

Time Series Simulation with Quasi Monte Carlo Methods

Jenny X. Li ^{*} Peter Winker [†]

September 27, 2000

Abstract

This paper compares quasi Monte Carlo methods, in particular so-called (t, m, s) -nets, with classical Monte Carlo approaches for simulating econometric time-series models. Quasi Monte Carlo methods have found successful application in many fields, such as physics, image processing, and the evaluation of finance derivatives. However, they are rarely used in econometrics. Here, we apply both traditional and quasi Monte Carlo simulation methods to time-series models that typically arise in macroeconometrics. The numerical experiments demonstrate that quasi Monte Carlo methods outperform traditional ones for all models we investigate.

1 Introduction

The traditional Monte Carlo method is widely used in many research fields due to its simplicity. Indeed, its computational ease and convergence rates are independent of the dimension of the problem. However, this method also exhibits disadvantages. In particular, it can be computationally onerous to achieve a high level of accuracy because the convergence rate is only $O(N^{-1/2})$ for N sample paths. For example, to reduce the error by a factor of 10, the

^{*}Departments of Mathematics and Economics, The Pennsylvania State University, University Park, PA, 16802, USA *Email:* <li@math.psu.edu>

[†]Department of Economics, University of Mannheim, D-68131 Mannheim, Germany. *Email:* <Peter.Winker@vwl.uni-mannheim.de>

number of simulations would have to increase by a factor of 100. Hence, high accuracy requirements may lead to long computation times. Furthermore, it might be difficult to obtain “randomly” generated sample paths in high dimensions (Ripley, 1987, pp. 23ff).

By contrast, quasi Monte Carlo simulation uses more uniformly distributed deterministic sequences and can provide a considerably improved convergence rate — close to $O(N^{-1})$ or even $O(N^{-3/2})$, in some special cases (Owen (1997), Caflisch and Morokoff (1996) and Fang *et al.* (2000)). This improved convergence rate can produce dramatic gains in both computational time and the range of applications of simulation methods for econometric problems.

We review both Monte Carlo and quasi Monte Carlo methods, and introduce the quasi Monte Carlo sequences, including Fauré, Halton, Sobol and the most recently developed (t, m, s) -nets (Niederreiter, 1992). Emphasis is given to problems that may be encountered in implementing quasi Monte Carlo methods and comparing their performance with the traditional Monte Carlo approach. In particular, we apply both methods to time-series models as they often arise in the modeling of macroeconomic data. We are especially interested in the uni- and multivariate autoregressive models (AR, VAR), error correction models (ECM, VECM), and nonlinear models of the type found in the minimum-condition or an aggregate matching function in models of temporary equilibrium (Franz *et al.*, 2000).

The rest of the paper is arranged as follows: In Section 2, we provide a basic description of Monte Carlo and quasi Monte Carlo simulation techniques. In Section 3, we introduce the econometric time-series models we investigate, and in Section 4 we discuss the results of an extensive numerical comparison study. Some final remarks are given in Section 5.

2 Classical Monte Carlo, Low-Discrepancy Sequences and (t, m, s) -Nets

2.1 The Monte Carlo Simulation Framework

There is a large literature on Monte Carlo simulation (Gentle, 1998), so we give only a brief overview and focus on applications to econometric time-series models.

The problem of simulating time-series models can be illustrated using the

simplest case of a stochastic univariate autoregressive process

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \varepsilon_t.$$

For a given initial value y_0 and given parameter (estimates) α_0 and α_1 , a sample path for y_t of length T can be simulated based on a sequence of ε_t of length T . However, usually a single realization of such a sample path is not sufficient, since one might be interested in the distribution of some y_t or some statistics derived from this distribution. Such results are conditional on the given or assumed joint distribution of all ε_t . Examples of statistics on y_t that are of interest in econometric research include its expectation and variance or confidence bands. In a simple linear setting with normally distributed ε_t , these statistics can be obtained analytically. In more general settings, however, these statistics have to be approximated using the empirical distribution of the y_t obtained by simulating a large number, N , of sample paths. The resulting empirical distribution of the y_t is taken as Monte Carlo approximation to the true distribution.

The sample paths ε_{it} ($i = 1, \dots, N$, $t = 1, \dots, T$) are obtained in three steps by classical Monte Carlo. First, a pseudo-random number generator is used to generate numbers uniformly distributed on the interval $(0, 1)$. Second, these uniformly distributed numbers are transformed to normally distributed numbers by some transformation method such as the Box–Muller or inverse method (Ripley, 1987, pp. 54ff). Finally, the vectors of length T for each $i = 1, \dots, N$ are formed by using T consecutive ε_{it} numbers. If the inverse method is used for the transformation to normal deviates,¹ the problem of generating “good” ε ’s is equivalent to the problem of generating “good” random numbers in the T -dimensional unit cube. By “good”, we mean that for each time period, the distribution of the simulated ε ’s closely approximates $\mathcal{N}(0, I)$, where I denotes the identity matrix. Correspondingly, “good” random numbers in the unit cube $(0, 1)^T$ should approximate the uniform distribution as closely as possible.

For multivariate and nonlinear time-series models, the situation is similar. However, the dimension of the sample paths ε_{it} increases. For example, in a two-dimensional autoregressive model, sample paths of length $2T$ must be generated: one of length T for the first equation and another for the second. Since the time-series models usually assume that these sample paths are

¹Different transformation methods, such as the Box–Muller method, should be avoided in this context, as they might introduce artificial correlation (Ripley, 1987, pp. 54ff).

independent, this requirement has to be satisfied by both Monte Carlo and quasi Monte Carlo methods.

So, the key to Monte Carlo methods is in generating “good” random points. However, we can only generate pseudo-random sequences, which, particularly in higher dimensions, can lead to clumping of points, which limits their uniformity. This clumping results from the fact that points in pseudo-random sequences are almost, but not completely independent. Thus, they have a chance of landing very close to each other (Ripley, 1987, pp. 23f). The accuracy of Monte Carlo methods can be improved by using more uniformly distributed pseudo-random sequences. One straightforward such method is antithetic variates: along with a random vector ε , $-\varepsilon$ is also used for the simulation. In linear models, this method reduces the approximation error of the estimation of the mean to zero. However, the effect of antithetic variates on estimates of variance and in nonlinear models is not clear, as the simulation results of Section 4 demonstrate. Hence, quasi Monte Carlo or low-discrepancy sequences methods strive for uniformity in a more general setting. Instead of using pseudo-random sequences, deterministic point sets, known as quasi-random numbers, are applied, thereby minimizing clustering and improving accuracy. In fact, by uniformly picking the points, higher accuracy may be achieved with a smaller number of simulations. However, the meaning of the uniformity in the context of time-series simulation and the choice of uniformity measure has not been addressed yet.

