

Introduction to (randomized) quasi-Monte Carlo

Pierre L'Ecuyer



MCQMC Conference, Stanford University, August 2016

Program

- ▶ Monte Carlo, Quasi-Monte Carlo, Randomized quasi-Monte Carlo
- ▶ QMC point sets and randomizations
- ▶ Error and variance bounds, convergence rates
- ▶ Transforming the integrand to make it more QMC friendly (smoother, smaller effective dimension, etc.).
- ▶ Numerical illustrations
- ▶ RQMC for Markov chains

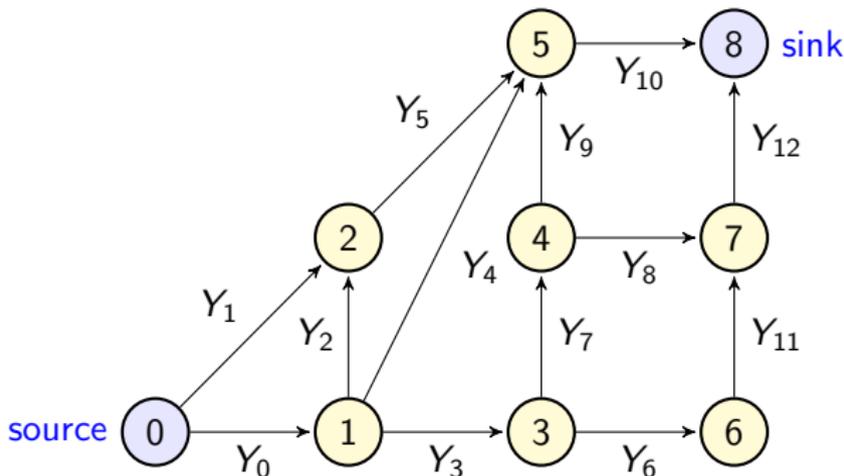
Focus on ideas, insight, and examples.

Example: A stochastic activity network

Gives precedence relations between activities. Activity k has random duration Y_k (also length of arc k) with known cumulative distribution function (cdf) $F_k(y) := \mathbb{P}[Y_k \leq y]$.

Project duration $T =$ (random) length of longest path from source to sink.

May want to estimate $\mathbb{E}[T]$, $\mathbb{P}[T > x]$, a quantile, density of T , etc.



Monte Carlo (simulation)

Algorithm: Monte Carlo to estimate $\mathbb{E}[T]$

for $i = 0, \dots, n - 1$ **do**

for $k = 0, \dots, 12$ **do**

 Generate $U_k \sim U(0, 1)$ and let $Y_k = F_k^{-1}(U_k)$

 Compute $X_i = T = h(Y_0, \dots, Y_{12}) = f(U_0, \dots, U_{12})$

 Estimate $\mathbb{E}[T] = \int_{(0,1)^s} f(\mathbf{u})d\mathbf{u}$ by $\bar{X}_n = \frac{1}{n} \sum_{i=0}^{n-1} X_i$, etc.

Can also compute confidence interval on $\mathbb{E}[T]$, a histogram to estimate the **distribution** of T , etc.

Numerical illustration from Elmaghraby (1977):

$Y_k \sim N(\mu_k, \sigma_k^2)$ for $k = 0, 1, 3, 10, 11$, and $V_k \sim \text{Expon}(1/\mu_k)$ otherwise.

μ_0, \dots, μ_{12} : 13.0, 5.5, 7.0, 5.2, 16.5, 14.7, 10.3, 6.0, 4.0, 20.0, 3.2, 3.2, 16.5.

We may pay a penalty if $T > 90$, for example.

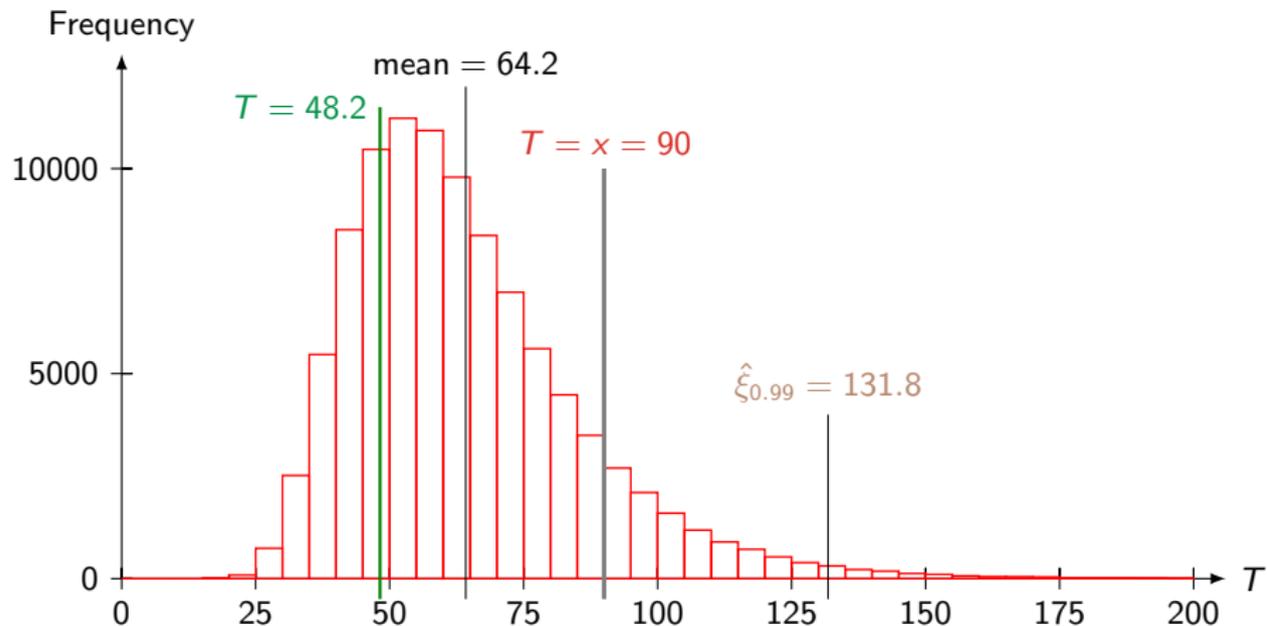
Naive idea: replace each Y_k by its expectation. Gives $T = 48.2$.

Naive idea: replace each Y_k by its expectation. Gives $T = 48.2$.

Results of an experiment with $n = 100\,000$.

Histogram of values of T gives more information than confidence interval on $\mathbb{E}[T]$ or $\mathbb{P}[T > x]$.

Values from 14.4 to 268.6; 11.57% exceed $x = 90$.



Sample path of hurricane Sandy for the next 5 days

As Forecasts Go, You Can Bet on Monte Carlo - WSJ

THE WALL STREET JOURNAL.

This copy is for your personal, non-commercial use only. To order presentation-ready copies for distribution to your colleagues, clients or customers visit <http://www.djreprints.com>.

<http://www.wsj.com/articles/as-forecasts-go-you-can-bet-on-monte-carlo-1470994203>

U.S. | THE NUMBERS

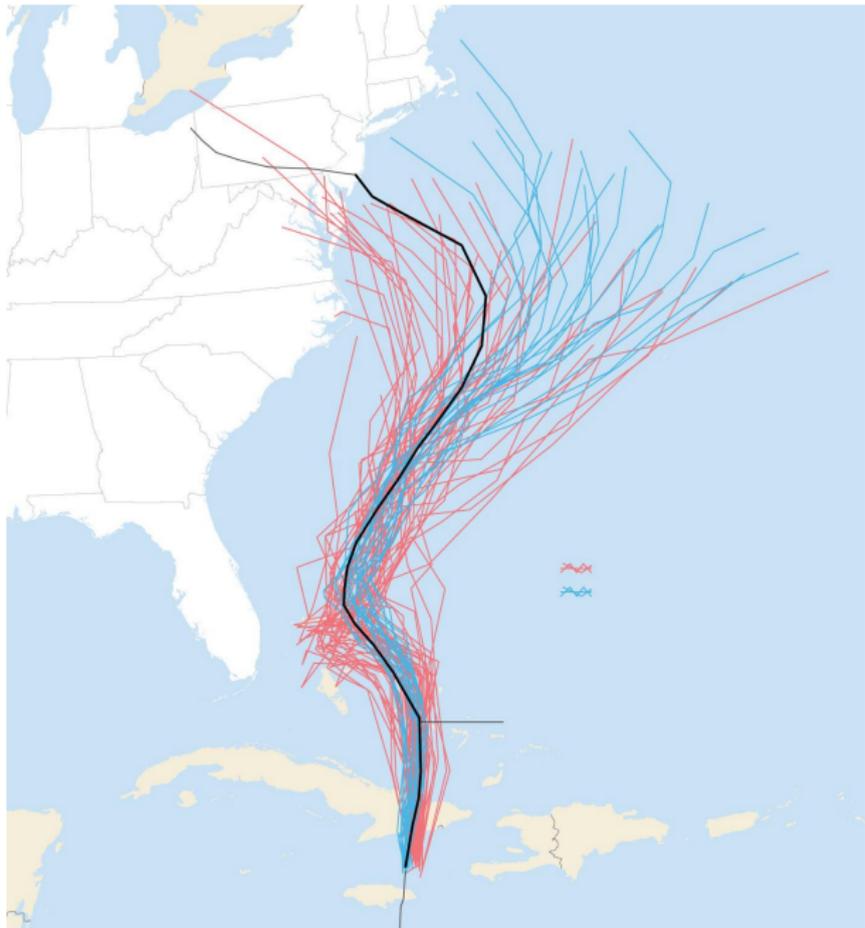
As Forecasts Go, You Can Bet on Monte Carlo

From Super Bowls to hurricanes, this simulation method helps predict them all



Monte Carlo simulations helped give emergency workers advance warning that Hurricane Sandy would make landfall in New Jersey and New York. Here are Oct. 31, 2010 file photo of homes in Ocean Beach, N.J. destroyed by the storm.

Sample path of hurricane Sandy for the next 5 days



Monte Carlo to estimate an expectation

Want to estimate $\mu = \mathbb{E}[X]$ where $X = f(\mathbf{U}) = f(U_0, \dots, U_{s-1})$, and the U_j are i.i.d. $U(0, 1)$ “random numbers.” We have

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

Monte Carlo estimator:

$$\bar{X}_n = \frac{1}{n} \sum_{i=0}^{n-1} X_i$$

where $X_i = f(\mathbf{U}_i)$ and $\mathbf{U}_0, \dots, \mathbf{U}_{n-1}$ i.i.d. uniform over $[0, 1]^s$.

We have $\mathbb{E}[\bar{X}_n] = \mu$ and $\text{Var}[\bar{X}_n] = \sigma^2/n = \text{Var}[X]/n$.

Convergence

Theorem. Suppose $\sigma^2 < \infty$. When $n \rightarrow \infty$:

(i) **Strong law of large numbers:** $\lim_{n \rightarrow \infty} \hat{\mu}_n = \mu$ with probability 1.

Convergence

Theorem. Suppose $\sigma^2 < \infty$. When $n \rightarrow \infty$:

- (i) Strong law of large numbers: $\lim_{n \rightarrow \infty} \hat{\mu}_n = \mu$ with probability 1.
- (ii) Central limit theorem (CLT):

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{S_n} \Rightarrow N(0, 1),$$

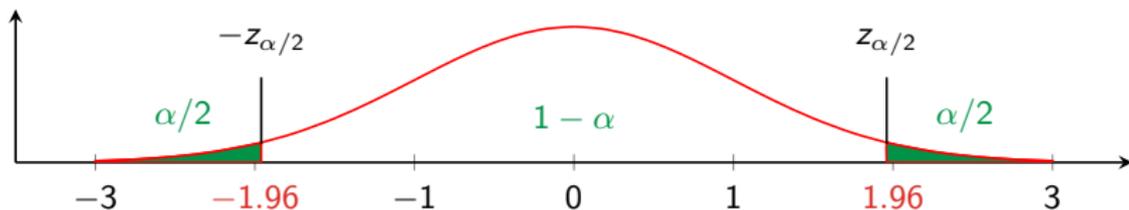
where

$$S_n^2 = \frac{1}{n-1} \sum_{i=0}^{n-1} (X_i - \bar{X}_n)^2.$$

Confidence interval at level α (we want $\Phi(x) = 1 - \alpha/2$):

$$(\hat{\mu}_n \pm z_{\alpha/2} S_n / \sqrt{n}), \text{ where } z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2).$$

Example: $z_{\alpha/2} \approx 1.96$ for $\alpha = 0.05$.



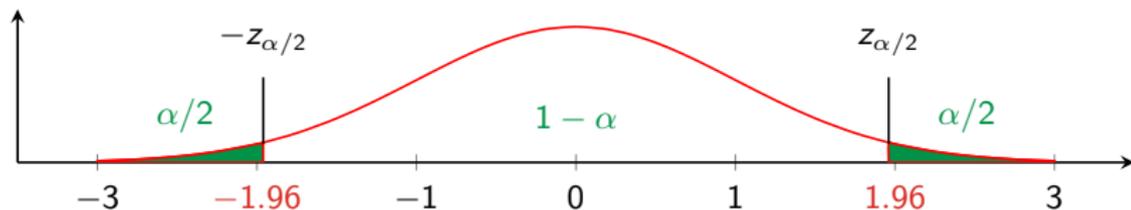
The width of the confidence interval is asymptotically proportional to σ/\sqrt{n} , so it converges as $O(n^{-1/2})$. **Relative error:** $\sigma/(\mu\sqrt{n})$.

For one more decimal digit of accuracy, we must multiply n by 100.

Confidence interval at level α (we want $\Phi(x) = 1 - \alpha/2$):

$$(\hat{\mu}_n \pm z_{\alpha/2} S_n / \sqrt{n}), \text{ where } z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2).$$

Example: $z_{\alpha/2} \approx 1.96$ for $\alpha = 0.05$.



The width of the confidence interval is asymptotically proportional to σ/\sqrt{n} , so it converges as $O(n^{-1/2})$. **Relative error:** $\sigma/(\mu\sqrt{n})$.

