Exact sampling with highly-uniform point sets

Christiane Lemieux
Department of Mathematics and Statistics
University of Calgary
2500 University Drive N.W.
Calgary, AB, T2N 1N4, CANADA
lemieux@math.ucalgary.ca

Paul Sidorsky
Canadian Pacific Railway
Gulf Canada Square
401-9th Avenue SW
Calgary, AB, T2P 4Z4, CANADA
paulsid@shaw.ca

Abstract

In 1996, Propp and Wilson came up with a remarkably clever method for generating exact samples from the stationary distribution of a Markov chain [18]. Their method, called "perfect sampling" or "exact sampling" avoids the inherent bias of samples that are generated by running the chain for a large but fixed number of steps. It does so by using a strategy called "coupling from the past". Although the sampling mechanism used in their method is typically driven by independent random points, more structured sampling can also be used. Recently, Craiu and Meng [3, 4] suggested to use different forms of antithetic coupling for that purpose. In this paper, we consider the use of highly-uniform point sets to drive the exact sampling in Propp and Wilson's method, and illustrate the effectiveness of the proposed method with a few numerical examples.

1 Introduction

The problem under consideration in this paper is that of sampling from a distribution that corresponds to the stationary distribution of an ergodic Markov chain, as done, for instance, in the context of Markov Chain Monte Carlo methods. The goal is to produce samples whose empirical distribution is as close as possible to the true distribution. The technique of exact sampling, introduced by Propp and Wilson in 1996 [18], has provided an important breakthrough in this area by effectively removing the problem of determining for how long the chain should be simulated in order to achieve an acceptable level of accuracy. Their approach makes use of a method called "coupling from the past" (CFTP), which instead of running the simulation of the chain forward, starts it far enough in the past so that eventually, the samples produced can be shown to follow the targeted stationary distribution. More details are given in Section 2.

While their technique removes a major hurdle, it still relies on random trials and therefore, it is subject to the usual pitfalls of random sampling. On a more positive note, this means that standard variance reduction techniques can be used to improve the quality of the samples generated. Research in this direction has been made by Craiu and Meng [3, 4], who use three different forms of antithetic coupling to drive the sampling in Propp and Wilson's algorithm. The antithetic coupling methods they consider are based on groups of $k \geq 2$ sample points that are *negatively associated*. This is a stronger condition than asking for negative correlation, because this requires that the negative correlation holds after monotone (component-wise) transformations have been applied to the sampling points. The

most promising method studied by Craiu and Meng appears to be their "Iterative Latin Hypercube Sampling" (ILHS), which is a variant of the well-known Latin Hypercube Sampling (LHS) method [11].

Antithetic coupling can be thought of as an alternative to random sampling based on more structured sampling schemes. Another kind of structured sampling that has been used successfully in the last few years in the context of multivariate integration – most noticeably in finance problems [17, 21, 1] – is *quasi-random* sampling. Here, the idea is to use highly-uniform (or low-discrepancy) point sets to sample the function to be integrated, a technique known as *quasi-Monte Carlo* (QMC).

In this paper, we study the use of highly-uniform point sets (HUPS) to drive the simulations in the exact sampling algorithm of Propp and Wilson. In the context of multivariate integration, a HUPS $P_k = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ is used to estimate an integral of the form

$$\mu = \int_{[0,1)^s} f(\mathbf{u}) d\mathbf{u} \tag{1}$$

by the average

$$\hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} f(\mathbf{u}_i).$$

By using a (deterministic) point set that is more uniformly distributed than a random point set, it is often possible to get a smaller error than with the Monte Carlo (MC) method. However, because of the deterministic nature of QMC methods, getting an estimate for the integration error is difficult. To circumvent this problem, one can use *randomized quasi-Monte Carlo* (RQMC) methods, which are based on randomized HUPS $\tilde{P}_k$ having the following properties: (i) each $\mathbf{u}_i$ in $\tilde{P}_k$ is uniformly distributed over $[0,1)^s$; (ii) $\tilde{P}_k$ has the same high uniformity as the original HUPS P_k. Property (i) ensures that $\hat{\mu}$ is an unbiased estimator of μ, while property (ii) suggests that $\hat{\mu}$ should have a smaller variance than a MC estimator based on a random point set. There exist theoretical results providing support for this claim [15], but it can also be verified empirically, as discussed in Section 2.

