New inputs and methods for Markov chain quasi-Monte Carlo

Su Chen and Makoto Matsumoto and Takuji Nishimura and Art B. Owen

Abstract We present some new results on incorporating quasi-Monte Carlo rules into Markov chain Monte Carlo. First, we present some new constructions of points, fully equidistributed LFSRs, which are small enough that the entire point set can be used in a Monte Carlo calculation. Second, we introduce some antithetic and round trip sampling constructions and show that they preserve the completely uniformly distributed property necessary for QMC in MCMC. Finally, we also give some new empirical results. We see large improvements in sampling some GARCH and stochastic volatility models.

1 Introduction

Simple Monte Carlo sampling has two limitations when used in practice. First, it converges only at a slow rate, with root mean squared error $O(n^{-1/2})$. Second, on many challenging problems there is no known way to generate independent samples from the desired target distribution. Quasi-Monte Carlo (QMC) methods have been developed to address the first problem, yielding greater accuracy, while Markov chain Monte Carlo (MCMC) methods have been developed for the second problem yielding wider applicability.

Su Chen
Stanford University, e-mail: suchenpku@gmail.com

Makoto Matsumoto
University of Tokyo, e-mail: matumoto@ms.u-tokyo.ac.jp

Takuji Nishimura
Yamagata University, e-mail: nisimura@sci.kj.yamagata-u.ac.jp

Art B. Owen
Stanford University e-mail: owen@stanford.edu

It is natural then to seek to combine these two approaches. There were some early attempts by Chentsov [2] and Sobol' [15] around 1970. The problem has been revisited more recently. See for example [11] and [13]. For a survey of recent combinations of QMC and MCMC see [1].

QMC uses n points in $[0,1)^d$, where typically $n \gg d$. MCMC uses one long stream of IID $U[0,1)$ inputs, which we call the 'driving sequence'. It has effectively $n=1$ with $d \to \infty$, quite unlike QMC. Chentsov's key insight was to use completely uniformly distributed points to drive the MCMC. That is the approach taken in [13].

The contributions of this paper are as follows. First, we present some new point sets, small fully equidistributed LFSRs, to use as driving sequences for MCMC. Second, we show how some antithetic sampling strategies within the driving sequence still give rise to valid driving sequences. Third, we present some new empirical findings.

The outline of the paper is as follows. Section 2 defines some key notions that we need. Section 3 describes the mini-twisters that we use. Section 4 presents our antithetic extensions of the driving sequence. We give new empirical results in Section 5. Our conclusions are in Section 6.

2 Background

In this section we describe completely uniformly distributed points and some generalizations that we need. We also give a sketch of MCMC. For more details on the latter, the reader may consult [12, 14].

2.1 Completely uniformly distributed sequences

Here we define some notions of completely uniformly distributed sequences. We assume that the reader is familiar with the star discrepancy D_n^{*d}.

Let $u_i \in [0,1]$ for $i \geq 1$. For integer $d \geq 1$, define

$$\bar{u}_i^{(d)} = (u_i, u_{i+1}, \dots, u_{i+d-1}), \quad \text{and,} \tag{1}$$

$$u_i^{(d)} = (u_{i(d-1)+1}, u_{i(d-1)+2}, \dots, u_{id}). \tag{2}$$

Both $\bar{u}_i^{(d)}$ and $u_i^{(d)}$ are made up of consecutive d-tuples from u_i, but the former are overlapping while the latter are non-overlapping.

Definition 1. The infinite sequence u_i is *completely uniformly distributed* (CUD), if

$$\lim_{n\to\infty} D_n^{*d}(\bar{u}_1^{(d)}, \dots, \bar{u}_n^{(d)}) = 0 \tag{3}$$

for all integer $d \geq 1$.

If u_i are CUD, then

$$\lim_{n\to\infty} D_n^{*d}(u_1^{(d)},\dots,u_n^{(d)}) = 0 \tag{4}$$

holds for all $d \geq 1$. Conversely (see [2]), if (4) holds for all $d \geq 1$ then u_i are CUD.

For randomized points u_i it is useful to have the following definition.

Definition 2. The infinite sequence u_i is *weakly completely uniformly distributed* (WCUD), if

$$\lim_{n\to\infty} \Pr\bigl(D_n^{*d}(\bar{u}_1^{(d)},\dots,\bar{u}_n^{(d)}) > \varepsilon\bigr) = 0 \tag{5}$$

for all $\varepsilon > 0$ and integer $d \geq 1$.

To better model driving sequences of finite length, there are also triangular array versions of these definitions. A triangular array has elements $u_{n,i} \in [0,1]$ for $i = 1,\dots,n$ and $n \in \mathscr{N}$ where $\mathscr{N}$ is an infinite set of nonnegative integers. This triangular array is CUD if $\lim_{n\to\infty} D_n^{*d}(\bar{u}_{n,1}^{(d)},\dots,\bar{u}_{n,n-d+1}^{(d)}) = 0$ for all integer $d \geq 1$ as $n \to \infty$ through values in $\mathscr{N}$. There is a similar definition for weakly CUD triangular arrays.

