

Randomized Quasi-Monte Carlo*

Pierre L'Ecuyer

DIRO, Université de Montréal, Canada (E-mail: lecuyer@iro.umontreal.ca)

Keywords: *Monte Carlo, quasi-Monte Carlo, RQMC, low-discrepancy sequence, numerical integration*

Abstract: Monte Carlo (MC) methods use *independent* uniform random numbers to sample realizations of random variables and sample paths of stochastic processes, often to estimate high-dimensional integrals that can represent mathematical expectations. Randomized quasi-Monte Carlo (RQMC) methods replace the independent random numbers by *dependent* vectors of uniform random numbers that cover the space more evenly. When estimating an integral, they can provide unbiased estimators whose variance converges at a faster rate than with Monte Carlo. RQMC can also be effective for the simulation of Markov chains, to approximate or optimize functions, to solve partial differential equations, for density estimation, etc.

Monte Carlo, QMC, and RQMC

Quasi-Monte Carlo (QMC) and randomized quasi-Monte Carlo (RQMC) methods were introduced to improve on standard Monte Carlo (MC) when estimating the mathematical expectation $\mu = \mathbb{E}[X]$ of a random variable X . In simulation applications, the expectation can be rewritten as a multivariate integral of the form

$$\mu = \mathbb{E}[X] = \mathbb{E}[f(\mathbf{U})] = \int_{(0,1)^s} f(\mathbf{u}) \, d\mathbf{u} \quad (1)$$

where $f : (0,1)^s \rightarrow \mathbb{R}$ is a function, \mathbf{U} is a vector of s uniform random variables that represent the noise injected into a simulation, and s is an upper bound on how many random numbers we need. We can allow s to be infinite.

With standard *Monte Carlo* (MC), one estimates μ by the average of n independent replicates of X , that is,

$$\hat{\mu}_{n,\text{mc}} = \frac{1}{n} \sum_{i=0}^{n-1} X_i \quad (2)$$

where $X_i = f(\mathbf{U}_i)$ and $\mathbf{U}_0, \dots, \mathbf{U}_{n-1}$ are n independent random vectors uniform over $(0,1)^s$, usually simulated using random number generators. One has $\mathbb{E}[\hat{\mu}_{n,\text{mc}}] = \mu$ (unbiasedness) and $\text{Var}[\hat{\mu}_{n,\text{mc}}] = \sigma^2/n$ with $\sigma^2 := \int_{(0,1)^s} f^2(\mathbf{u}) \, d\mathbf{u} - \mu^2$. When $\sigma^2 < \infty$, this estimator obeys the central limit theorem (CLT) $\sqrt{n}(\hat{\mu}_{n,\text{mc}} - \mu)/S_n \Rightarrow \mathcal{N}(0,1)$ when $n \rightarrow \infty$, where $S_n^2 = \frac{1}{n-1} \sum_{i=0}^{n-1} (X_i - \bar{X}_n)^2$ is the sample variance. This CLT implies

*To appear in Wiley StatsRef: Statistics Reference Online, John Wiley, Article 08240, 2020.

that the width of a confidence interval on μ is asymptotically proportional to σ/\sqrt{n} . This $\mathcal{O}(n^{-1/2})$ convergence is slow.

Quasi-Monte Carlo (QMC) methods replace the independent random vectors \mathbf{U}_i by a *deterministic low-discrepancy* set of n points $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\}$ that cover $[0, 1]^s$ more evenly. The estimate is

$$\hat{\mu}_{n,\text{qmc}} = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{u}_i).$$

Roughly speaking, a point set P_n is said to have *low-discrepancy* if a given measure of discrepancy between the empirical distribution of P_n and the uniform distribution converges to 0 at a faster rate than for independent points. A discrepancy is usually defined jointly with a measure of variation of f in some functional space \mathcal{H} (often a Hilbert space), in a way that the following type of worst-case error bound can be obtained via the Cauchy-Schwarz inequality:

$$|\hat{\mu}_{n,\text{qmc}} - \mu| \leq D(P_n)V(f) \tag{3}$$

for all $f \in \mathcal{H}$, where $V(f) = \|f - \mu\|_{\mathcal{H}}$ measures the *variation* of f in \mathcal{H} and $D(P_n)$ is the *discrepancy* of P_n [6, 12, 44]. This bound is a product of two terms, one that depends only on the point set P_n and the other that depends only on f . For any $f \in \mathcal{H}$ with $V(f) \neq 0$, the bound (3) converges at the same rate as $D(P_n)$. The error itself may converge faster than the bound.

