

**On Array-RQMC for Markov Chains:
Mapping Alternatives and
Convergence Rates**

P. L'Ecuyer, C. Lécot,
A. L'Archevêque-Gaudet

G-2009-03

January 2009

On Array-RQMC for Markov Chains: Mapping Alternatives and Convergence Rates

Pierre L'Ecuyer

*GERAD and DIRO
Université de Montréal
C.P. 6128, Succ. Centre-ville
Montréal (Québec) Canada, H3C 3J7
lecuyer@iro.umontreal.ca*

Christian Lécot

*LAMA
Université de Savoie
73376 Le Bourget-du-Lac Cedex, France
christian.lecot@univ-savoie.fr*

Adam L'Archevêque-Gaudet

*DIRO
Université de Montréal
C.P. 6128, Succ. Centre-ville
Montréal (Québec) Canada, H3C 3J7
larcheva@iro.umontreal.ca*

January 2009

Les Cahiers du GERAD

G-2009-03

Copyright © 2009 GERAD

Abstract

We study the convergence behavior of a randomized quasi-Monte Carlo (RQMC) method for the simulation of discrete-time Markov chains, known as array-RQMC. The goal is to estimate the expectation of a smooth function of the sample path of the chain. The method simulates n copies of the chain in parallel, using highly uniform point sets randomized independently at each step. The copies are sorted after each step, according to some multidimensional order, for the purpose of assigning the RQMC points to the chains. In this paper, we discuss and compare different ways of realizing this sort and assignment, and report empirical experiments on the convergence rate of the variance as a function of n . In these experiments, the variance reduction with respect to standard Monte Carlo is substantial and we observe (approximately) an $O(n^{-2})$ convergence for the variance. On the other hand, for most standard discrepancies, the mean square discrepancy between the empirical and theoretical distributions of the states at any given step converges at a slower rate, approximately $O(n^{-3/2})$ in some examples.

Résumé

Nous étudions la convergence d'une méthode quasi-Monte Carlo randomisée (RQMC), appelée *array-RQMC*, pour la simulation de chaînes de Markov en temps discret. Le but est d'estimer l'espérance mathématique d'une variable aléatoire définie comme une fonction de la trajectoire de la chaîne. La méthode simule n copies de la chaîne en parallèle, en utilisant des ensembles de points hautement uniformes randomisés indépendamment à chaque étape. Les copies sont triées après chaque étape, selon un ordre multidimensionnel choisi, afin de coupler les n points RQMC avec les n chaînes. Dans cet article, nous discutons et comparons différentes façons de réaliser ce tri et cette affectation, et rapportons les résultats d'expériences numériques sur le taux de convergence de la variance de l'estimateur en fonction de n . Dans nos expériences, la réduction de variance par rapport à Monte Carlo standard est substantielle et nous observons (approximativement) un taux de convergence de $O(n^{-2})$ pour la variance. Par ailleurs, pour les définitions les plus usuelles de la discrédance, le carré moyen de la discrédance entre la loi théorique et la loi empirique des états de la chaîne à une étape donnée semble converger plus lentement, à un taux de $O(n^{-3/2})$ dans nos exemples.

Acknowledgments: This research has been supported by NSERC-Canada grant No. ODGP0110050 and a Canada Research Chair to the first author, and an NSERC scholarship to the third author.

1 Introduction

Quasi-Monte Carlo (QMC) and randomized QMC (RQMC) methods can be quite effective to estimate an integral when the integrand is reasonably smooth and has low effective dimension [9, 15, 18]. But when we simulate a system (modeled as a Markov chain) that evolves over several time steps, and the integrand is a function of the sample path, the dimension is typically very large, and the effective dimension can also be large. RQMC is often not very effective in this type of situation.

A different type of QMC and RQMC methodology, whose RQMC version is called array-RQMC, has been introduced and developed in [7, 8, 12, 13]. This array-RQMC algorithm simulates n copies of the chain in parallel. It advances all copies by one step at each iteration, using an RQMC point set of cardinality n to generate the transitions of these chains at the given step, using a clever matching of the RQMC points to the chains. This matching is done by sorting both the chains and the points according to their successive coordinates. The idea is to induce negative dependence between the n copies, so that the empirical distribution of the n states at any given step provides a much more accurate approximation of the true distribution than if the n copies were simulated independently. Empirical experiments have shown that this can improve the simulation efficiency for Markov chains simulated over several hundred steps, sometimes by factors of over 1000. Potential applications include queueing systems, option pricing in finance, reliability and risk assessment models, image generation in computer graphics, and more [1, 11, 13, 19].

The aim of this paper is to examine and compare alternative ways of matching the RQMC points to the chains at each step, and report empirical experiments on the convergence rate of the variance and of the discrepancy between the empirical and theoretical distribution of the states, as a function of n .

The remainder is organized as follows. The Markov chain setting and the estimation problems are defined in Section 2. In Section 3, we recall the array-RQMC algorithm and discuss possibilities for bounding its convergence rate. In Section 4, we examine how to map the chains to the RQMC points at each step. Empirical investigations of the convergence rate are reported in Section 5. A conclusion is given in Section 6.