2.2 Measuring the Uniformity of Point Sets

Intuitively, for a uniformly distributed sequence \mathbf{x}_n in the s -dimensional unit cube $I^s = [0, 1]^s$, we would expect the same number of points to be in all subsets of I^s having the same volume. So, we can measure the uniformity of \mathbf{x}_n in terms of its discrepancy, simply defined by considering the number of points in the subsets of I^s . For N points $\{\mathbf{x}_n\}_1^N$ in the s -dimensional unit cube $I^s = [0, 1]^s$, $s \geq 0$, and a subset J of I^s , the *local discrepancy* $D(J; N)$ is defined by

$$D(J; N) = A(J; N) - V(J)N,$$

where $A(J; N)$ is the number of $n, 1 \leq n \leq N$, with $\mathbf{x}_n \in J$ and $V(J)$ is the volume of the subinterval J . If the N points are uniformly distributed, the local discrepancy should be very small for all J 's. This leads to a global definition: the *discrepancy* (also called the *star-discrepancy*) $\Delta(N)$ of the N

points is defined to be

$$\Delta(N) = \sup_J |D(J; N)|,$$

where the supremum is taken over all subsets J of the form $J = \prod_{i=1}^s [0, u_i)$. Unfortunately, it appears impossible to calculate this measure for large point sets with moderate dimension s (L’Ecuyer and Hellekalek, 1998, p. 230).² Further, this measure has some theoretical shortcomings since it is not invariant to certain natural transformations of the unit cube. Alternative measures can be derived by replacing the supremum norm with such norms as L_2 and its corresponding subsets J (Hickernell, 1998). Here, we employ the centred L_2 -discrepancy (CL_2) proposed by Hickernell (1998), which, according to Fang *et al.* (2000), outperforms the star-discrepancy when searching for uniform designs for at least two reasons. First, it is more sensitive than the star-discrepancy: designs with identical star-discrepancies can differ markedly in their CL_2 -discrepancies. Second, employing the CL_2 -discrepancy results in low values for all lower dimensional projections of the point sets. Finally, Fang *et al.* (2000) establish a connection between low CL_2 -discrepancy and orthogonality of designs.

2.3 Low-Discrepancy Sequences and (t, m, s) -Nets

Any point set that has very “small” discrepancy is called a low-discrepancy point set. There exist many well-known low-discrepancy sets that are uniformly distributed in $(0, 1)^s$ – for example Fauré-, Halton-, and Sobol-sequences (Press *et al.*, 1992, pp. 300ff) and (t, m, s) -nets. Halton-sequences describe a class of multidimensional infinite sequences that fill the interval $[0, 1)$. It is easy to generate a multidimensional Halton-sequence. Begin first with a consecutive sequence of non-negative integers, such as $n = 0, 1, 2, \dots, N - 1$ if N replications are required. Each integer n is converted to its representation in the base p number system, where p is any prime number and $p \geq 2$. Thereby, for each dimension, a different base p is chosen. Second, the base p representation is transformed into a number in the interval $[0, 1)$ by reflection about the decimal point. Interested readers are referred to Halton (1960) for a detailed example. Fauré sequences are essentially permutations of Halton sequences, and the Sobol sequence is a reordering of the Halton

²An approximation based on the optimization heuristic threshold accepting is proposed in Winker and Fang (1997).

sequence (Bratley and Fox, 1988). The low-discrepancy sequences proposed by Halton, Fauré, and Sobol are all called (t, s) -sequences, for which certain finite segments form (t, m, s) -nets. A (t, m, s) -net is a point set in I^s for which the local discrepancy equals zero for many subsets. More specifically, let $0 \leq t \leq m$ be integers. A (t, m, s) -net in base b is a point set \mathbf{x}_n of b^m points in $[0, 1]^s$ such that every elementary interval E in base b of volume $1/b^{m-t}$ contains exactly b^t points, where an *elementary interval* in base b is a subinterval E of $[0, 1]^s$ of the form

$$E = \prod_{i=1}^s [a_i b^{-d_i}, (a_i + 1) b^{-d_i}),$$

with integers $a_i, d_i \geq 0$, and $0 \leq a_i < b^{d_i}$ for $1 \leq i \leq s$.

The following Koksma-Hlawka inequality establishes the relationship between low-discrepancy sequences and expectation of random variables with distribution function f .

$$\left| \int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x} - \frac{1}{N} \sum_{n=1}^N f(\mathbf{x}_n) \right| \leq V(f) D_N^*(\mathbf{x}_1, \dots, \mathbf{x}_N), \quad (1)$$

which holds for any function f of s variables that has bounded variation $V(f)$ on $[0, 1]^s$ and for any $\mathbf{x}_1, \dots, \mathbf{x}_N \in [0, 1]^s$. The inequality provides a better error bound when low-discrepancy points are used to approximate the desired expectation. Indeed, the N^{-1} factor in the convergence formula for low-discrepancy points may be contrasted with the $N^{1/2}$ convergence for Monte Carlo and suggests that low-discrepancy methods are superior to Monte Carlo methods.

There are many methods to construct (t, m, s) -nets (see the survey by Clayman *et al.* (2000)). The most commonly used methods include direct constructions using various properties of finite fields and polynomials over finite fields, error-correcting codes including both linear and nonlinear ones such as Kerdock codes, combinatorial methods including generalized orthogonal arrays, and a method that uses linear combinations of the rows of a so-called *generator matrix*, see Bierbrauer and Edel (1999) and Li and Mullen (2000) for more details.

Obviously, (t, s) -sequences and (t, m, s) -nets are closely related. Indeed, (t, s) -sequences provide an effective way to construct (t, m, s) -nets, since the existence of a (t, s) -sequence in base b implies the existence of a $(t, m, s+1)$ -net for all $m \geq t$. However, by using techniques other than (t, s) -sequences,

we can often construct a net with a smaller value of t and hence a net with more uniformly distributed points.

For example, using the method described as construction 18 in Clayman *et al.* (2000), it is known that there is a $(6, 9)$ -sequence in base 2 (and there is no known $(5, 9)$ -sequence). Hence there is a $(6, m, 10)$ -net in base 2 for all $m > 6$. In particular, there is a $(6, 14, 10)$ -net in base 2. However, with the construction method indicated in Bierbrauer and Edel (1999) — discussed in considerable detail in Sections 4 and 5 of that paper — generator matrices can be used to produce a $(5, 14, 10)$ -net in base 2. A $(5, 14, 10)$ -net has a more uniform distribution of points than does a $(6, 14, 10)$ -net.

2.4 Uniformity of Pseudo-Random and Quasi Monte Carlo Point Sets

In this section we consider a number of pseudo-random (GAUSS, UNIF, ESSL) and quasi Monte Carlo (FAURE, HALTON, SOBOL, TMS) point sets:

GAUSS	Uniform pseudo-random number generator of GAUSS 3.2.4
UNIF	Uniform pseudo-random number generator from Bratley and Fox (1988)
ESSL	Uniform pseudo-random number generator from ESSL: $s_n = (a(s_{n-1})) \bmod m = (a^n s_0) \bmod m$ $x_n = s_n/m$, where s_0 is the initial seed, $a = 16807$ and $m = 2^{31} - 1$
FAURE	Fauré sequence generator from Bratley and Fox (1988)
HALTON	Halton sequence generator from Bratley and Fox (1988)
SOBOL	Sobol sequence generator from Bratley and Fox (1988)
TMS	TMS net generator from Li and Mullen (2000)

All those point sets provide good approximations to the uniform distribution. While the pseudo-random number generators are univariate by construction, the quasi Monte Carlo methods provide multivariate point sets explicitly. For the pseudo-random number generators, higher dimensional vectors are obtained by stacking the corresponding number of drawings into one vector. The quality of the approximation of the uniform distribution is measured using the centred L_2 -discrepancy (CL_2). Table 1 shows the CL_2 -discrepancy of some point sets obtained by pseudo-random number generators and quasi Monte Carlo methods. Since the point sets provided by

pseudo-random number generators depend on the initial seed, the mean of 10 different point sets is reported for these generators. For some instances, different (t, m, s) -nets are available. Then, the table provides the smallest discrepancy found for all (t, m, s) -nets used in the simulation study in section 4.