The best-known special case of (3) is the classical Koksma-Hlawka inequality, in which $D(P_n)$ is the *star discrepancy* $D^*(P_n)$ defined as follows: for each $\mathbf{u} \in (0, 1)^s$, let $\Delta(\mathbf{u})$ be the absolute difference between the volume of the box $[\mathbf{0}, \mathbf{u})$ and the fraction of P_n that fall in that box, and let $D^*(P_n)$ be the supremum of $\Delta(\mathbf{u})$ over all $\mathbf{u} \in (0, 1)^s$. The corresponding variation $V(f)$ is the Hardy-Krause variation of f , $V_{\text{HK}}(f)$, whose definition and discussion can be found in [1, 6]. Note that $V_{\text{HK}}(f)$ is infinite whenever f has a discontinuity not aligned with the axes, in two or more dimensions. There are known explicit point sets P_n for which $D^*(P_n) = \mathcal{O}(n^{-1}(\ln n)^{s-1})$, and also infinite sequences of points for which $D^*(P_n) = \mathcal{O}(n^{-1}(\ln n)^s)$ if P_n is defined as the first n points of the sequence. There are many other useful choices for the discrepancy and variation in (3). They are usually defined in a way that the discrepancy is easy to compute for the selected type of point set construction, that one can search efficiently for point sets having a small discrepancy, and they often give different weights to the projections of the points (and of f) on different subsets of coordinates. See the next section for one more example. Classical references on QMC include [6, 5, 44, 51, 52].

In applications, the bound in (3) is usually much too hard to compute; only its convergence rate is known, and it is very hard to estimate the QMC integration error. *Randomized QMC* (RQMC) addresses this issue by providing an unbiased estimator of μ whose variance converges at worst at the same rate as the square of the QMC error bound (and sometimes at a better rate). This is achieved by randomizing the point set P_n in a way that each individual point has the uniform distribution over the unit cube $(0, 1)^s$ while the point set as a whole preserves its low discrepancy. The RQMC estimator of μ is

$$\hat{\mu}_{n,\text{rqmc}} = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{U}_i).$$

where $\mathbf{U}_0, \dots, \mathbf{U}_{n-1}$ are now the randomized points. The variance of this estimator can itself be estimated by replicating the RQMC scheme m times independently, to obtain m independent realizations of $\hat{\mu}_{n,\text{rqmc}}$, and computing their empirical mean and variance. This can be used to estimate the error and perhaps compute a confidence interval. However, one must keep in mind that the distribution of this mean does not always converge to

a normal for $n \rightarrow \infty$ [35]. It does for $m \rightarrow \infty$, and also for $n \rightarrow \infty$ for specific types of (more costly) randomizations such as nested scrambling [40]; otherwise one must be careful when computing confidence intervals.

One may wonder how it is possible to fill the unit hypercube $(0, 1)^s$ very uniformly when s is large. For $s = 100$, for instance, it already takes 2^{100} points to have one in each quadrant. This is far too many points to be practical. So how can RQMC possibly work for high-dimensional functions, say for $s > 30$ with no more than 2^{20} points? The reason is that when it works, it is typically because f can be well approximated by a sum of low-dimensional functions: $f(\mathbf{u}) = \sum_{\mathbf{u} \subseteq \{1, \dots, s\}} f_{\mathbf{u}}(\mathbf{u})$, where $f_{\mathbf{u}}(\mathbf{u})$ depends only on the coordinates of \mathbf{u} that belong to the set \mathbf{u} , and the variance σ^2 decomposes accordingly as the sum of the variances $\sigma_{\mathbf{u}}^2 = \text{Var}[f_{\mathbf{u}}(\mathbf{U})]$. This is known as an ANOVA decomposition of f . It often occurs that only a small fraction of the $\sigma_{\mathbf{u}}^2$'s contribute to almost all the variance σ^2 . Then, to obtain a significant variance reduction, it is sufficient that the projections of the point P_n on those important subsets \mathbf{u} be very uniform. This can be achieved by giving more weights to these projections in the discrepancy criterion used to construct the points. In practice, these variance components do not need to be known accurately, only very rough estimates suffice. The faster convergence rates of RQMC vs MC are proved under conditions that do not always hold, but RQMC can nevertheless reduce the variance by large factors even when these conditions do not hold.