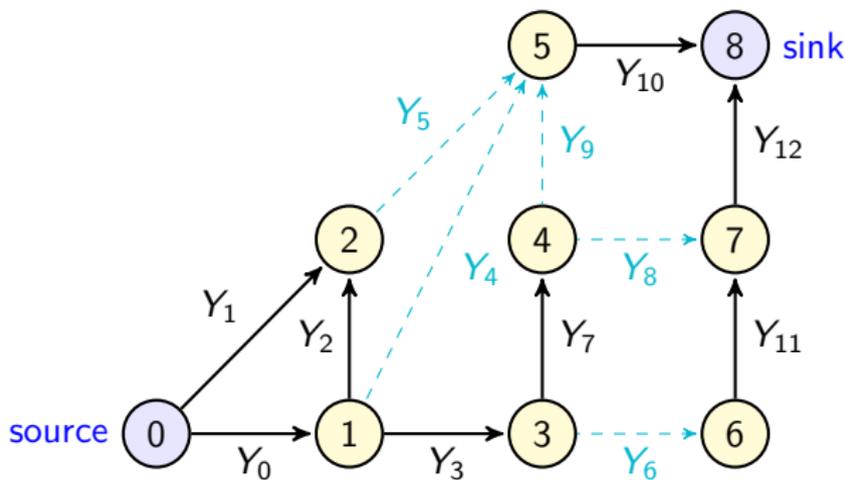
For one more decimal digit of accuracy, we must multiply n by 100.

Warning: If the X_i have an asymmetric law, these confidence intervals can have very bad coverage (convergence to normal can be very slow).

Alternative estimator of $\mathbb{P}[T > x] = \mathbb{E}[\mathbb{I}(T > x)]$ for SAN.

Naive estimator: Generate T and compute $X = \mathbb{I}[T > x]$.

Repeat n times and average.



Conditional Monte Carlo estimator of $\mathbb{P}[T > x]$. Generate the Y_j 's ¹² only for the 8 arcs that **do not** belong to the cut $\mathcal{L} = \{4, 5, 6, 8, 9\}$, and replace $\mathbb{I}[T > x]$ by its **conditional expectation** given those Y_j 's,

$$X_e = \mathbb{P}[T > x \mid \{Y_j, j \notin \mathcal{L}\}].$$

This makes the integrand **continuous in the U_j 's**.

Conditional Monte Carlo estimator of $\mathbb{P}[T > x]$. Generate the Y_j 's ¹² only for the 8 arcs that **do not** belong to the cut $\mathcal{L} = \{4, 5, 6, 8, 9\}$, and replace $\mathbb{I}[T > x]$ by its **conditional expectation** given those Y_j 's,

$$X_e = \mathbb{P}[T > x \mid \{Y_j, j \notin \mathcal{L}\}].$$

This makes the integrand **continuous in the U_j 's**.

To compute X_e : for each $l \in \mathcal{L}$, say from a_l to b_l , compute the length α_l of the longest path from 1 to a_l , and the length β_l of the longest path from b_l to the destination.

The longest path that passes through link l does not exceed x iff $\alpha_l + Y_l + \beta_l \leq x$, which occurs with probability $\mathbb{P}[Y_l \leq x - \alpha_l - \beta_l] = F_l[x - \alpha_l - \beta_l]$.

Conditional Monte Carlo estimator of $\mathbb{P}[T > x]$. Generate the Y_j 's¹² only for the 8 arcs that **do not** belong to the cut $\mathcal{L} = \{4, 5, 6, 8, 9\}$, and replace $\mathbb{I}[T > x]$ by its **conditional expectation** given those Y_j 's,

$$X_e = \mathbb{P}[T > x \mid \{Y_j, j \notin \mathcal{L}\}].$$

This makes the integrand **continuous in the U_j 's**.

To compute X_e : for each $l \in \mathcal{L}$, say from a_l to b_l , compute the length α_l of the longest path from 1 to a_l , and the length β_l of the longest path from b_l to the destination.

The longest path that passes through link l does not exceed x iff $\alpha_l + Y_l + \beta_l \leq x$, which occurs with probability $\mathbb{P}[Y_l \leq x - \alpha_l - \beta_l] = F_l[x - \alpha_l - \beta_l]$.

Since the Y_l are independent, we obtain

$$X_e = 1 - \prod_{l \in \mathcal{L}} F_l[x - \alpha_l - \beta_l].$$

Can be faster to compute than X , and always has less variance.

Example: Pricing a financial derivative.

Market price of some asset (e.g., one share of a stock) evolves in time as stochastic process $\{S(t), t \geq 0\}$ with (supposedly) known probability law (estimated from data).

A financial contract gives owner net payoff $g(S(t_1), \dots, S(t_d))$ at time $T = t_d$, where $g : \mathbb{R}^d \rightarrow \mathbb{R}$, and $0 \leq t_1 < \dots < t_d$ are fixed observation times.

Under a no-arbitrage assumption, present value (fair price) of contract at time 0, when $S(0) = s_0$, can be written as

$$v(s_0, T) = \mathbb{E}^* [e^{-rT} g(S(t_1), \dots, S(t_d))],$$

where \mathbb{E}^* is under a risk-neutral measure and e^{-rT} is the discount factor.

This expectation can be written as an integral over $[0, 1]^s$ and estimated by the average of n i.i.d. replicates of $X = e^{-rT} g(S(t_1), \dots, S(t_d))$.

A simple model for S : **geometric Brownian motion (GBM)**:

$$S(t) = s_0 e^{(r - \sigma^2/2)t + \sigma B(t)}$$

where r is the interest rate, σ is the **volatility**, and $B(\cdot)$ is a **standard Brownian motion**: for any $t_2 > t_1 \geq 0$, $B(t_2) - B(t_1) \sim N(0, t_2 - t_1)$, and the increments over disjoint intervals are independent.

A simple model for S : **geometric Brownian motion (GBM)**:

$$S(t) = s_0 e^{(r - \sigma^2/2)t + \sigma B(t)}$$

where r is the interest rate, σ is the **volatility**, and $B(\cdot)$ is a **standard Brownian motion**: for any $t_2 > t_1 \geq 0$, $B(t_2) - B(t_1) \sim N(0, t_2 - t_1)$, and the increments over disjoint intervals are independent.

Algorithm: Option pricing under GBM model

for $i = 0, \dots, n - 1$ **do**

Let $t_0 = 0$ and $B(t_0) = 0$

for $j = 1, \dots, d$ **do**

Generate $U_j \sim U(0, 1)$ and let $Z_j = \Phi^{-1}(U_j)$

Let $B(t_j) = B(t_{j-1}) + \sqrt{t_j - t_{j-1}} Z_j$

Let $S(t_j) = s_0 \exp[(r - \sigma^2/2)t_j + \sigma B(t_j)]$

Compute $X_i = e^{-rT} g(S(t_1), \dots, S(t_d))$

Return $\bar{X}_n = \frac{1}{n} \sum_{i=0}^{n-1} X_i$, estimator of $v(s_0, T)$.

Example of contract: Discretely-monitored **Asian call option**:

$$g(S(t_1), \dots, S(t_d)) = \max \left(0, \frac{1}{d} \sum_{j=1}^d S(t_j) - K \right).$$

Option price written as an integral over the unit hypercube:

Let $Z_j = \Phi^{-1}(U_j)$ where the U_j are i.i.d. $U(0, 1)$. Here we have $s = d$ and

$$\begin{aligned} v(s_0, T) &= \int_{[0,1]^s} e^{-rT} \max \left(0, \frac{1}{s} \sum_{i=1}^s s_0 \cdot \right. \\ &\quad \left. \exp \left[(r - \sigma^2/2)t_i + \sigma \sum_{j=1}^i \sqrt{t_j - t_{j-1}} \Phi^{-1}(u_j) \right] - K \right) du_1 \dots du_s \\ &= \int_{[0,1]^s} f(u_1, \dots, u_s) du_1 \dots du_s. \end{aligned}$$

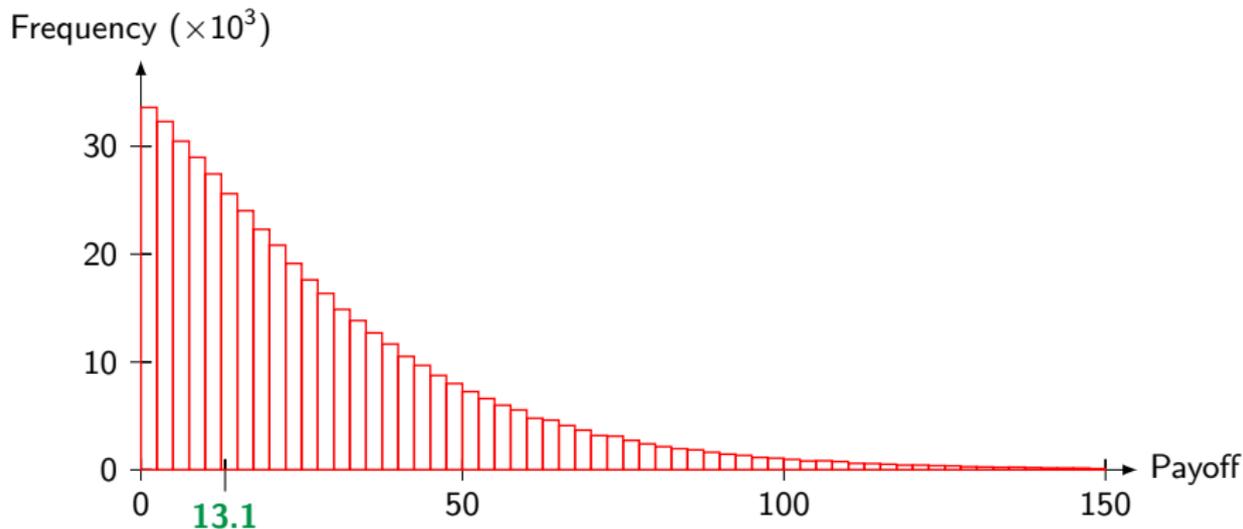
Numerical illustration: Bermudean Asian option with $d = 12$, $T = 1$ (one year), $t_j = j/12$ for $j = 0, \dots, 12$, $K = 100$, $s_0 = 100$, $r = 0.05$, $\sigma = 0.5$.

We performed $n = 10^6$ independent simulation runs.

In 53.47% of cases, the payoff is 0.

Mean: 13.1. Max = 390.8

Histogram of the 46.53% **positive values:**



Reducing the variance by changing f

If we replace the arithmetic average by a **geometric average** in the payoff, we obtain

$$C = e^{-rT} \max \left(0, \prod_{j=1}^d (S(t_j))^{1/d} - K \right),$$

whose expectation $\nu = \mathbb{E}[C]$ has a closed-form formula.

When estimating the mean $\mathbb{E}[X] = v(s_0, T)$, we can then use C as a **control variate** (CV): Replace the estimator X by the “corrected” version

$$X_c = X - \beta(C - \nu)$$

for some well-chosen constant β . Optimal β is $\beta^* = \text{Cov}[C, X] / \text{Var}[C]$.

Using a CV makes the integrand f smoother. Can provide a huge variance reduction, e.g., by a factor of over **a million** in some examples.

Quasi-Monte Carlo (QMC)

Replace the independent random points \mathbf{U}_i by a set of **deterministic** points $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\}$ that cover $[0, 1]^s$ **more evenly**.

Estimate

$$\mu = \int_{[0,1]^s} f(\mathbf{u})d\mathbf{u} \quad \text{by} \quad \bar{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{u}_i).$$

Integration error $E_n = \bar{\mu} - \mu$.

P_n is called a **highly-uniform point set** or **low-discrepancy point set** if some measure of **discrepancy** between the empirical distribution of P_n and the uniform distribution converges to 0 faster than $O(n^{-1/2})$ (the typical rate for independent random points).

Quasi-Monte Carlo (QMC)

Replace the independent random points \mathbf{U}_i by a set of **deterministic** points $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\}$ that cover $[0, 1]^s$ **more evenly**.

Estimate

$$\mu = \int_{[0,1]^s} f(\mathbf{u})d\mathbf{u} \quad \text{by} \quad \bar{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{u}_i).$$

Integration error $E_n = \bar{\mu} - \mu$.

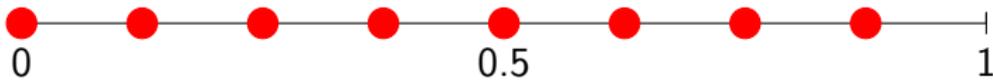
P_n is called a **highly-uniform point set** or **low-discrepancy point set** if some measure of **discrepancy** between the empirical distribution of P_n and the uniform distribution converges to 0 faster than $O(n^{-1/2})$ (the typical rate for independent random points).

Main construction methods: **lattice rules** and **digital nets**
(Korobov, Hammersley, Halton, Sobol', Faure, Niederreiter, etc.)

Simple case: one dimension ($s = 1$)

Obvious solutions:

$P_n = \mathbb{Z}_n/n = \{0, 1/n, \dots, (n-1)/n\}$ (left Riemann sum):



which gives $\bar{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} f(i/n)$, and $E_n = \mathcal{O}(n^{-1})$ if f' is bounded,

Simple case: one dimension ($s = 1$)

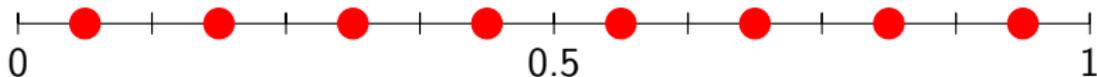
Obvious solutions:

$P_n = \mathbb{Z}_n/n = \{0, 1/n, \dots, (n-1)/n\}$ (left Riemann sum):



which gives $\bar{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} f(i/n)$, and $E_n = \mathcal{O}(n^{-1})$ if f' is bounded,

or $P'_n = \{1/(2n), 3/(2n), \dots, (2n-1)/(2n)\}$ (midpoint rule):



for which $E_n = \mathcal{O}(n^{-2})$ if f'' is bounded.

If we allow different **weights** on the $f(\mathbf{u}_i)$, we have the **trapezoidal rule**:



$$\frac{1}{n} \left[\frac{f(0) + f(1)}{2} + \sum_{i=1}^{n-1} f(i/n) \right],$$

for which $|E_n| = O(n^{-2})$ if f'' is bounded,

If we allow different **weights** on the $f(\mathbf{u}_i)$, we have the **trapezoidal rule**:



$$\frac{1}{n} \left[\frac{f(0) + f(1)}{2} + \sum_{i=1}^{n-1} f(i/n) \right],$$

for which $|E_n| = O(n^{-2})$ if f'' is bounded, or the **Simpson rule**,

$$\frac{f(0) + 4f(1/n) + 2f(2/n) + \cdots + 2f((n-2)/n) + 4f((n-1)/n) + f(1)}{3n},$$

which gives $|E_n| = O(n^{-4})$ if $f^{(4)}$ is bounded, etc.

