 100 or more. These were generated from the primes starting from 2. From this initial set of sequences Operations Research was applied to to solve 2 different optimization problems:

1. Find the fixed subset of sequences (10, 20 or more) with the least correlation between all pairs
2. Find the maximum subset of sequences where none of the sequences are correlated with each other

Both problems are very difficult to solve. We have thus set an upper time limit on the applied solver (cplex). The results are the best found solution within the 8 hour time limit we imposed. For almost all of the small (100 and 200) initial sequence sets the results are optimal. As predicted the 500 sequence sets were exponentially harder and the results are the best found when the time limit was exceeded.

6 Runtime analysis

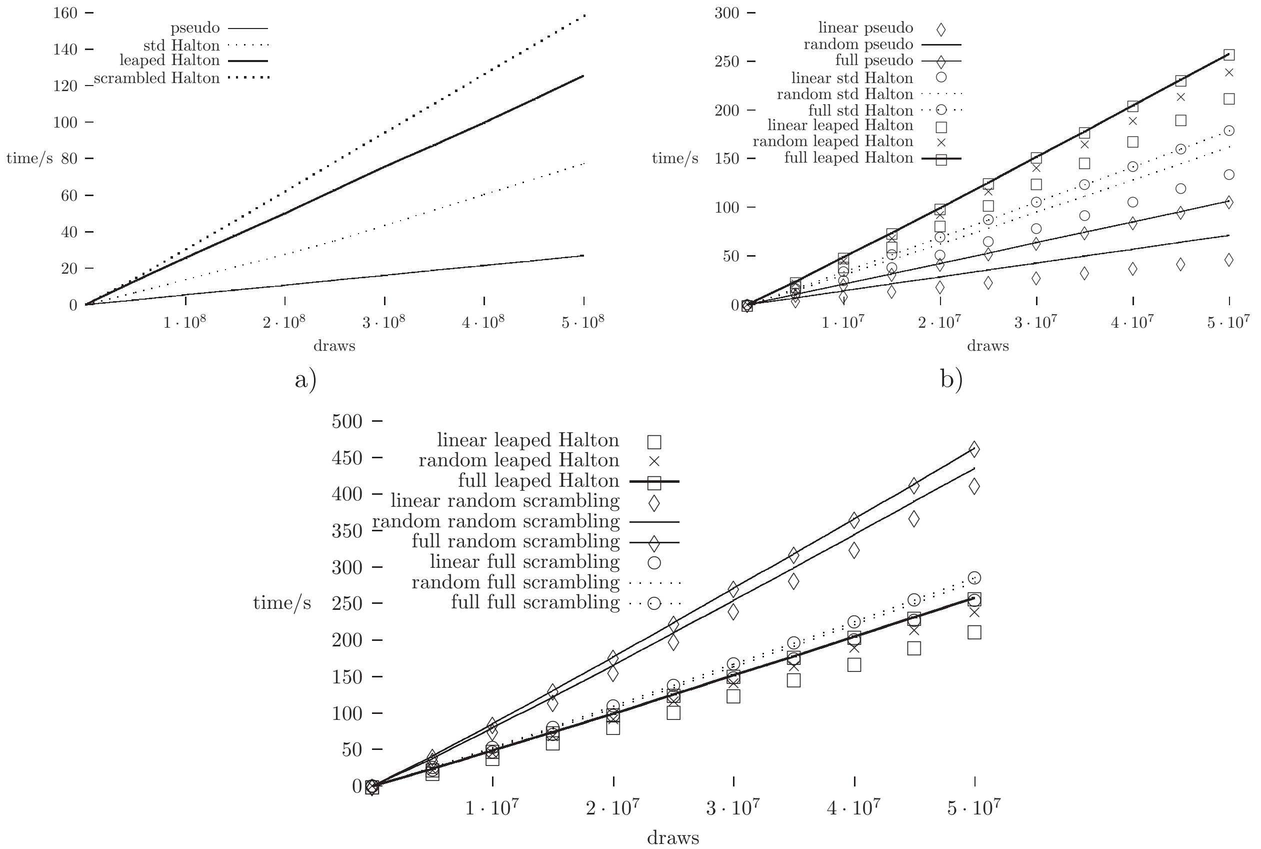
As the basics of pseudo random and Halton sequences have been covered, we now turn to an interesting question. Above the draws were seemingly made more and more perfect, but here we try to shed some light on the implementation performance of the different draws.

As shown in [Bhat, 2001] and [Bhat, 2003] applying the Halton sequences through the Quasi-Monte Carlo method to the mixed logit is highly effective. His research however only uses short sequences (25 to 150 numbers) and few dimensions (1 to 10). It is evident that the models applied in transportation research will become increasingly complex and thus require substantially more from the random sequences used in the estimation of the different models. We thus consider sequences of thousands numbers and up to 500 dimensions as the runtime issue will become even more important. Eventhough the numbers in some cases can be generated a-priori many neglect the use of quasi random numbers due to the extra time it requires. However comparing the runtime analysis in this paper with the effect on MAPE[3] and RMSE[4] in [Bhat, 2001] and [Bhat, 2003] shows that neglecting Halton sequences is a poor disposition.