2 A Markov chain setting

We consider a Markov chain model with state space $\mathcal{X} \subseteq \mathbb{R}^\ell$, whose state evolves according to the stochastic recursion

$$X_0 = x_0, \quad X_j = \varphi_j(X_{j-1}, \mathbf{U}_j), \quad j \geq 1,$$

where $\mathbf{U}_1, \mathbf{U}_2, \dots$ are i.i.d. uniform random variables over the unit hypercube $(0, 1)^d$. We want to estimate

$$\mu = \mathbb{E}[Y] \quad \text{where} \quad Y = \sum_{j=1}^{\tau} c_j(X_j)$$

and τ is a random stopping time with respect to the filtration generated by $\mathcal{F}\{(j, X_j), j \geq 0\}$. We also assume that $c_j(X_j) = 0$ for $j > \tau$.

To estimate μ by ordinary Monte Carlo (MC), we proceed as follows. For each i , $i = 0, \dots, n-1$, we generate a sample path of the chain via

$$X_{i,j} = \varphi_j(X_{i,j-1}, \mathbf{U}_{i,j}), \quad j = 1, \dots, \tau_i,$$

where $\mathbf{U}_{i,1}, \dots, \mathbf{U}_{i,\tau_i}$'s are i.i.d. uniform over $(0, 1)^d$ and τ_i is the realization of τ , and we compute $Y_i = \sum_{j=1}^{\tau_i} c_j(X_{i,j})$, the realization number i of Y . These sample paths are independent. The MC estimator of μ is then

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} Y_i.$$

For the classical RQMC method, let $s = \inf\{s' : \mathbb{P}[\tau d \leq s'] = 1\}$ (which could be infinite) and put $\mathbf{V}_i = (\mathbf{U}_{i,1}, \mathbf{U}_{i,2}, \dots, \mathbf{U}_{i,s/d})$ (which is an infinite sequence if $s = \infty$). Let $P_n = \{\mathbf{V}_0, \dots, \mathbf{V}_{n-1}\} \subset (0, 1)^s$ be an s -dimensional *RQMC point set*, defined as a point set with the following properties [9, 14, 16]: (a) each point \mathbf{V}_i has the uniform distribution over $(0, 1)^s$, and (b) P_n has low discrepancy in some sense (whose specific definition is left open). The RQMC estimator of μ is defined as

$$\hat{\mu}_{\text{rqmc},n} = \frac{1}{n} \sum_{i=0}^{n-1} Y_i = \frac{1}{n} \sum_{i=0}^{n-1} \sum_{j=1}^{\tau_i} c_j(X_{i,j}) \quad (1)$$

as before, where the $X_{i,j}$ and τ_i are defined as in the MC estimator. One difficulty here is that the dimension s of P_n can be very large, sometimes infinite.

3 The Array-RQMC Algorithm

With the array-RQMC method introduced in [12, 13], we simulate n chains in parallel, and use a d -dimensional RQMC point set P_n at each step to advance all the chains by one step, while inducing global negative dependence across the chains. The goal is that at each step j , the empirical distribution of the set of states $S_{n,j} = \{X_{0,j}, \dots, X_{n-1,j}\}$ is a very accurate approximation of the theoretical distribution of X_j . We want the discrepancy between these two distributions to be as small as possible, for a given measure of discrepancy that needs to be chosen.

Discrepancies for the uniform distribution over the unit hypercube $[0, 1]^\ell$ have been widely studied and are known to provide error bounds on the integration error and on the variance. This suggests the following strategy for defining a discrepancy and analyzing the convergence, under the assumption that X_j has a continuous distribution. Define a bijection $\psi_j : \mathcal{X} \rightarrow [0, 1]^\ell$ such that $\psi_j(X_j)$ has the uniform distribution over $[0, 1]^\ell$ in the Markov chain model, and define

$$D_j = D_j(S_{n,j}) = D_j(X_{0,j}, \dots, X_{n-1,j}) \stackrel{\text{def}}{=} D(\psi_j(X_{0,j}), \dots, \psi_j(X_{n-1,j})),$$

where D is a selected measure of discrepancy with respect to the uniform distribution over $[0, 1]^\ell$. One possible (conceptual) way of defining ψ_j is as follows. Given $X_j = (X_j^{(1)}, \dots, X_j^{(\ell)})$, let $U_j^{(1)} = F_{j,1}^{-1}(X_j^{(1)})$ where $F_{j,1}$ is the (cumulative) distribution function of $X_j^{(1)}$, then let $U_j^{(2)} = F_{j,2}^{-1}(X_j^{(2)} | X_j^{(1)})$ where $F_{j,2}^{-1}(\cdot | X_j^{(1)})$ is the distribution function of $X_j^{(2)}$ conditional on $X_j^{(1)}$, and so on. Then put $\psi_j(X_j) = U_j = (U_j^{(1)}, \dots, U_j^{(\ell)})$. When the distribution of X_j is not continuous, then this does not define a bijection, but one could still define ψ_j by taking $U_j^{(1)}$ as an arbitrary solution of $F_{j,1}(U_j^{(1)}) = X_j^{(1)}$, and so on.