For further background on CUD sequences see [10]. For triangular arrays and sufficient conditions for weak CUD see [18]. The usual construction for WCUD sequences applies Cranley-Patterson [4] rotation to a CUD sequence [18].

2.2 *Markov chain Monte Carlo*

A typical MCMC run begins with a starting point X_0. Then, for $i \geq 1$

$$X_i = \phi(X_{i-1}, u_i^{(m)}) \tag{6}$$

where $u_i^{(m)}$ is defined at (2) in terms of an IID driving sequence $u_i \sim U[0,1]$. This version of MCMC assumes that each update consumes exactly m elements of the driving sequence. MCMC sometimes uses more general schemes, and it's QMC version can too. See [18]. In this paper we will suppose that (6) holds. The CUD property for a driving sequence has to apply to all integer values $d \geq 1$, not just $d = m$.

The update function $\phi(\cdot,\cdot)$ is chosen so that as $n \to \infty$, the distribution of X_n approaches a desired distribution π. If we are interested in the quantity

$$\mu = \int f(x)\pi(x)\,\mathrm{d}x$$

we estimate it by

$$\hat{\mu} = \frac{1}{n}\sum_{i=b+1}^{b+n} f(X_i)$$

where $b \geq 0$ is a burn-in parameter. For simplicity, we take $b = 0$.

The typical behavior of MCMC is that $f(X_i)$ and $f(X_{i+k})$ have a correlation that decreases as ρ^k, where $|\rho| < 1$. As a result $\hat{\mu}$ ordinarily approaches μ with an RMSE of $O(1/\sqrt{n})$. There are however pathologies in which the chain can get stuck. Such failure to mix can result in lack of convergence. Considerable creativity goes into constructing the update function ϕ, to obtain a rapidly mixing Markov chain. The details are beyond the scope of this article. See [12, 14]. Our focus is on replacing IID driving sequences by CUD ones in chains that do mix well. CUD driving sequences do not repair faulty choices of $\phi()$.

2.3 QMC in MCMC results

Much of the literature combining QMC with MCMC is empirical. Here we provide a short summary of the theoretical results that underpin the work described in this paper.

Running an MCMC algorithm with deterministic inputs gives output that is not Markovian. As a result, there is potential for error. There is however a safe harbor in replacing IID points by (W)CUD points.

Suppose first that $X_i \in \Omega = \{\omega_1, \dots, \omega_M\}$. Such finite state spaces are technically simpler. If X_i is sampled by inversion and $\min_{1\le j,k\le M} \Pr(X_i = \omega_j \mid X_{i-1} = \omega_k) > 0$ then [2] shows that a CUD driving sequence gives consistency, i.e.,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} 1_{X_i=\omega_j} = \pi(\omega_j) \tag{7}$$

for $j = 1, \dots, M$. Conversely non-CUD points will fail for some Markov chain. For random driving sequences, the consistency condition is

$$\lim_{n\to\infty} \Pr\Bigl(|\frac{1}{n} \sum_{i=1}^{n} 1_{X_i=\omega_j} - \pi(\omega_j)| > \varepsilon\Bigr) = 0, \quad \forall \varepsilon > 0. \tag{8}$$

It is seldom possible to sample the transitions by inversion. The Metropolis-Hastings update [8] is usually used instead. For the Metropolis-Hastings update, consistency (7) still holds (see [13]) under three conditions. First, the driving sequence must be CUD. Second, the function ϕ must be one for which an IID $U(0,1)$ driving sequence achieves weak consistency (8). (It could include some zero transition probabilities.) Finally, there is a technical condition that pre-images in $[0,1]^m$ for transitions from one state to another must all give Jordan measurable sets of $u_i^{(m)}$. To summarize, if Metropolis-Hastings sampling on a finite state space is weakly consistent with IID sampling, then it is consistent with CUD sampling. It is then also weakly consistent with weakly CUD sampling.

The case of continuous state spaces was taken up by [1]. The same conclusion holds. If an MCMC algorithm, either Metropolis-Hastings (their Theorem 2) or Gibbs sampling (Theorem 3), is weakly consistent when driven by IID $U(0,1)$

inputs, then it is consistent when driven by CUD inputs and is weakly consistent when driven by WCUD inputs. In the continuous state space setting consistency means having the empirical probability of hyperrectangles match their probability under π. The dimension of these hyperrectangles equals that of the point X_i, which is not necessarily m. The technical conditions for Metropolis-Hastings involve Jordan measurability of pre-images for multistage transitions, while those for Gibbs sampling require a kind of contraction mapping.