Table 1: Centered L_2 -discrepancy of Point Sets

Method	$s = 10$				
	N=1024	N=4096	N=16384	N=32768	N=65536
GAUSS	$0.715 \cdot 10^{-2}$	$0.181 \cdot 10^{-2}$	$0.452 \cdot 10^{-3}$	$0.216 \cdot 10^{-3}$	$0.107 \cdot 10^{-3}$
UNIF	$0.699 \cdot 10^{-2}$	$0.170 \cdot 10^{-2}$	$0.425 \cdot 10^{-3}$	$0.194 \cdot 10^{-3}$	$0.107 \cdot 10^{-3}$
ESSL	$0.698 \cdot 10^{-2}$	$0.170 \cdot 10^{-2}$	$0.425 \cdot 10^{-3}$	$0.194 \cdot 10^{-3}$	$0.107 \cdot 10^{-3}$
FAURE	$0.121 \cdot 10^{-2}$	$0.148 \cdot 10^{-3}$	$0.195 \cdot 10^{-4}$	$0.845 \cdot 10^{-5}$	$0.344 \cdot 10^{-5}$
HALTON	$0.133 \cdot 10^{-2}$	$0.204 \cdot 10^{-3}$	$0.250 \cdot 10^{-4}$	$0.825 \cdot 10^{-5}$	$0.197 \cdot 10^{-5}$
SOBOL	$0.832 \cdot 10^{-3}$	$0.137 \cdot 10^{-3}$	$0.194 \cdot 10^{-4}$	$0.812 \cdot 10^{-5}$	$0.288 \cdot 10^{-5}$
TMS	$0.364 \cdot 10^{-2}$	$0.207 \cdot 10^{-3}$	$0.186 \cdot 10^{-4}$	–	$0.509 \cdot 10^{-5}$

In Table 1, s is the dimension and N the number of points of the sample point sets. The results for these instances clearly indicate that the CL_2 -discrepancy of all quasi Monte Carlo point sets is smaller than the expected value for point sets obtained by Monte Carlo methods.³ A clear ranking of the quasi Monte Carlo methods is not provided by this evidence. Only the (t, m, s) -net with 16 384 points has a slightly smaller centred L_2 -discrepancy than the other quasi Monte Carlo point sets. However, Niederreiter (1992) points out that (t, m, s) -nets yield the smallest discrepancy bound and therefore, by the Koksma–Hlawka inequality, the smallest error bound (within the class of functions of bounded variation in the sense of Hardy and Krause) among all known constructions of point sets. Especially, within the class of functions with rapidly converging Walsh series, Larcher and Traunfellner (1994) have shown that digital (t, m, s) -nets yield an error bound of the optimal order of magnitude. Thus, the findings of table 1 have to be attributed either to the use of the centred L_2 -discrepancy instead of the star-discrepancy or to some shortcomings of the specific construction of (t, m, s) -nets for these comparatively small sets.

³See also Fang *et al.* (2000) for a formal derivation of expectation values for CL_2 of Monte Carlo and quasi Monte Carlo point sets.

3 Time–Series Models

In the previous section we discussed different Monte Carlo and quasi Monte Carlo methods for generating stochastic error paths for time–series models. Furthermore, a comparison of the generated point sets based on a measure of discrepancy has been provided. In this section, we add evidence on the performance of the methods in both linear and nonlinear stochastic time–series models as they typically appear, e.g., in macroeconomic modeling (Winker, 1999). While it is possible to obtain analytical solutions for the linear models, the benchmark solutions for the nonlinear models have to be obtained by a huge number of replications either in a standard Monte Carlo framework or — and this is the approach followed in this paper — in a quasi Monte Carlo framework. We discuss the number of replications used and accuracy problems in Section 4.

Tables 2 and 3 provide an overview of the models used in our simulation setup. While models (1)–(6) are linear, models (7)–(9) exhibit nonlinearities. Thus, analytical solutions for the mean response and its variance can be obtained for the first models, while the benchmarks for the latter are obtained using Sobol sequences with a large number of replications N . All one–dimensional models are simulated over 10 time periods and all two–dimensional models over 5, fixing initial values to one. Thus, for all models, the dimension of the error space is 10; hence 10–dimension pseudo–random or low–discrepancy points sets are needed for the simulation.

The numbers of replications used in the simulations are powers of 2, which is a result of the construction of (t, m, s) –nets in base 2. Thus, results are provided for 1024, 4096, 16 384, and 65 536 points. When antithetic variates are employed, only half the number of points is generated, the other half being obtained by multiplying each vector ε by minus one.

Of course, the accuracy of the approximations could be improved using a larger number of replications. However, in macroeconomic modeling, the solution of the model for a given set of errors is often quite time consuming. Therefore, the number of replications is limited by available computer resources. The actual number of replications used in stochastic policy simulation is often in the range 1000 to 2000 rather than larger than 10 000 (Franz *et al.*, 2000). Furthermore, the gain from switching from Monte Carlo to quasi Monte Carlo methods is often much larger than that from increasing the number of replications in a Monte Carlo setting.

Table 2: Linear models used for Simulation

No.	Model	Equation	Parameters
(1)	AR(1)	$x_t = \alpha_0 + \alpha_1 x_{t-1} + \varepsilon_t$	$\alpha_0, \alpha_1, \sigma_\varepsilon$
(2)	VAR(1) 2-dim.	$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 y_{t-1} + \varepsilon_{1,t}$ $y_t = \beta_0 + \beta_1 x_{t-1} + \beta_2 y_{t-1} + \varepsilon_{2,t}$	$\alpha_0, \alpha_1, \alpha_2, \sigma_{\varepsilon_1},$ $\beta_0, \beta_1, \beta_2, \sigma_{\varepsilon_2}$
(3)	AR(2)	$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \varepsilon_t$	$\alpha_0, \alpha_1, \alpha_2, \sigma_\varepsilon$
(4)	VAR(2) 2-dim.	$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} +$ $\alpha_3 y_{t-1} + \alpha_4 y_{t-2} + \varepsilon_{1,t}$ $y_t = \beta_0 + \beta_1 x_{t-1} + \beta_2 x_{t-2} +$ $\beta_3 y_{t-1} + \beta_4 y_{t-2} + \varepsilon_{2,t}$	$\alpha_0, \alpha_1, \alpha_2,$ $\alpha_3, \alpha_4, \sigma_{\varepsilon_1},$ $\beta_0, \beta_1, \beta_2,$ $\beta_3, \beta_4, \sigma_{\varepsilon_2}$
(5)	ECM(1)	$\Delta x_t = \alpha_0 + \alpha_1 \Delta x_{t-1} + \alpha_2 \Delta y_{t-1}$ $+ \lambda(x_{t-1} - \beta y_{t-1}) + \varepsilon_t$	$\alpha_0, \alpha_1, \alpha_2,$ $\lambda, \beta, \sigma_\varepsilon$
(6)	VECM(1) 2-dim.	$\Delta x_t = \alpha_0 + \alpha_1 \Delta x_{t-1} + \alpha_2 \Delta y_{t-1}$ $+ \lambda_1(x_{t-1} - \gamma y_{t-1}) + \varepsilon_{1,t}$ $\Delta y_t = \beta_0 + \beta_1 \Delta x_{t-1} + \beta_2 \Delta y_{t-1}$ $+ \lambda_2(x_{t-1} - \gamma y_{t-1}) + \varepsilon_{2,t}$	$\alpha_0, \alpha_1, \alpha_2,$ $\lambda_1, \gamma, \sigma_{\varepsilon_1}$ $\beta_0, \beta_1, \beta_2,$ $\lambda_2, \sigma_{\varepsilon_2}$

4 Simulation

Despite of the small number of models considered for the comparison study, a large number of qualitatively different parameter settings can be considered. Furthermore, the simulation outcomes can be evaluated using different measures. Finally, the results depend on the size of the point sets, i.e., the number of replications. To keep the comparison study tractable, some *a priori* choices are made.