RQMC methods usually produce better results than the simple antithetic variates method widely used in simulation [6], for which $k = 2$ and $P_2 = \{\mathbf{u}, 1 - \mathbf{u}\}$. Also, it often does better than LHS because the high uniformity of the point sets used in quasi-random sampling goes beyond the one-dimensional uniformity achieved by LHS. More precisely, let $P_k = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ be an LHS point set in $[0,1)^s$, where $\mathbf{u}_i = (u_i^1, \ldots, u_i^s)$, for $i = 1, \ldots, k$. Then, for each one-dimensional projection $P_k(\{j\}) = \{u_1^j, \ldots, u_k^j\}$, the LHS property is such that there is exactly one coordinate u_i^j in each interval $[l/k, (l+1)/k)$, for $l = 0, \ldots, k-1$. For highly-uniform point sets (HUPS), this type of *equidistribution* holds for higher-dimensional projections as well. To explain what we mean by that, it is useful to introduce the concept of $(q_1, \ldots, q_s)$-*equidistribution in base* b [7]: for non-negative integers $q_1, \ldots, q_s$, we say a point set P_k with $k = b^m$ is $(q_1, \ldots, q_s)$-equidistributed in base b if each rectangular box of the form

$$\prod_{j=1}^{s} \left[\frac{l_j}{b^{q_j}}, \frac{l_j + 1}{b^{q_j}} \right), \qquad 0 \le l_j < b^{q_j}, \qquad j = 1, \ldots, s,$$

contains exactly b^{m-Q} points, where $Q = q_1 + \ldots + q_s$. For a given point set P_k, if ρ is the largest integer such that P_k is $(q_1, \ldots, q_s)$-equidistributed in base b for all non-negative vectors $(q_1, \ldots, q_s)$ such that $Q \le \rho$, we say P_k is *equidistributed up to* ρ *in base* b.

By definition, an LHS point set P_k is equidistributed up to $\rho = 1$ in base k. In contrast, a (t, m, s)-*net in base* b [12] is a kind of HUPS with $k = b^m$ points that is equidistributed up to $m - t$. Hence the

smallest t is, the better is the quality of the (t, m, s)-net. Figure 1 compares an LHS point set and a $(0, 4, 2)$-net in base 3. The figure shows the $(2, 2)$-equidistribution in base 3 of the net: we find $b^{m-Q} = 3^{4-4} = 1$ point in each of the 81 squares thus obtained. For comparison, we draw the same squares for the LHS point set. In this case, several squares contain no points, while others contain two or even three. This is to be expected, since LHS does not guarantee any uniformity property beyond one dimension. We could make similar comparisons by looking at the $(3, 1)$ or $(1, 3)$-equidistribution of the net.

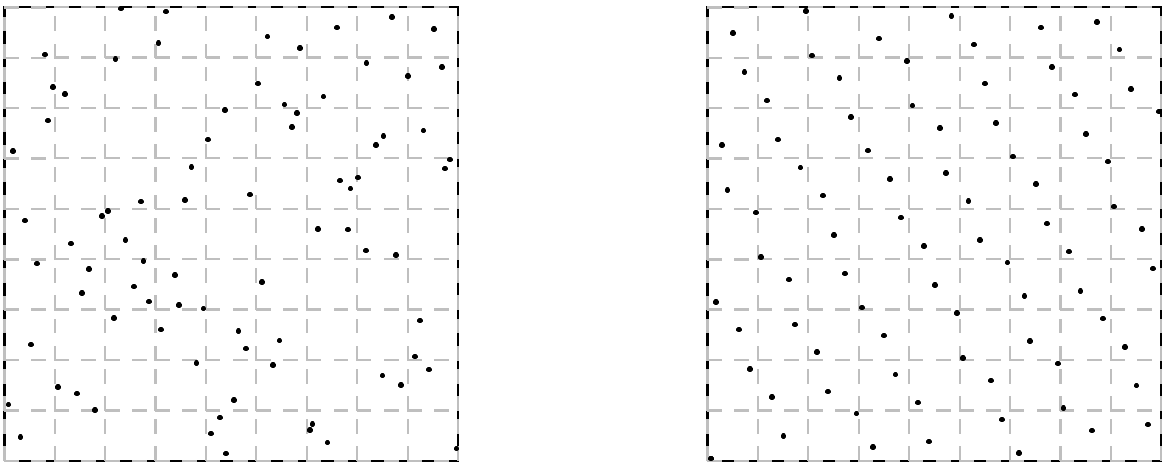

Figure 1: Left-hand side: 81 points from a LHS sample; Right-hand side: 81 points of a (0,4,3)-net in base 3.

The rest of this paper is organized as follows. In Section 2, we give the required background on exact sampling, and briefly explain the ILHS method used by Craiu and Meng [3, 4]. Properties of HUPS relevant in the context of exact sampling are discussed in Section 3, where we also describe the implementation of our method. Numerical experiments where the proposed combination is compared with MC sampling and the ILHS method are reported in Section 4. A summary and ideas for future research are briefly discussed in Section 5.