3 New small LFSR constructions

Levin [10] gives several constructions for CUD points, but they are not convenient to implement. Tribble [17] used small versions of multiple congruential generators and linear feedback shift registers (LFSRs). His best results were for LFSRs but he had only a limited number of them.

In this section we present some new LFSR type sequences with lengths 2^d-1 for all integers $10 \le d \le 32$. Their consecutive blocks of various lengths obey an equidistribution property. That makes them suitable for applications which require low discrepancy for vectors formed by taking overlapping consecutive points.

Let P be an integer and u_i for $i=0,1,2,\dots$ be a sequence of real numbers in the half-open interval $[0,1)$ with period P. Let

$$u_i = \sum_{j=1}^{\infty} b_{i,j} 2^{-j} \tag{9}$$

be 2-adic expansion of u_i.

We associate to the sequence (u_i) a multi-set (namely, a set with multiplicity of each element counted) Ψ_k as follows:

$$\Psi_k := \{\bar{u}_i^{(k)} \mid 0 \le i \le P-1\}.$$

The multi-set Ψ_k consists of k-dimensional points obtained as overlapping k-tuples in the sequence for one period. For some positive integer v, we divide the interval $[0,1)$ into 2^v equal pieces. This yields a partition of the unit hyper cube $[0,1)^k$ into 2^{kv} cubic cells of equal size. Following [16] (cf. [9]), we say that the sequence (x_i) is *k-dimensionally equidistributed* with v-bit accuracy if each cell contains exactly same number of points of Ψ_k, except for the cell at the origin that contains one less. The largest value of such k is called the dimension of equidistribution with k-bit accuracy and denoted by $k(v)$.

Let $M(k,v)$ denote the set of $k \times v$ binary matrices. The above condition is equivalent to that the multiset of $k \times v$ matrices

$$\Phi_{k,v} := \{(b_{i+s,j})_{s=0,\dots,k-1;j=1,\dots,v} \mid 0 \le i \le P-1\}$$

contains every element of $M(k,v)$ with the same multiplicity, except the 0 matrix with one less multiplicity. Since there are $2^{kv}-1$ nonzero such matrices, we have an inequality $2^{kv}-1 \leq P$. In the following examples, $P = 2^d - 1$, and hence $k(v) \leq \lfloor d/v \rfloor$. A sequence (x_i) of period 2^d-1 is said to be *fully equidistributed* (FE) if the equality holds for all $1 \leq v \leq d$. This is a special case of the maximal equidistribution property [16, 9] where d is equal to the number of binary digits of the elements of the sequence.

Next we define $GF(2)$-linear sequence generators. Let $S := GF(2)^d$ be the state space, $F : S \to S$ be a $d \times d$ $GF(2)$-matrix F (multiplication from left) representing the state transition, and $o : S \to GF(2)^d$ be another $d \times d$-matrix for the output function. Choose an initial state $s_0 \neq 0$. The state transition is given by $s_i = F(s_{i-1})$ for $i \geq 1$. The i-th output $o(s_i) = (b_{i,1}, \ldots, b_{i,d})$ is regarded as a real number u_i by

$$u_i = \sum_{j=1}^{d} b_{i,j} 2^{-j}.$$

We could add random digits beyond the d'th, but they would not affect the FE property.

Assume that F has the maximal period 2^d-1. Then, every nonzero element of S is on one orbit. Thus, the multiset $\Psi_{k,v} \cup \{0\}$ is the image of the $GF(2)$-linear map

$$o_{k,v} : S \to M(k,v); \quad s_i \mapsto (b_{i+s,j})_{0 \leq s \leq k-1, 1 \leq j \leq v}, \ \ 0 \mapsto 0$$

and k-dimensional equidistribution with v-bit accuracy is equivalent to the surjectivity of this linear map (since the inverse image of any element is an affine space of the same dimension), and hence easy to check for small d such as $d < 100$.

Let $(a_{d-1}, a_{d-2}, \ldots, a_0) \in GF(2)^d$. Let $f : S \to S$ be the matrix defined by $(b_0, b_1, \ldots, b_{d-1}) \mapsto (b_1, b_2, \ldots, b_{d-1}, \sum_{i=0}^{d-1} a_i b_i)$. This transition is known as a linear feedback shift register. It attains the maximal period 2^d-1 if and only if its characteristic polynomial $t^d + a_{d-1}t^{d-1} + \cdots + a_1 t + a_0$ is primitive. For each $d = 10, 11, \ldots, 32$, we take a primitive polynomial of degree d from a list in [7] and let f_d be the associated transition function as above.

Let $F := f_d^s$ for some integer s. Then F has the maximal period 2^d-1 if and only if s and 2^d-1 are coprime. We choose the output function o to be the identity, and search for s in ascending order among the integers coprime to 2^d-1 such that $F = f_d^s$ satisfies the FE condition. For each d, we found such s in the range $1 < s < 4000$. We select one s for each d, and call it s_d. See Table 1 for the values we used. We compute $F_d = f_d^{s_d}$ as a $d \times d$ matrix, and then implement the FE $GF(2)$-linear generator with transition function F_d and identity output function.