Suppose that a Koksma-Hlawka-type inequality holds for the discrepancy D together with the corresponding measure of variation $V(f)$ for functions $f : [0, 1]^\ell \rightarrow \mathbb{R}$ [3]. This implies that if $\mu_j = \mathbb{E}[c_j(X_j)]$ and

$$\hat{\mu}_{\text{rqmc},j,n} = \frac{1}{n} \sum_{i=0}^{n-1} c_j(X_{i,j}),$$

the average cost at step j , then we have $\mathbb{E}[\hat{\mu}_{\text{rqmc},j,n}] = \mu_j$ and

$$\text{Var}[\hat{\mu}_{\text{rqmc},j,n}] \leq \mathbb{E}[D_j^2] V^2(c_j \circ \psi_j^{-1}). \quad (2)$$

We would like to prove, most likely by induction on j , that

$$\mathbb{E}[D_j^2] \leq \kappa_j n^{-\alpha+\epsilon}, \quad (3)$$

for any $\epsilon > 0$, for some $\alpha \geq 1$, where κ_j does not depend on n and grows only very slowly (or not at all) with j . From this, assuming that the $c_j \circ \psi_j^{-1}$ have bounded variation, it would follow that $\text{Var}[(Y_0 + \dots + Y_{n-1})/n]$ converges as $O(n^{-\alpha+\epsilon})$. The key issue is then to have an algorithm that preserves the low mean-square

discrepancy $\mathbb{E}[D_j^2]$ from one step to the next. This is the aim of the array-RQMC method, which we now summarize.

For simplicity, we assume in the remainder of the paper (except in the examples at the end) that X_j is a random variable uniformly distributed over $[0, 1]^\ell$, which we denote $X_j \sim U[0, 1]^\ell$. This is equivalent to always taking the image of the state by the transformation ψ_j and working in the transformed space. Roughly speaking, at each step of the algorithm, we view $\mathbb{E}[D_j^2]$ as an $(\ell + d)$ -dimensional integral, with respect to (X_{j-1}, \mathbf{U}_j) , and estimate it by RQMC. For this, we select an $(\ell + d)$ -dimensional low-discrepancy point set

$$\tilde{Q}_n = \{(\mathbf{w}_0, \tilde{\mathbf{u}}_0), \dots, (\mathbf{w}_{n-1}, \tilde{\mathbf{u}}_{n-1})\},$$

where $\mathbf{w}_i \in [0, 1]^\ell$ and $\tilde{\mathbf{u}}_i \in [0, 1]^d$, and we define a randomization of $\tilde{P}_n = \{\tilde{\mathbf{u}}_0, \dots, \tilde{\mathbf{u}}_{n-1}\}$ with the following property. If $P_n = (\mathbf{U}_0, \dots, \mathbf{U}_{n-1})$ denotes (a realization of) the randomized version and if Q_n is the version of \tilde{Q}_n in which \tilde{P}_n is replaced by its randomized version P_n , then we must have: (a) each \mathbf{U}_i is uniformly distributed over $(0, 1)^d$ and (b) Q_n has low discrepancy (in a sense to be defined). The algorithm simulates (in parallel) n copies of the chain; it can be summarized as follows.

Array-RQMC algorithm:

For $i = 0, \dots, n - 1$, do $X_{i,0} = x_0$;

For $j = 1, 2, \dots$, until $j > \max_{0 \leq i < n} \tau_i$ {

Randomize \tilde{P}_n afresh (independently of the previous randomizations) into $P_n = \{\mathbf{U}_{0,j}, \dots, \mathbf{U}_{n-1,j}\}$;

For $i = 0, \dots, n - 1$, do $X_{i,j} = \varphi_j(X_{i,j-1}, \mathbf{U}_{i,j})$;

Map the n chains to the n points, and renumber the chains accordingly, so that $X_{i,j}$ is “close” to \mathbf{w}_i for each i (more on this later);

}

Estimate μ by the same average $\bar{Y}_n = \hat{\mu}_{\text{rqmc},n}$ as in (1).

This can be replicated m times to estimate the variance and compute a confidence interval on μ . The following is proved in [13]:

Proposition 1 (a) \bar{Y}_n is an unbiased estimator of μ and (b) the empirical variance of the m copies of \bar{Y}_n is an unbiased estimator of $\text{Var}[\bar{Y}_n]$.

A natural way of proving a bound of the form (3) would be as follows. Assume that D is defined via a reproducing kernel Hilbert space (RKHS) [2, 3] and that V is the corresponding variation. Suppose that Q_n is defined so that

$$\mathbb{E}[D_{(2)}^2(Q_n)] \leq \kappa' n^{-\alpha+\epsilon} \quad (4)$$

for any $\epsilon > 0$, where $\alpha \geq 1$, $\kappa' = \kappa'(\epsilon)$ may depend on ϵ but not on n , and where $D_{(2)}$ is a discrepancy defined over the $(\ell + d)$ -dimensional unit hypercube, with corresponding variation $V_{(2)}$. That is, the mean square integration error of a function $g : [0, 1]^{\ell+d} \rightarrow \mathbb{R}$ by Q_n would be bounded by $\mathbb{E}[D_{(2)}^2(Q_n)] \cdot V_{(2)}^2(g)$ if $V_{(2)}^2(g)$ is well defined.