If we allow different **weights** on the $f(\mathbf{u}_i)$, we have the **trapezoidal rule**:



$$\frac{1}{n} \left[\frac{f(0) + f(1)}{2} + \sum_{i=1}^{n-1} f(i/n) \right],$$

for which $|E_n| = O(n^{-2})$ if f'' is bounded, or the **Simpson rule**,

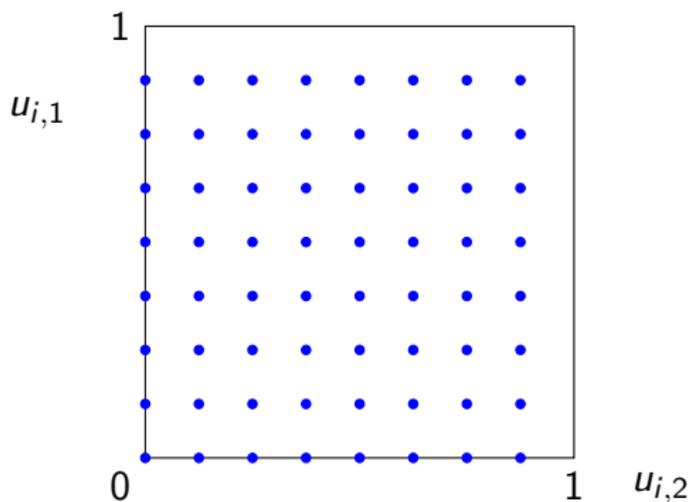
$$\frac{f(0) + 4f(1/n) + 2f(2/n) + \cdots + 2f((n-2)/n) + 4f((n-1)/n) + f(1)}{3n},$$

which gives $|E_n| = O(n^{-4})$ if $f^{(4)}$ is bounded, etc.

Here, for QMC and RQMC, we restrict ourselves to **equal weight** rules. For the RQMC points that we will examine, one can prove that equal weights are optimal.

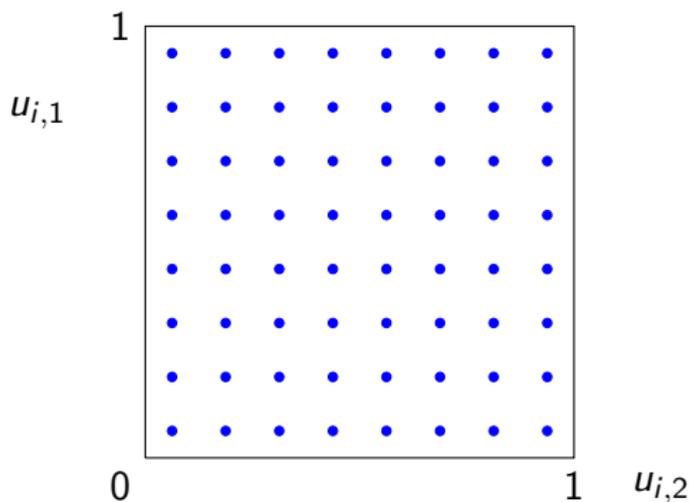
Simplistic solution for $s > 1$: rectangular grid

$P_n = \{(i_1/d, \dots, i_s/d) \text{ such that } 0 \leq i_j < d \ \forall j\}$ where $n = d^s$.



Simplistic solution for $s > 1$: rectangular grid

$$P_n = \{(i_1/d, \dots, i_s/d) \text{ such that } 0 \leq i_j < d \ \forall j\} \text{ where } n = d^s.$$



Midpoint rule in s dimensions.

Quickly becomes impractical when s increases.

Moreover, each one-dimensional projection has only d distinct points, each two-dimensional projections has only d^2 distinct points, etc.

Lattice rules (Korobov, Sloan, etc.)

Integration lattice:

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^s z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \dots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over \mathbb{R} and where L_s contains \mathbb{Z}^s . **Lattice rule:** Take $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\} = L_s \cap [0, 1)^s$.

Lattice rules (Korobov, Sloan, etc.)

Integration lattice:

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^s z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \dots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over \mathbb{R} and where L_s contains \mathbb{Z}^s . **Lattice rule:** Take $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\} = L_s \cap [0, 1)^s$.

Lattice rule of **rank 1**: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \dots, n-1$, where $n\mathbf{v}_1 = \mathbf{a} = (a_1, \dots, a_s) \in \{0, 1, \dots, n-1\}^s$.

Korobov rule: $\mathbf{a} = (1, a, a^2 \bmod n, \dots)$.

Lattice rules (Korobov, Sloan, etc.)

Integration lattice:

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^s z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \dots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over \mathbb{R} and where L_s contains \mathbb{Z}^s . **Lattice rule:** Take $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\} = L_s \cap [0, 1)^s$.

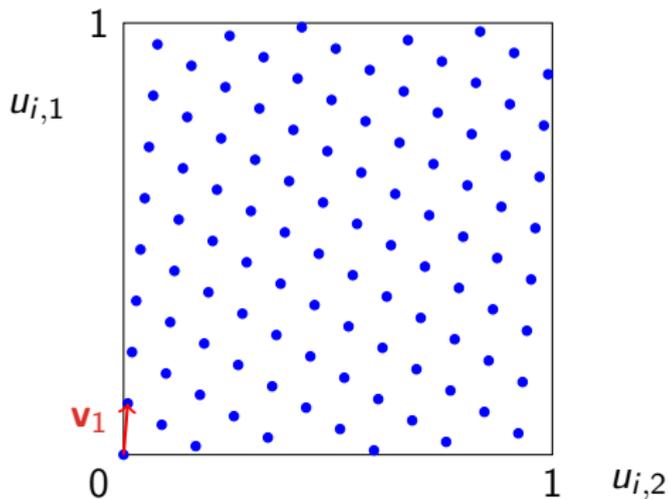
Lattice rule of **rank 1**: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \dots, n-1$, where $n\mathbf{v}_1 = \mathbf{a} = (a_1, \dots, a_s) \in \{0, 1, \dots, n-1\}^s$.

Korobov rule: $\mathbf{a} = (1, a, a^2 \bmod n, \dots)$.

For any $u \subset \{1, \dots, s\}$, the **projection** $L_s(u)$ of L_s is also a lattice.

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/n$

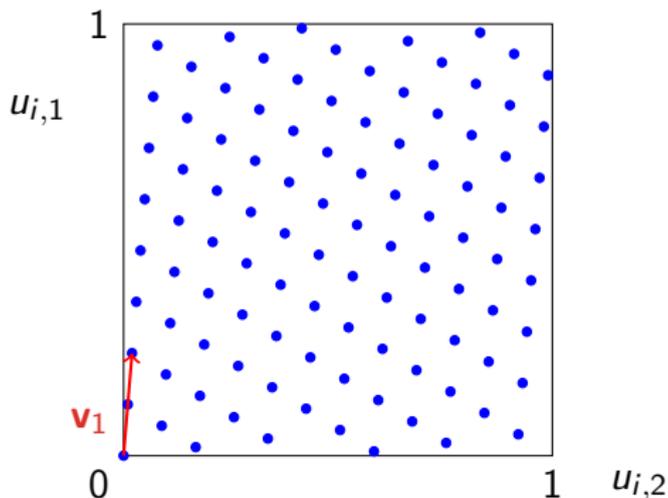
$$\begin{aligned} P_n &= \{\mathbf{u}_i = i\mathbf{v}_1 \bmod 1 : i = 0, \dots, n-1\} \\ &= \{(0, 0), (1/101, 12/101), (2/101, 43/101), \dots\}. \end{aligned}$$



Here, each one-dimensional projection is $\{0, 1/n, \dots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/n$

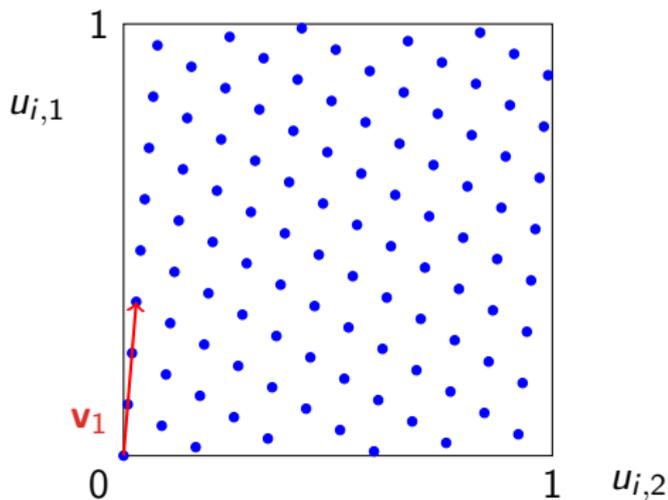
$$\begin{aligned} P_n &= \{\mathbf{u}_i = i\mathbf{v}_1 \bmod 1 : i = 0, \dots, n-1\} \\ &= \{(0, 0), (1/101, 12/101), (2/101, 43/101), \dots\}. \end{aligned}$$



Here, each one-dimensional projection is $\{0, 1/n, \dots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/n$

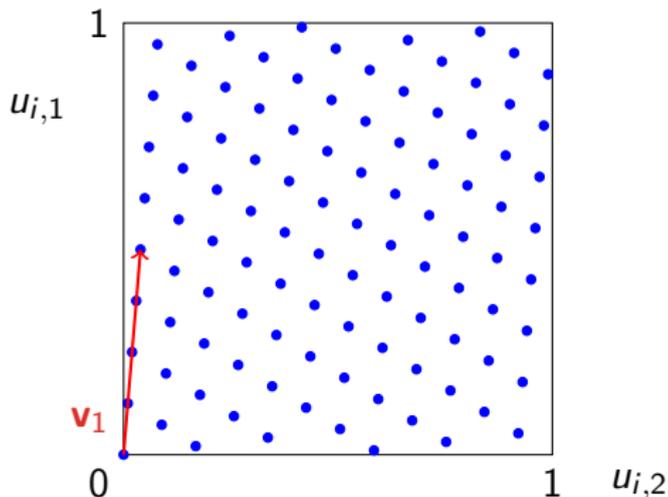
$$\begin{aligned}
 P_n &= \{\mathbf{u}_i = i\mathbf{v}_1 \bmod 1 : i = 0, \dots, n-1\} \\
 &= \{(0, 0), (1/101, 12/101), (2/101, 43/101), \dots\}.
 \end{aligned}$$



Here, each one-dimensional projection is $\{0, 1/n, \dots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/n$

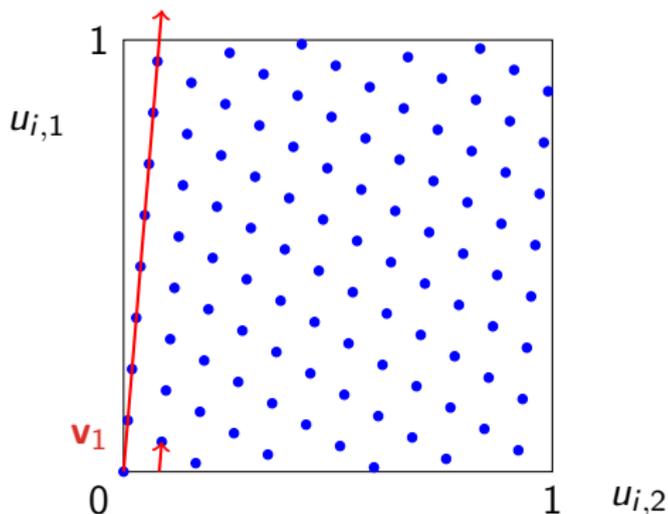
$$\begin{aligned}
 P_n &= \{\mathbf{u}_i = i\mathbf{v}_1 \bmod 1 : i = 0, \dots, n-1\} \\
 &= \{(0, 0), (1/101, 12/101), (2/101, 43/101), \dots\}.
 \end{aligned}$$



Here, each one-dimensional projection is $\{0, 1/n, \dots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/n$

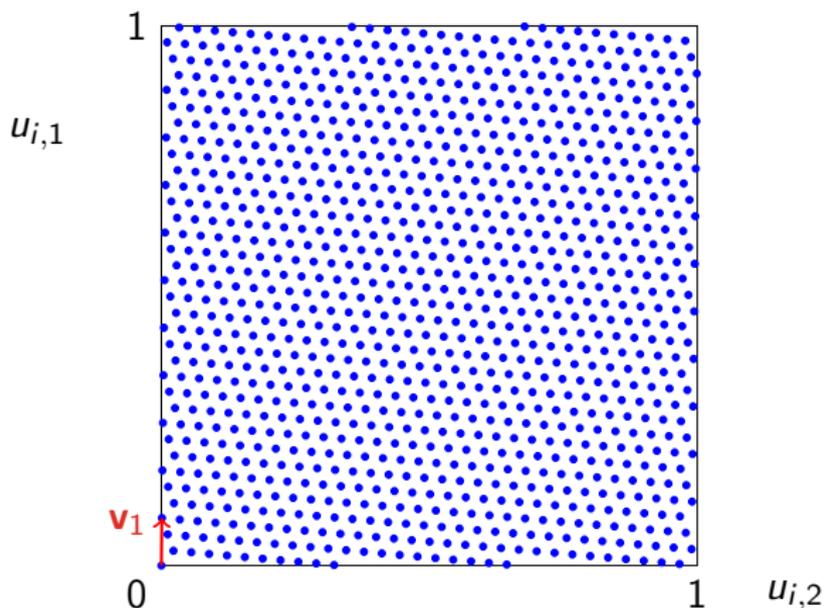
$$\begin{aligned}
 P_n &= \{\mathbf{u}_i = i\mathbf{v}_1 \bmod 1 : i = 0, \dots, n-1\} \\
 &= \{(0, 0), (1/101, 12/101), (2/101, 43/101), \dots\}.
 \end{aligned}$$



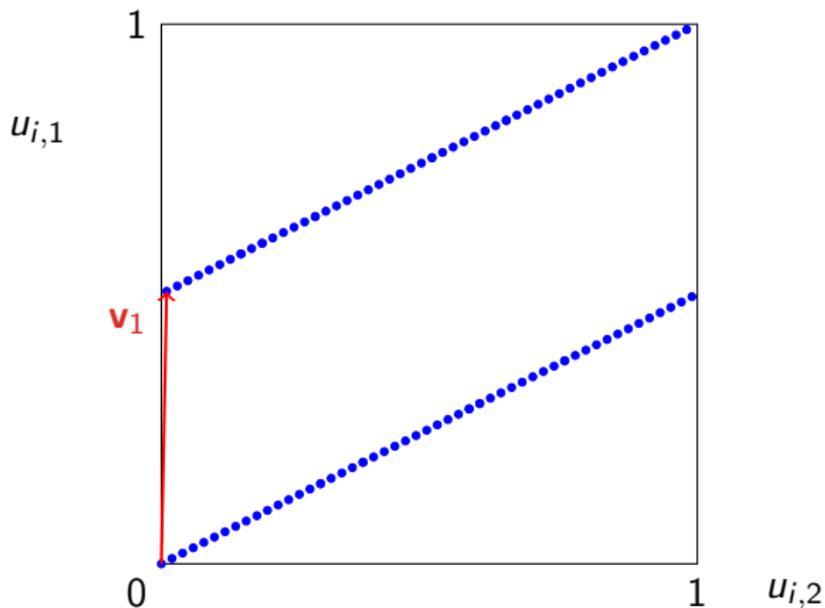
Here, each one-dimensional projection is $\{0, 1/n, \dots, (n-1)/n\}$.