Except for model (1), we consider three different parameter sets for each model, corresponding to high negative autocorrelation (AC=-1), almost negligible autocorrelation (AC=0), and high positive autocorrelation (AC=1). The complete set of parameters for all 27 simulated models is provided in Tables 8 and 9 in the appendix. For model (1), a larger number of different parameter values is considered.

The simulation results are evaluated based on the approximation error of

Table 3: Non linear models used for Simulation

No.	Model	Equation	Parameters
(7)	VAR(1) 2-dim. min/max	$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 y_{t-1} + \varepsilon_{1,t}$ $y_t = \beta_0 + \beta_1 x_{t-1} + \beta_2 y_{t-1} + \varepsilon_{2,t}$ $z_t = \min\{x_t, y_t\}$	$\alpha_0, \alpha_1, \alpha_2, \sigma_{\varepsilon_1},$ $\beta_0, \beta_1, \beta_2, \sigma_{\varepsilon_2}$
(8)	VAR(1) 2-dim. CES	$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 y_{t-1} + \varepsilon_{1,t}$ $y_t = \beta_0 + \beta_1 x_{t-1} + \beta_2 y_{t-1} + \varepsilon_{2,t}$ $z_t = \{x_t^{-\rho} + y_t^{-\rho}\}^{-\frac{1}{\rho}}$	$\alpha_0, \alpha_1, \alpha_2, \sigma_{\varepsilon_1},$ $\beta_0, \beta_1, \beta_2, \sigma_{\varepsilon_2}$ ρ
(9)	VECM(1) 2-dim. CES	$\Delta x_t = \alpha_0 + \alpha_1 \Delta x_{t-1} + \alpha_2 \Delta y_{t-1}$ $\quad + \lambda_1 (x_{t-1} - \gamma y_{t-1}) + \varepsilon_{1,t}$ $\Delta y_t = \beta_0 + \beta_1 \Delta x_{t-1} + \beta_2 \Delta y_{t-1}$ $\quad + \lambda_2 (x_{t-1} - \gamma y_{t-1}) + \varepsilon_{2,t}$ $z_t = \{x_t^{-\rho} + y_t^{-\rho}\}^{-\frac{1}{\rho}}$	$\alpha_0, \alpha_1, \alpha_2,$ $\lambda_1, \gamma, \sigma_{\varepsilon_1}$ $\beta_0, \beta_1, \beta_2,$ $\lambda_2, \sigma_{\varepsilon_2}$ ρ

$E(x_T)$ for models (1) to (6) and $E(z_T)$ for models (7) to (9), where T denotes the last simulated period.⁴ Furthermore, the approximation errors of the estimated variances of x_T and z_T are also considered. In future research, we will also include measures like the estimated 10- and 90-percent quantiles or the MSFE discussed in Ericsson and Marquez (1998).

Although a high number of replications eventually results in high quality results for all methods, we concentrate on rather small numbers of replications as they typically appear when simulating econometric models or estimators. Therefore, we report results for $N = 4096$, $N = 16384$ and $N = 65536$. Table 4 shows the approximation error for $E(x_{10})$ in percent for model (1). For the pseudo-random generators, the mean of the absolute approximation error of ten different runs is reported.

The results for the approximation error of $E(x_{10})$ in model (1) are clearly in favour of quasi Monte Carlo methods. In particular, (t, m, s) -nets provide unbiased estimates without using antithetic variates. Therefore, in the sequel antithetic variates are used only for the pseudo-random numbers and

⁴A similar approach is chosen by Acworth *et al.* (1998), when comparing the performance of Monte Carlo and quasi Monte Carlo methods for option pricing.

Table 4: Approximation error (p.c.) of simulated $E(x_{10})$ for Model (1)
 $(\alpha_0 = 0.1, \sigma_\varepsilon = 0.2)$

α_1	N	Pseudo Random Numbers			Quasi Monte Carlo Methods			
		GAUSS	UNIF	ESSL	FAURE	HALTON	SOBOL	TMS
-0.99	4096	0.5001	0.5084	1.4363	0.0706	-0.1199	0.0149	0.0000
-0.99	16384	0.4605	0.4610	0.3922	0.0972	-0.0459	0.0083	0.0000
-0.99	65536	0.1501	0.1485	0.1607	0.0054	-0.0066	-0.0012	0.0000
-0.90	4096	1.0157	1.0340	2.3534	0.1150	-0.2730	0.0181	0.0000
-0.90	16384	0.6601	0.6631	0.7683	0.1358	-0.1084	0.0127	0.0000
-0.90	65536	0.2422	0.2412	0.2765	0.0007	-0.0167	-0.0029	0.0000
-0.50	4096	2.6063	2.6622	5.7335	0.2872	-1.4954	0.0990	0.0000
-0.50	16384	1.2873	1.2970	2.4894	0.0860	-0.6421	0.0370	0.0000
-0.50	65536	0.7355	0.7407	1.2406	-0.0391	-0.1133	-0.0072	0.0000
0.00	4096	1.7426	1.7329	2.7782	0.0861	-1.4771	0.1090	0.0000
0.00	16384	0.7956	0.7927	1.2153	-0.0124	-0.5336	0.0224	0.0000
0.00	65536	0.6722	0.6731	0.5722	-0.0177	-0.1208	-0.0009	0.0000
0.50	4096	1.4046	1.4059	1.1047	0.0179	-1.4133	0.0576	0.0000
0.50	16384	0.5487	0.5494	0.7799	-0.0207	-0.4590	0.0095	0.0000
0.50	65536	0.4142	0.4141	0.2831	-0.0241	-0.1293	0.0012	0.0000
0.90	4096	0.5447	0.5485	0.4930	-0.0368	-0.6581	0.0036	0.0000
0.90	16384	0.2138	0.2159	0.2554	-0.0180	-0.2164	-0.0003	0.0000
0.90	65536	0.1636	0.1631	0.1298	-0.0232	-0.0653	0.0002	0.0000
0.99	4096	0.4082	0.4111	0.3987	-0.0385	-0.4599	-0.0029	0.0000
0.99	16384	0.1845	0.1848	0.2113	-0.0186	-0.1510	-0.0014	0.0000
0.99	65536	0.1206	0.1201	0.1130	-0.0192	-0.0456	-0.0002	0.0000

the other quasi Monte Carlo methods. Further, the speed of convergence, i.e., the improvement in the approximation quality when the number of replications N is increased, is much higher for the quasi Monte Carlo methods, mirroring the findings of a smaller discrepancy of these point sets. Both for the pseudo-random number generators and the Halton sequences, the approximation error becomes smaller when autocorrelation is high. For the pseudo-random numbers this finding can be explained by the fact that the artificial correlation stemming from their construction becomes relatively less important when the autocorrelation of the process increases.