Let $X_{0,j-1}, \dots, X_{n-1,j-1}$ be the states at step $j - 1$ sorted by the selected mapping (we suppose that ψ_j is the identity). We have $X_{i,j} = \varphi_j(X_{i,j-1}, U_{i,j})$. Define $\tilde{X}_{i,j} = \varphi_j(\mathbf{w}_i, U_{i,j})$ for all i and let $\tilde{D}_j = D(\tilde{X}_{0,j}, \dots, \tilde{X}_{n-1,j})$ be the discrepancy of this set of states. By the RKHS property, \tilde{D}_j is the integration error for some worst-case function ξ_j of bounded variation $V(\xi_j) < \infty$ by the point set $\{\tilde{X}_{0,j}, \dots, \tilde{X}_{n-1,j}\}$. But this \tilde{D}_j is also the integration error of the function $g_j = \xi_j \circ \varphi_j$ by Q_n , which implies that $\mathbb{E}[\tilde{D}_j^2] = O(n^{-\alpha+\epsilon})$ if $V_{(2)}(g_j) < \infty$. Then we would need to show that $\mathbb{E}[D_j^2]$ also converges at this rate, which could presumably be done by bounding $\mathbb{E}[\tilde{D}_j^2 - D_j^2]$ under an induction assumption of the form $\mathbb{E}[D_{j-1}^2] \leq \kappa_{j-1} n^{-\alpha+\epsilon}$ for some constant $\kappa_{j-1} < \infty$. Induction on j would complete the proof. The problem is to find appropriate definitions of D and $D_{(2)}$ such that the details can be filled up.

We tried this with $\ell = d = 1$, $\alpha = 2$, and both D and $D_{(2)}$ defined as the \mathcal{L}_2 -star discrepancy, motivated by the fact that it has a very simple expression in one dimension, and its mean square converges is $O(n^{-2+\epsilon})$ for good point sets. It is instructive to point out why this choice does not work. In this case, we have

$$D^2(x_0, \dots, x_{n-1}) = \frac{1}{12n^2} + \frac{1}{n} \sum_{i=0}^{n-1} (w_i - x_i)^2$$

where $w_i = (i + 1/2)/n$ and $0 \leq x_0 \leq x_1 \leq \dots \leq x_{n-1}$, and [2]:

$$\xi_j(x) = -\frac{1}{n} \sum_{i=0}^{n-1} \left[\mu(\tilde{X}_{i,j}) + B_2((x - \tilde{X}_{i,j}) \bmod 1) + B_1(x)B_1(\tilde{X}_{i,j}) \right],$$

for some (unimportant) function μ , and where B_1 and B_2 are the Bernoulli polynomials defined by $B_1(x) = x - 1/2$ and $B_2(x) = x^2 - x + 1/6$. We also have

$$\frac{d\xi_j(x)}{dx} = -x + \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{I}[\tilde{X}_{i,j} < x]$$

where \mathbb{I} denotes the indicator function. This derivative is square integrable over $(0, 1)$, so $V(\xi_j) < \infty$. However, $g_j = \xi_j \circ \varphi_j$ is a two-dimensional function and its variation $V_{(2)}(g_j)$ can be finite only if its mixed derivative with respect to the two coordinates is square integrable. But it is not, because the indicators in the above expression are discontinuous in x , and the mixed derivative of g_j involves the second derivative of ξ_j . Therefore, g_j does not have finite variation.

In our numerical experiments, the mean square \mathcal{L}_2 -star discrepancy never converged at a rate near $O(n^{-2})$ even for the smoothest functions φ_j that we could try, except for the trivial cases where $\varphi_j(X_{j-1}, \mathbf{U}_j)$ depends only on X_{j-1} or only on \mathbf{U}_j . The rates observed empirically were closer to $O(n^{-3/2})$. On the other hand, in many examples, we observed a convergence rate of $O(n^{-2})$ for the variance.

4 Mapping the chains to the points

Recall that both the chain's states and the points are assumed to be in $[0, 1]^\ell$. At each step, we want to select a one-to-one mapping that assigns each state to a representative point that is close to it. If the corresponding states $X_{i,j}$ and points \mathbf{w}_i were identical, then $\hat{\mu}_{\text{rQMC},j,n}$ would be exactly the same as a QMC estimator of μ_j based on the point set $S_{n,j} = \{\mathbf{w}_0, \dots, \mathbf{w}_{n-1}\} \subset [0, 1]^\ell$.

We consider the following way of mapping the chains to the points, called a *multivariate sort* [6, 1]. Select some positive integers n_1, \dots, n_ν such that $\nu \geq \ell$ and $n_1 \cdots n_\nu = n$. Sort the states (i.e., the chains) by their first coordinate, in n_1 packets of size n/n_1 . This means that any state in a given packet will have its first coordinate smaller or equal to the first coordinate of any other state in the next packet. Then sort each packet by the second coordinate, in n_2 packets of size $n/n_1 n_2$, and so on. When we reach coordinate ℓ , we sort each packet in n_ℓ packets of size $n/n_1 \cdots n_\ell$ by the last coordinate. If $\nu > \ell$, then at the next step we sort each packet into $n_{\ell+1}$ packets according to the first coordinate, and so on. As a special case of this, one can take $n_j = 2$ for all j , with n equal to a power of 2. This corresponds to splitting each packet of states in two with respect to the next coordinate, and doing this for each coordinate in a round-robin fashion.