2 Background

The problem considered here is to sample from a distribution that corresponds to the stationary distribution π of an ergodic Markov chain X with N states. Since the chain is ergodic, we can start it in any state $X_0 \in \{1, \ldots, N\}$, and if we run the chain long enough, then the probability that it will end up in state i converges to $\pi(i)$, for $i = 1, \ldots, N$. Using this fact, a straightforward approach to generate samples that (approximately) follow π is to choose some large integer T, simulate the Markov chain for T steps, and output X_T. Figure 2 gives pseudocode for this *forward simulation* approach. The function `Unif01()` used there is assumed to be returning independent observations from the uniform distribution over (0,1), denoted $U(0, 1)$. The function `GenP(`u, x`)` is assumed to be returning independent observations from the distribution $P(X_t | X_{t-1} = x)$, which can be obtained from the transition probability matrix P of the Markov chain X. More precisely, if $P_{i,j} = P(X_t = i | X_{t-1} = j)$, then `GenP(`$u, x$`)` returns the smallest l such that $u \leq P_{1,x} + \ldots + P_{l,x}$. We could have combined the two functions `Unif01()` and `GenP(`$\cdot, \cdot$`)` into one, but we didn't because it will be useful later to separate

```
ForwardSim(P,N,X_0)
for t ← 1 to T
    u ← Unif01()
    s ← GenP(u, X_{t-1})
    X_t ← s
output X_T
```

Figure 2: Forward simulation approach

these two steps.

The main disadvantage of this forward simulation approach is that because T is fixed, the observation X_T output only approximately has distribution π. Of course if T is large enough, the approximation should be quite good, but how do we know if T is large enough?

The exact sampling method proposed by Propp and Wilson [18] removes this problem by using a *coupling from the past* (CFTP) approach. Here, we only briefly describe this method, and refer the reader to [18, 2] for more details. Instead of simulating forward only one path of the Markov chain – starting at time 0 and in some arbitrary state X_0 – the idea is to simulate N paths from some time $-T$ until time 0, where path l starts in state l, for $l = 1, \ldots, N$. The starting time $-T$ should be chosen far enough in the past so that by time 0, all N chains have *coalesced*. That is, they are all in the same state $X_0 \in \{1, \ldots, N\}$. How should we choose $-T$? It turns out we do not need to choose it: we can simply initialize T at 1 and increase it until coalescence is reached at time 0. As Propp and Wilson shows in [18], this approach produces an observation X_0 that is distributed according to π.

Before giving pseudocode for the exact sampling approach, let us first introduce the concept of maps used by Propp and Wilson to describe their approach. For $t \leq u \leq 0$, in the CFTP approach, we get N paths of the form $X_t^l, X_{t+1}^l, \ldots, X_u^l$, where $X_t^l = l$, for $l = 1, \ldots, N$. Define the random map F_t^u from $\{1, \ldots, N\}$ to $\{1, \ldots, N\}$ as follows: $F_t^u(l) = X_u^l$. Also, let $f_t = F_t^{t+1}$. Then, it is easy to see that $F_t^u = F_{t+1}^u \circ f_t$, where $\circ$ denotes the composition of functions. From this point of view, the idea of CFTP is to decrease t until the map F_t^0 becomes a constant map. More details are given in Figure 3. Note that in our description of the exact sampling algorithm, we have assumed that the same uniform number was used to generate all the components of f_t at a given time t. The method of Propp and Wilson as described in [18] allows for more general sampling mechanisms than this particular one.

Once we are able to generate observations from π, we are generally interested in estimating expectations with respect to π. That is, we want to estimate quantities of the form

$$\mu = \sum_{l=1}^{N} g(l)\pi(l), \tag{2}$$

for some real-valued function g. A straightforward way to do this is to generate an i.i.d. sample $\{y_1, \ldots, y_k\}$ from π using the exact sampling approach. Then, the estimator

$$\hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} g(y_i) \tag{3}$$

ExactSim(P, N)
$t \leftarrow 0$
$F_t^0 \leftarrow I$ (the identity map over $\{1, \ldots, N\}$)
repeat
    $t \leftarrow t - 1$
    $u \leftarrow$ `Unif01()`
    for $l \leftarrow 1$ to N
        $f_t(l) =$ `GenP(`u, l`)`
    $F_t^0 \leftarrow F_{t+1}^0 \circ f_t$
until F_t^0 is constant
return $F_t^0(1)$

Figure 3: Exact sampling approach

for (2) is unbiased.