It must be mentioned that all benchmarks have been optimized and executed on the same uni-processor linux box and the results are thus biased only by the hardware dependent performance of both pseudo random and Halton sequences.

[3] Mean Absolute Percentage Error (MAPE)

[4] Root Mean Square Error (RMSE)



a)

b)

c)

Figure 4: Execution times for a) 1-D draws and b) 5-D draws using different shufflings. c) compares the linear leaped Halton with different shuffling/scrambling combinations (eg full random scrambled means fully shuffled randomly scrambled Halton)

As expected figure 4 a) shows the the pseudo random draw is approximately three times faster than the standard Halton. Leaping adds roughly 60% to the execution time and finally scrambling doubles it.

Inspecting the multidimensional draws in figure 4 b) the impression is much less different. The pseudo random draw is fastest, but the difference is not as significant as with the unidimensional draws.

Adding shuffling to the pseudo random numbers is irrelevant which is why pseudo random numbers are left out in figure 4 c). These compare the impact of adding shuffling to the draw. The "full" refers to full permutation which is based on a slower, but theoretically sound, stateful approach consuming $O(b)$ space where as the "random" refers to a generic permutation based on projection requiring $O(1)$ space.

It can be concluded that the performance impact of the full permutation as opposed to generic permutation, is negligible and that the shuffling adds approximately 10 percent to the generation time of the fully shuffled and scrambled sequence.

7 Correlation

To evaluate the influence of the shuffling, leaping or scrambling on a multidimensional draw several issues have to be considered. First we consider the influence of leaping and scrambling compared to the length of the sequence. After this we will attend to the multidimensional draws and shuffling.

Figure 5 indicates that the correlation generally is reduced as the sequences are extended. The problem is that the reduction is less than n^{-2}, which indicates that the significance of the correlation increases as the sequences get longer.

There is one very important exception from the above rule; which is the illustrations in rows two and four where the leaped sequences have been used. At a specific point the correlation suddenly starts increasing. This is because the sequences were leaped with the same length and that this leap length becomes dominant in the number generation. This increase happens much earlier for the highly correlated pairs than for the less correlated pairs.

The small periodic "bubbles" on the standard graphs are caused by the cyclic nature of the Halton sequence.