RQMC sampling can be useful in other settings than estimating an integral. It can also improve efficiency when estimating a quantile, a density, a function of several expectations, the derivative of an expectation, and the solution of an optimization problem (e.g., for maximum likelihood estimation) [2, 17, 43]. Active areas of application of RQMC include option pricing in finance [8, 10, 17, 24, 56], simulating partial and stochastic differential equations [18, 17], statistics, and computer graphics [55]. Further details on RQMC in general can be found in [31, 24, 25, 37, 49, 50, 54].

Construction and Randomization of Points Sets

The two main types of point set constructions are lattice rules and digital nets. We now look at how they are defined and how the points are randomized.

Lattice rules

In a *rank-1 lattice rule*, the points are defined as $\mathbf{u}_i = (i/n)\mathbf{a} \bmod 1$, $i = 0, \dots, n-1$, where $\mathbf{a} \in \{0, \dots, n-1\}^s$ is called the generating vector, and “mod” means that we take the fractional part of each vector coordinate. This lattice point set P_n can be randomized by a random shift modulo 1, which means that a single random vector \mathbf{U} is generated uniformly over $(0, 1)^s$ and added to all the points \mathbf{u}_i , modulo 1. This yields an RQMC method known as a *randomly-shifted lattice rule*. The variance of the corresponding RQMC estimator is exactly (see [30]):

$$\text{Var}[\hat{\mu}_{n,\text{rqmc}}] = \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} |\hat{f}(\mathbf{h})|^2, \quad (4)$$

where $L_s^* \subset \mathbb{Z}^s$ is the dual of the lattice generated by the points of P_n and the $\hat{f}(\mathbf{h})$ are the Fourier coefficients of f [30]. To minimize the variance, ideally we would like to select a lattice that minimizes (4), which is very hard in general, because there is an infinite number of Fourier coefficients in the sum, and they are usually unknown. A more practical approach is the following. Suppose that for an even integer $\alpha > 0$, the mixed partial derivatives of f up to order $\alpha/2 > 0$ are square-integrable, and the periodic continuations of the derivatives of f up to order $\alpha/2 - 1$ are continuous across the boundaries of the unit hypercube modulo 1, then one has

$|\hat{f}(\mathbf{h})|^2 = \mathcal{O}((\max(1, h_1) \cdots \max(1, h_s))^{-\alpha})$. Moreover, for any $\epsilon > 0$, there is always a choice of \mathbf{a} such that

$$\mathcal{P}_\alpha := \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} (\max(1, h_1) \cdots \max(1, h_s))^{-\alpha} = \mathcal{O}(n^{-\alpha+\epsilon}). \quad (5)$$

This \mathcal{P}_α is the variance for a worst-case f with square Fourier coefficients $|\hat{f}(\mathbf{h})|^2 = (\max(1, |h_1|) \cdots \max(1, |h_s|))^{-\alpha}$. The larger is α , the smoother is f and the faster is the convergence rate.

A more general version of \mathcal{P}_α gives different weights $\gamma_{\mathbf{u}}$ to the different projections (subsets of coordinates) $\mathbf{u} \subseteq \{1, \dots, s\}$. Let

$$\mathcal{P}_{\gamma, \alpha} = \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} \gamma_{\mathbf{u}(\mathbf{h})} (\max(1, |h_1|) \cdots \max(1, |h_s|))^{-\alpha},$$

in which $\mathbf{u}(\mathbf{h}) = \mathbf{u}(h_1, \dots, h_s) = \{j : h_j \neq 0\}$. The idea here is to give larger weights to the projections deemed more important (having larger $\sigma_{\mathbf{u}}^2$).