Another example: $s = 2$, $n = 1021$, $\mathbf{v}_1 = (1, 90)/n$

$$\begin{aligned} P_n &= \{\mathbf{u}_i = i\mathbf{v}_1 \bmod 1 : i = 0, \dots, n-1\} \\ &= \{(i/1021, (90i/1021) \bmod 1) : i = 0, \dots, 1020\}. \end{aligned}$$



A bad lattice: $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 51)/n$



Good uniformity in one dimension, but not in two!

Digital net in base b (Niederreiter)

Gives $n = b^k$ points. For $i = 0, \dots, b^k - 1$ and $j = 1, \dots, s$:

$$i = a_{i,0} + a_{i,1}b + \dots + a_{i,k-1}b^{k-1} = a_{i,k-1} \dots a_{i,1}a_{i,0},$$

$$\begin{pmatrix} u_{i,j,1} \\ \vdots \\ u_{i,j,w} \end{pmatrix} = \mathbf{C}_j \begin{pmatrix} a_{i,0} \\ \vdots \\ a_{i,k-1} \end{pmatrix} \pmod{b},$$

$$\mathbf{u}_{i,j} = \sum_{\ell=1}^w u_{i,j,\ell} b^{-\ell}, \quad \mathbf{u}_i = (u_{i,1}, \dots, u_{i,s}),$$

where the **generating matrices** \mathbf{C}_j are $w \times k$ with elements in \mathbb{Z}_b .

In practice, w and k are finite, but there is no limit.

Digital sequence: infinite sequence. Can stop at $n = b^k$ for any k .

Digital net in base b (Niederreiter)

Gives $n = b^k$ points. For $i = 0, \dots, b^k - 1$ and $j = 1, \dots, s$:

$$i = a_{i,0} + a_{i,1}b + \dots + a_{i,k-1}b^{k-1} = a_{i,k-1} \dots a_{i,1}a_{i,0},$$

$$\begin{pmatrix} u_{i,j,1} \\ \vdots \\ u_{i,j,w} \end{pmatrix} = \mathbf{C}_j \begin{pmatrix} a_{i,0} \\ \vdots \\ a_{i,k-1} \end{pmatrix} \pmod{b},$$

$$u_{i,j} = \sum_{\ell=1}^w u_{i,j,\ell} b^{-\ell}, \quad \mathbf{u}_i = (u_{i,1}, \dots, u_{i,s}),$$

where the **generating matrices** \mathbf{C}_j are $w \times k$ with elements in \mathbb{Z}_b .

In practice, w and k are finite, but there is no limit.

Digital sequence: infinite sequence. Can stop at $n = b^k$ for any k .

Can also multiply in some ring R , with bijections between \mathbb{Z}_b and R .

Each one-dim projection truncated to first k digits is

$\mathbb{Z}_n/n = \{0, 1/n, \dots, (n-1)/n\}$. Each \mathbf{C}_j defines a permutation of \mathbb{Z}_n/n .

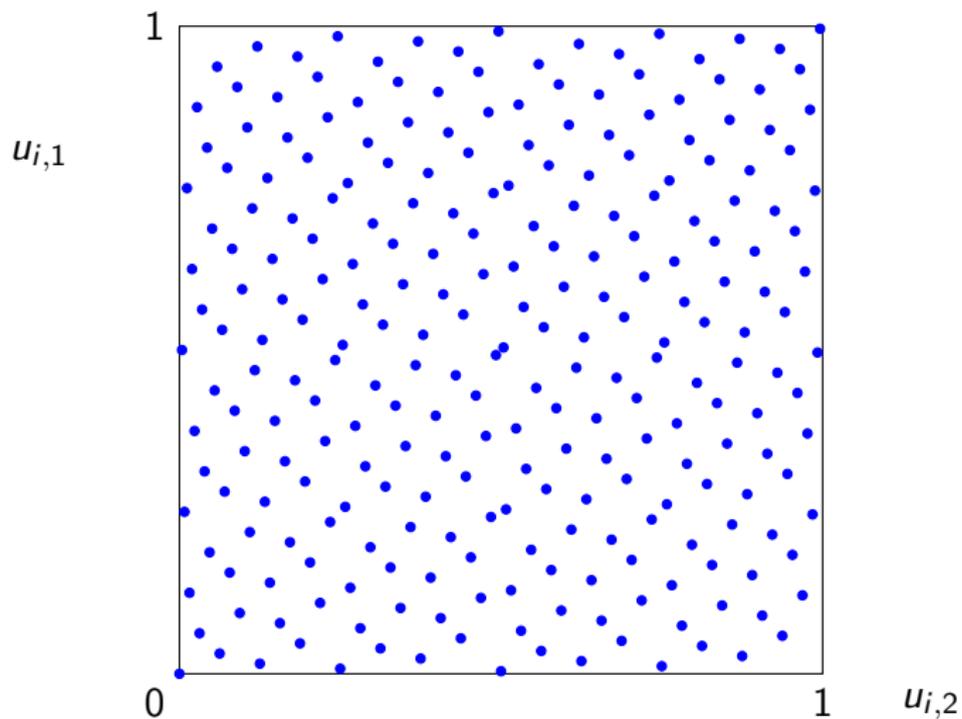
Small example: Hammersley in two dimensions

Let $n = 2^8 = 256$ and $s = 2$. Take the points (in binary):

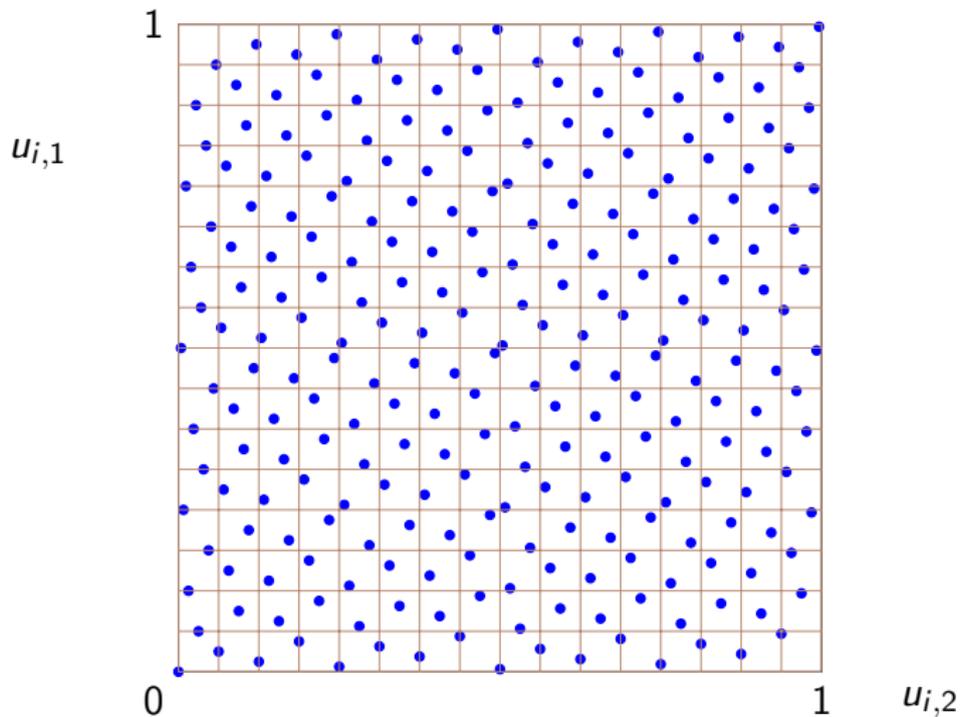
| i | $u_{1,i}$ | $u_{2,i}$ |
|----------|-----------|-----------|
| 0 | .00000000 | .0 |
| 1 | .00000001 | .1 |
| 2 | .00000010 | .01 |
| 3 | .00000011 | .11 |
| 4 | .00000100 | .001 |
| 5 | .00000101 | .101 |
| 6 | .00000110 | .011 |
| \vdots | \vdots | \vdots |
| 254 | .11111110 | .01111111 |
| 255 | .11111111 | .11111111 |

Right side: [van der Corput](#) sequence in base 2.

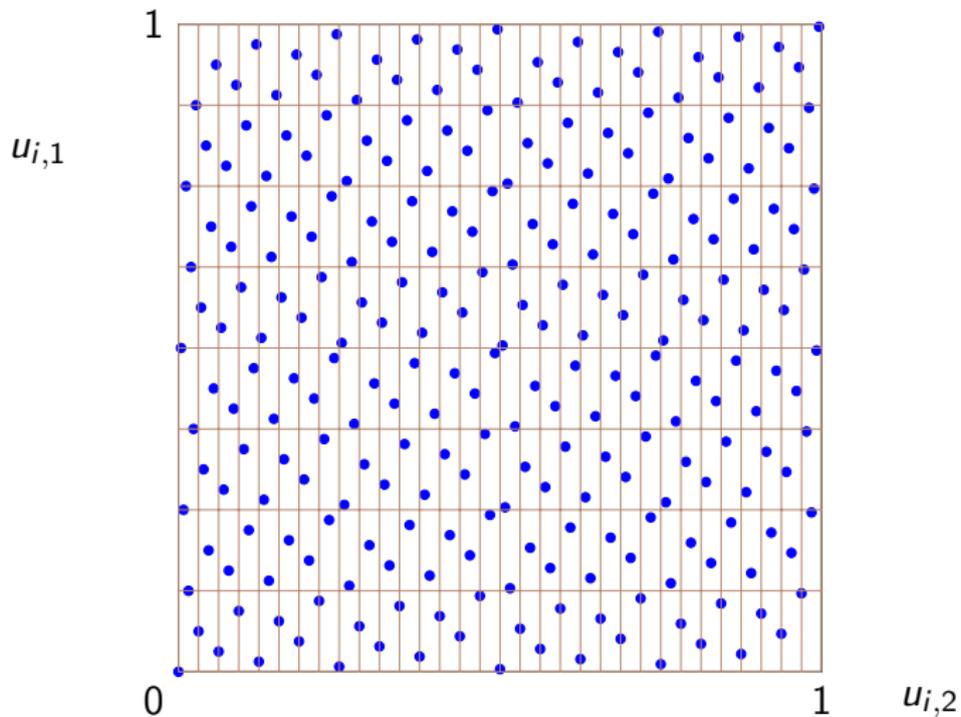
Hammersley point set, $n = 2^8 = 256$, $s = 2$.



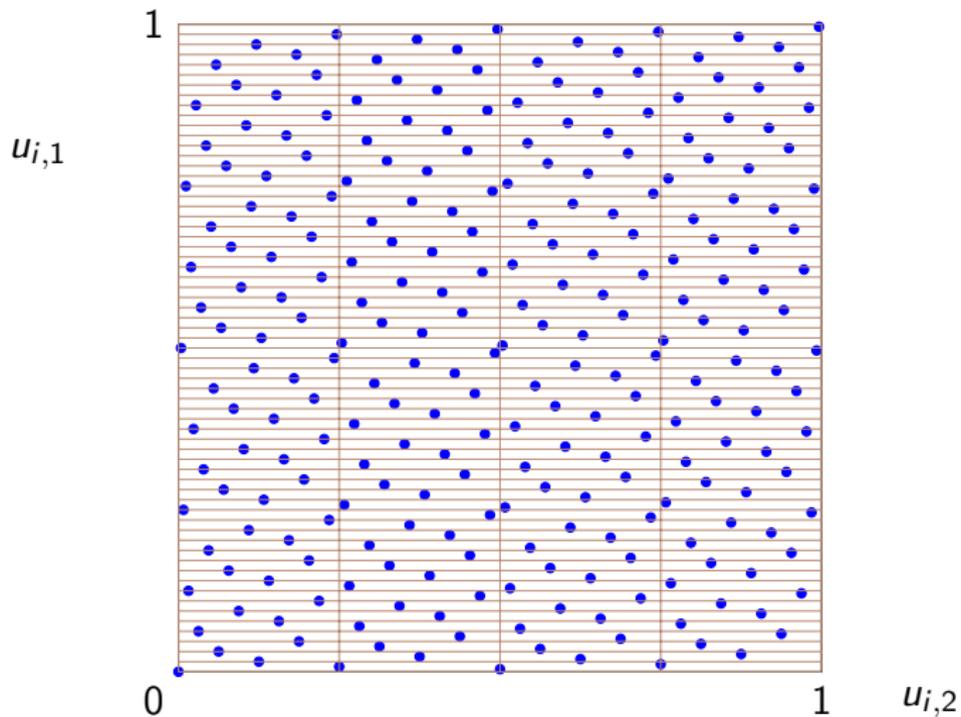
Hammersley point set, $n = 2^8 = 256$, $s = 2$.



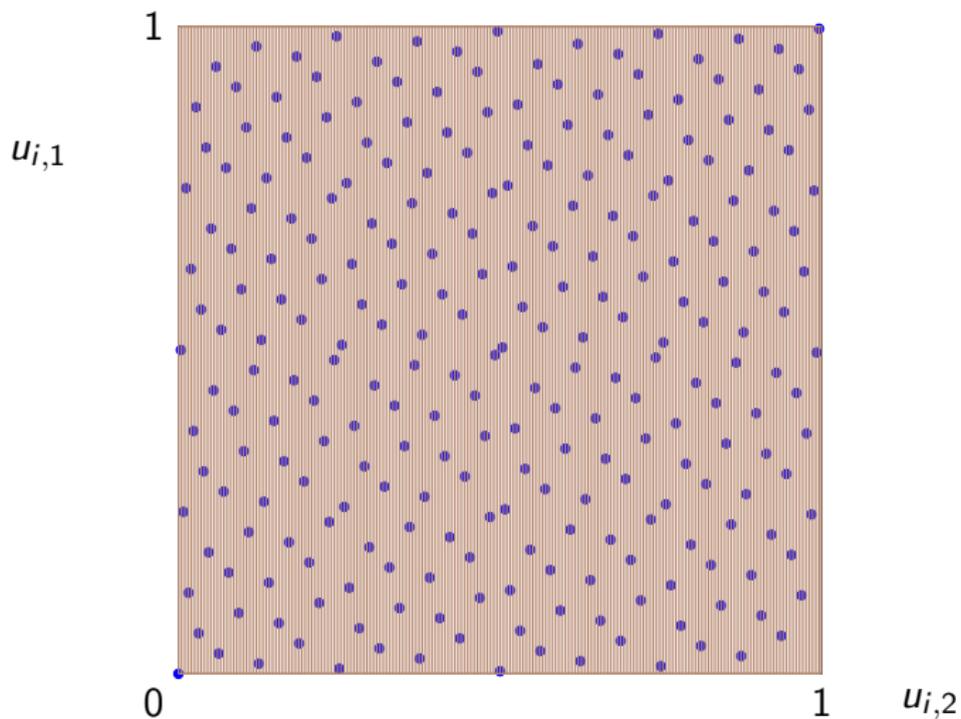
Hammersley point set, $n = 2^8 = 256$, $s = 2$.