Table 5: Approximation error (p.c.) of simulated $\text{VAR}(x_{10})$ for Model (1) ($\alpha_0 = 0.1, \sigma_\varepsilon = 0.2$)

α_1	N	Pseudo Random Numbers			Quasi Monte Carlo Methods			
		GAUSS	UNIF	ESSL	FAURE	HALTON	SOBOL	TMS
-0.99	4096	1.5927	1.6104	2.4209	-0.2865	0.1226	-0.1347	-0.3434
-0.99	16384	0.8573	0.8663	2.0180	-0.2887	0.0407	-0.4396	0.2600
-0.99	65536	0.6623	0.6592	0.7449	0.0102	-0.0096	0.0238	-0.0824
-0.90	4096	1.5637	1.4916	2.7374	-0.4077	0.2308	-0.4753	-0.5611
-0.90	16384	0.9292	0.9283	1.9133	-0.2374	0.0086	-0.7404	0.0575
-0.90	65536	0.4530	0.4513	0.6688	0.0302	-0.0174	0.0458	0.0318
-0.50	4096	1.1716	1.1780	2.9215	-0.1082	-0.3710	-1.6932	-0.0431
-0.50	16384	1.0441	1.0436	1.7506	-0.0969	-0.1426	-1.4697	-0.1200
-0.50	65536	0.5836	0.5831	0.5430	0.0445	-0.0341	-0.0077	0.0832
0.00	4096	2.3790	2.3593	2.8036	0.1360	-0.3785	-0.6623	-0.0076
0.00	16384	1.1960	1.2007	1.5163	-0.1688	-0.2354	-0.1870	-0.0020
0.00	65536	0.3141	0.3161	0.5072	0.0242	-0.0344	-0.0582	-0.0005
0.50	4096	3.2600	3.2365	2.1190	0.2575	-0.9234	0.1708	-0.0827
0.50	16384	0.9213	0.9015	1.1403	-0.1175	-0.5971	1.1219	0.1205
0.50	65536	0.3308	0.3296	0.5459	-0.0505	-0.1182	-0.0970	-0.0637
0.90	4096	2.2593	2.2643	2.7785	0.9080	-3.5490	-2.1190	-0.7205
0.90	16384	1.0401	1.0462	1.5611	0.0807	-1.3127	-0.1957	0.5963
0.90	65536	0.4219	0.4223	0.8896	-0.0735	-0.4119	-0.2254	0.2277
0.99	4096	1.3713	1.4312	2.7272	0.9600	-3.9162	-2.8039	-1.7599
0.99	16384	0.8311	0.8470	1.5107	0.1454	-1.3521	-0.6654	0.6146
0.99	65536	0.3943	0.3949	0.9864	-0.0441	-0.4352	-0.2664	0.4673

Table 5 shows the approximation error for the simulated variance of x_{10}

($\text{VAR}(x_{10})$) for model (1) using antithetic variates for all methods except (t, m, s) -nets. The qualitative results do not change when employing original sequences (results are available on request). However, the approximation errors for the other quasi Monte Carlo methods become smaller without antithetic variates, in particular for a small number of replications. For high negative autocorrelation, HALTON sequences outperform the other methods, while for small negative autocorrelation, FAURE and SOBOL sequences seem competitive. Only when there is no autocorrelation in the data generating process do (t, m, s) -nets clearly outperform the other methods. With positive autocorrelation FAURE sequences result in the smallest approximation error for the simulated variance.

Since the results for the different pseudo-random number generators do not differ much, only results for the ESSL generator are reported for models (2) – (9). Further, for the nonlinear models (7) – (9), when analytical solutions for the true estimates are not available, reference values for expected value and variance are obtained by simulation. Therefore, SOBOL-sequences of increasing length are used until the change in these values from doubling the number of replications becomes smaller than 10^{-5} .⁵

For the linear models (1) – (6), the approximation error of all methods becomes zero when using antithetic variates. Therefore, Table 6 reports summary information on approximation errors of simulated $E(x_{10})$ only for the nonlinear models (7) – (9). Detailed results can be found in Tables 10, 11 and 12 in the appendix. Results are based on antithetic variates except for the (t, m, s) -nets.

The approximation error of simulated $E(x_{10})$ is smaller than 1 percent in almost all instances and quasi Monte Carlo point sets for $N = 4096$. However, it differs considerably across methods. Thus, it is difficult to derive general conclusions from these results. For models with a high negative or positive autocorrelation ($\text{AC}=-1$ or $\text{AC}=1$), the (t, m, s) -nets used in this simulation study do not provide the smallest approximation error. For negative autocorrelation, HALTON and SOBOL, result in the smallest approximation errors, while FAURE-sequences are superior if high positive autocorrelation is present. If autocorrelation is small, (t, m, s) -nets provide the smallest mean approximation error for $N = 4096$, but other quasi Monte Carlo methods

⁵In order to obtain this high accuracy up to several 100 million replications are required depending on the model and the parameter set, while, typically, some 10 million replications are sufficient.

Table 6: Mean absolute approximation error (p.c.) of simulated $E(x_{10})$ for models (7)–(9)

AC	N	ESSL	FAURE	HALTON	SOBOL	TMS
-1	4096	0.1668	0.1035	0.0256	0.0293	0.4288
-1	16384	0.0793	0.0834	0.0324	0.0412	0.0254
-1	65536	0.0496	0.0091	0.0082	0.0005	0.0170
0	4096	0.9017	0.1161	0.1455	0.1545	0.0348
0	16384	0.4195	0.0555	0.0178	0.0339	0.0604
0	65536	0.1270	0.0262	0.0206	0.0040	0.0343
1	4096	0.3100	0.1832	0.3481	0.3780	0.4206
1	16384	0.1954	0.0070	0.1458	0.1146	0.1425
1	65536	0.0821	0.0256	0.0302	0.0236	0.0981

perform better for a higher number of replications.

In general, the approximation errors resulting from Monte Carlo methods are much larger than the best results obtained by quasi Monte Carlo sequences. Since the results of the Monte Carlo simulation are stochastic, approximation errors of the order of magnitude of more than 1 percent may appear frequently in this setting. It should be noted that simulated effects in a macroeconomic simulation setup are often smaller than one percent. Consequently, the approximation error of Monte Carlo simulation can be larger than the simulated effect! The FAURE-sequences seem to be most robust in avoiding large approximation errors, which is confirmed by the findings for $N = 16\,384$ and $N = 65\,536$, with a single exception of model (8), $AC = -1$ and $N = 16\,384$ (see appendix for detailed results). Again, the (t, m, s) -net sometimes provides high quality approximations (and for model (8), $AC = -1$, it is even the overall best approximation) but fails for other instances, resulting in approximation errors even larger than the mean for the pseudo-random number generator. This puzzling result requires further analysis of the implementation of (t, m, s) -nets to simulations of this kind of time-series processes.

Instead of looking at the mean approximation error, one could also consider the number of times each simulation method provides the smallest approximation error. This criterion is in favour of FAURE-sequences for $N = 4096$ and $N = 16384$, with 4 and 6 out of 9 cases, respectively, while for

$N = 65536$ the SOBOL-sequence obtains the best results in 5 out of 9 cases.

Although the mean response is of crucial importance, further statistics of simulated time series are also of interest. As an example, Table 7 shows summary results for the simulated variance of x_{10} for the linear models and z_{10} for the nonlinear models, respectively. The entries are mean absolute approximation errors, while detailed results for individual models are reported in Tables 13, 14 and 15 in the appendix.