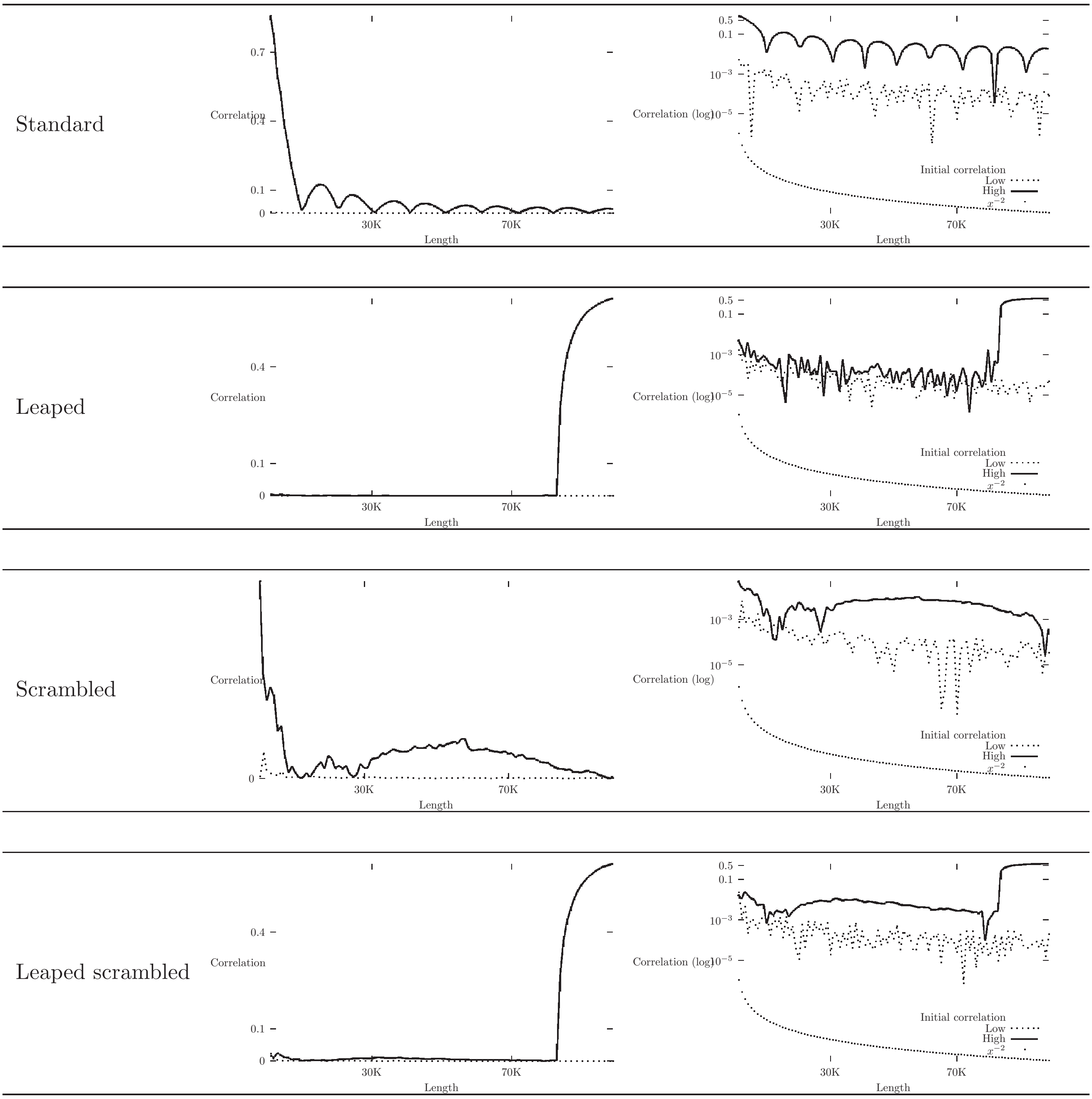


Figure 5: Length and correlation

Dimensions: 200	Selection			
draws: 1000	10	20	30	50
halton-linear	0.00617	~ 0.14617	~ 0.79661	0.96039
halton-shuffled	~ 0.15547	~ 0.19574	~ 0.22030	~ 0.28451
leaped-linear	~ 0.00931	~ 0.04764	~ 0.30843	0.94363
leaped-shuffled	~ 0.06414	~ 0.09291	~ 0.11466	~ 0.14040
scrambled-linear	~ 0.01485	~ 0.04588	~ 0.07084	~ 0.09108
scrambled-shuffled	~ 0.02432	~ 0.04563	~ 0.06466	~ 0.10614

This table summarizes the result of optimization problem 1 with initial set size 200 and sequence length of 1000. It can be seen that the lowest pairwise correlation is found with standard halton sequences with out shuffling for the small set. In all other cases scrambled halton sequences yeild the best result. This was also the case with all other initial set sizes 100, 300, 500 and 1000.

Figure 6: Minimal correlation with preset number of generated sequences

The graphs show that the low correlated pair is relatively unaffected by choice of generation whereas the highly correlated pair becomes better when using leaping, scrambling or both.

As leaping has the possible correlation increase we would recommend using the scrambled Halton sequences.

Figure 6 and figure 8 shows the main findings of the multidimensional tests. These were found by solving the optimization problems described in section 5

Several guidelines can be deducted by inspecting the numbers.

First, increasing the number of initially generated draws does not reduce the correlation of the selected set. Which means that the least correlated pair are found among the smaller primes. Second, scrambling is the most effective measure to eliminate correlation in larger subset selections.

Selection: 50	Dimensions				
draws: 1000	100	200	300	500	1000
halton-linear	$\sim$ 0.53868	0.96039	1.00000	1.00000	1.00000
halton-shuffled	$\sim$ 0.12817	$\sim$ 0.28451	$\sim$ 0.32453	$\sim$ 0.31940	0.26208
leaped-linear	$\sim$ 0.30618	0.94363	1.00000	$\sim$ 0.99348	1.00000
leaped-shuffled	$\sim$ 0.08794	$\sim$ 0.14040	$\sim$ 0.18867	$\sim$ 0.20099	0.31337
scrambled-linear	$\sim$ 0.06287	$\sim$ 0.09108	$\sim$ 0.10321	$\sim$ 0.10040	0.10532
scrambled-shuffled	$\sim$ 0.06814	$\sim$ 0.10614	$\sim$ 0.09947	$\sim$ 0.10114	0.14266

This table shows the same results as figure 6, but the influence of the initial set size is clearer here. The linear multidimensional draws shows that the solver is having an increasingly hard time as the initial set size increases. This can be seen from the fact that the correlation rises with the size of the initial set even though the initial set of 200 contains the sequences in the size 100 initial set.

The shuffled draws however shows that increasing the initial size tends to increase the found correlation. This could be caused by the fact that the halton sequences of the larger size initial sets are more correlated than the smaller initial sets.