This \mathcal{P}_α or $\mathcal{P}_{\gamma, \alpha}$ criterion is defined for any real number $\alpha > 1$, but we know how to compute it exactly only when α is an even integer. In that case, we have

$$\mathcal{P}_{\gamma, \alpha} = \sum_{\emptyset \neq \mathbf{u} \subseteq \{1, \dots, s\}} \frac{1}{n} \sum_{i=0}^{n-1} \gamma_{\mathbf{u}} \left[\frac{-(-4\pi^2)^{\alpha/2}}{(\alpha)!} \right]^{|\mathbf{u}|} \prod_{j \in \mathbf{u}} B_\alpha(u_{i,j}), \quad (6)$$

where B_α is the Bernoulli polynomial of degree α (e.g., $B_1(u) = u - 1/2$, $B_2(u) = u^2 - u + 1/6$, etc.). The corresponding (squared) *variation* for $f : [0, 1]^s \rightarrow \mathbb{R}$ smooth enough is

$$V_\gamma^2(f) = \sum_{\emptyset \neq \mathbf{u} \subseteq \{1, \dots, s\}} \frac{1}{\gamma_{\mathbf{u}} (4\pi^2)^{\alpha|\mathbf{u}|/2}} \int_{[0,1]^{|\mathbf{u}|}} \left| \frac{\partial^{\alpha|\mathbf{u}|/2}}{\partial \mathbf{u}^{\alpha/2}} f_{\mathbf{u}}(\mathbf{u}) \right|^2 d\mathbf{u},$$

and one has

$$\text{Var}[\hat{\mu}_{n, \text{rqmc}}] = \sum_{\mathbf{u} \subseteq \{1, \dots, s\}} \text{Var}[\hat{\mu}_{n, \text{rqmc}}(f_{\mathbf{u}})] \leq V_\gamma^2(f) \mathcal{P}_{\gamma, \alpha}. \quad (7)$$

This $\mathcal{P}_{\gamma, \alpha}$ is a good practical choice of figure of merit for lattice rules [7, 13, 32]. The weights are usually chosen to have a specific form with a small number of parameters, such as order-dependent weights for which $\gamma_{\mathbf{u}}$ depends only on the cardinality of \mathbf{u} , or product weights for which a weight γ_j is selected for each coordinate j and $\gamma_{\mathbf{u}} = \prod_{j \in \mathbf{u}} \gamma_j$, for example [7, 32, 52]. Generating vectors \mathbf{a} with a small $\mathcal{P}_{\gamma, \alpha}$ can be found with the *Lattice Builder* software [33] for arbitrary n , s , and weights. The software supports other criteria as well.

The bound on the square Fourier coefficients here holds under conditions (e.g., f is periodic and sufficiently smooth) that do not always hold in applications. But often, f can be changed into a function having the same integral and that satisfies the conditions, via a change of variable. For example, if f is discontinuous at the boundary, i.e., $f(\dots, u_j = 0, \dots) \neq f(\dots, u_j = 1, \dots)$, one can make the change of variable $\tilde{u}_j = 2u_j$ if $u_j \leq 1/2$ and $1 - 2u_j$ if $u_j > 1/2$ [14]. It stretches the points by a factor of 2 from $[0, 1]$ to $[0, 2]$, then folds the segment $[1, 2]$ back to $[1, 0]$. This is equivalent to shrinking f horizontally by a factor of $1/2$ and making a mirror copy on the interval $[1/2, 1)$, which makes f continuous. Higher-order transformations can also make the derivatives continuous and sometimes further improve the convergence rate of the variance. However, they may also increase the variation of f and the variance for the values of n that are used [24, 51].

Digital nets

To define a *digital net in base b* , select a base $b \geq 2$, usually a prime and most often $b = 2$, an integer $k > 0$, and a set of s *generating matrices* $\mathbf{C}_1, \dots, \mathbf{C}_s$, which are $w \times k$ matrices with elements in \mathbb{Z}_b . To define the $n = b^k$ points \mathbf{u}_i , for $i = 0, \dots, b^k - 1$ and $j = 1, \dots, s$, put $i = a_{i,0} + a_{i,1}b + \dots + a_{i,k-1}b^{k-1} = a_{i,k-1} \dots a_{i,1}a_{i,0}$,

$$\begin{pmatrix} u_{i,j,1} \\ \vdots \\ u_{i,j,w} \end{pmatrix} = \mathbf{C}_j \begin{pmatrix} a_{i,0} \\ \vdots \\ a_{i,k-1} \end{pmatrix} \pmod{b}, \quad u_{i,j} = \sum_{\ell=1}^w u_{i,j,\ell} b^{-\ell}, \quad \text{and } \mathbf{u}_i = (u_{i,1}, \dots, u_{i,s}).$$