Hammersley point set, $n = 2^8 = 256$, $s = 2$.



Hammersley point set, $n = 2^8 = 256$, $s = 2$.



In general, can take $n = 2^k$ points.

If we partition $[0, 1)^2$ in rectangles of sizes 2^{-k_1} by 2^{-k_2} where $k_1 + k_2 \leq k$, each rectangle will contain exactly the same number of points. We say that the points are **equidistributed** for this partition.

In general, can take $n = 2^k$ points.

If we partition $[0, 1)^2$ in rectangles of sizes 2^{-k_1} by 2^{-k_2} where $k_1 + k_2 \leq k$, each rectangle will contain exactly the same number of points. We say that the points are **equidistributed** for this partition.

For a digital net in base b in s dimensions, we choose s permutations of $\{0, 1, \dots, 2^b - 1\}$, then divide each coordinate by b^k .

Can also have $s = \infty$ and/or $n = \infty$ (**infinite sequence** of points).

Suppose we divide axis j in b^{q_j} equal parts, for each j . This determines a partition of $[0, 1)^s$ into $2^{q_1 + \dots + q_s}$ rectangles of equal sizes. If each rectangle contains exactly the same number of points, we say that the point set P_n is (q_1, \dots, q_s) -equidistributed in base b .

This occurs iff the matrix formed by the first q_1 rows of \mathbf{C}_1 , the first q_2 rows of \mathbf{C}_2 , \dots , the first q_s rows of \mathbf{C}_s , is of full rank (mod b). To verify equidistribution, we can construct these matrices and compute their rank.

P_n is a (t, k, s) -net iff it is (q_1, \dots, q_s) -equidistributed whenever $q_1 + \dots + q_s = k - t$. This is possible for $t = 0$ only if $b \geq s - 1$.
 t -value of a net: smallest t for which it is a (t, k, s) -net.

Suppose we divide axis j in b^{q_j} equal parts, for each j . This determines a partition of $[0, 1)^s$ into $2^{q_1 + \dots + q_s}$ rectangles of equal sizes. If each rectangle contains exactly the same number of points, we say that the point set P_n is (q_1, \dots, q_s) -equidistributed in base b .

This occurs iff the matrix formed by the first q_1 rows of \mathbf{C}_1 , the first q_2 rows of \mathbf{C}_2 , \dots , the first q_s rows of \mathbf{C}_s , is of full rank (mod b). To verify equidistribution, we can construct these matrices and compute their rank.

P_n is a (t, k, s) -net iff it is (q_1, \dots, q_s) -equidistributed whenever $q_1 + \dots + q_s = k - t$. This is possible for $t = 0$ only if $b \geq s - 1$.
 t -value of a net: smallest t for which it is a (t, k, s) -net.

An infinite sequence $\{\mathbf{u}_0, \mathbf{u}_1, \dots\}$ in $[0, 1)^s$ is a (t, s) -sequence in base b if for all $k > 0$ and $\nu \geq 0$, $Q(k, \nu) = \{\mathbf{u}_i : i = \nu b^k, \dots, (\nu + 1)b^k - 1\}$, is a (t, k, s) -net in base b .

Suppose we divide axis j in b^{q_j} equal parts, for each j . This determines a partition of $[0, 1)^s$ into $2^{q_1 + \dots + q_s}$ rectangles of equal sizes. If each rectangle contains exactly the same number of points, we say that the point set P_n is (q_1, \dots, q_s) -equidistributed in base b .

This occurs iff the matrix formed by the first q_1 rows of \mathbf{C}_1 , the first q_2 rows of \mathbf{C}_2 , \dots , the first q_s rows of \mathbf{C}_s , is of full rank (mod b). To verify equidistribution, we can construct these matrices and compute their rank.

P_n is a (t, k, s) -net iff it is (q_1, \dots, q_s) -equidistributed whenever $q_1 + \dots + q_s = k - t$. This is possible for $t = 0$ only if $b \geq s - 1$.
 t -value of a net: smallest t for which it is a (t, k, s) -net.

An infinite sequence $\{\mathbf{u}_0, \mathbf{u}_1, \dots\}$ in $[0, 1)^s$ is a (t, s) -sequence in base b if for all $k > 0$ and $\nu \geq 0$, $Q(k, \nu) = \{\mathbf{u}_i : i = \nu b^k, \dots, (\nu + 1)b^k - 1\}$, is a (t, k, s) -net in base b . This is possible for $t = 0$ only if $b \geq s$.

Sobol' nets and sequences

Sobol' (1967) proposed a digital net in base $b = 2$ where

$$\mathbf{c}_j = \begin{pmatrix} 1 & v_{j,2,1} & \cdots & v_{j,c,1} & \cdots \\ 0 & 1 & \cdots & v_{j,c,2} & \cdots \\ \vdots & 0 & \ddots & \vdots & \\ & \vdots & & 1 & \end{pmatrix}.$$

Sobol' nets and sequences

Sobol' (1967) proposed a digital net in base $b = 2$ where

$$\mathbf{C}_j = \begin{pmatrix} 1 & v_{j,2,1} & \cdots & v_{j,c,1} & \cdots \\ 0 & 1 & \cdots & v_{j,c,2} & \cdots \\ \vdots & 0 & \ddots & \vdots & \\ & \vdots & & 1 & \end{pmatrix}.$$

Column c of \mathbf{C}_j is represented by an odd integer

$$m_{j,c} = \sum_{l=1}^c v_{j,c,l} 2^{c-l} = v_{j,c,1} 2^{c-1} + \cdots + v_{j,c,c-1} 2 + 1 < 2^c.$$

The integers $m_{j,c}$ are selected as follows.

For each j , we choose a primitive polynomial over \mathbb{F}_2 ,

$$f_j(z) = z^{d_j} + a_{j,1}z^{d_j-1} + \cdots + a_{j,d_j},$$

and we choose d_j integers $m_{j,0}, \dots, m_{j,d_j-1}$ (the first d_j columns).

For each j , we choose a primitive polynomial over \mathbb{F}_2 ,

$$f_j(z) = z^{d_j} + a_{j,1}z^{d_j-1} + \cdots + a_{j,d_j},$$

and we choose d_j integers $m_{j,0}, \dots, m_{j,d_j-1}$ (the first d_j columns).

Then, $m_{j,d_j}, m_{j,d_j+1}, \dots$ are determined by the recurrence

$$m_{j,c} = 2a_{j,1}m_{j,c-1} \oplus \cdots \oplus 2^{d_j-1}a_{j,d_j-1}m_{j,c-d_j+1} \oplus 2^{d_j}m_{j,c-d_j} \oplus m_{j,c-d_j}$$

Proposition. If the polynomials $f_j(z)$ are all distinct, we obtain a (t, s) -sequence with $t \leq d_0 + \cdots + d_{s-1} + 1 - s$.

For each j , we choose a primitive polynomial over \mathbb{F}_2 ,

$$f_j(z) = z^{d_j} + a_{j,1}z^{d_j-1} + \cdots + a_{j,d_j},$$

and we choose d_j integers $m_{j,0}, \dots, m_{j,d_j-1}$ (the first d_j columns).

Then, $m_{j,d_j}, m_{j,d_j+1}, \dots$ are determined by the recurrence

$$m_{j,c} = 2a_{j,1}m_{j,c-1} \oplus \cdots \oplus 2^{d_j-1}a_{j,d_j-1}m_{j,c-d_j+1} \oplus 2^{d_j}m_{j,c-d_j} \oplus m_{j,c-d_j}$$

Proposition. If the polynomials $f_j(z)$ are all distinct, we obtain a (t, s) -sequence with $t \leq d_0 + \cdots + d_{s-1} + 1 - s$.

Sobol' suggests to list all primitive polynomials over \mathbb{F}_2 by increasing order of degree, starting with $f_0(z) \equiv 1$ (which gives $\mathbf{C}_0 = \mathbf{I}$), and to take $f_j(z)$ as the $(j+1)$ -th polynomial in the list.

There are many ways of selecting the first $m_{j,c}$'s, which are called the **direction numbers**. They can be selected to minimize some discrepancy (or figure of merit). The values proposed by Sobol' give an (s, ℓ) -equidistribution for $\ell = 1$ and $\ell = 2$ (only the first two bits).

For $n = 2^k$ fixed, we can gain one dimension as for the Faure sequence.

Joe and Kuo (2008) tabulated direction numbers giving the best t -value for the two-dimensional projections, for given s and k .

Other constructions

Faure nets and sequences

Niederreiter-Xing point sets and sequences

Polynomial lattice rules (special case of digital nets)

Halton sequence

Etc.

Worst-case error bounds

Koksma-Hlawka-type inequalities (Koksma, Hlawka, Hickernell, etc.):

$$|\hat{\mu}_{n,\text{rqmc}} - \mu| \leq V(f) \cdot D(P_n)$$

for all f in some Hilbert space or Banach space \mathcal{H} , where

$V(f) = \|f - \mu\|_{\mathcal{H}}$ is the variation of f , and $D(P_n)$ is the discrepancy of P_n .

Worst-case error bounds

Koksma-Hlawka-type inequalities (Koksma, Hlawka, Hickernell, etc.):

$$|\hat{\mu}_{n,\text{rqmc}} - \mu| \leq V(f) \cdot D(P_n)$$

for all f in some Hilbert space or Banach space \mathcal{H} , where

$V(f) = \|f - \mu\|_{\mathcal{H}}$ is the variation of f , and $D(P_n)$ is the discrepancy of P_n .

Lattice rules: For certain Hilbert spaces of smooth periodic functions f with square-integrable partial derivatives of order up to α :

$$D(P_n) = O(n^{-\alpha+\epsilon}) \text{ for arbitrary small } \epsilon.$$

Digital nets: "Classical" Koksma-Hlawka inequality for QMC: f must have finite variation in the sense of Hardy and Krause (implies no discontinuity not aligned with the axes). Popular constructions achieve

$$D(P_n) = O(n^{-1}(\ln n)^s) = O(n^{-1+\epsilon}) \text{ for arbitrary small } \epsilon.$$

More recent constructions offer better rates for smooth functions.

Worst-case error bounds

Koksma-Hlawka-type inequalities (Koksma, Hlawka, Hickernell, etc.):

$$|\hat{\mu}_{n,\text{rqmc}} - \mu| \leq V(f) \cdot D(P_n)$$

for all f in some Hilbert space or Banach space \mathcal{H} , where

$V(f) = \|f - \mu\|_{\mathcal{H}}$ is the variation of f , and $D(P_n)$ is the discrepancy of P_n .

Lattice rules: For certain Hilbert spaces of smooth periodic functions f with square-integrable partial derivatives of order up to α :

$$D(P_n) = O(n^{-\alpha+\epsilon}) \text{ for arbitrary small } \epsilon.$$

Digital nets: “Classical” Koksma-Hlawka inequality for QMC: f must have finite variation in the sense of Hardy and Krause (implies no discontinuity not aligned with the axes). Popular constructions achieve

$$D(P_n) = O(n^{-1}(\ln n)^s) = O(n^{-1+\epsilon}) \text{ for arbitrary small } \epsilon.$$

More recent constructions offer better rates for smooth functions.

Bounds are conservative and too hard to compute in practice.

Randomized quasi-Monte Carlo (RQMC)

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

with $P_n = \{\mathbf{U}_0, \dots, \mathbf{U}_{n-1}\} \subset (0, 1)^s$ an RQMC point set:

- (i) each point \mathbf{U}_i has the uniform distribution over $(0, 1)^s$;
- (ii) P_n as a whole is a low-discrepancy point set.

$$\mathbb{E}[\hat{\mu}_{n,\text{rqmc}}] = \mu \quad (\text{unbiased}).$$

$$\text{Var}[\hat{\mu}_{n,\text{rqmc}}] = \frac{\text{Var}[f(\mathbf{U}_i)]}{n} + \frac{2}{n^2} \sum_{i < j} \text{Cov}[f(\mathbf{U}_i), f(\mathbf{U}_j)].$$

We want to make the last sum as negative as possible.

Randomized quasi-Monte Carlo (RQMC)

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

with $P_n = \{\mathbf{U}_0, \dots, \mathbf{U}_{n-1}\} \subset (0, 1)^s$ an RQMC point set:

- (i) each point \mathbf{U}_i has the uniform distribution over $(0, 1)^s$;
- (ii) P_n as a whole is a low-discrepancy point set.

$$\mathbb{E}[\hat{\mu}_{n,\text{rqmc}}] = \mu \quad (\text{unbiased}).$$

$$\text{Var}[\hat{\mu}_{n,\text{rqmc}}] = \frac{\text{Var}[f(\mathbf{U}_i)]}{n} + \frac{2}{n^2} \sum_{i < j} \text{Cov}[f(\mathbf{U}_i), f(\mathbf{U}_j)].$$

We want to make the last sum as negative as possible.

Weaker attempts to do the same: antithetic variates ($n = 2$), Latin hypercube sampling (LHS), stratification, ...

Variance estimation:

Can compute m independent realizations X_1, \dots, X_m of $\hat{\mu}_{n,\text{rqmc}}$, then estimate μ and $\text{Var}[\hat{\mu}_{n,\text{rqmc}}]$ by their sample mean \bar{X}_m and sample variance S_m^2 . Could be used to compute a confidence interval.

Temptation: assume that \bar{X}_m has the normal distribution.