Figure 7: Minimal correlation with preset selection size

Maximal uncorrelated	Dimensions				
subset (draws: 1000)	100	200	300	500	1000
halton-linear	25	26	26	26	26
halton-shuffled	7	2	1	1	2
leaped-linear	30	31	32	32	33
leaped-shuffled	23	12	≥ 6	3	2
scrambled-linear	49	67	≥ 74	≥ 84	≥ 93
scrambled-shuffled	42	≥ 62	≥ 71	≥ 82	≥ 90

This table shows the results from optimization problem 2. It can easily be seen that shuffling is deteriorating and that scrambling is significantly better if getting the maximum uncorrelated set is most important. The results where $\geq$ # is used means that the solver reached the time limit before optimality, but had found a feasible solution with # uncorrelated sequences. As the solver is highly efficient and quickly gets close to optimality it is unlikely that sets with significantly more sequences exists in the initial set.

Inspecting the subset sizes for the scrambled linear generator shows that it will be extremely demanding (huge initial set and very long computation time) to find a subset of more than 100 uncorrelated sequences when using halton sequences.

Figure 8: Maximal sets

A third observation, that is less obvious, is that the introduction of shuffling has negative impact on small sets, while yielding lower correlation on large sets. The explanation for this is that the shuffling evens the correlation among the generated sequences. In other words; it sacrifices the low correlation pairs in order to get better high correlation pairs.

Furthermore it can be concluded that using both shuffling and scrambling seems needless as the scrambling leaves little room for improvement by the shuffling.

It can be concluded that standard Halton sequences are best for draws with less than approximately 15 dimensions. Further sequence inspection indicated that selecting bases among the first 40 primes is preferable. From 15 to around 25 dimensions and limited length leaped Halton sequences should be used. For draws of larger dimension scrambling is the most effective measure for reducing the correlation.

If uniform correlation is important, shuffling can be used to bring the pairwise correlation closer to average.

Appendices A.1, A.2, A.3 and A.4 gives visual examples on a highly correlated set.

8 Conclusion

Building from the standard Halton draw we have now been through the two other single dimension number sequences: leaped and scrambled and we have discussed the multidimensional shuffling.

Several combinations of these have been compared and it can be concluded that the usage of the multidimentional sequence must be considered when deciding how to generate the numbers.

As basic sequence the scrambled Halton sequence provide the least interdimensional correlation. Leaping is also a possibility, but the leaping distances must be chosen carefully as they introduce correlation in long sequences.

Shuffling does not guarantee a reduction in the correlation of a draw, but instead it evens the correlation such that every pair of sequences in a multidimensional draw becomes closer to equally correlated.

Considering a single measure to avoid correlation it can be concluded scrambling gives lowest correlation at the cost of slightly longer generation time.

This paper has demonstrated that there are immediately unobservable – though detectable, differences between the performance of random numbers, contingent on their generation.

The differences in the uniformity and bias will inevitably impact the modelling of any model, where random numbers are incorporated by use of simulation in the estimations as eg. the Mixed Logit estimation.

Coefficients estimated and even distributions thought valid for a given model may be impacted by the use of different random number generators. The impact will be via the calibration, hence the estimation provided an iterative method is applied for the estimation of the model.

Therefore, use of method for random number generation should actively be considered and at least, be documented along with other model formulation considerations.

References

[Ben, 2002] (2002). Hybrid choice models: Progress and challenges. *Marketing Letters*, 13 (3):165–175.

[Ben-Akiva et al., 1993] Ben-Akiva, M., Bolduc, D., and Bradley, M. (1993). Estimation of travel choice models with random distributed values of time. *Transportation Research Record*, 1413:88–97.

[Ben-Akiva et al., 2001] Ben-Akiva, M., Bolduc, D., and Walker, J. L. (2001). Specification, identification & estimation of the logit kernel (or continuous mixed logit) model. Draft presented at 5th tri-annual Invitational Choice Symposium Hosted by UC Berkeley, June 1-5, 2001.

[Bhat, 2003] Bhat, C. (2003). Simulation estimation of mixed discrete choice models using randomized and scrambled halton sequences. *Transportation Research Part B: Methodological*, 37(9):837–855.

[Bhat, 2001] Bhat, C. R. (2001). Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model. *Transportation Research Part B: Methodological*, 35(7):677–693.

[Bierlaire, 2002] Bierlaire, M. (2002). Biogeme website. http://roso.epfl.ch/biogeme.

[Hess and Polak, 2003] Hess, S. and Polak, J. W. (2003). On the performance of the shuffled halton sequence in the estiamtion od discrete choice models. Technical report, Association for European Transport 2003.

[Kocis and Whiten, 1997] Kocis, L. and Whiten, W. J. (1997). Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 23(2):266–294.

[McFadden and Train, 2000] McFadden, D. and Train, K. (2000). Mixed mnl models for discrete response. *Journal of Applied Econometrics*, 15(5):447–70.

[Nunes et al., 2001] Nunes, L. C., Cunha-e Sa, M. A., Ducla-Soares, M. M., Rosado, M. A., and Day, B. H. (2001). Identifying non-consistent choice behavior in recreation demand models. *Economics Letters*, 72(3):403–410.