The FE condition gives stratification over congruent subcubes. Because any rectangle in $[0,1]^d$ can be closely approximated by subcubes, the d dimensional discrepancy tends to 0 for points formed from an LFSR satisfying the FE condition. Thus an infinite sequence of FE-LFSRs provides a triangular array that is CUD.

Table 1 Parameters s_d for LFSRs of length $P = 2^d - 1$.

d	s_d	d	s_d	d	s_d	d	s_d
10	115	16	283	22	1336	28	2573
11	291	17	514	23	1236	29	2633
12	172	18	698	24	1511	30	2423
13	267	19	706	25	1445	31	3573
14	332	20	1304	26	1906	32	3632
15	388	21	920	27	1875		

4 Antithetic and round trip sampling

Some Markov chains are closely connected to random walks. For example, Metropolis samplers accept or reject proposals made by a random walk process. For a random walk with increments of mean zero, the expected value of X_n is X_0. Similarly, for an autoregressive process such as $X_i = \rho X_{i-1} + \sqrt{1-\rho^2} Z_i$ for Gaussian Z_i, we have $\mathbb{E}(X_n) = X_0$.

We can sample an autoregression by taking

$$X_i = \rho X_{i-1} + \sqrt{1-\rho^2}\Phi^{-1}(u_i) \tag{10}$$

where the driving sequence u_i are IID $U(0,1)$.

In an *antithetic driving sequence*, we take

$$u_1, u_2, \dots, u_n, 1-u_1, 1-u_2, \dots, 1-u_n.$$

That is, the second half of the sequence simply replays the ones complement of the first half. In a *round trip driving sequence*, we take

$$u_1, u_2, \dots, u_n, 1-u_n, 1-u_{n-1}, \dots, 1-u_1.$$

The sequence steps backwards the way it came.

With either of these driving sequences, an autoregression (10) would satisfy $X_{2n} = X_0 \equiv \mathbb{E}(X_{2n} \mid X_0)$. A random walk would also end where it started. A Markov chain driven by symmetric random walk proposals would be expected to end up close to where it started if most of its proposals were accepted.

Inducing the chain to end up at or near to its expected value should bring a variance reduction. To ensure that the points asymptotically cover the space properly, we require the driving sequence to be (W)CUD. The sampling methods we use are similar to antithetic sampling. The antithetic sampling here differs from that of [6] who sample two chains. A related method in [3] also runs two chains, the second time-reversed one driven by $u_n, \dots, u_1$. The second half of the round trip sequence is time reversed and antithetic to the first half.

If the updates take points $u_i^{(m)} \in [0,1]^m$ for $m > 1$ and $i = 1, \dots, \lfloor n/m \rfloor$, then a reasonable alternative to both of these sampling methods is to use a driving sequence of $2m\lfloor n/m\rfloor$ numbers constructed from all m components of the points

$$u_1^{(m)}, u_2^{(m)}, \dots, u_{\lfloor n/m\rfloor}^{(m)}, 1-u_1^{(m)}, 1-u_2^{(m)}, \dots, 1-u_{\lfloor n/m\rfloor}^{(m)}, \quad \text{or,}$$
$$u_1^{(m)}, u_2^{(m)}, \dots, u_{\lfloor n/m\rfloor}^{(m)}, 1-u_{\lfloor n/m\rfloor}^{(m)}, 1-u_{\lfloor n/m\rfloor-1}^{(m)}, \dots, 1-u_1^{(m)}.$$

We call these m-fold antithetic and m-fold round trip driving sequences.

For round trip and antithetic sequences, we will use some results about discrepancies. If $v_1,\dots,v_n$ and $w_1,\dots,w_n$ are points in $[0,1]^d$ then

$$D_{2n}^{*d}(v_1,\dots,v_n,w_1,\dots,w_n) \le \frac{1}{2}\bigl(D_n^{*d}(v_1,\dots,v_n) + D_n^{*d}(w_1,\dots,w_n)\bigr), \tag{11}$$
$$D_n^{*d}(1-v_1,\dots,1-v_n) \le 2^d D_n^{*d}(v_1,\dots,v_n), \quad \text{and} \tag{12}$$
$$\Bigl| D_{n+k}^{*d}(v_1,\dots,v_{n+k}) - D_n^{*d}(v_1,\dots,v_n)\Bigr| \le \frac{k}{n+k}. \tag{13}$$

Equation (11) is simple to prove, equation (12) follows from the well known bound relating discrepancy to star discrepancy and equation (13) is Lemma 4.2.2 of [17].