As mentioned in the introduction, a natural idea to reduce the variance of $\hat{\mu}$ is to use a structured sample $\{y_1, \ldots, y_k\}$, rather than a random one. To describe this idea in more details, let us introduce some notation. First, let $(u_i^1, \ldots, u_i^s)$ be the vector of independent uniform numbers used when Exact(P, N) is called the ith time, to produce y_i. Here, the dimension s is equal to T, the number of backward steps required for coalescence to occur, which means s is random. Note also that we can view y_i as being a certain function h of $(u_i^1, \ldots, u_i^s)$. Hence the estimator (3) can be rewritten as

$$\hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} f(\mathbf{u}_i), \tag{4}$$

where $f = g \circ h$ and $\mathbf{u}_i = (u_i^1, \ldots, u_i^s)$. Similarly, we can rewrite (2) as

$$\mu = \int_{[0,1)^s} f(\mathbf{u}) d\mathbf{u}, \tag{5}$$

which is equivalent to the integration problem described by (1).

From this point of view, it becomes clear that by using a structured or highly-uniform point set $P_k = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ to generate $\{y_1, \ldots, y_k\}$, we should get an estimator $\hat{\mu}$ with smaller variance than if we use a set of independent points. For example, for LHS, we know that the variance of the resulting estimator is upper bounded by $k/(k-1)$ times the variance of the MC estimator based on the same number k of points [15]. For nearly linear functions, the variance of the LHS estimator can be expected to be much lower than that of a MC estimator [15]. For RQMC methods, under some conditions, it is possible to get a variance in $O(k^{-3} \log^s k)$ [14]. As a rough guide, we can expect RQMC methods to do well when the integrand f has a *low effective dimension*, which means that the interaction between small subsets of variables in $\{u^1, \ldots, u^s\}$ are those that contribute the most to the variance of f.

An important observation about the formulation described by (4) and (5) is that the integrand f is generally discontinuous, since $f = g \circ h$, and h is a piecewise constant function. Hence theoretical

results that rely on the smoothness of f (such as those providing bounds for the integration error of deterministic QMC methods) can generally not be applied in this context. What is interesting about RQMC methods is that results on the variance of the corresponding estimators can be obtained with the sole assumption that f is square-integrable [13, 16, 7].

In addition to theoretical results, the efficiency of a particular choice for P_k can be assessed by estimating the variance of $\hat{\mu}$. When the $\mathbf{u}_i$ (and thus the y_i) are not independent, this can be done by repeating the whole procedure described above, say, M times. That is, M i.i.d. estimators $\hat{\mu}_1, \ldots, \hat{\mu}_M$ of the form (3) are computed, and then the variance of the resulting estimator

$$\bar{\mu} = \frac{1}{M} \sum_{p=1}^{M} \hat{\mu}_p$$

can be estimated by

$$\bar{\sigma}^2 = \frac{1}{M} \sum_{p=1}^{M} \frac{(\hat{\mu}_p - \bar{\mu})^2}{M-1}.$$

We conclude this section by describing the ILHS method proposed in [3]. First, recall that with LHS, the coordinates of the points in P_k are obtained as follows:

$$u_{i,j} = \frac{1}{k} \left(\pi_j[i] + v_{i,j}\right), \qquad i = 1, \ldots, k,\ j = 1, \ldots, s,$$

where π_j is a permutation of $[1, \ldots, k]$, and $v_{i,j} \sim U(0,1)$. In the ILHS method, the shifts $v_{i,j}$ are replaced by $u_{i,j-1}$ for $j \geq 2$, and $u_{i,1} \sim U(0,1)$ for $i = 1, \ldots, k$.

3 Using highly-uniform point sets for exact sampling

It is reasonable to expect that since HUPS have good uniformity properties that are not restricted to one-dimensional projections, they should provide a better sampling mechanism for exact sampling than LHS and its variants. For example, the uniformity properties of a (t, m, s)-net in base b with $k = b^m$ points is such that any r-dimensional projection – that is, any projection of the form $P_k(I)$ for $I \subseteq \{1, \ldots, s\}$ such that $|I| = r$ – with $r \leq m$ is equidistributed up to (at least) $m - t$. The reason why we say "at least" is because the parameter t only gives an upper bound for the quality of those projections. That is, if we define t_I to be the smallest integer such that $P_k(I)$ is a $(t_I, m, |I|)$-net, then t is the maximum value of t_I over all subsets I. For $(0, m, s)$-nets, this means that all projections $P_k(I)$ with $|I| \leq m$ have $t_I = 0$. This suggests that nets with $t = 0$ are far superior to nets with $t > 0$. However, to build a net with $t = 0$, the base b must be chosen so that $b \geq s$ [12]. Therefore, nets with $t = 0$ tend to have small values of m, which means that only a few projections are known to have $t_I = 0$, and nothing is known about the quality of the other projections. For example, if $s = 60$, then the smallest base b for which we can have a net with $t = 0$ is $b = 61$. If we let $k = 61$, then only for the one-dimensional projections $P_k(\{j\})$, $j = 1, \ldots, 60$, can we say that $t_I = 0$. So in this case, the uniformity of P_k is not superior to that of LHS. If we take $k = b^2 = 3721$, then we also know the two-dimensional projections have $t_I = 0$, but nothing is known about the quality of $P_k(I)$ with $|I| > 2$.