Table 7: Mean absolute approximation error (p.c.) of simulated $\text{VAR}(x_{10})$

AC	N	ESSL	FAURE	HALTON	SOBOL	TMS
linear models (1)-(6)						
-1	4096	2.6273	0.7445	1.8689	1.5617	1.1986
-1	16384	1.1000	0.1314	0.6628	0.3087	0.3457
-1	65536	0.5494	0.0628	0.2248	0.1474	0.2837
0	4096	2.6243	0.3144	0.6463	0.6301	0.2247
0	16384	1.0583	0.1445	0.2698	0.4471	0.1968
0	65536	0.5389	0.0336	0.0794	0.0809	0.0972
1	4096	2.6273	0.3062	0.9441	1.1128	0.7183
1	16384	1.1267	0.1229	0.2871	0.6198	0.2394
1	65536	0.6000	0.0262	0.1032	0.0829	0.1950
nonlinear models (7)-(9)						
-1	4096	2.3814	0.2177	0.3148	0.8182	0.6829
-1	16384	0.9093	0.1162	0.5064	0.5879	0.2441
-1	65536	0.4365	0.1334	0.1166	0.1203	0.1819
0	4096	2.0841	0.4220	1.4011	1.1623	0.3926
0	16384	1.1210	0.2506	0.6225	0.5045	0.3405
0	65536	0.4326	0.0553	0.1848	0.1767	0.6144
1	4096	2.0572	0.2889	0.2302	0.7470	1.4410
1	16384	1.1314	0.2431	0.3016	0.1399	0.4159
1	65536	0.5872	0.1005	0.1501	0.1236	0.2262

The order of magnitude of the approximation errors does not differ much between the linear and nonlinear models. Using antithetic variates does not preclude approximation errors of the variance. In fact, the results for ESSL indicate high and persistent approximation errors of the order of 2 to 3 percent for $N = 4096$, which decreases only slowly to about 1 percent for

$N = 16\,384$ and 0.4 to 0.6 percent for $N = 65\,536$ corresponding to the slow convergence rate of Monte Carlo methods.

In comparing among the quasi Monte Carlo methods, once again no clear ranking is obtained. Both for linear and nonlinear models, (t, m, s) -nets provide the best approximations for $N = 4096$, if autocorrelation is small. However, as soon as autocorrelation becomes relevant, other quasi Monte Carlo methods produce smaller approximation errors. In particular, for an increasing number of iterations, FAURE-sequences result in the smallest approximation errors for 12 and 15 out of 27 simulated models for $N = 16384$ and $N = 65536$, respectively. The (t, m, s) -net results in the smallest approximation error only 3 times for $N = 65536$. The higher convergence rate of quasi Monte Carlo methods is also mirrored by the simulation results, as the typical approximation error of the variance estimate for the FAURE-sequences for the linear models decreases from 0.3 to 0.75 percent for $N = 4096$ to 0.02 to 0.06 percent for $N = 65\,536$. For the nonlinear models this trend is less pronounced.

5 Conclusion

In this paper, the use of quasi Monte Carlo methods for the purpose of simulating time-series processes is analyzed. Based on theoretical results on the discrepancy and integration error bounds, different quasi Monte Carlo methods are compared to standard pseudo-random number generators. In particular, the use of (t, m, s) -nets is motivated by a strong theoretical background. However, the centred L_2 -discrepancy does not indicate a clear ranking among the quasi Monte Carlo methods, nevertheless they all behave much better than Monte Carlo methods.

The application of Monte Carlo and quasi Monte Carlo methods to the simulation of linear and nonlinear time-series models provides further evidence of the superiority of quasi Monte Carlo methods. In the case of linear models, the approximation error of simulated expectation can be reduced to zero by the use of antithetic variates. However, (t, m, s) -nets provide unbiased results in this case without using antithetic variates. For the approximation error of expectation for the nonlinear models and the approximation error of the simulated variance, the quasi Monte Carlo methods are clearly superior to the Monte Carlo approaches. However, again there is no clear ranking among the quasi Monte Carlo methods. Any superiority of (t, m, s) -

nets, as might be expected on theoretical grounds, is not found.⁶ Rather, it is FAURE and SOBOL-sequences that provide the overall best approximations here.

We concentrate only on the simulated distribution of x_{10} , y_{10} and z_{10} in this paper. However, further work could extend the simulation period and, in due course, the dimensionality of the error space. Further, a detailed investigation of potential improvements of (t, m, s) -nets is important, since one of the advantages of (t, m, s) -nets is that, with “nice” tweaking, uniformity can be improved while all the good theoretical properties remain the same. Such testing is now underway.

⁶Hellekalek (1998, pp. 68f) reports on results from a comparison of good lattice point sets and (t, m, s) -nets for numerical integration. He also finds that despite of the nice theoretical properties of (t, m, s) -nets, no superiority can be detected, quite to the contrary.

A Parameters of Simulated Models

Table 8: Parameters of simulated linear models

Model	AC	Parameters
(1)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.9, \sigma_\varepsilon = 0.2$
	0	$\alpha_0 = 0.1, \alpha_1 = 0, \sigma_\varepsilon = 0.2$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.9, \sigma_\varepsilon = 0.2$
(2)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.8, \alpha_2 = -0.1, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05$
	0	$\alpha_0 = 0.1, \alpha_1 = 0.1, \alpha_2 = -0.1, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.8, \alpha_2 = 0.1, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05$
(3)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.8, \alpha_2 = -0.1, \sigma_\varepsilon = 0.2$
	0	$\alpha_0 = 0.1, \alpha_1 = -0.5, \alpha_2 = 0.5, \sigma_\varepsilon = 0.2$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.8, \alpha_2 = 0.1, \sigma_\varepsilon = 0.2$
(4)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.8, \alpha_2 = -0.1, \alpha_3 = 0.5, \alpha_4 = 0.1, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = -0.3, \beta_2 = 0.2, \beta_3 = -0.5, \beta_4 = -0.3, \sigma_{\varepsilon_2} = 0.05$
	0	$\alpha_0 = 0.1, \alpha_1 = -0.5, \alpha_2 = 0.4, \alpha_3 = 0.5, \alpha_4 = 0.1, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = 0.5, \beta_2 = -0.1, \beta_3 = -0.2, \beta_4 = 0.3, \sigma_{\varepsilon_2} = 0.05$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.8, \alpha_2 = 0.1, \alpha_3 = -0.5, \alpha_4 = -0.1, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = -0.1, \beta_1 = 0.3, \beta_2 = -0.2, \beta_3 = 0.5, \beta_4 = 0.3, \sigma_{\varepsilon_2} = 0.05$
(5)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.9, \alpha_2 = 0, \lambda = -0.1, \beta = 0.5, \sigma_\varepsilon = 0.2$
	0	$\alpha_0 = 0.1, \alpha_1 = 0.1, \alpha_2 = 0, \lambda = -0.4, \beta = 1.0, \sigma_\varepsilon = 0.2$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.9, \alpha_2 = 0, \lambda = -0.1, \beta = 0.5, \sigma_\varepsilon = 0.2$
(6)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.8, \alpha_2 = -0.1, \lambda_1 = -0.1, \gamma = 0.5, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = -0.1, \beta_2 = -0.8, \lambda_2 = 0.2, \sigma_{\varepsilon_2} = 0.05$
	0	$\alpha_0 = 0.1, \alpha_1 = 0.2, \alpha_2 = -0.2, \lambda_1 = -0.5, \gamma = 1.0, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.2, \lambda_2 = 0.2, \sigma_{\varepsilon_2} = 0.05$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.8, \alpha_2 = 0.1, \lambda_1 = -0.1, \gamma = 0.5, \sigma_{\varepsilon_1} = 0.2$
		$\beta_0 = 0.1, \beta_1 = -0.1, \beta_2 = -0.8, \lambda_2 = 0.2, \sigma_{\varepsilon_2} = 0.05$