[Revelt and Train, 2000] Revelt, D. and Train, K. E. (2000). Customer-specific taste parameters and mixed logit. Working Paper 00-274, Department of Economics, University of California, Berkeley.

[Ross, 1997] Ross, S. M. (1997). *Simulation.* Academic Press, Burlington, MA, USA, 2nd edition.

[Sørensen, 2003a] Sørensen, M. V. (2003a). *Discrete Choice Models. Estimation of passenger traffic.* PhD thesis, Danish Technical University, DK-2800 Kgs. Lyngby, Denmark.

[Sørensen, 2003b] Sørensen, M. V. (2003b). Msl for mixed logit model estimation - on shape of distributions. In *Proceedings of European Transport Conference, Strasbourg, France.* CD-rom.

[Sørensen, 2004] Sørensen, M. V. (2004). Impact of a priori distributions on mixed logit model estimation. tests on synthetic data. In *Proceedings of World Conference on Transport Research, Istanbul , Turkey.*

[Train, 1998] Train, K. E. (1998). Recreation demand models with taste differences over people. *Land Economics*, 74(2):230–239.

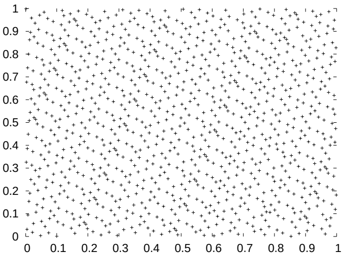
[Train, 2001] Train, K. E. (2001). A comparison of hierarchical bayes and maximum simulated likelihood for mixed logit. Working paper, Department of Economics, University of California, Berkeley.

[Train, 2002] Train, K. E. (2002). Website with gauss shareware code. http://elsa.berkeley.edu/˜train/software.html.

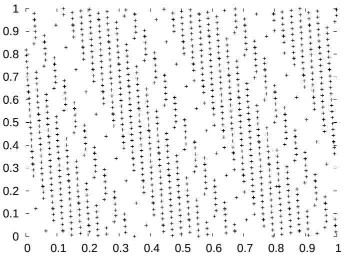
A Example draws

A.1 The pseudorandom draw

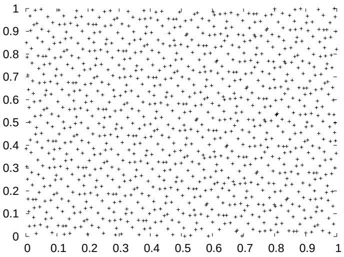
Visualization of the linear combination where the sequences are statically assigned to dimensions.



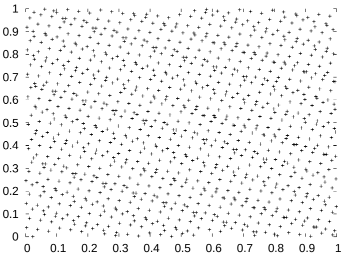
sequences 1 and 2



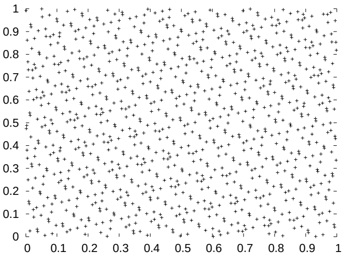
sequences 1 and 3



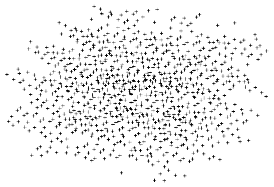
sequences 1 and 4



sequences 1 and 5

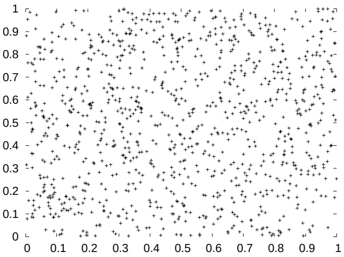


sequences 2 and 3

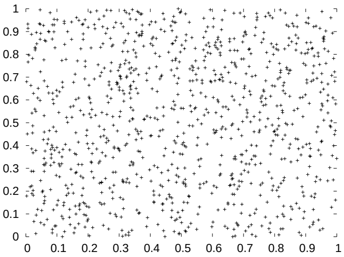


sequences 1, 2 and 3

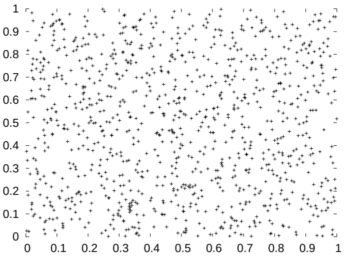
The shuffled approach where the sequences are shuffled at each draw.