Now, recall that in the context of exact sampling, the dimension s is random and could be quite large in some cases. To keep the computation time under control, we thus prefer to use small HUPS whose points can be generated quickly. From this point of view, using nets with $t = 0$ is inappropriate. In

addition, these nets have a base that depends on the dimension s. This makes them unsuitable for problems where the dimension is unknown. In our experiments, we instead use two different types of nets in base 2, and another HUPS construction called a *Korobov lattice*. This type of point set can be generated very easily, and depends only on one parameter a – called the *generator* – as follows:

$$P_k = \left\{ \frac{1}{k}(i, i \cdot a \bmod k, i \cdot a^2 \bmod k, \ldots, i \cdot a^{s-1} \bmod k), i = 0, \ldots, k-1 \right\}.$$

A Korobov lattice can be randomized using the following method [5]: generate a random uniform vector $\mathbf{v} \in [0,1)^s$, and then use it to shift each point $\mathbf{u}_i$ in P_k. That is, let

$$\tilde{P}_k = \{(\mathbf{u}_i + \mathbf{v}) \bmod 1, i = 1, \ldots, k\}.$$

The two nets used are a *Sobol' net* [20] and a *polynomial Korobov lattice* [10]. They are both randomized using a digital shift in base 2 [7]. This randomization technique is similar to the one described above, except the random shift $\mathbf{v}$ is added to each $\mathbf{u}_i$ using bitwise addition modulo 2. That is, for a digital net $P_k = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ in base 2, let

$$\tilde{P}_k = \{\mathbf{u}_i \oplus \mathbf{v}, i = 1, \ldots, k\},$$

where

$$\mathbf{u} \oplus \mathbf{v} = \left(u^1 \oplus v^1, \ldots, u^s \oplus v^s\right),$$

and for each $j = 1, \ldots, s$,

$$u^j \oplus v^j = \sum_{l=1}^{\infty} ((u_l^j + v_l^j) \bmod 2) 2^{-l},$$

where $u_l^j, v_l^j \in \{0,1\}$ are from the binary expansion of u^j and v^j, respectively. That is, $u^j = \sum_{l=1}^{\infty} u_l^j 2^{-l}$ and $v^j = \sum_{l=1}^{\infty} v_l^j 2^{-l}$ (where we assume that u_l^j and v_l^j are 0 for infinitely many l).

We mentioned before that $(0, m, s)$-nets cannot be used for problems where the dimension s is random since their base b depends on s. More generally, this holds for any HUPS construction for which the number of parameters or their range depends on the dimension. For example, Sobol' nets in dimension s have a set of parameters for each coordinate $j = 1, \ldots, s$. Therefore, in practice they cannot be used for problems where the dimension is random, since in this case there is no *a priori* limit on the number of parameters we need. Most implementations of this method actually have a limit of $s \leq 360$ (see [9] and the references therein).

A convenient way to handle problems with a random dimension is to use *recurrence-based point sets* [7], which include both Korobov and polynomial Korobov lattices. To illustrate how these point sets can deal with a random dimension, we describe in Figure 4 how to use a (randomly shifted) Korobov lattice in the context of exact sampling. The reader is referred to [7, 8] for more details. Note that both LHS and ILHS can handle a random dimension as well, but are more computationally expensive than the HUPS we chose for our experiments.