For m-fold versions we need another result. In the case $m=3$ the second half of the driving sequence has entries

$$u_3, u_2, u_1,\ \ u_6, u_5, u_4,\ \ u_9, u_8, u_7,\ \ \cdots,\ \ u_{\lfloor n/m\rfloor}, u_{\lfloor n/m\rfloor-1}, u_{\lfloor n/m\rfloor-2}.$$

The $\lfloor n/m\rfloor$ entries are grouped into blocks of size m and a fixed permutation (here a simple reversal) is applied within each such block. If u_i are CUD then so are the block permuted points. The reasoning is as follows. Consider integers d that are multiples of m. The discrepancy of (nonoverlapping) points $u_i^{(d)}$ is preserved by the permutation. Therefore it vanishes for all such d. Because there are infinitely many such d, the permuted points are CUD by Theorem 3 of [13].

Theorem 1. *Suppose that $u_{n,1},\dots,u_{n,n}$ are from a triangular array that is CUD or weakly CUD. Then the points of an antithetic sequence or a round trip sequence in either original or m-fold versions are CUD (respectively, weakly CUD).*

Proof. First consider the antithetic construction. Pick any integer $d \ge 1$ and let $u_{n,n+j} = 1-u_{n,j}$ for $n \ge d$ and $j=1,\dots,n$. Then using u_j for $u_{n,j}$,

$$\begin{aligned}
&D_{2n-d+1}^{*d}(\bar u_1^{(d)},\dots,\bar u_{2n-d+1}^{(d)})\\
&\le D_{2n-2d+2}^{*d}(\bar u_1^{(d)},\dots,\bar u_{n-d+1}^{(d)},\bar u_{n+1}^{(d)},\dots,\bar u_{2n-d+1}^{(d)}) + \frac{d-1}{2n-d+1}\\
&= D_{2n-2d+2}^{*d}(\bar u_1^{(d)},\dots,\bar u_{n-d+1}^{(d)},1-\bar u_1^{(d)},\dots,1-\bar u_{n-d+1}^{(d)}) + \frac{d-1}{2n-d+1}\\
&\le \frac{2^d+1}{2} D_{n-d+1}^{*d}(\bar u_1^{(d)},\dots,\bar u_{n-d+1}^{(d)}) + \frac{d-1}{2n-d+1}\\
&\to 0,
\end{aligned}$$

using (13) at the first inequality and (11) and (12) at the second. The proof for the round trip construction is similar. For the m-fold versions, we apply Theorem 3 of [13] as described above, to show that the second half of the sequence is CUD. □

5 Empirical results

We tried four methods on each of four problems. The methods used are IID, CUD, ANT and RND. In these, the driving sequences are IID, CUD based on the construction from Section 3, CUD with antithetics, and CUD with round trip sampling, respectively.

The four problems we tried were: bivariate Gaussian Gibbs sampling using various correlations and tracking the estimated mean, the same but tracking the estimated correlation, a Garch model, and a stochastic volatility model. We label these GMU, GRHO, GARCH and SV respectively.

What we report are root mean square errors based on 100 independent replications generated by Cranley-Patterson rotations. In the Gaussian-Gibbs problem we used 2–fold versions of ANT and RND. For GARCH and SV we used ordinary (1–fold) ANT and RND.

The bivariate Gaussian Gibbs sampler is a simple test case for algorithms. It has $X_i \in \mathbb{R}^2$. The sampling proceeds via

$$X_{i,1} = \rho X_{i-1,2} + \sqrt{1-\rho^2}\Phi^{-1}(u_{2i-1}), \quad \text{and} \tag{14}$$

$$X_{i,2} = \rho X_{i,1} + \sqrt{1-\rho^2}\Phi^{-1}(u_{2i}), \tag{15}$$

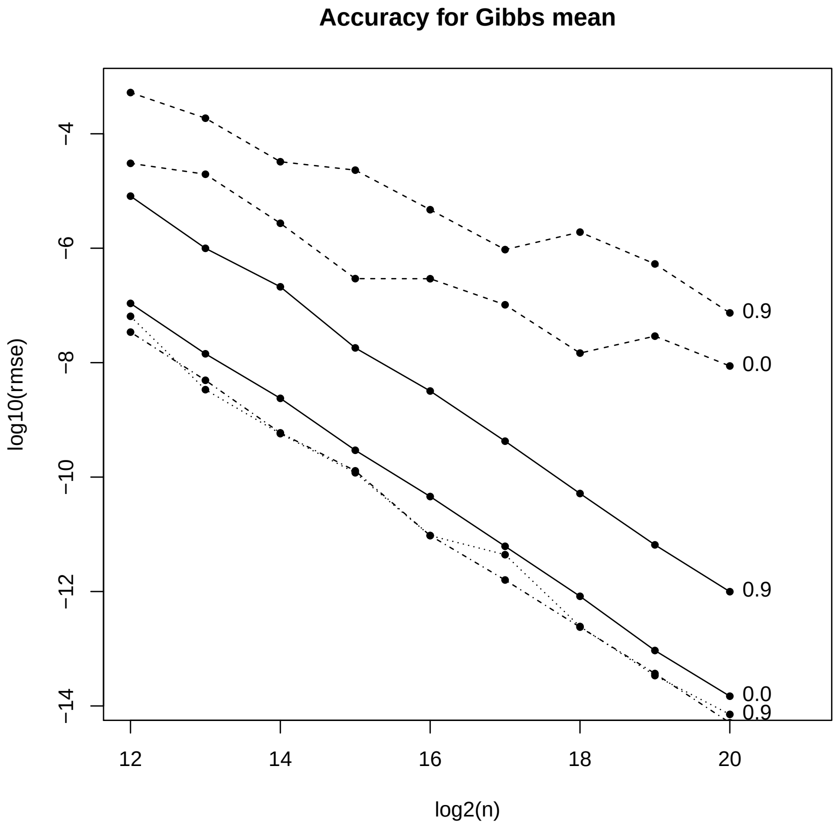
starting with $X_0 = (0,0)^T$. We then use $2n$ driving variables to generate $X_1,\dots,X_n$. We varied the true correlation ρ over the range from -0.9 to 0.9.