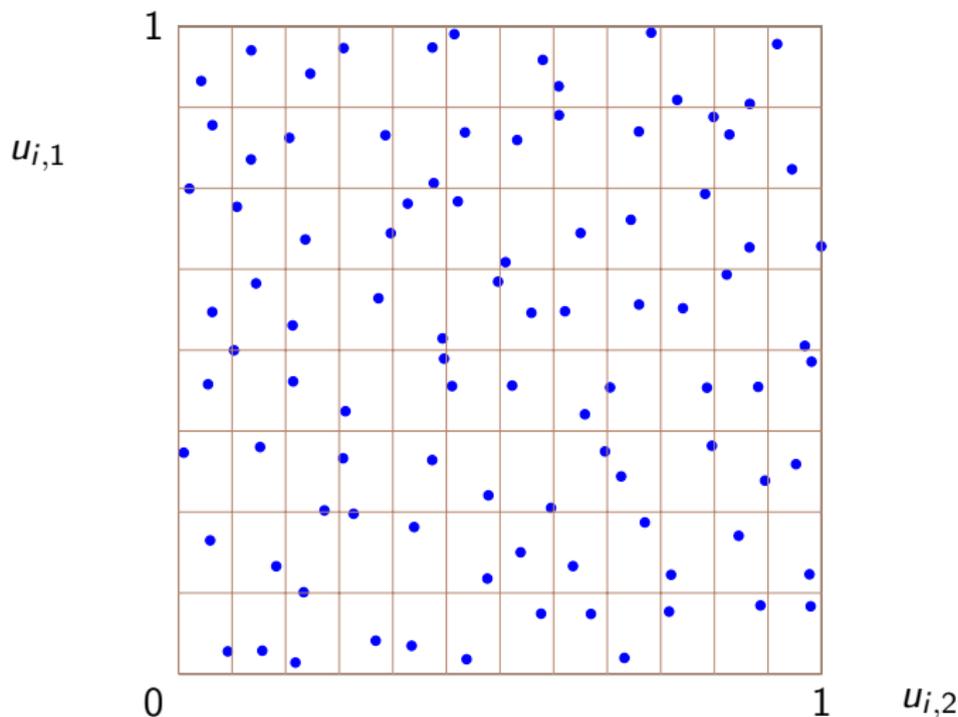
Beware: usually wrong unless $m \rightarrow \infty$.

Stratification of the unit hypercube

Partition axis j in $k_j \geq 1$ equal parts, for $j = 1, \dots, s$.

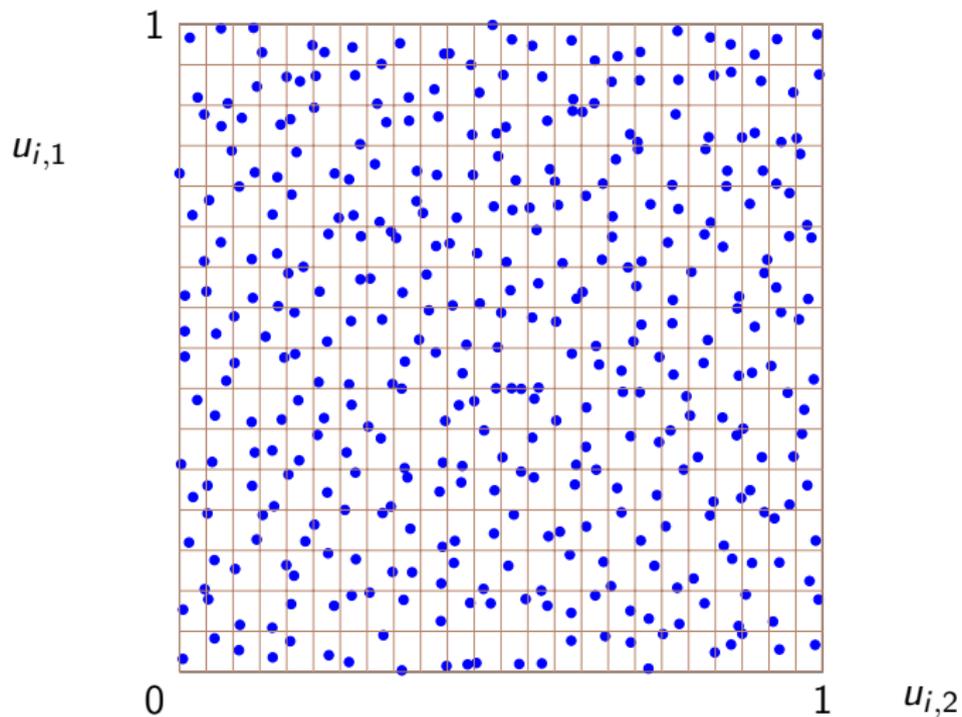
Draw $n = k_1 \cdots k_s$ random points, one per box, independently.

Example, $s = 2$, $k_1 = 12$, $k_2 = 8$, $n = 12 \times 8 = 96$.



Stratification of the unit hypercube

Example, $s = 2$, $k_1 = 24$, $k_2 = 16$, $n = 384$.



Stratified estimator:

$$X_{s,n} = \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{U}_j).$$

The crude MC variance with n points can be decomposed as

$$\text{Var}[\bar{X}_n] = \text{Var}[X_{s,n}] + \frac{1}{n} \sum_{j=0}^{n-1} (\mu_j - \mu)^2$$

where μ_j is the mean over box j .

The more the μ_j differ, the more the variance is reduced.

Stratified estimator:

$$X_{s,n} = \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{U}_j).$$

The crude MC variance with n points can be decomposed as

$$\text{Var}[\bar{X}_n] = \text{Var}[X_{s,n}] + \frac{1}{n} \sum_{j=0}^{n-1} (\mu_j - \mu)^2$$

where μ_j is the mean over box j .

The more the μ_j differ, the more the variance is reduced.

If f' is continuous and bounded, and all k_j are equal, then

$$\text{Var}[X_{s,n}] = \mathcal{O}(n^{-1-2/s}).$$

Stratified estimator:

$$X_{s,n} = \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{U}_j).$$

The crude MC variance with n points can be decomposed as

$$\text{Var}[\bar{X}_n] = \text{Var}[X_{s,n}] + \frac{1}{n} \sum_{j=0}^{n-1} (\mu_j - \mu)^2$$

where μ_j is the mean over box j .

The more the μ_j differ, the more the variance is reduced.

If f' is continuous and bounded, and all k_j are equal, then

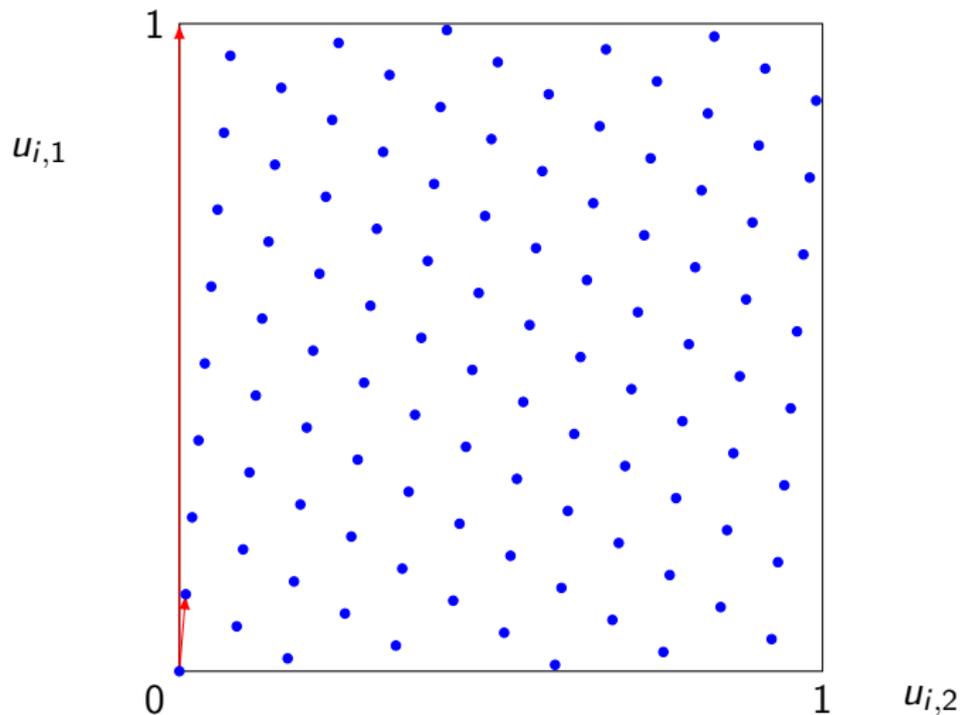
$$\text{Var}[X_{s,n}] = \mathcal{O}(n^{-1-2/s}).$$

For large s , not practical. For small s , not really better than midpoint rule with a grid when f is smooth. But can still be applied to a few important random variables.

Also, gives an unbiased estimator, and variance can be estimated by replicating $m \geq 2$ times.

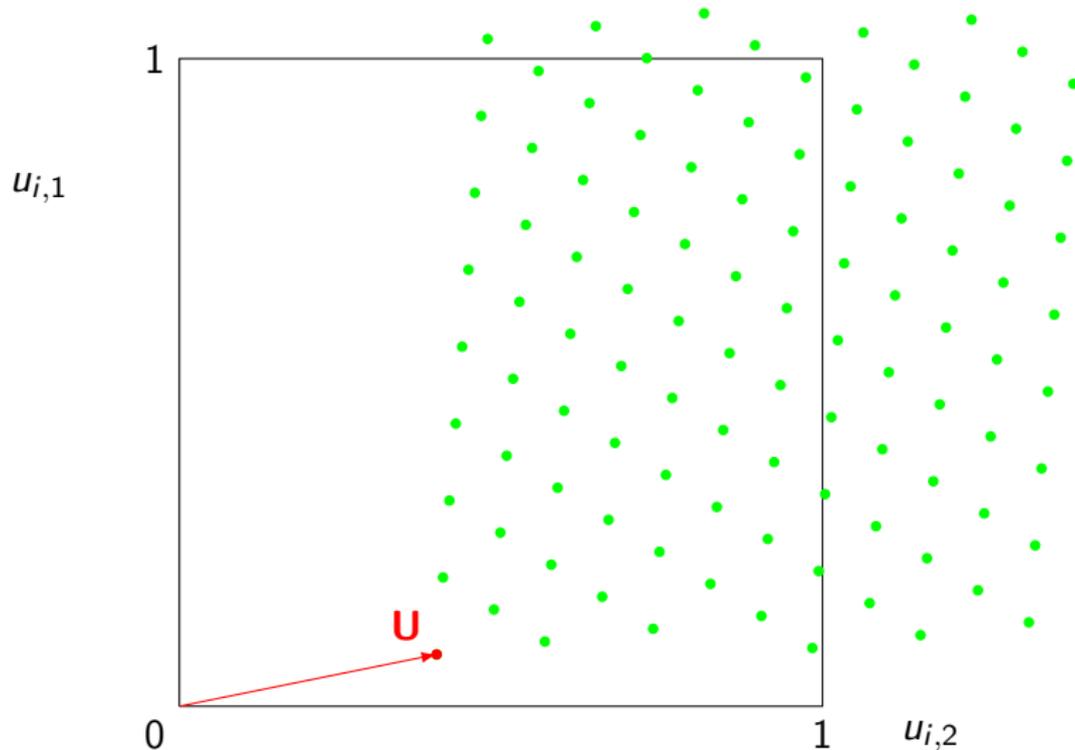
Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/101$



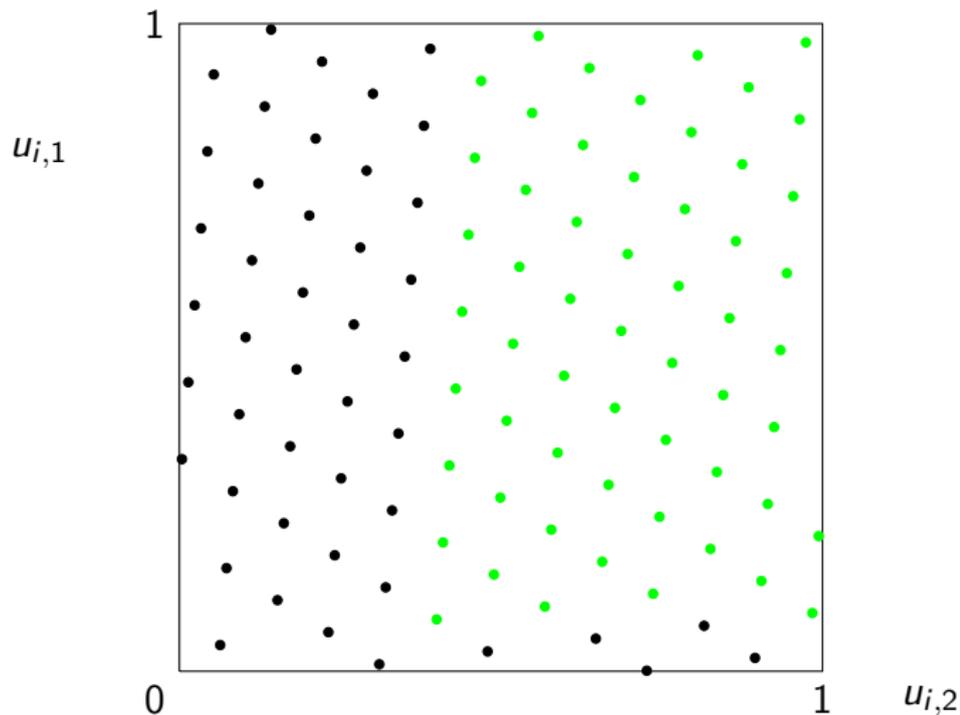
Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/101$



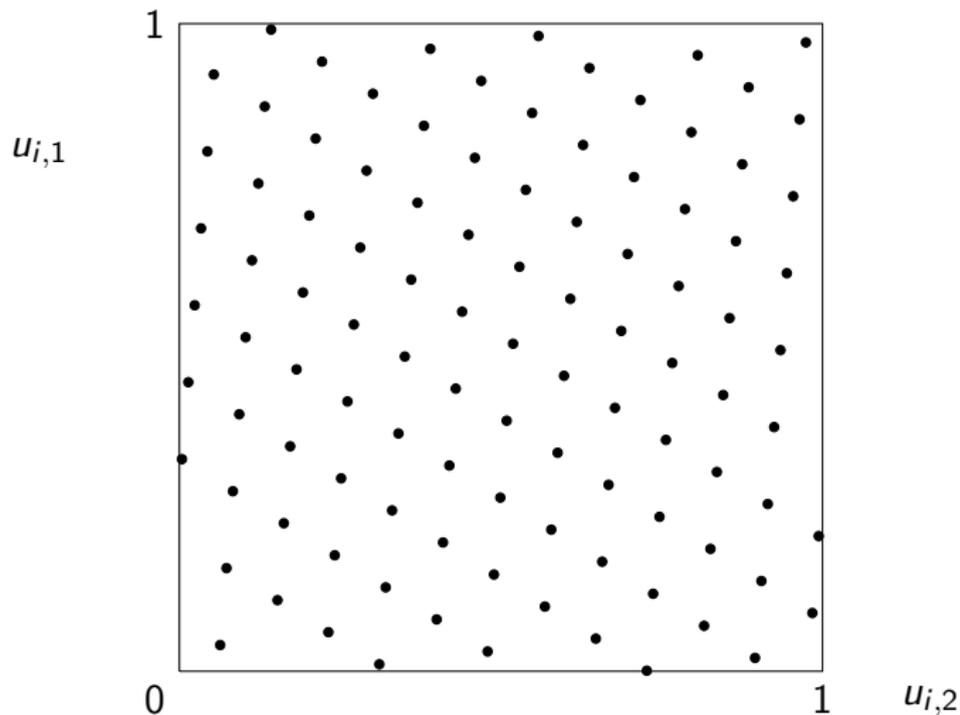
Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/101$



Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $\mathbf{v}_1 = (1, 12)/101$



Random digital shift for digital net

Equidistribution in digital boxes is lost with random shift modulo 1, but can be kept with a **random digital shift** in base b .