In practice, w and k are finite, but there is no limit. The definition in [44] is slightly more general. Assuming that each \mathbf{C}_j has full rank, each one-dimensional projection truncated to its first k digits is $\mathbb{Z}_n/n = \{0, 1/n, \dots, (n-1)/n\}$. If each \mathbf{C}_j has an infinite number of columns, this gives an infinite sequence of points, called a *digital sequence in base b* . One can always take the first $n = b^k$ points of a digital sequence to define a digital net, for any k .

The point set P_n is called a (t, k, s) -*net in base b* if for all non-negative integers q_1, \dots, q_s for which $q_1 + \dots + q_s = k - t$, if we partition $[0, 1)^s$ into b^{k-t} rectangular boxes by dividing the interval $[0, 1)$ into b^{q_j} equal parts in the j axis for each j , each box contains exactly b^{k-t} points. The smallest t for which a digital net is a (t, k, s) -net is called its t -value. A smaller t guarantees better uniformity of the points, so the ultimate value is $t = 0$, but this can be achieved only for $b \geq s - 1$. An infinite sequence of points in $[0, 1)^s$ is a (t, s) -*sequence in base b* if for any $k > 0$ and $\ell \geq 0$, the point set $\{\mathbf{u}_i, i = \ell b^k, \dots, (\ell + 1)b^k - 1\}$ is a (t, k, s) -net in base b . Beyond its intuitive interpretation, the t -value also has theoretical relevance: if P_n is a (t, k, s) -net in base b for $n = b^k$ and $k = 1, 2, 3, \dots$ and t is fixed or bounded, then $D^*(P_n) = \mathcal{O}(n^{-1}(\log n)^{s-1})$. Using Koksma-Hlawka, this provides the same convergence rate for the worst-case error, and $\mathcal{O}(n^{-2+\epsilon})$ rate for the RQMC variance, for any $\epsilon > 0$, when $V_{\text{HK}}(f) < \infty$. Moreover, there are known digital net constructions that achieve this property. The most popular type is in base $b = 2$ and was proposed by Sobol' [53]. It requires a selection of parameters called direction numbers. Specific ones are proposed in [16, 38], for example. Instead of selecting the parameters based on a single t -value in s dimensions, one can also use a criterion based on a weighted sum of the t -values of projections on subsets of coordinates, as we saw for the lattice rules. An algorithm that does this is given in [41].

Applying a random shift modulo 1 to a digital net does not preserve its (t, k, s) -net property. but applying a *random digital shift* does so and provides an RQMC point set. One generates a single $\mathbf{U} = (U_1, \dots, U_s)$ uniformly over $(0, 1)^s$ as before, find the digits in base b of the expansion of each coordinate U_j , and add these digits modulo b to the corresponding digits of the same coordinate of each point \mathbf{u}_i . For $b = 2$, the shift is applied by a bitwise xor, which is very fast. See [6, 25, 31] for more details and illustrations.

Another important type of randomization for digital nets is the *nested uniform scramble* (NUS) proposed by Owen [47, 48]. Let $b = 2$. As for the digital shift, for any given coordinate, with probability $1/2$ we flip the first bit for all the points, then we also flip the second bit with with probability $1/2$, and so on. But instead of flipping the second bit the same way for all the points, we make two independent flipping decisions, one for the points whose first bit is 0 and another for the points whose first bit is 1. And we perform this recursively: there are four independent flip decisions for the third bit, eight for the fourth bit, and so on. We can stop at the k th bit (or perhaps earlier) and generate the other bits randomly and independently across the points and coordinates. This also works for a general base b : replace the flip by a random permutation of the elements of \mathbb{Z}_b applied to each digit. NUS is much more costly than a random digital shift, but also more powerful. For sufficiently smooth f , with NUS applied to digital (t, k, s) -nets with fixed s and bounded t , the RQMC variance converges as $\mathcal{O}(n^{-3}(\log n)^{s-1}) = \mathcal{O}(n^{-3+\epsilon})$ [47, 48, 50]. Other types of scrambles that provide only a slower rate but are faster to apply are discussed in [15, 42, 50], for example.