If ℓ is deemed too large, we can map the state space to a lower-dimensional space as follows. Define a *sorting function* $v : \mathcal{X} \rightarrow [0, 1]^c$, for $c < \ell$, and apply the multivariate sort to the transformed points $v(X_{i,j})$, in c dimensions. The function v should be selected so that two states mapped to nearby values in $[0, 1]^c$ should be approximately equivalent in terms of the probability distribution of future costs when we are in these two states. In [13], it was assumed that such a mapping was always used, with $c = 1$, so v uniquely determined the sort, whence the appellation ‘‘sorting function.’’

Figure 1: A mapping with $n_1 = n = 16$ in two dimensions

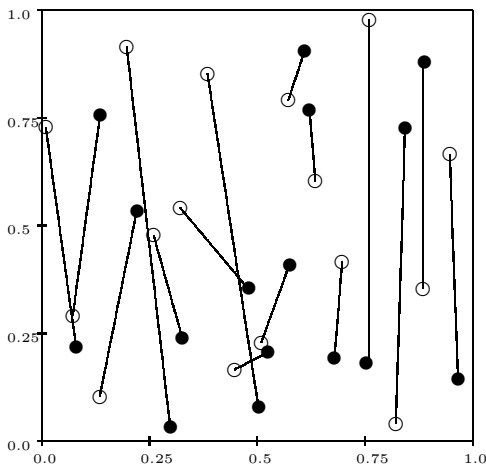
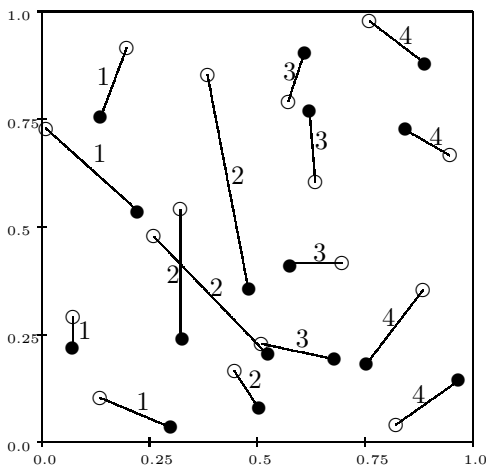


Figure 2: A mapping with $n_1 = n^{1/2} = 4$ in two dimensions



Figures 1 and 2 illustrate the mappings obtained for two choices of n_1 , for an example with $\ell = 2$ and $n = 16$. The black dots represent the states of the chains, and the white dots represent the first 16 points of the two-dimensional Sobol' sequence. The lines indicate the mapping between the two sets of points.

In Figure 1, we see the case where $n_1 = n$, which means that we sort according to the first coordinate only. Here, the leftmost state is mapped to the leftmost point, the second leftmost state is mapped to the second leftmost point, and so on.

Figure 2 shows the mapping with $n_1 = n^{1/2} = 4$, for the same points. Here, we first sort both the points and the states in four packets according to the first coordinate. The numbers from 1 to 4 indicate the packet number in which each pair ended up in this first sort. Within each packet, the states are mapped to the points according to the second coordinate.

In the more general (and realistic) case where the state space is not $[0, 1]^\ell$ but \mathbb{R}^ℓ (or a subset) and the ψ_j cannot be computed explicitly, then a reasonable heuristic is to simply sort the states in the real space in exactly the same way as in the unit hypercube. This is what we will do in our examples.

5 Empirical investigations of the convergence rate

We now show how the mean square \mathcal{L}_2 -star discrepancy $\mathbb{E}[D_j^2]$ and the variance behave as functions of j and n , for small examples. All mean square discrepancies and variances were estimates from 100 independent replications of the array-RQMC estimator.

Example 1 Consider a Markov chain defined over the real line by

$$Y_1 = Z_1, \quad Y_j = \rho Y_{j-1} + (1 - \rho)Z_j \quad \text{for } j \geq 2,$$

where $0 < \rho < 1$ (a constant) and Z_1, Z_2, \dots are i.i.d. $N(0, 1)$ (standard normal). Then, $Y_j \sim N(0, 1)$ and $X_j = \Phi(Y_j) \sim U(0, 1)$, where Φ is the standard normal (cumulative) distribution function. That is, we transform the state so that it has the uniform distribution at each step j , to be able to compute explicitly the mean square discrepancy $\mathbb{E}[D_j^2]$ and see how it evolves with j and n . The Markov chain can also be defined directly in terms of a stochastic recurrence for X_j , namely $X_1 = U_1$ and

$$X_j = \varphi_j(X_{j-1}, U_j) = \Phi(\rho\Phi^{-1}(X_{j-1}) + (1 - \rho)\Phi^{-1}(U_j)) \quad \text{for } j \geq 2,$$

where U_1, U_2, \dots are i.i.d. $U(0, 1)$. In this example, we will examine $\mathbb{E}[D_j^2]$ as a function of j and of n , and also the variance of $c_j(X_j) = X_j$ (that is, the variance of the average cost at step j for this simple cost function) as a function of n , for $j = 20$ and $j = 100$ steps.

The RQMC point set used at each stage was the first n points of the two-dimensional Sobol' sequence, where the second coordinate of the points was randomized by a (random) left matrix scramble followed by a random digital shift [17]. The simulations were done using SSJ [10].

Figure 3 shows our estimate of $\mathbb{E}[D_j^2]$ as a function of j , with $n = 4096$ points, for $\rho = 1/11, 1/2$, and $10/11$. The mean square discrepancy turns out to be quite stable even when we simulate this chain over a large number of steps. This is very encouraging.