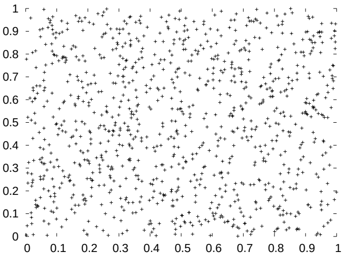
sequences 1 and 2



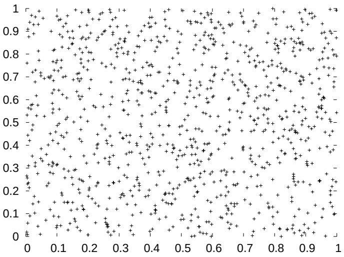
sequences 1 and 3



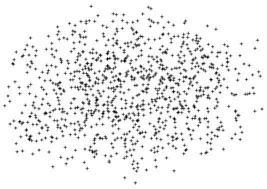
sequences 1 and 4



sequences 1 and 5



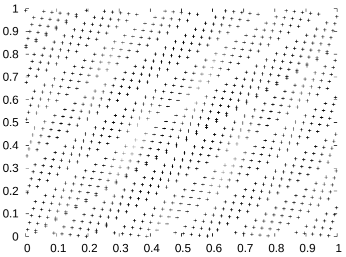
sequences 2 and 3



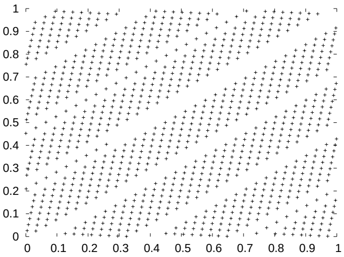
sequences 1, 2 and 3

A.2 The standard Halton draw

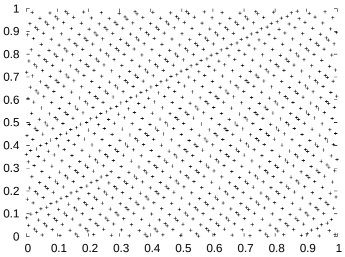
Visualization of the linear combination where the sequences are statically assigned to dimensions.



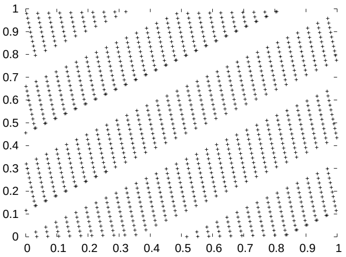
sequences Φ_{37} and Φ_{41}



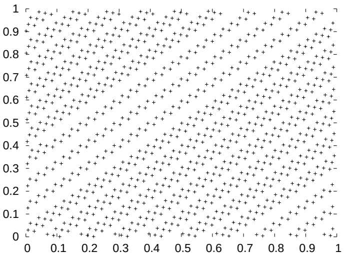
sequences Φ_{37} and Φ_{43}



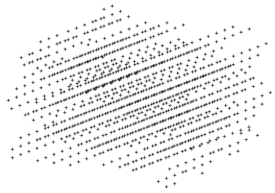
sequences Φ_{37} and Φ_{47}



sequences Φ_{37} and Φ_{53}

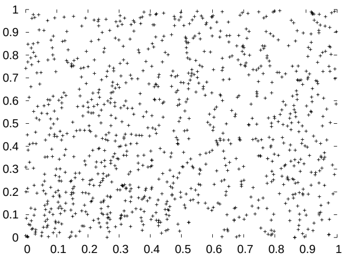


sequences Φ_{41} and Φ_{43}

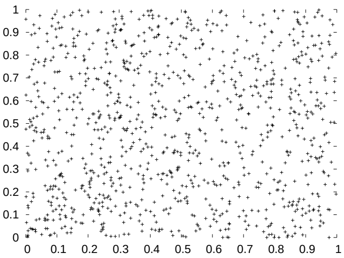


sequences Φ_{37}, Φ_{41} and Φ_{43}

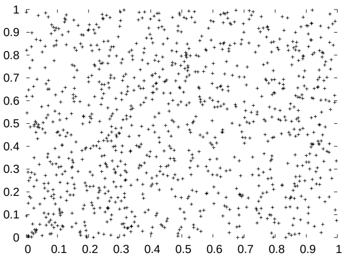
The shuffled approach where the sequences are shuffled at each draw.