Polynomial Lattice Rules

One important way of constructing digital nets is via lattice rules in spaces of polynomials or formal series, say with coefficients in the finite field \mathbb{F}_2 or more generally in the residue ring \mathbb{Z}_b of integers modulo b . Most practical implementations use $b = 2$ for efficiency reasons, so we take $b = 2$ in the following. A *polynomial lattice rule* of rank 1 in base 2 with $n = 2^k$ points in s dimensions is defined as follows [6, 44]. Let $\mathbb{F}_2[z]$ be the ring of polynomials with coefficients in \mathbb{F}_2 . Select a polynomial $Q(z)$ of degree k and a vector $\mathbf{a}(z) = (a_1(z), \dots, a_s(z))$ of polynomials of degree less than k which are relatively prime with $Q(z)$, all in $\mathbb{F}_2[z]$. For any other polynomial $h(z) \in \mathbb{F}_2[z]$ and $1 \leq j \leq s$, $h(z)a_j(z)/Q(z)$ is a formal power series of the form $\sum_{\ell=w}^{\infty} x_\ell z^{-\ell}$ with coefficients $x_\ell \in \mathbb{F}_2$, for some integer w . We define a mapping φ from the set of such formal series to the interval $[0, 1]$ by

$$\varphi\left(\sum_{\ell=w}^{\infty} x_\ell z^{-\ell}\right) = \sum_{\ell=\max(w,1)}^{\infty} x_\ell 2^{-\ell}.$$

Then P_n is the set of all points of the form $(\varphi(h(z)a_1(z)/Q(z)), \dots, \varphi(h(z)a_s(z)/Q(z)))$ where $h(z) \in \mathbb{F}_2[z]$ has degree less than k . Note that there are $n = 2^k$ distinct choices for $h(z)$. This P_n turns out to be a digital net in base 2 [23, 31, 39, 44] and its uniformity can be analyzed in the same way as a digital net. A very similar theory as for ordinary lattice rules has also been developed for polynomial lattice rules, in which the Fourier coefficients of f are replaced by Walsh coefficients, and counterparts of (7) and its other versions have been obtained by adapting the weighted P_α criterion in (6); see [6, 11, 45].

Latnet Builder [27] permits one to construct different types of point sets, including polynomial lattice rules and Sobol' points, with a variety of criteria. The Magic Point Shop [46] offers another set of tools.

RQMC for Markov Chains

In many applications, the expectation of interest is with respect to the distribution of the state of a Markov chain after several steps. If each step requires d uniform random numbers and there are τ steps, then we have an integral in $s = \tau d$ dimensions, which can be very large, in which case straightforward RQMC is likely to be ineffective. Special variants of RQMC have been designed to address this situation, the main one being *Array-RQMC* [28, 29, 34], based on ideas from [19, 21, 22]. The method simulates n dependent realizations of the chain, using an RQMC point set at each step to advance all the chains by one step, after sorting the chains in a specific order in terms of their states. For the sort, each state can be mapped to a real number using some well-selected *value function* and then the states are sorted by order of value, or one can use some form of multivariate sort [34]. The basic idea is to maintain an empirical distribution of the states, from step to step, that is closer to the exact theoretical distribution than if the n realizations were independent. Several successful applications and numerical examples can be found in [3, 4, 8, 20, 26, 29, 34, 36], where convergence rates of $\mathcal{O}(n^{-2})$ or even better for the variance are observed empirically on some examples. A closely related method was studied in [9] in the context of particle filters. These authors proved an $o(1)$ convergence rate for the variance.

Acknowledgements

This work has been supported by the Natural Sciences and Engineering Research Council of Canada Grant No. RGPIN-2018-05795.