In Figure 4, we see our estimate of $\log_2 \mathbb{E}[D_j^2]$ as a function of $\log_2 n$, for selected values of ρ and j . In all cases, $\mathbb{E}[D_j^2]$ seems to converge approximately as $O(n^{-3/2})$ as a function of n . It is also practically independent of j , at least in these examples.

In these experiments, we also computed other types of discrepancies and the results were similar. The results were also similar when we tried other types of RQMC point sets, such as a randomly shifted lattice rule with a baker's transformation [4]. In this case, a specialized discrepancy for this particular type of RQMC point set, defined in [4] was much smaller than the discrepancy shown here (by a constant factor), but had the same convergence rate.

Figure 5 shows our estimate of $\log_2 \text{Var}[\hat{\mu}_{\text{rqmc},j,n}]$ as a function of $\log_2 n$. Here, we observe a convergence rate of approximately $O(n^{-2})$, which is faster than for the mean square discrepancy. Here, the n chains were simulated for j steps, the cost was then averaged over the n chains (at step j) to get one realization of the estimator $\hat{\mu}_{\text{rqmc},j,n}$. This was repeated $m = 100$ times, and the variance shown is the empirical variance of those m observations. The variance reduction factor, defined as the Monte Carlo variance divided by the array-RQMC variance when the two estimators are based on an average for n chains, is very roughly $600n$ when $\rho = 0.1$ and $j = 100$ (although there is significant fluctuation around this value when we change n and especially the RQMC point set that is used). The variance is also practically independent of j .

Example 2 In this example, let $0 < t_1 < t_2 < \dots < t_s = T$ be fixed numbers (observation times), r and σ be positive constants, $S_0 = s_0$ (a constant), and

$$S_j = S_{j-1} \exp[(r - \sigma^2/2)(t_j - t_{j-1}) + \sigma(t_j - t_{j-1})^{1/2}\Phi^{-1}(U_j)] \quad (5)$$

where $U_j \sim U[0, 1)$, for $j = 1, \dots, s$. Define

$$\bar{S}_j = \frac{1}{j} \sum_{i=1}^j S_i.$$

Figure 3: Estimate of $\mathbb{E}[D_j^2]$ as a function of j for Example 1.

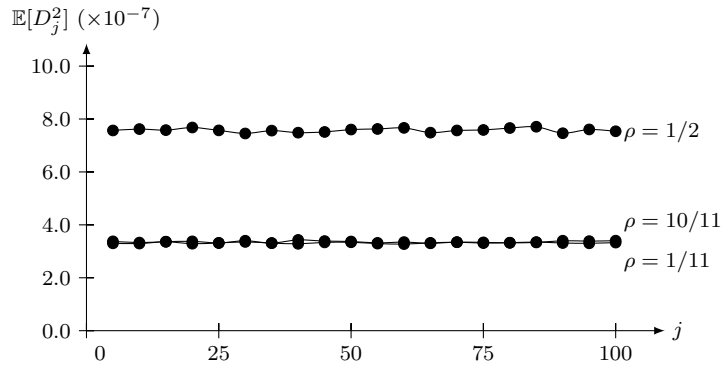


Figure 4: Estimate of $\log_2 \mathbb{E}[D_j^2]$ as a function of $\log_2 n$ for Example 1.

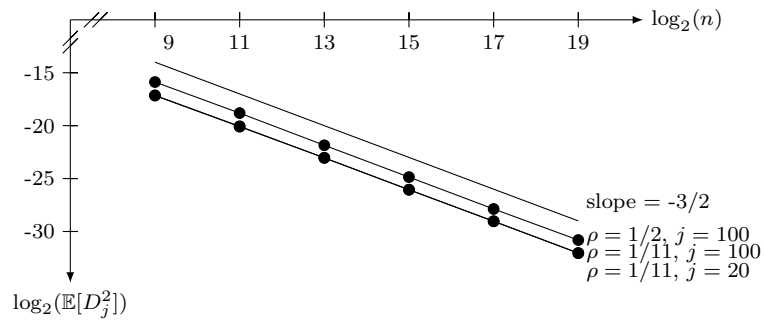
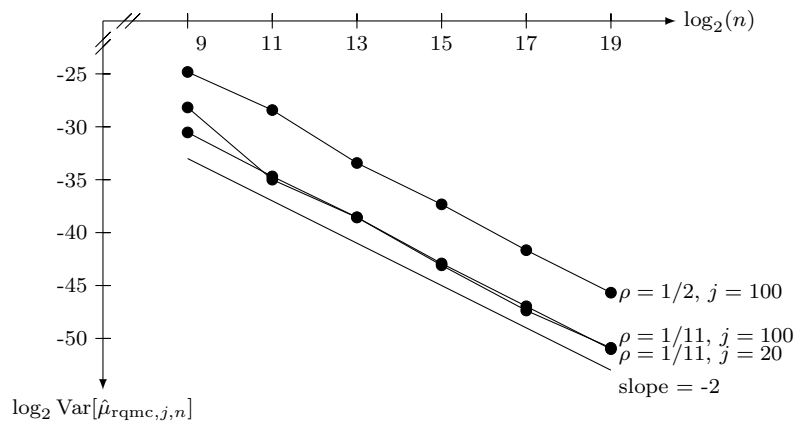


Figure 5: Estimate of $\log_2 \text{Var}[\hat{\mu}_{\text{rqmc},j,n}]$ as a function of $\log_2 n$, for Example 1.