In **base 2**: Generate $\mathbf{U} \sim U(0, 1)^s$ and XOR it bitwise with each \mathbf{u}_i .

Example for $s = 2$:

$$\begin{aligned} \mathbf{u}_i &= (0.01100100\dots, 0.10011000\dots)_2 \\ \mathbf{U} &= (0.01001010\dots, 0.11101001\dots)_2 \\ \mathbf{u}_i \oplus \mathbf{U} &= (0.00101110\dots, 0.01110001\dots)_2. \end{aligned}$$

Each point has $U(0, 1)$ distribution.

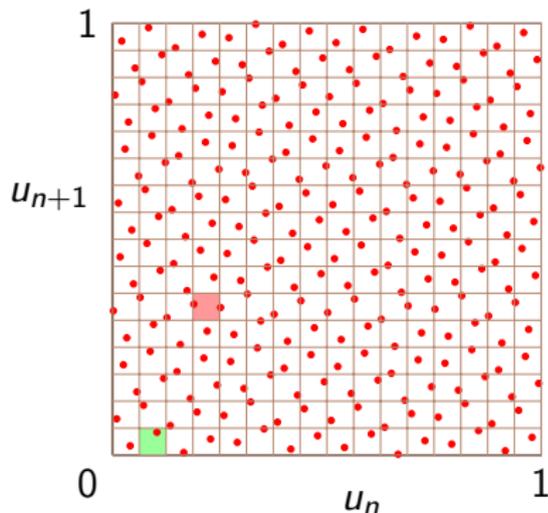
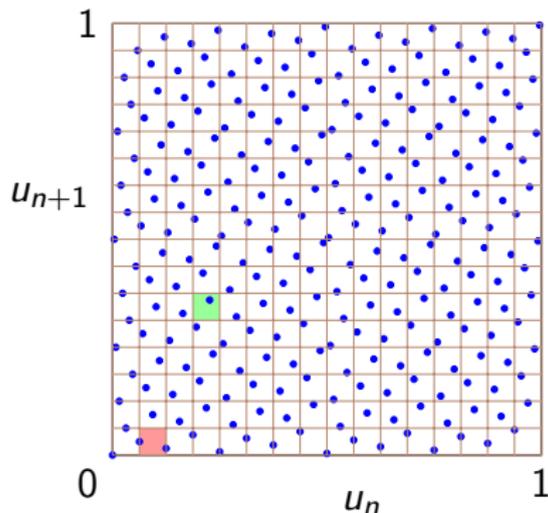
Preservation of the equidistribution ($k_1 = 3, k_2 = 5$):

$$\begin{aligned} \mathbf{u}_i &= (0.***, 0.*****) \\ \mathbf{U} &= (0.010, 0.11101)_2 \\ \mathbf{u}_i \oplus \mathbf{U} &= (0.***, 0.*****) \end{aligned}$$

Example with

$$\begin{aligned} \mathbf{U} &= (0.1270111220, 0.3185275653)_{10} \\ &= (0.0010\ 0000100000111100, 0.0101\ 0001100010110000)_2. \end{aligned}$$

Changes the bits 3, 9, 15, 16, 17, 18 of $u_{i,1}$
and the bits 2, 4, 8, 9, 13, 15, 16 of $u_{i,2}$.



Red and green squares are permuted ($k_1 = k_2 = 4$, first 4 bits of \mathbf{U}).

Random digital shift in base b

We have $u_{i,j} = \sum_{\ell=1}^w u_{i,j,\ell} b^{-\ell}$.

Let $\mathbf{U} = (U_1, \dots, U_s) \sim U[0, 1]^s$ where $U_j = \sum_{\ell=1}^w U_{j,\ell} b^{-\ell}$.

We replace each $u_{i,j}$ by $\tilde{U}_{i,j} = \sum_{\ell=1}^w [(u_{i,j,\ell} + U_{j,\ell}) \bmod b] b^{-\ell}$.

Proposition. \tilde{P}_n is (q_1, \dots, q_s) -equidistributed in base b iff P_n is.

For $w = \infty$, each point $\tilde{\mathbf{U}}_i$ has the uniform distribution over $(0, 1)^s$.

Other permutations that preserve equidistribution and may help reduce the variance further:

Linear matrix scrambling (Matoušek, Hickernell et Hong, Tezuka, Owen):

We left-multiply each matrix \mathbf{C}_j by a random $w \times w$ matrix \mathbf{M}_j , non-singular and lower triangular, mod b . Several variants.

We then apply a random digital shift in base b to obtain uniform distribution for each point (unbiasedness).

Other permutations that preserve equidistribution and may help reduce the variance further:

Linear matrix scrambling (Matoušek, Hickernell et Hong, Tezuka, Owen):

We left-multiply each matrix \mathbf{C}_j by a random $w \times w$ matrix \mathbf{M}_j , non-singular and lower triangular, mod b . Several variants.

We then apply a random digital shift in base b to obtain uniform distribution for each point (unbiasedness).

Nested uniform scrambling (Owen 1995).

More costly. But provably reduces the variance to $O(n^{-3}(\log n)^s)$ when f is sufficiently smooth!

Asian option example

$T = 1$ (year), $t_j = j/d$, $K = 100$, $s_0 = 100$, $r = 0.05$, $\sigma = 0.5$.

$s = d = 2$. Exact value: $\mu \approx 17.0958$. MC Variance: **934.0**.

Lattice: Korobov with a from old table + random shift.

Sobol: left matrix scramble + random digital shift.

Variance estimated from $m = 1000$ indep. randomizations.

VRF = (MC variance) / ($n\text{Var}[X_{s,n}]$)

| method | n | \bar{X}_m | nS_m^2 | VRF |
|----------|----------|-------------|----------|---------|
| stratif. | 2^{10} | 17.100 | 232.8 | 4 |
| lattice | 2^{10} | 17.092 | 20.8 | 45 |
| Sobol | 2^{10} | 17.094 | 1.66 | 563 |
| stratif. | 2^{16} | 17.046 | 135.3 | 7 |
| lattice | 2^{16} | 17.096 | 4.38 | 213 |
| Sobol | 2^{16} | 17.096 | 0.037 | 25,330 |
| stratif. | 2^{20} | 17.085 | 117.6 | 8 |
| lattice | 2^{20} | 17.096 | 0.112 | 8,318 |
| Sobol | 2^{20} | 17.096 | 0.0026 | 360,000 |

$s = d = 12$. $\mu \approx 13.122$. MC variance: **516.3**.

Lattice: Korobov + random shift.

Sobol: left matrix scramble + random digital shift.

Variance estimated from $m = 1000$ indep. randomizations.

| method | n | \bar{X}_m | nS_m^2 | VRF |
|---------|----------|-------------|----------|-----|
| lattice | 2^{10} | 13.114 | 39.3 | 13 |
| Sobol | 2^{10} | 13.123 | 5.9 | 88 |
| lattice | 2^{16} | 13.122 | 6.61 | 78 |
| Sobol | 2^{16} | 13.122 | 1.63 | 317 |
| lattice | 2^{20} | 13.122 | 8.59 | 60 |
| Sobol | 2^{20} | 13.122 | 0.89 | 579 |

Variance for randomly-shifted lattice rules

Suppose f has Fourier expansion

$$f(\mathbf{u}) = \sum_{\mathbf{h} \in \mathbb{Z}^s} \hat{f}(\mathbf{h}) e^{2\pi\sqrt{-1}\mathbf{h}^t\mathbf{u}}.$$

For a **randomly shifted lattice**, the exact variance is always

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

where $L_s^* = \{\mathbf{h} \in \mathbb{R}^s : \mathbf{h}^t\mathbf{v} \in \mathbb{Z} \text{ for all } \mathbf{v} \in L_s\} \subseteq \mathbb{Z}^s$ is the **dual lattice**.

From the viewpoint of variance reduction, an **optimal lattice for f** minimizes $\text{Var}[\hat{\mu}_{n,\text{rqmc}}]$.

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

Let $\alpha > 0$ be an even integer. If f has **square-integrable mixed partial derivatives** up to order $\alpha/2 > 0$, and the periodic continuation of its derivatives up to order $\alpha/2 - 1$ is **continuous across the unit cube boundaries**, then

$$|\hat{f}(\mathbf{h})|^2 = \mathcal{O}((\max(1, h_1) \cdots \max(1, h_s))^{-\alpha}).$$

Moreover, there is a vector $\mathbf{v}_1 = \mathbf{v}_1(n)$ such that

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

This \mathcal{P}_α has been proposed long ago as a figure of merit, often with $\alpha = 2$. It is the variance for a **worst-case f** having

$$|\hat{f}(\mathbf{h})|^2 = (\max(1, |h_1|) \cdots \max(1, |h_s|))^{-\alpha}.$$

A larger α means a smoother f and a faster convergence rate.

For even integer α , this worst-case f is

$$f^*(\mathbf{u}) = \sum_{u \subseteq \{1, \dots, s\}} \prod_{j \in u} \frac{(2\pi)^{\alpha/2}}{(\alpha/2)!} B_{\alpha/2}(u_j).$$

where $B_{\alpha/2}$ is the Bernoulli polynomial of degree $\alpha/2$.

In particular, $B_1(u) = u - 1/2$ and $B_2(u) = u^2 - u + 1/6$.

Easy to compute P_α and search for good lattices in this case!

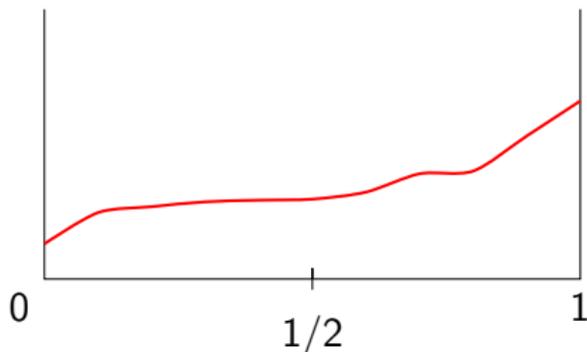
However: This worst-case function is not necessarily representative of what happens in applications. Also, the **hidden factor in \mathcal{O} increases quickly with s** , so this result is not very useful for large s .

To get a bound that is uniform in s , the Fourier coefficients must decrease faster with the dimension and “size” of vectors \mathbf{h} ; that is, f must be “smoother” in high-dimensional projections. This is typically what happens in applications for which RQMC is effective!

Baker's (or tent) transformation

To make the periodic continuation of f **continuous**.

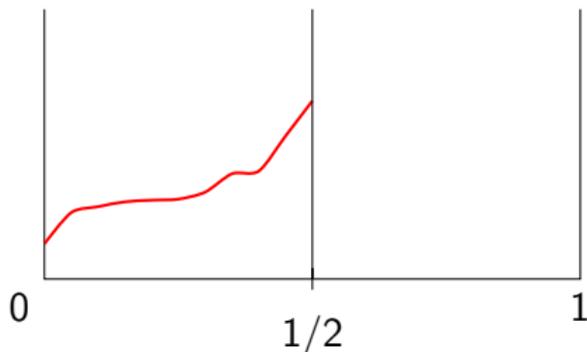
If $f(0) \neq f(1)$, define \tilde{f} by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$. This \tilde{f} has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's (or tent) transformation

To make the periodic continuation of f **continuous**.

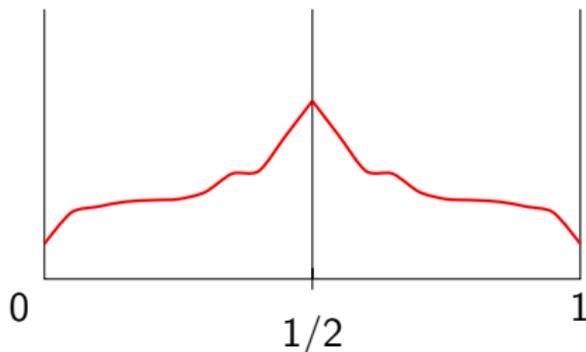
If $f(0) \neq f(1)$, define \tilde{f} by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$. This \tilde{f} has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's (or tent) transformation

To make the periodic continuation of f **continuous**.

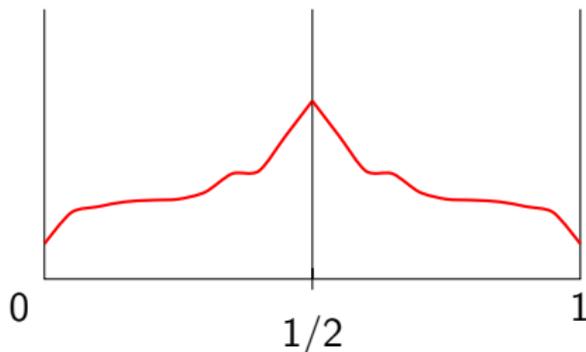
If $f(0) \neq f(1)$, define \tilde{f} by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$. This \tilde{f} has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's (or tent) transformation

To make the periodic continuation of f **continuous**.

If $f(0) \neq f(1)$, define \tilde{f} by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$. This \tilde{f} has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



For smooth f , can reduce the variance to $O(n^{-4+\epsilon})$ (Hickernell 2002). The resulting \tilde{f} is symmetric with respect to $u = 1/2$.

In practice, we transform the points \mathbf{U}_i instead of f .

One-dimensional case

Random shift followed by [baker's transformation](#).

Along each coordinate, stretch everything by a factor of 2 and fold.

Same as replacing U_j by $\min[2U_j, 2(1 - U_j)]$.