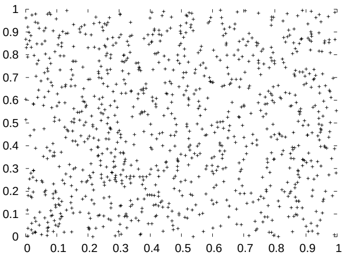
sequences Φ_{37} and Φ_{41}



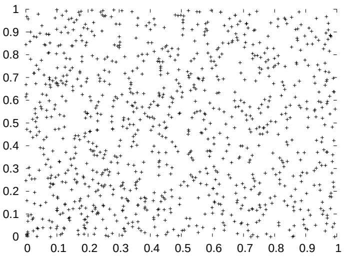
sequences Φ_{37} and Φ_{43}



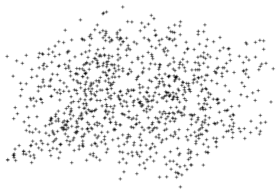
sequences Φ_{37} and Φ_{47}



sequences Φ_{37} and Φ_{53}



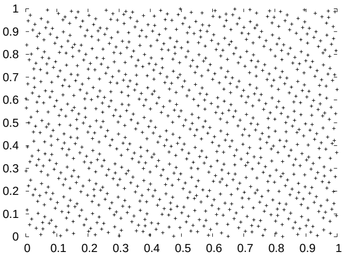
sequences Φ_{41} and Φ_{43}



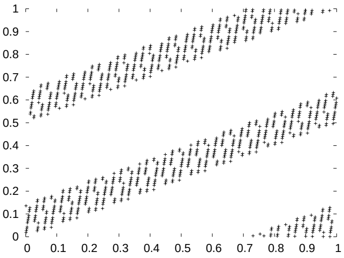
sequences Φ_{37}, Φ_{41} and Φ_{43}

A.3 The leaped Halton draw

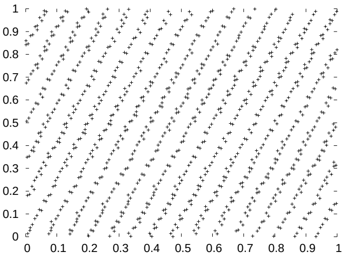
Visualization of the linear combination where the sequences are statically assigned to dimensions.



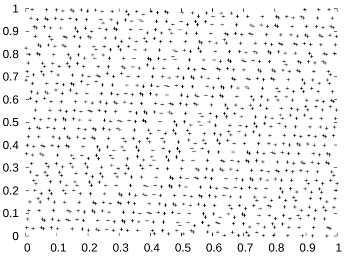
sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{83\curvearrowright 89}$



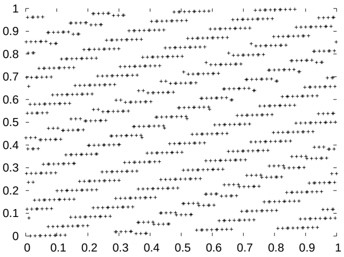
sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{97\curvearrowright 101}$



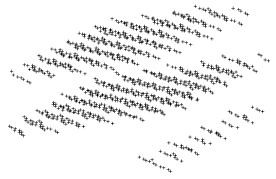
sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{103\curvearrowright 107}$



sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{109\curvearrowright 113}$

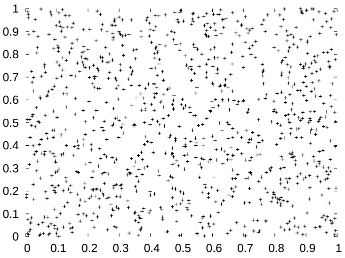


sequences $\Phi_{83\curvearrowright 89}$ and $\Phi_{97\curvearrowright 101}$

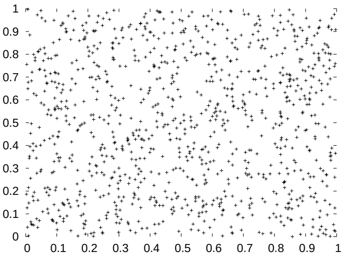


sequences $\Phi_{73\curvearrowright 79}$, $\Phi_{83\curvearrowright 89}$ and $\Phi_{97\curvearrowright 101}$

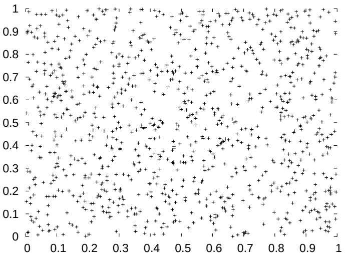
The shuffled approach where the sequences are shuffled at each draw.



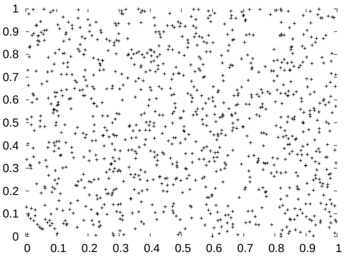
sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{83\curvearrowright 89}$



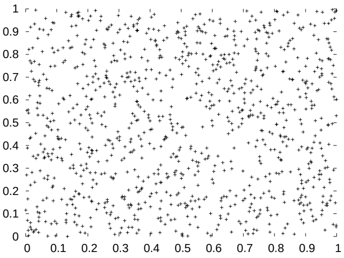
sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{97\curvearrowright 101}$