We want to estimate

$$\mu = \mathbb{E} [\max(0, \bar{S}_s - K)].$$

This estimation problem occurs in pricing an Asian call option for a single asset whose price evolves as a geometric Brownian motion [5, 14]. Note that μ is then multiplied by a constant discount factor, which we ignore here.

To put this model in our framework, we define a Markov chain with state $X_j = (S_j, \bar{S}_j)$ at step j , and whose transitions obey $(S_j, \bar{S}_j) = \varphi_j(S_{j-1}, \bar{S}_{j-1}, U_j)$ where φ_j is defined via (5) and $\bar{S}_j = [(j-1)\bar{S}_{j-1} + S_j]/j$. The function c_j is zero for $j < s$ and we have $c_s(S_s, \bar{S}_s) = \max(0, \bar{S}_s - K)$. Here, $\tau = s$, a constant. We have a two-dimensional state space, and we use a two-dimensional sort at each step: we first sort the states in n_1 packets of size n/n_1 based on $S(t_j)$, then we sort the packets based on \bar{S}_j .

In contrast with the previous example, we have no explicit mapping ψ_j available to transform the state into a uniform point over the unit square, so we cannot compute the discrepancy D_j explicitly. However, we can estimate the variance and examine its convergence speed as a function of n . Our RQMC point set at each step is still the first n points of a Sobol' sequence, this time in three dimensions, with coordinates 2 and 3 randomized by a left matrix scramble followed by a random digital shift.

For a numerical example, we take $S(0) = 100$, $K = 90$, $T = 240/365$, $t_1 = T - (s-1)/365$, $t_j - t_{j-1} = 1/365$, $r = \ln 1.09$, $\sigma = 0.2$, and $s = 10$ and 60 .

Figures 6 and 7 show the variance as a function of n , again in a log-log scale, for different choices of n_1 as a function of n , for $s = 10$ and $s = 60$, respectively. The best results are with $n_1 \approx n^{1/2}$, for which the variance seems to converge approximately as $O(n^{-2})$. For $n_1 \approx n^{1/3}$ and $n_1 \approx n^{2/3}$, the variance is larger (by a factor of about 10 for $s = 60$, $n \approx 2^{18}$, and $n_1 \approx n^{1/3}$, for example). The results are even worse if we take $n_1 = 1$ or $n_1 = n$, which corresponds to sorting the states by one of their two coordinates, and is the strategy that was used for this same example in [13]. For $s = 60$ and $n \approx 2^{18}$, for example, the variance with the best two-dimensional sort adopted here is about 400 times smaller than with a sort based on the second coordinate only. We also observe that sorting only (or more) on the first coordinate gives better results than sorting only (or more) on the second coordinate.

Figure 8 shows a similar plot, but with the coordinates of the state reversed. This means that we sort first in n_1 packets with respect to \bar{S}_j , and then sort the packets with respect to S_j . The results are similar.

Figure 6: Estimate of $\log_2 \text{Var}[\hat{\mu}_{\text{rqmc},s,n}]$ as a function of $\log_2 n$, for Example 2 with $s = 10$. The dotted line shows $\log_2 \text{Var}[\hat{\mu}_n]$, for comparison.

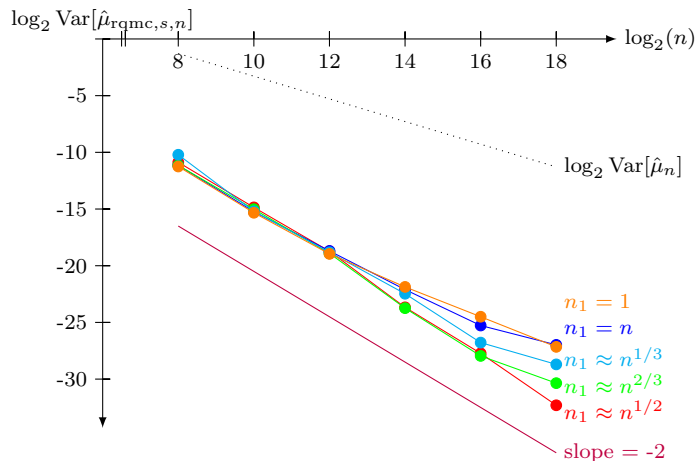


Figure 7: Estimate of $\log_2 \text{Var}[\hat{\mu}_{\text{rqmc},s,n}]$ as a function of $\log_2 n$, for Example 2 with $s = 60$. The dotted line shows $\log_2 \text{Var}[\hat{\mu}_n]$, for comparison.