References

1. K. Basu and A. B. Owen. Transformations and Hardy-Krause variation. *SIAM Journal on Numerical Analysis*, 54(3):1946–1966, 2016.
2. A. Ben Abdellah, P. L’Ecuyer, A. Owen, and F. Puchhammer. Density estimation by randomized quasi-Monte Carlo. Manuscript, 2019.
3. A. Ben Abdellah, P. L’Ecuyer, and F. Puchhammer. Array-RQMC for option pricing under stochastic volatility models. In *Proceedings of the 2019 Winter Simulation Conference*. IEEE Press, 2019.
4. V. Demers, P. L’Ecuyer, and B. Tuffin. A combination of randomized quasi-Monte Carlo with splitting for rare-event simulation. In *Proceedings of the 2005 European Simulation and Modeling Conference*, pages 25–32, Ghent, Belgium, 2005. EUROSIS.
5. J. Dick, F. Y. Kuo, and I. H. Sloan. High dimensional integration—the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.
6. J. Dick and F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, Cambridge, U.K., 2010.
7. J. Dick, I. H. Sloan, X. Wang, and H. Woźniakowski. Good lattice rules in weighted Korobov spaces with general weights. *Numerische Mathematik*, 103:63–97, 2006.
8. M. Dion and P. L’Ecuyer. American option pricing with randomized quasi-Monte Carlo simulations. In *Proceedings of the 2010 Winter Simulation Conference*, pages 2705–2720, 2010.
9. M. Gerber and N. Chopin. Sequential quasi-Monte Carlo. *Journal of the Royal Statistical Society, Series B*, 77(Part 3):509–579, 2015.
10. M. B. Giles, F. Y. Kuo, I. H. Sloan, and B. J. Waterhouse. Quasi-Monte Carlo for finance applications. *ANZIAM Journal*, 50:C308–C323, 2008.
11. T. Goda. Good interlaced polynomial lattice rules for numerical integration in weighted Walsh spaces. *Journal of Computational and Applied Mathematics*, 285:279–294, 2015.
12. F. J. Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67(221):299–322, 1998.
13. 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.
14. 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.
15. H. S. Hong and F. H. Hickernell. Algorithm 823: Implementing scrambled digital sequences. *ACM Transactions on Mathematical Software*, 29:95–109, 2003.
16. S. Joe and F. Y. Kuo. Constructing Sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30(5):2635–2654, 2008.
17. F. Y. Kuo and D. Nuyens. Hot new directions for quasi-Monte Carlo research in step with applications. In P. W. Glynn and A. B. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2016*, pages 123–144, Berlin, 2018. Springer-Verlag.
18. Frances Y. Kuo and Dirk Nuyens. Application of quasi-Monte Carlo methods to elliptic pdes with random diffusion coefficients: A survey of analysis and implementation. *Foundations of Computational Mathematics*, 16(6):1631–1696, 2016.
19. C. Lécot and I. Coulibaly. A quasi-Monte Carlo scheme using nets for a linear Boltzmann equation. *SIAM Journal on Numerical Analysis*, 35(1):51–70, 1998.
20. C. Lécot, P. L’Ecuyer, R. El Haddad, and A. Tarhini. Quasi-Monte Carlo simulation of coagulation-fragmentation. *Mathematics and Computers in Simulation*, 161:113–124, 2019.
21. 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.
22. 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.
23. P. L’Ecuyer. Polynomial integration lattices. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 73–98, Berlin, 2004. Springer-Verlag.
24. P. L’Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 13(3):307–349, 2009.
25. P. L’Ecuyer. Randomized quasi-Monte Carlo: An introduction for practitioners. In P. W. Glynn and A. B. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods: MCQMC 2016*, pages 29–52, Berlin, 2018. Springer.
26. 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, 45 pages, 2007.
27. P. L’Ecuyer, M. Godin, A. Jemel, P. Marion, and D. Munger. LatNet Builder: A general software tool for constructing highly uniform point sets. <https://github.com/umontreal-simul/latnetbuilder>, 2019.
28. P. L’Ecuyer, C. Lécot, and A. L’Archevêque-Gaudet. On array-RQMC for Markov chains: Mapping alternatives and convergence rates. In P. L’Ecuyer and A. B. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 485–500, Berlin, 2009. Springer-Verlag.