For problem GMU, we studied estimation of $\mathbb{E}(X_{1,1})$. This is somewhat of a toy problem. In the case $\rho = 0$, the round trip and antithetic sampling algorithms got the answer exactly. The CUD method seemed to attain a better rate than did IID sampling. For $\rho = 0.9$, we also saw an apparently better rate for CUD than IID, while the ANT and RND methods seem to have a better constant than the CUD method. See Figure 1.

The mean under Gibbs sampling is much easier than most problems we will face. To make it a bit more difficult we considered estimating the correlation itself from the data. This GRHO problem is artificial because we have to know that correlation in order to do the sampling. But a badly mixing chain would not allow us to properly estimate the correlation and so this is a reasonable test. In IID sampling the closer $|\rho|$ is to 1, the easier ρ is to estimate. In Gibbs sampling large $|\rho|$ makes the data values more dependent, but we will see $\rho = 0.9$ is still easier than $\rho = 0$.

We found that CUD outperformed IID on this case. The ANT and RND methods did about the same as CUD for most correlations but seemed to be worse than CUD

Fig. 1 Numerical results for bivariate Gaussian Gibbs sampling. CUD = solid and IID = dashed. The goal is to estimate the mean. The correlation is marked at the right. For $\rho = 0$ the ANT and RND methods had no error due to symmetry. For $\rho = 0.9$ they were essentially equal and much better than CUD, lying below even the CUD $\rho = 0$ curve. For $\rho = 0.9$, ANT is shown in dotted lines and RND in dot-dash lines.

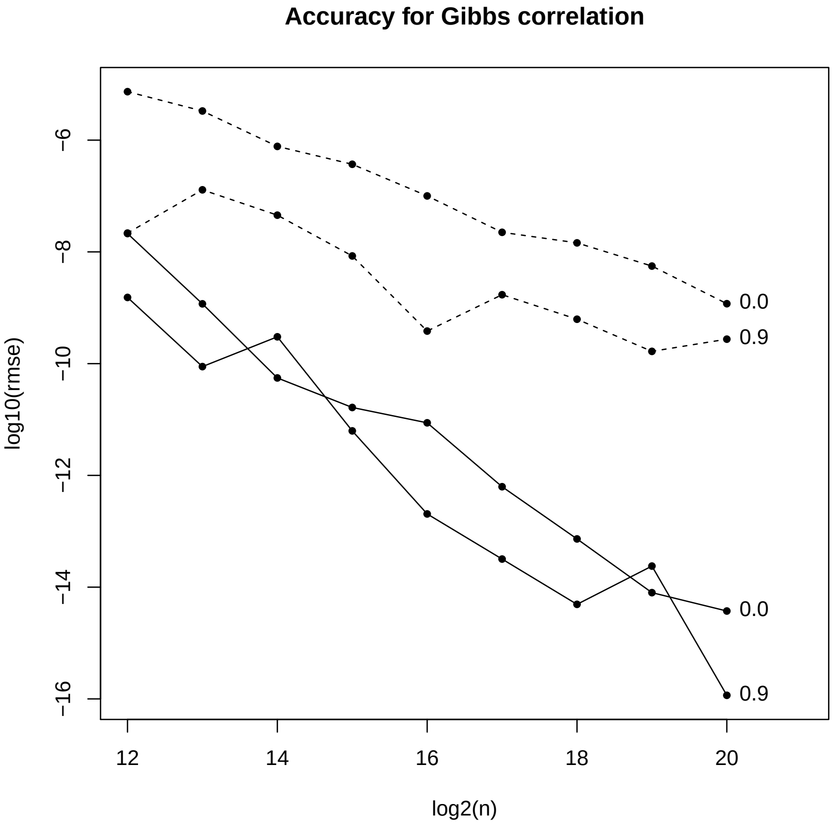


for the most extreme values ±0.9. The results comparing CUD to IID are shown in Figure 2.