4 Experiments

We now look at three different simple Markov chain examples, and compare the performance of three different RQMC methods with MC. We also report results obtained with the ILHS method of Craiu

```
ExactSim(P, N, a, k)
for l ← 1 to N
    π[l] = 0
for i ← 0 to k − 1
    InitUnif01 // Initializes Unif01
    t ← 0
    F_t^0 ← I
    repeat
        t ← t − 1
        s ← −t
        u ← (((i · a^{s−1}) mod k)/k + Unif01()) mod1
        for l ← 1 to N
            f_t(l) ← GenP(u, l)
        F_t^0 ← F_{t+1} ∘ f_t
    until F_t^0 is constant
    x ← F_t^0(1)
    π[x] ← π[x] + 1
for l ← 1 to N
    π[l] ← π[l]/N
```

Figure 4: Using shifted Korobov for exact sampling

and Meng [3]. For each example, the stationary distribution $\pi = (\pi_1, \ldots, \pi_N)$ of the chain is estimated. Then, each method's performance is measured by looking at the variance reduction factor obtained with respect to MC when estimating the quantity μ given in (2) for some function g. For the function g, we consider the three same examples as in [3], which are $g_1(x) = x$, $g_2(x) = (x-2)(x-5)$, and $g_3(x) = \sin(3x)$.

The three RQMC methods chosen were shifted Korobov, digitally shifted polynomial Korobov, and digitally shifted Sobol'. In the examples we look at, although the dimension is random, it is often reasonably small. Therefore, in most cases we can fix $s = 360$ and use Sobol' nets without any problem. Also, we have chosen $k \approx 127$ ($k = 127$ for MC, Korobov and ILHS; $k = 128$ for Sobol' and polynomial Korobov since these two are nets in base 2), and $M = 100$ in all our experiments. Thus each variance is estimated from a sample of size 100. We also report the average value of s obtained in these examples.

The first two examples are simple Markov chains taken from [19]. In the first case, $N = 3$ and the transition matrix is given by

$$\begin{pmatrix} 0.5 & 0.4 & 0.1 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{pmatrix},$$

while in the second, $N = 4$ and the transition matrix is given by

$$\begin{pmatrix} 0.7 & 0.0 & 0.3 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.4 & 0.0 & 0.6 \\ 0.0 & 0.2 & 0.0 & 0.8 \end{pmatrix}.$$

Figure 5 gives the reduction factors obtained for these two examples. The numbers on the horizontal axis refer to the function type. As we can see in this figure, for both the 3-state and 4-state Markov chain, the three RQMC methods consistently reduce the variance compared to MC, by factors ranging between 2 and 18. Also, the two digital nets seem to work better than the Korobov rule, and among the two, the Sobol' net seems the best. The ILHS method also does significantly better than MC, but the reduction factors are not as large as for the three RQMC methods. The average value for s was 2 and 3.7 for these two examples, respectively.

The third example is a random walk over $N = 16$ states with semi-absorbent barriers and probability p, which was used in [3]. More precisely, the corresponding transition matrix is defined by

$$P_{i,j} = \begin{cases} p & \text{if } i+1 = j \le N \text{ or } i = j = N \\ 1-p & \text{if } i-1 = j \le N \text{ or } i = j = 1 \\ 0 & \text{otherwise} \end{cases}$$

We performed experiments for $p = 1/10$, $p = 1/3$, and $p = 1/2$. When $p = 1/2$, we have not used the Sobol' net since in this case, the dimension was often exceeding 360. The variance reduction factors obtained are shown on Figure 6. In the case where $p = 0.1$, the three RQMC methods reduce the variance by factors ranging between 6 and 18, and do better than ILHS. When $p = 1/3$ or $1/2$, generally the best reduction factors are obtained by one of the RQMC methods, but in some cases, ILHS is as good or better, for example with $g_2(x)$ when $p = 1/3$ or with $g_3(x)$ when $p = 1/2$. In this latter case, the variance reduction factors are all close to 1, which means ILHS and the RQMC methods do not improve much upon MC. The average dimension s was 18, 42, and 120 for these three examples.