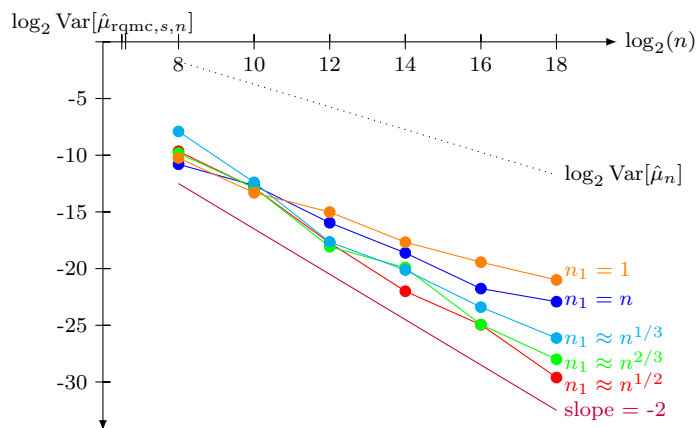
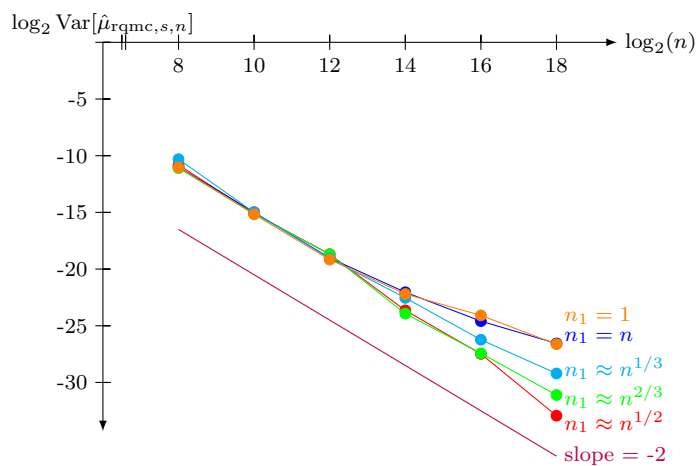


Figure 8: Estimate of $\log_2 \text{Var}[\hat{\mu}_{\text{rqmc},s,n}]$ as a function of $\log_2 n$, for Example 2 with $s = 10$, when the state is taken as $(\bar{S}_j, S(t_j))$.



Interestingly, it is empirically better to take n_1 larger than n/n_1 rather than the opposite, regardless of how we order the coordinates of the state.

6 Conclusion

The array-RQMC algorithm is a promising methodology for reducing the variance in the simulation of Markov chains. So far, we have only scratched the surface for its convergence analysis. We believe that plenty of interesting results are just waiting to be established in this direction, in particular for multidimensional state spaces, for various choices of discrepancy measures for the set of states of the chains and corresponding smoothness assumptions on the cost functions. More empirical experimentation is also needed, with large examples, alternative sorting strategies, and various classes of applications.

References

- [1] R. El Haddad, C. Lécot, and P. L'Ecuyer. Quasi-Monte Carlo simulation of discrete-time Markov chains on multidimensional state spaces. In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 413–429, Berlin, 2008. Springer-Verlag.
- [2] F. J. Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67:299–322, 1998.
- [3] F. J. Hickernell. What affects the accuracy of quasi-Monte Carlo quadrature? In H. Niederreiter and J. Spanier, editors, *Monte Carlo and Quasi-Monte Carlo Methods 1998*, pages 16–55, Berlin, 2000. Springer-Verlag.
- [4] F. J. Hickernell. Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules. In K.-T. Fang, F. J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 274–289, Berlin, 2002. Springer-Verlag.
- [5] J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice-Hall, Upper Saddle River, N.J., sixth edition, 2006.
- [6] C. Lécot and I. Coulibaly. A quasi-Monte Carlo scheme using nets for a linear Boltzmann equation. *SIAM Journal on Numerical Analysis*, 35:51–70, 1998.
- [7] C. Lécot and S. Ogawa. Quasirandom walk methods. In K.-T. Fang, F. J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 63–85, Berlin, 2002. Springer-Verlag.
- [8] C. Lécot and B. Tuffin. Quasi-Monte Carlo methods for estimating transient measures of discrete time Markov chains. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 329–343, Berlin, 2004. Springer-Verlag.
- [9] P. L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 2008. To appear.
- [10] P. L'Ecuyer. *SSJ: A Java Library for Stochastic Simulation*, 2008. Software user's guide, Available at <http://www.iro.umontreal.ca/~lecuyer>.
- [11] P. L'Ecuyer, V. Demers, and B. Tuffin. Rare-events, splitting, and quasi-Monte Carlo. *ACM Transactions on Modeling and Computer Simulation*, 17(2):Article 9, 2007.
- [12] P. L'Ecuyer, C. Lécot, and B. Tuffin. Randomized quasi-Monte Carlo simulation of Markov chains with an ordered state space. In H. Niederreiter and D. Talay, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 331–342, 2006.
- [13] P. L'Ecuyer, C. Lécot, and B. Tuffin. A randomized quasi-Monte Carlo simulation method for Markov chains. *Operations Research*, 56(4):958–975, 2008.
- [14] P. L'Ecuyer and C. Lemieux. Variance reduction via lattice rules. *Management Science*, 46(9):1214–1235, 2000.
- [15] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, PA, 1992.
- [16] A. B. Owen. Latin supercube sampling for very high-dimensional simulations. *ACM Transactions on Modeling and Computer Simulation*, 8(1):71–102, 1998.
- [17] A. B. Owen. Variance with alternative scramblings of digital nets. *ACM Transactions on Modeling and Computer Simulation*, 13(4):363–378, 2003.
- [18] I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford, 1994.
- [19] C. Wächter and A. Keller. Efficient simultaneous simulation of Markov chains. In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 669–684, Berlin, 2008. Springer-Verlag.