The next two models are more challenging. They are stochastic volatility and Garch models. We apply them to a European call option. Under geometric Brownian motion that problem requires one dimensional quadrature and has a closed form solution due to Black and Scholes. For these models the value is a higher dimensional integral.

The SV model we used, from Zhu [19], is generated as follows:

Fig. 2 Numerical results for bivariate Gaussian Gibbs sampling. CUD = solid and IID = dashed. The goal is to estimate the correlation, which is marked at the right. There was little difference between CUD and its balanced alternatives ANT and RND (not shown).



$$dS = rs\,\mathrm{d}t + \sqrt{V}S\,\mathrm{d}W_1, \quad 0 < t < T \tag{16}$$
$$dV = \kappa(\theta - V)\,\mathrm{d}t + \sigma\sqrt{V}\,\mathrm{d}W_2, \tag{17}$$

for parameters $T = 6$ (years), $r = 0.04$, $\theta = 0.04$, $\kappa = 2$ and $\sigma = 0.3$. The initial conditions were $S(0) = 100$ and $V(0) = 0.025$. The processes W_1 and W_2 to the price and volatility were correlated Brownian motions with $\rho(\mathrm{d}W_1, \mathrm{d}W_2) = -0.5$. We priced a European call option, the discounted value of $\mathbb{E}((S(T) - K)_+)$ where the strike price K was 100. That is, the option starts at the money. Each sample path had 2^8 time points, requiring 2^9 elements u_i to generate it. The results are in Table 2.

The GARCH$(1,1)$ model we used had

Table 2 Log (base 10) of root mean squared error in the Heston stochastic volatility model for the four sampling methods and sample sizes 2^{11} to 2^{17}.

$\log_2(n)$	IID	CUD	ANT	RND
11	0.287	−0.089	−0.511	−0.545
12	−0.137	−0.534	−0.311	−0.327
13	0.112	−0.697	−1.017	−0.973
14	−0.594	−0.954	−1.013	−1.085
15	−0.611	−1.245	−1.099	−1.118
16	−1.150	−1.704	−1.770	−1.749
17	−0.643	−1.760	−1.892	−1.927

$$\log\Bigl(\frac{X_i}{X_{t-1}}\Bigr) = r + \lambda\sqrt{h_t} - \frac{1}{2}h_t + \varepsilon_t, \quad 1 \le t \le T, \quad \text{where} \tag{18}$$

$$\varepsilon_t \sim N(0, h_t), \quad \text{and} \tag{19}$$

$$h_t = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 h_{t-1}. \tag{20}$$

The parameter values, from Duan [5] were $r = 0$, $\lambda = 7.452 \times 10^{-3}$, $T = 30$, $\alpha_0 = 1.525 \times 10^{-5}$, $\alpha_1 = 0.1883$ and $\beta_1 = 0.7162$. The process starts with $h = 0.64\sigma^2$ where $\sigma^2 = 0.2413$ is the stationary variance of X_t.

Once again, the quantity we simulated was the value of a European call option. The strike price was $K = 1$. We started the process at values of $X_0 \in \{0.8, 0.9, 1.0, 1.2\}$.

In this example there was little difference between CUD sampling and either ANT or RND. Plain CUD sampling did better at sample sizes $2^{11} \le n \le 2^{18}$. It seemed to do slightly worse at sample sizes 2^{19} and 2^{20}. The CUD points outperformed IID sampling by a large margin and because the Garch model is interesting and important we show that result in Figure 3.