Table 9: Parameters of simulated nonlinear models

Model	AC	Parameters
(7)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.8, \alpha_2 = -0.1, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05$
	0	$\alpha_0 = 0.1, \alpha_1 = 0.1, \alpha_2 = -0.1, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.8, \alpha_2 = 0.1, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05$
(8)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.8, \alpha_2 = -0.1, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05, \rho = 10$
	0	$\alpha_0 = 0.1, \alpha_1 = 0.1, \alpha_2 = -0.1, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05, \rho = 10$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.8, \alpha_2 = 0.1, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.6, \sigma_{\varepsilon_2} = 0.05, \rho = 10$
(9)	-1	$\alpha_0 = 0.1, \alpha_1 = -0.8, \alpha_2 = -0.1, \lambda_1 = -0.2, \gamma = 0.5, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = -0.7, \lambda_2 = 0.05, \sigma_{\varepsilon_2} = 0.05, \rho = 10$
	0	$\alpha_0 = 0.1, \alpha_1 = -0.1, \alpha_2 = 0.2, \lambda_1 = -0.1, \gamma = 0.2, \sigma_{\varepsilon_1} = 0.1$ $\beta_0 = 0.1, \beta_1 = -0.2, \beta_2 = 0.1, \lambda_2 = 0.05, \sigma_{\varepsilon_2} = 0.05, \rho = 10$
	1	$\alpha_0 = 0.1, \alpha_1 = 0.8, \alpha_2 = 0.1, \lambda_1 = -0.2, \gamma = 0.5, \sigma_{\varepsilon_1} = 0.2$ $\beta_0 = 0.1, \beta_1 = 0.2, \beta_2 = 0.7, \lambda_2 = 0.05, \sigma_{\varepsilon_2} = 0.05, \rho = 10$

B Detailed Simulation Results

Table 10: Approximation error (p.c.) of simulated $E(x_{10})$ for models (7)–(9), $N = 4096$

Model	AC	ESSL	FAURE	HALTON	SOBOL	TMS
7	-1	0.0640	-0.0174	-0.0105	-0.0412	0.1322
7	0	1.6692	-0.2725	0.3756	0.4321	0.0471
7	1	0.2461	-0.2403	-0.7360	-0.4444	0.6314
8	-1	0.3990	0.2870	-0.0538	0.0316	1.1276
8	0	1.0200	0.0700	-0.0291	0.0060	0.0495
8	1	0.5591	-0.2816	-0.0870	0.4040	-0.5546
9	-1	0.0375	-0.0061	0.0126	0.0152	0.0267
9	0	0.0160	0.0059	0.0318	0.0256	-0.0079
9	1	0.1247	0.0278	-0.2213	-0.2855	-0.0757

Table 11: Approximation error (p.c.) of simulated $E(x_{10})$ for models (7)–(9),
 $N = 16\,384$

Model	AC	ESSL	FAURE	HALTON	SOBOL	TMS
7	-1	0.0167	-0.0137	0.0154	0.0141	0.0330
7	0	0.6914	-0.0909	-0.0019	0.0906	0.1036
7	1	0.0936	-0.0140	-0.2466	-0.1619	0.3389
8	-1	0.2033	0.2361	0.0795	-0.1070	-0.0359
8	0	0.5587	-0.0746	-0.0400	0.0019	0.0722
8	1	0.4347	-0.0041	-0.1862	0.1513	-0.0263
9	-1	0.0178	0.0006	0.0023	0.0025	0.0073
9	0	0.0084	-0.0011	0.0114	0.0092	-0.0055
9	1	0.0579	-0.0027	-0.0045	-0.0304	0.0623

Table 12: Approximation error (p.c.) of simulated $E(x_{10})$ for models (7)–(9),
 $N = 65\,536$

Model	AC	ESSL	FAURE	HALTON	SOBOL	TMS
7	-1	0.0202	-0.0180	-0.0045	-0.0012	-0.0049
7	0	0.2626	-0.0460	0.0537	0.0027	-0.0400
7	1	0.0590	-0.0265	-0.0375	-0.0334	0.1183
8	-1	0.1231	0.0090	-0.0198	0.0003	0.0453
8	0	0.1156	-0.0321	0.0049	-0.0067	-0.0534
8	1	0.1677	-0.0487	-0.0471	-0.0304	0.1142
9	-1	0.0054	-0.0002	-0.0004	0.0001	0.0006
9	0	0.0029	-0.0004	0.0032	0.0027	-0.0096
9	1	0.0194	0.0016	0.0060	0.0070	0.0618

Table 13: Approximation error (p.c.) of simulated $\text{VAR}(x_{10})$ for models (1)–(9), $N = 4096$

Model	AC	ESSL	FAURE	HALTON	SOBOL	TMS
1	-1	2.7944	-0.4088	0.1896	-0.4764	-0.5622
1	0	2.7069	0.1361	-0.2262	-0.6622	-0.0075
1	1	2.8995	0.9081	-2.7780	-2.1189	-0.7204
2	-1	2.8031	-0.3518	-0.3942	-0.2245	0.0852
2	0	2.6641	0.4078	-0.6673	-0.7425	0.1207
2	1	3.0371	0.5895	-2.0199	-2.0955	-0.7507
3	-1	2.5023	-0.3141	-0.0608	-1.4554	-0.2188
3	0	2.8201	-0.2133	-0.0154	-0.1013	-0.2043
3	1	2.4065	0.9170	-2.7790	-2.1114	-0.8185
4	-1	2.5210	0.3218	-0.2787	-0.1623	0.7929
4	0	2.9119	0.2802	-0.3744	-0.2554	0.3128
4	1	2.1302	1.3037	-0.7596	-0.7898	1.3810
5	-1	1.9472	0.0619	-1.8169	-1.0942	-0.4224
5	0	2.8734	0.2311	-0.7314	0.1552	-0.0404
5	1	2.5937	0.9666	-2.5146	-2.2225	1.4534
6	-1	2.9617	0.1412	-1.2990	-1.3933	-1.5628
6	0	1.7691	0.6178	-1.8631	-1.8643	-0.6622
6	1	2.9305	-0.0199	-1.9878	-1.9028	-2.7333
7	-1	2.4556	-0.3220	-0.2353	-0.0114	-0.5203
7	0	2.3937	0.0079	-0.3324	-0.5165	-0.0129
7	1	2.0748	0.1224	-0.0603	0.0285	0.2755
8	-1	2.0179	-0.0472	-0.0096	1.5756	0.0509
8	0	1.0749	-0.7839	0.0513	0.0941	-0.2327
8	1	1.8836	0.5618	-0.4989	-1.7387	1.0353
9	-1	2.6708	0.2837	-0.6996	-0.8674	-1.4774
9	0	2.7836	-0.4743	-3.8196	-2.8764	-0.9321
9	1	2.2133	-0.1825	0.1315	0.4739	-3.0122

Table 14: Approximation error (p.c.) of simulated $\text{VAR}(x_{10})$ for models (1)–(9), $N = 16\,384$