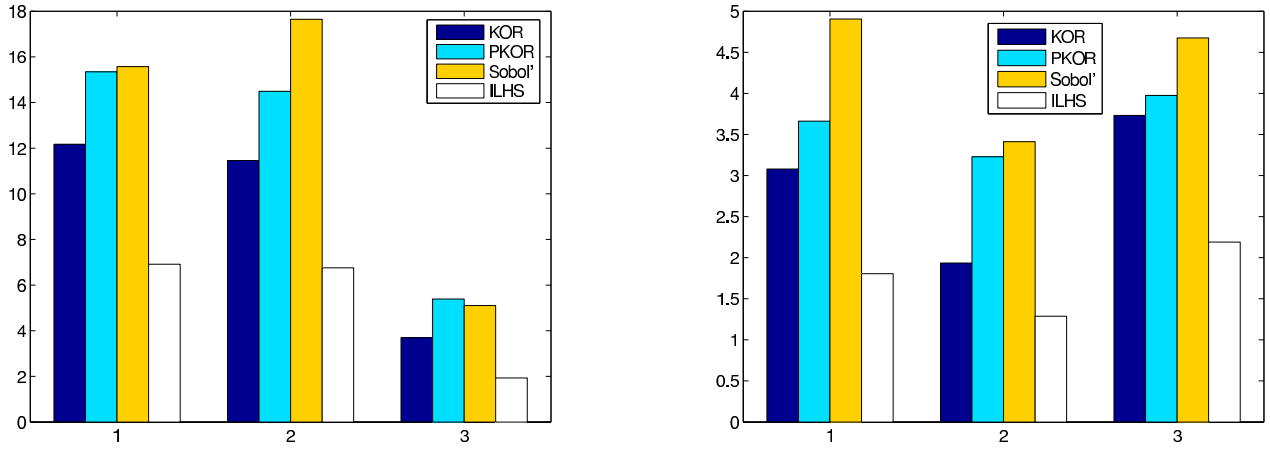

Figure 5: Variance reduction factors for the Markov chain examples

5 Conclusion

In this paper, we have explained how to use HUPS within the exact sampling method introduced by Propp and Wilson [18]. We have presented numerical results where in some cases, the proposed approach reduces the variance by factors close to 20 compared to random sampling.

For future work, we plan to look at more complex Markov chains than the ones considered here. Also, we would like to use the functional ANOVA decomposition to try to better understand the properties of the simulated Markov chains, and hopefully gain insight on techniques that could be used to improve the performance of our approach. Finally, we have ignored in this paper several methods that have been proposed to improve the efficiency of the exact sampling algorithm: we refer the reader to David Wilson's comprehensive web site on exact sampling at `http://www.dbwilson.com/exact/` for more on this. Obviously, these should be incorporated in our "highly-uniform exact sampling" method in the future. Of special interest will be methods that can help reduce the effective dimension of the sampling function h mentioned in Section 2.

Acknowledgments

This work was supported by NSERC grant #RGP238959.

References

[1] P. Acworth, M. Broadie, and P. Glasserman. A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. In P. Hellekalek and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, number 127 in Lecture Notes in Statistics, pages 1–18. Springer-Verlag, 1997.

[2] G. Casella, M. Lavine, and C. Robert. Explaining the perfect sampler. *The American Statistician*, 55:299–305, 2001.