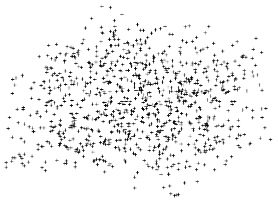
sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{103\curvearrowright 107}$



sequences $\Phi_{73\curvearrowright 79}$ and $\Phi_{109\curvearrowright 113}$



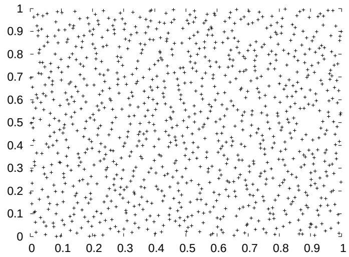
sequences $\Phi_{83\curvearrowright 89}$ and $\Phi_{97\curvearrowright 101}$



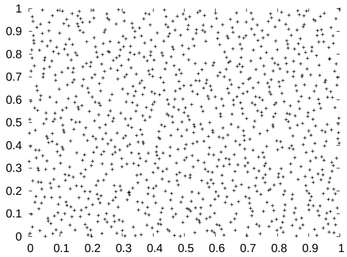
sequences $\Phi_{73\curvearrowright 79}$, $\Phi_{83\curvearrowright 89}$ and $\Phi_{97\curvearrowright 101}$

A.4 The scrambled Halton draw

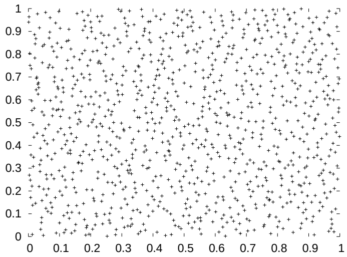
Visualization of the linear combination where the sequences are statically assigned to dimensions.



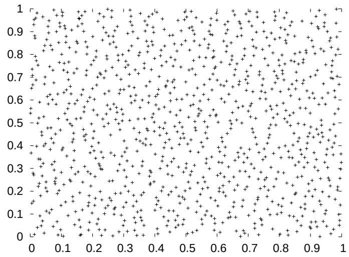
sequences Φ_{37} and Φ_{41}



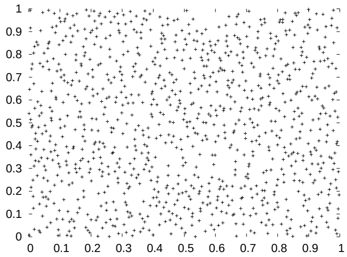
sequences Φ_{37} and Φ_{43}



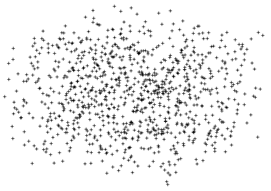
sequences Φ_{37} and Φ_{47}



sequences Φ_{37} and Φ_{53}

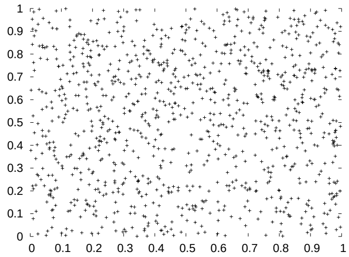


sequences Φ_{41} and Φ_{43}

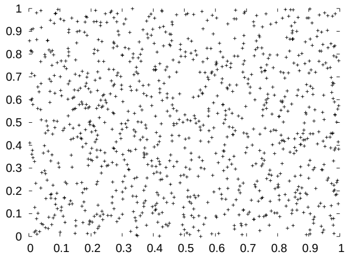


sequences Φ_{37}, Φ_{41} and Φ_{43}

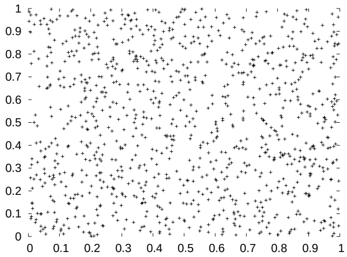
The shuffled approach where the sequences are shuffled at each draw.



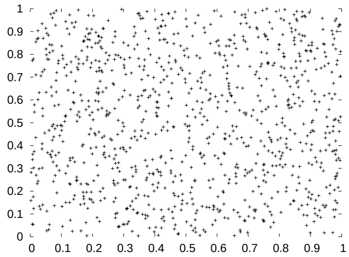
sequences Φ_{37} and Φ_{41}



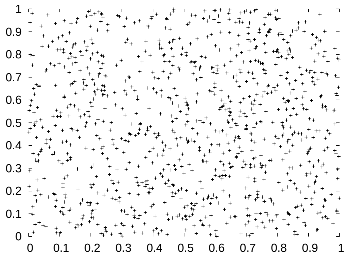
sequences Φ_{37} and Φ_{43}



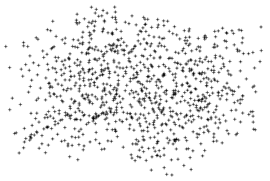
sequences Φ_{37} and Φ_{47}



sequences Φ_{37} and Φ_{53}



sequences Φ_{41} and Φ_{43}



sequences Φ_{37}, Φ_{41} and Φ_{43}