Model	AC	ESSL	FAURE	HALTON	SOBOL	TMS
1	-1	0.7462	-0.2385	-0.0133	-0.7415	0.0564
1	0	0.7390	-0.1687	-0.1951	-0.1869	-0.0019
1	1	1.3371	0.0809	-1.1133	-0.1955	0.5964
2	-1	0.6537	-0.2610	-0.0883	-0.0924	0.1590
2	0	0.7441	-0.1958	-0.2110	-0.2051	0.0139
2	1	1.1396	0.0645	-0.5994	-0.6130	0.2772
3	-1	1.1345	-0.1101	-0.0858	-1.4008	-0.2342
3	0	1.3185	-0.2736	0.0088	-0.3996	0.2676
3	1	1.0921	0.0685	-1.1174	-0.1947	0.5845
4	-1	1.3799	-0.1380	-0.0353	-0.0790	0.2750
4	0	1.5720	-0.1517	-0.0932	-0.1349	0.4148
4	1	1.0020	-0.0463	-0.1946	-0.1802	-0.0260
5	-1	0.7736	-0.1524	-0.5398	-0.4296	0.0554
5	0	1.0681	-0.0589	-0.5764	1.2126	0.2193
5	1	0.9761	0.1426	-0.8784	-0.5790	0.4200
6	-1	1.5864	0.0337	-0.4488	-0.4549	0.5381
6	0	0.9081	0.0184	-0.5344	-0.5437	0.2633
6	1	1.5391	0.1893	-0.5847	-0.6104	0.2880
7	-1	1.0579	-0.2570	-0.1835	-0.0680	0.0064
7	0	1.1878	-0.2875	-0.2010	-0.0772	0.1276
7	1	1.4324	0.1263	-0.2725	0.2138	0.7707
8	-1	0.5488	-0.0829	-1.1372	1.4788	0.5664
8	0	0.6057	-0.2426	-0.1964	0.2356	0.0793
8	1	1.0049	0.3990	0.3297	-0.0186	-0.0913
9	-1	1.1212	-0.0087	-0.1986	-0.2169	0.1594
9	0	1.5695	0.2219	-1.4699	-1.2007	0.8147
9	1	0.9568	0.2039	-0.3026	-0.1873	-0.3857

Table 15: Approximation error (p.c.) of simulated $\text{VAR}(x_{10})$ for models (1)–(9), $N = 65\,536$

Model	AC	ESSL	FAURE	HALTON	SOBOL	TMS
1	-1	0.6406	0.0291	-0.0216	0.0447	0.0307
1	0	0.3249	0.0243	-0.0495	-0.0581	-0.0004
1	1	0.5642	-0.0733	-0.3719	-0.2253	0.2279
2	-1	0.4866	-0.0525	-0.0239	-0.0400	-0.2568
2	0	0.3073	-0.0318	-0.0524	-0.0627	-0.0090
2	1	0.4158	-0.0144	-0.1986	-0.1770	0.3003
3	-1	0.5937	0.0456	-0.0430	0.0307	0.1257
3	0	0.7287	0.0085	-0.0125	0.0359	-0.0406
3	1	0.5063	-0.0699	-0.3696	-0.2238	0.2416
4	-1	0.9000	-0.0464	-0.0168	-0.0458	-0.2188
4	0	0.8518	-0.0479	-0.0303	-0.0603	-0.2000
4	1	0.5283	-0.0488	-0.0687	-0.0434	-0.1970
5	-1	0.5721	-0.0044	-0.2062	-0.0605	0.1290
5	0	0.3829	-0.0733	-0.1536	-0.1093	-0.0844
5	1	0.4607	-0.0724	-0.3209	-0.1992	0.0999
6	-1	0.4778	-0.0175	-0.1331	-0.1389	0.3652
6	0	0.6379	-0.0161	-0.1783	-0.1593	0.2488
6	1	0.7502	0.0600	-0.1939	-0.1529	0.6792
7	-1	0.5035	0.0254	-0.0122	-0.0302	-0.2431
7	0	0.2948	-0.0562	-0.0166	-0.0833	-0.1012
7	1	0.5982	0.0372	-0.2093	-0.2279	0.4391
8	-1	0.4029	-0.3629	-0.3061	0.2798	0.0904
8	0	0.1959	0.0145	0.0464	-0.0530	-0.3880
8	1	0.4402	0.2163	0.0912	-0.0319	0.0944
9	-1	0.4030	-0.0120	-0.0316	-0.0511	0.2121
9	0	0.8071	0.0951	-0.4915	-0.3940	1.3542
9	1	0.7232	0.0480	-0.1499	-0.1111	0.1449

References

- Acworth, P., M. Broadie and P. Glasserman (1998). A comparison of some Monte Carlo and quasi-Monte Carlo methods 1996. In: *Lecture Notes in Statistics* (H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof, Eds.). Vol. 127. pp. 1–18.
- Bierbrauer, J. and Y. Edel (1999). *Some good binary (t, m, s) -nets*. Preprint.
- Bratley, P. and B. L. Fox (1988). Sobol’s quasirandom sequence generator for multivariate quadrature and optimization. *ACM TOMS* **14**(1), 88–100.
- Caffisch, R. E. and W. Morokoff (1996). Valuation of mortgage-backed securities using the quasi-Monte Carlo method. Technical report. Department of Mathematics UCLA.
- Clayman, A. T., K. M. Lawrence, G. L. Mullen, H. Niederreiter and N. J. A. Sloane (2000). Updated tables of parameters of (t, m, s) -nets. *Journal of Combinatorial Designs* p. forthcoming.
- Ericsson, N. R. and J. Marquez (1998). A framework for economic forecasting. *Econometrics Journal* **1**, C228–C266.
- Fang, K.-T., C.-X. Ma and P. Winker (2000). Centered l_2 -discrepancy of random sampling and latin hypercube design, and construction of uniform designs. *Mathematical Computation* p. forthcoming.
- Franz, W., K. Göggelmann, M. Schellhorn and P. Winker (2000). Quasi-Monte Carlo methods in stochastic simulations. *Empirical Economics* **25**, 247–259.
- Gentle, J. E. (1998). *Random Number Generation and Monte Carlo Methods*. Springer. New York.
- Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik* **2**, 84–90.
- Hellekalek, P. (1998). On the assessment of random and quasi-random point sets. In: *Random and Quasi-Random Point Sets. Lecture Notes in Statistics* (P. Hellekalek and G. Larcher, Eds.). Vol. 138. Springer. New York. pp. 49–108.

- Hickernell, F. J. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of Computation* **67**, 299–322.
- Larcher and Traunfellner (1994). On the numerical integration of walsh series by number-theoretic methods. *Math. Comp.* **63**, 277–291.
- L'Ecuyer, P. and P. Hellekalek (1998). Random number generators: Selection criteria and testing. In: *Random and Quasi-Random Point Sets. Lecture Notes in Statistics* (P. Hellekalek and G. Larcher, Eds.). Vol. 138. Springer. New York. pp. 223–265.
- Li, J. and G. Mullen (2000). Parallel computing of a quasi-Monte Carlo algorithm for valuing derivatives. *Parallel Computing* **26**(5), 641–653.
- Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM. Philadelphia, PE.
- Owen, A. B. (1997). Monte Carlo variance of scrambled net quadrature. *SIAM Journal of Numerical Analysis* **34**, 1884–1910.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery (1992). *Numerical Recipes in Fortran 77*. 2nd ed.. Cambridge University Press. New York, NY.
- Ripley, B. D. (1987). *Stochastic Simulation*. Wiley. New York.
- Winker, P. (1999). Quasi-Monte Carlo policy simulations in a macroeconomic disequilibrium model. In: *Proceedings of the 14th World Congress International Federation of Automatic Control*. Elsevier. Oxford. pp. 45–50.
- Winker, P. and K.-T. Fang (1997). Application of threshold accepting to the evaluation of the discrepancy of a set of points. *SIAM Journal on Numerical Analysis* **34**(5), 2028–2042.