29. 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.
30. P. L'Ecuyer and C. Lemieux. Variance reduction via lattice rules. *Management Science*, 46(9):1214–1235, 2000.
31. P. L'Ecuyer and C. Lemieux. Recent advances in randomized quasi-Monte Carlo methods. In M. Dror, P. L'Ecuyer, and F. Szidarovszky, editors, *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, pages 419–474. Kluwer Academic, Boston, 2002.
32. P. L'Ecuyer and D. Munger. On figures of merit for randomly-shifted lattice rules. In H. Woźniakowski and L. Plaskota, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2010*, pages 133–159, Berlin, 2012. Springer-Verlag.
33. P. L'Ecuyer and D. Munger. Algorithm 958: Lattice builder: A general software tool for constructing rank-1 lattice rules. *ACM Transactions on Mathematical Software*, 42(2):Article 15, 2016.
34. P. L'Ecuyer, D. Munger, C. Lécot, and B. Tuffin. Sorting methods and convergence rates for Array-RQMC: Some empirical comparisons. *Mathematics and Computers in Simulation*, 143:191–201, 2018.
35. P. L'Ecuyer, D. Munger, and B. Tuffin. On the distribution of integration error by randomly-shifted lattice rules. *Electronic Journal of Statistics*, 4:950–993, 2010.
36. P. L'Ecuyer and C. Sanvido. Coupling from the past with randomized quasi-Monte Carlo. *Mathematics and Computers in Simulation*, 81(3):476–489, 2010.
37. C. Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer-Verlag, 2009.
38. C. Lemieux, M. Cieslak, and K. Luttmmer. *RandQMC User's Guide: A Package for Randomized Quasi-Monte Carlo Methods in C*, 2004. Software user's guide, available at <http://www.math.uwaterloo.ca/~clemieux/randqmc.html>.
39. C. Lemieux and P. L'Ecuyer. Randomized polynomial lattice rules for multivariate integration and simulation. *SIAM Journal on Scientific Computing*, 24(5):1768–1789, 2003.
40. W.-L. Loh. On the asymptotic distribution of scramble nets quadratures. *Annals of Statistics*, 31:1282–1324, 2003.
41. P. Marion, M. Godin, and P. L'Ecuyer. An algorithm to compute the t -value of a digital net and of its projections. *Journal of Computational and Applied Mathematics*, 371, 2020.
42. J. Matousek. On the L_2 -discrepancy for anchored boxes. *J. of Complexity*, 14:527–556, 1998.
43. D. Munger, P. L'Ecuyer, F. Bastin, C. Cirillo, and B. Tuffin. Estimation of mixed logit likelihood function by randomized quasi-Monte Carlo. *Transportation Research Part B: Methodological*, 4(2):305–320, 2012.
44. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *SIAM CBMS-NSF Reg. Conf. Series in Applied Mathematics*. SIAM, 1992.
45. D. Nuyens. The construction of good lattice rules and polynomial lattice rules. In Peter Kritzer, Harald Niederreiter, Friedrich Pillichshammer, and Arne Winterhof, editors, *Uniform Distribution and Quasi-Monte Carlo Methods: Discrepancy, Integration and Applications*, pages 223–255. De Gruyter, 2014.
46. D. Nuyens. The magic point shop, 2020. <https://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/>.
47. A. B. Owen. Monte Carlo variance of scrambled equidistribution quadrature. *SIAM Journal on Numerical Analysis*, 34(5):1884–1910, 1997.
48. A. B. Owen. Scrambled net variance for integrals of smooth functions. *Annals of Statistics*, 25(4):1541–1562, 1997.
49. A. B. Owen. Monte Carlo, quasi-Monte Carlo, and randomized quasi-Monte Carlo. In Harald Niederreiter and Jerome Spanier, editors, *Monte-Carlo and Quasi-Monte Carlo Methods 1998*, pages 86–97, Berlin, Heidelberg, 2000. Springer-Verlag.
50. A. B. Owen. Variance with alternative scramblings of digital nets. *ACM Transactions on Modeling and Computer Simulation*, 13(4):363–378, 2003.
51. I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford, 1994.
52. I. H. Sloan and H. Woźniakowski. When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals. *Journal of Complexity*, 14:1–33, 1998.
53. 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(4):86–112, 1967.
54. B. Tuffin and P. L'Ecuyer, editors. *Monte Carlo and Quasi-Monte Carlo Methods 2018*. Springer-Verlag, Berlin, 2020.
55. C. Wächter and A. Keller. Quasi-Monte Carlo light transport simulation by efficient ray tracing. US Patent 7,952,583, 2011.
56. X. Wang and I. H. Sloan. Brownian bridge and principal component analysis: Toward removing the curse of dimensionality. *IMA Journal of Numerical Analysis*, 27:631–654, 2007.