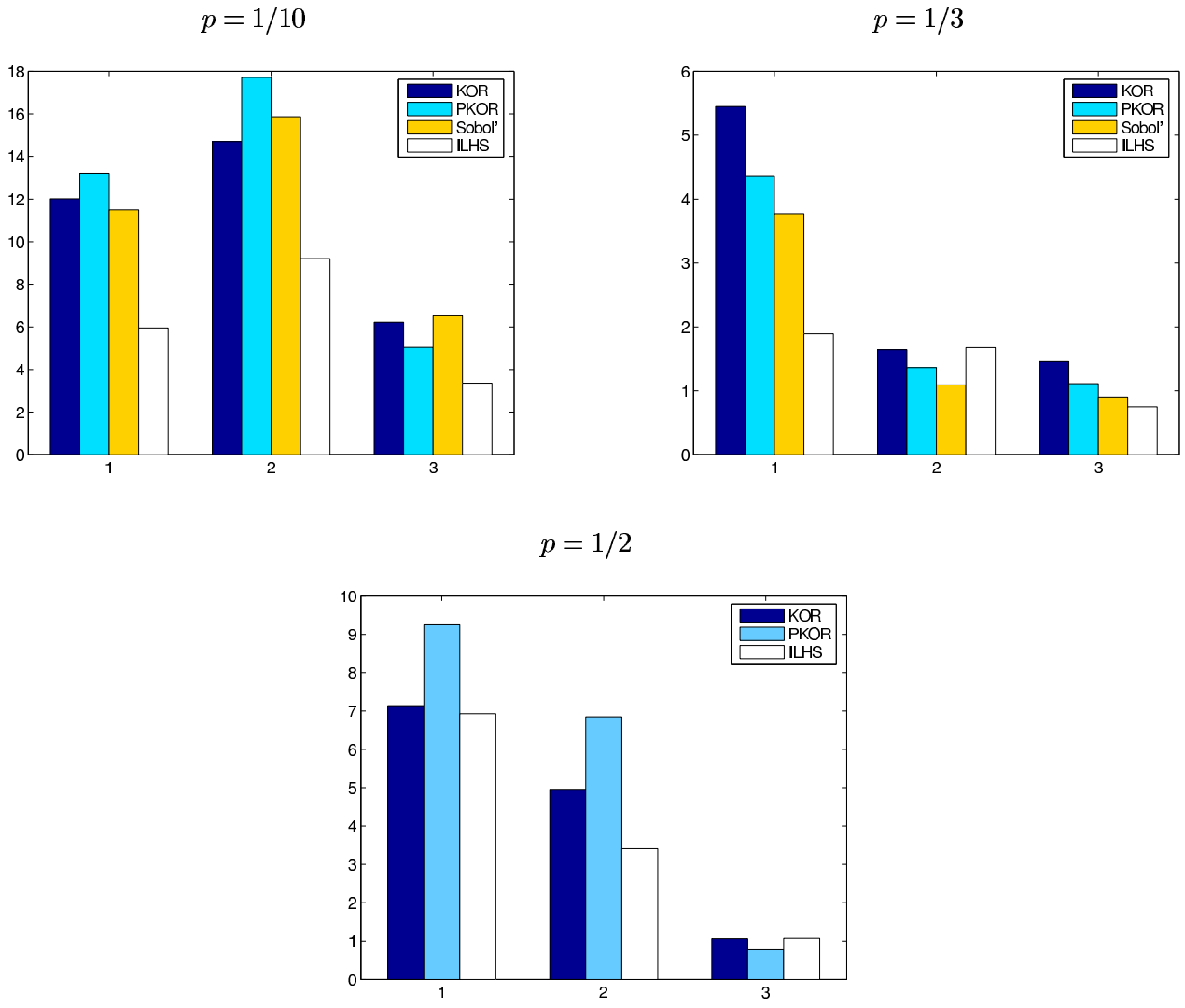

Figure 6: Variance reduction factors for the random walk examples

[3] R. V. Craiu and X.-L. Meng. Antithetic coupling for perfect sampling. In E.I. George, editor, *Bayesian Methods with Applications to Science, Policy, and Official Statistics (Selected Papers from ISBA 2000)*, pages 99–108, 2000.

[4] R. V. Craiu and X.-L. Meng. Multi-process parallel antithetic coupling for forward and backward Markov Chain Monte Marlo. *Annals of Statistics*, 2004. To appear.

[5] R. Cranley and T. N. L. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM J. Numer. Anal.*, 13(6):904–914, 1976.

[6] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, third edition, 2000.

[7] P. L'Ecuyer and C. Lemieux. Recent advances in randomized quasi-Monte Carlo methods. In M. Dror, P. L'Ecuyer, and F. Szidarovszki, editors, *Modeling Uncertainty: An Examination*

of Stochastic Theory, Methods, and Applications, pages 419–474. Kluwer Academic Publishers, Boston, 2002.

[8] C. Lemieux. Quasi-random numbers techniques. Book chapter to appear in *Handbook of Simulation*, Elsevier, 2005.

[9] C. Lemieux, M. Cieslak, and K. Luttmer. RandQMC user's guide: A package for randomized quasi-Monte Carlo methods in C. Technical Report 2002-712-15, Department of Computer Science,University of Calgary, 2002.

[10] C. Lemieux and P. L'Ecuyer. Randomized polynomial lattice rules for multivariate integration and simulation. *SIAM J. Sci. Comp.*, 24(5):1768–1789, 2003.

[11] M. D. Mckay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

[12] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1992.

[13] A. B. Owen. Monte Carlo variance of scrambled equidistribution quadrature. *SIAM J. Numer. Anal.*, 34(5):1884–1910, 1997.

[14] A. B. Owen. Scrambled net variance for integrals of smooth functions. *Annals of Statistics*, 25(4):1541–1562, 1997.

[15] A. B. Owen. Latin supercube sampling for very high-dimensional simulations. *ACM Trans. Model. Comp. Simul.*, 8(1):71–102, 1998.

[16] A.B. Owen. Scrambling Sobol and Niederreiter-Xing points. *J. Complexity*, 14:466–489, 1998.

[17] S. Paskov and J. Traub. Faster valuation of financial derivatives. *Journal of Portfolio Management*, 22:113–120, 1995. See also `www.cs.columbia.edu/~ap/html/information.html`.

[18] J. G. Propp and D. B. Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1-2):223–252, 1996.

[19] S. M. Ross. *Introduction to Probability Models*. Academic Press, fifth edition, 1993.

[20] I. M. Sobol'. The distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R. Comput. Math. and Math. Phys.*, 7:86–112, 1967.

[21] J. Spanier. Quasi-Monte Carlo methods for particle transport problems. In H. Niederreiter and P. J.-S. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106 of *Lecture Notes in Statistics*, pages 121–148, New York, 1995. Springer-Verlag.