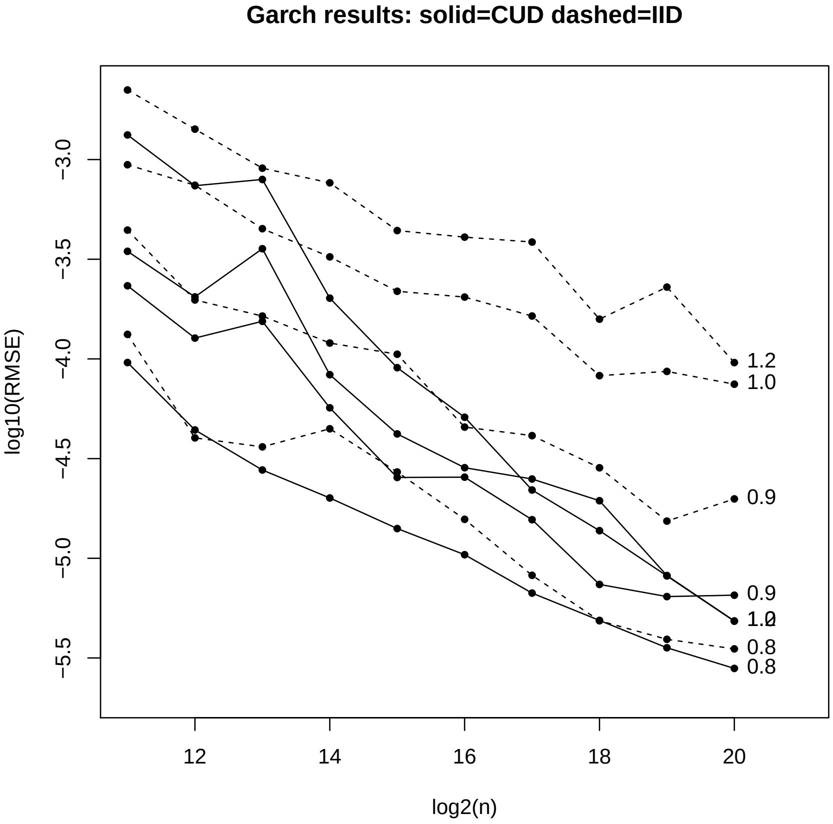
6 Conclusions

We have presented some new LFSRs and seen that they yield improved Markov chain quasi-Monte Carlo algorithms on some problems. Other problems do not show much improvement with the introduction of QMC ideas. This pattern is already familiar in finite dimensional applications.

We have also developed some ways to construct new (W)CUD sequences from old ones. The new sequences have a reflection property that we find is sometimes helpful and sometimes not, just as antithetic sampling is sometimes helpful and sometimes not in IID sampling.

The (W)CUD constructions sometimes appear to be achieving a better convergence rate than the IID ones do. There is therefore a need for a theoretical understanding of these rates of convergence.

Fig. 3 Numerical results for the Garch$(1,1)$ model described in the text. The initial price is marked on each trajectory, with the CUD trajectories for $X_0 = 0.9$ and 1.0 getting overlapping labels.



Acknowledgements This work was supported by grant DMS-0906056 from the U.S. National Science Foundation and by JSPS Grant-in-Aid for Scientific Research No.19204002, No.21654017, and JSPS Core-to-Core Program No.18005. We thank the organizers of MCQMC 2010, Leszek Pleskota and Henryk Wozniakowski, for providing an excellent scientific venue.

References

1. Chen, S., Dick, J., Owen, A.B.: A quasi-Monte Carlo Metropolis algorithm. Annals of Applied Statistics (2011). In press
2. Chentsov, N.N.: Pseudorandom numbers for modelling Markov chains. Computational Mathematics and Mathematical Physics **7**, 218–2332 (1967)

3. Craiu, R.V., Meng, X.L.: Multi-process parallel antithetic coupling for backward and forward Markov chain Monte Carlo. Annals of Statistics **33**, 661–697 (2005)
4. Cranley, R., Patterson, T.: Randomization of number theoretic methods for multiple integration. SIAM Journal of Numerical Analysis **13**, 904–914 (1976)
5. Duan, J.C.: The garch option pricing model. Mathematical Finance **5**(1), 13–32 (1995)
6. Frigessi, A., Gäsemyr, J., Rue, H.H.: Antithetic coupling of two Gibbs sampler chains. Annals of Statistics **28**, 1128–1149 (2000)
7. Hansen, T., Mullen, G.L.: Primitive polynomials over finite fields. Mathematics of Computation **59**, 639–643 (1992)
8. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97–109 (1970)
9. L'Ecuyer, P.: Tables of maximally-equidistributed combined lfsr generators. Mathematics of Computation **68**, 261–269 (1999)
10. Levin, M.B.: Discrepancy estimates of completely uniformly distributed and pseudo-random number sequences. International Mathematics Research Notices pp. 1231–1251 (1999)
11. Liao, L.G.: Variance reduction in Gibbs sampler using quasi random numbers. Journal of Computational and Graphical Statistics **7**, 253–266 (1998)
12. Liu, J.S.: Monte Carlo strategies in scientific computing. Springer, New York (2001)
13. Owen, A.B., Tribble, S.D.: A quasi-Monte Carlo Metropolis algorithm. Proceedings of the National Academy of Sciences **102**(25), 8844–8849 (2005)
14. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer, New York (2004)
15. Sobol', I.M.: Pseudo-random numbers for constructing discrete Markov chains by the Monte Carlo method. USSR Computational Mathematics and Mathematical Physics **14**(1), 36–45 (1974)
16. Tootill, J.P.R., Robinson, W.D., Eagle, D.J.: An asymptotically random Tausworthe sequence. Journal of the ACM **20**(3), 469–481 (1973)
17. Tribble, S.D.: Markov chain Monte Carlo algorithms using completely uniformly distributed driving sequences. Ph.D. thesis, Stanford University (2007)
18. Tribble, S.D., Owen, A.B.: Construction of weakly CUD sequences for MCMC sampling. Electronic Journal of Statistics **2**, 634–660 (2008)
19. Zhu, J.: A simple and exact simulation approach to Heston model. Tech. rep., Lucht Probst Associates (2008)