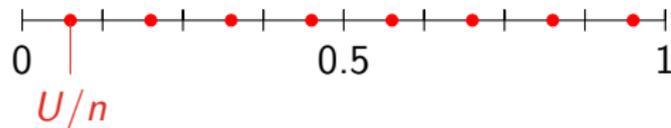


One-dimensional case

Random shift followed by [baker's transformation](#).

Along each coordinate, stretch everything by a factor of 2 and fold.

Same as replacing U_j by $\min[2U_j, 2(1 - U_j)]$.



One-dimensional case

Random shift followed by [baker's transformation](#).

Along each coordinate, stretch everything by a factor of 2 and fold.

Same as replacing U_j by $\min[2U_j, 2(1 - U_j)]$.

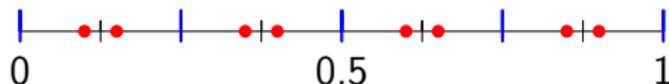


One-dimensional case

Random shift followed by [baker's transformation](#).

Along each coordinate, stretch everything by a factor of 2 and fold.

Same as replacing U_j by $\min[2U_j, 2(1 - U_j)]$.



Gives [locally antithetic](#) points in intervals of size $2/n$.

This implies that linear pieces over these intervals are integrated exactly.

Intuition: when f is smooth, it is well-approximated by a piecewise linear function, which is integrated exactly, so the error is small.

ANOVA decomposition

The Fourier expansion has too many terms to handle. As a cruder expansion, we can write $f(\mathbf{u}) = f(u_1, \dots, u_s)$ as:

$$f(\mathbf{u}) = \sum_{\mathbf{u} \subseteq \{1, \dots, s\}} f_{\mathbf{u}}(\mathbf{u}) = \mu + \sum_{i=1}^s f_{\{i\}}(u_i) + \sum_{i,j=1}^s f_{\{i,j\}}(u_i, u_j) + \dots$$

where

$$f_{\mathbf{u}}(\mathbf{u}) = \int_{[0,1]^{|\bar{\mathbf{u}}|}} f(\mathbf{u}) d\mathbf{u}_{\bar{\mathbf{u}}} - \sum_{\mathbf{v} \subset \mathbf{u}} f_{\mathbf{v}}(\mathbf{u}_{\mathbf{v}}),$$

and the Monte Carlo variance decomposes as

$$\sigma^2 = \sum_{\mathbf{u} \subseteq \{1, \dots, s\}} \sigma_{\mathbf{u}}^2, \quad \text{where } \sigma_{\mathbf{u}}^2 = \text{Var}[f_{\mathbf{u}}(\mathbf{U})].$$

The $\sigma_{\mathbf{u}}^2$'s can be estimated by MC or RQMC.

Heuristic intuition: Make sure the projections $P_n(\mathbf{u})$ are very uniform for the important subsets \mathbf{u} (i.e., with larger $\sigma_{\mathbf{u}}^2$).

Weighted $\mathcal{P}_{\gamma,\alpha}$ with projection-dependent weights $\gamma_{\mathbf{u}}$

Denote $\mathbf{u}(\mathbf{h}) = \mathbf{u}(h_1, \dots, h_s)$ the set of indices j for which $h_j \neq 0$.

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

For $\alpha/2$ integer > 0 , with $\mathbf{u}_i = (u_{i,1}, \dots, u_{i,s}) = i\mathbf{v}_1 \bmod 1$,

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

and the corresponding variation is

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

for $f : [0, 1]^s \rightarrow \mathbb{R}$ smooth enough. Then,

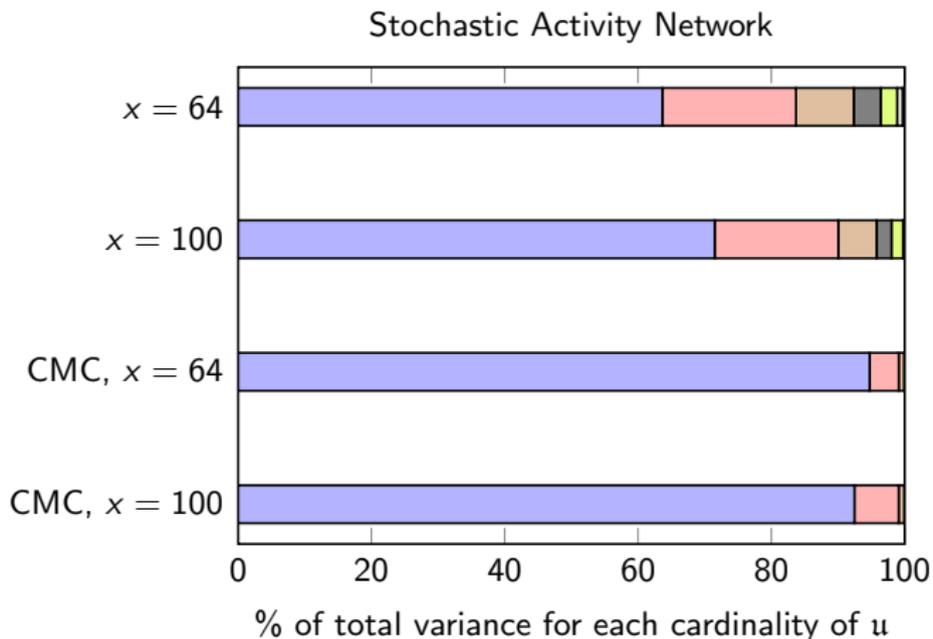
$$\text{Var}[\hat{\mu}_{n,\text{rqmc}}] = \sum_{\mathbf{u} \subseteq \{1, \dots, s\}} \text{Var}[\hat{\mu}_{n,\text{rqmc}}(f_{\mathbf{u}})] \leq V_{\gamma}^2(f) \mathcal{P}_{\gamma,\alpha}.$$

$\mathcal{P}_{\gamma,\alpha}$ with $\alpha = 2$ and properly chosen weights γ is a good practical choice of figure of merit.

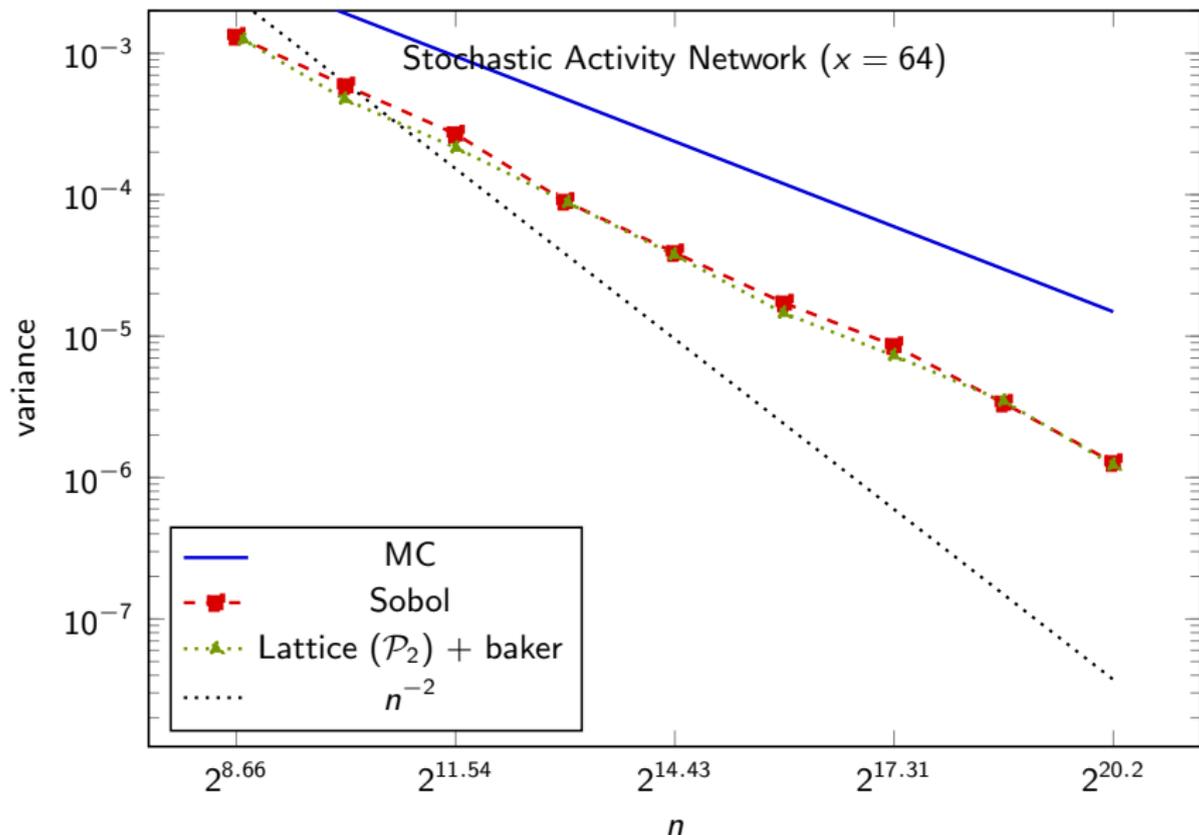
Simple choices of weights: order-dependent or product.

Lattice Builder: Software to search for good lattices with arbitrary n , s , weights, etc. See my web page.

ANOVA Variances for estimator of $\mathbb{P}[T > x]$ in Stochastic Activity Network

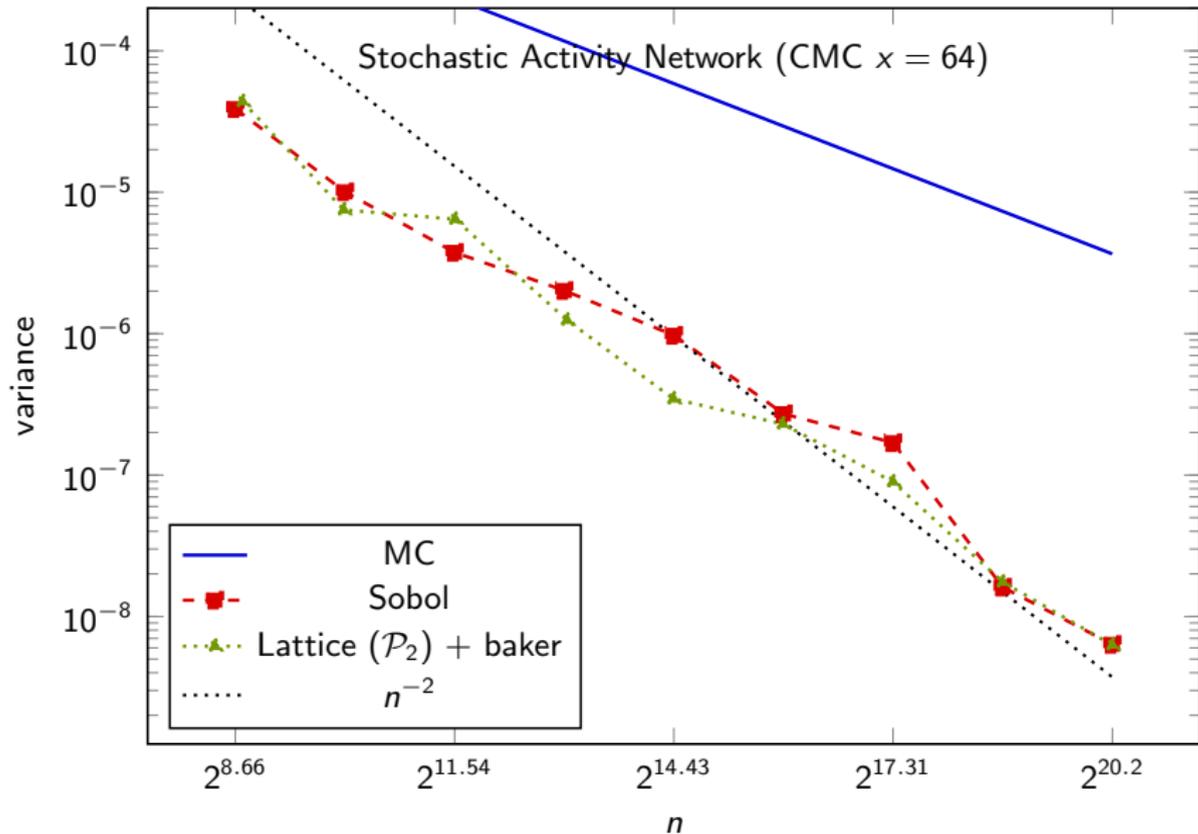


Variance for estimator of $\mathbb{P}[T > x]$ for SAN



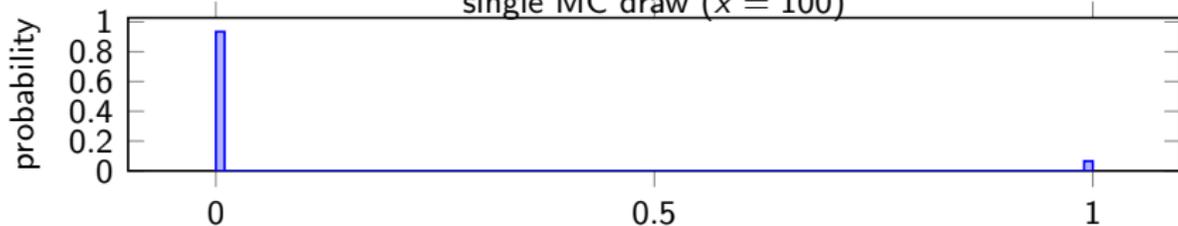
Variance decreases roughly as $\mathcal{O}(n^{-1.2})$. For $\mathbb{E}[T]$, we observe $\mathcal{O}(n^{-1.4})$.

Variance for estimator of $\mathbb{P}[T > x]$ with CMC

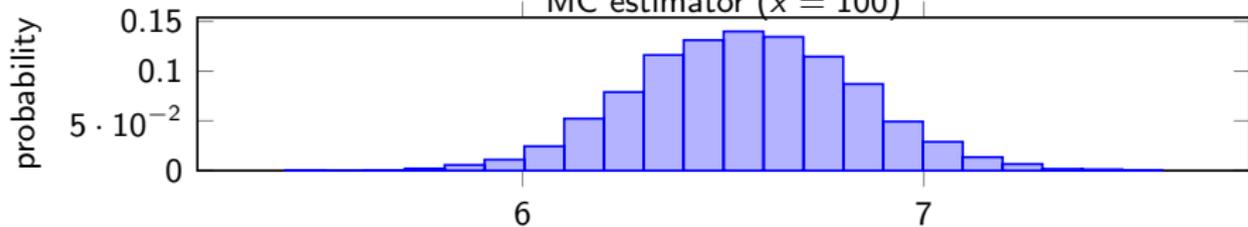


Histograms

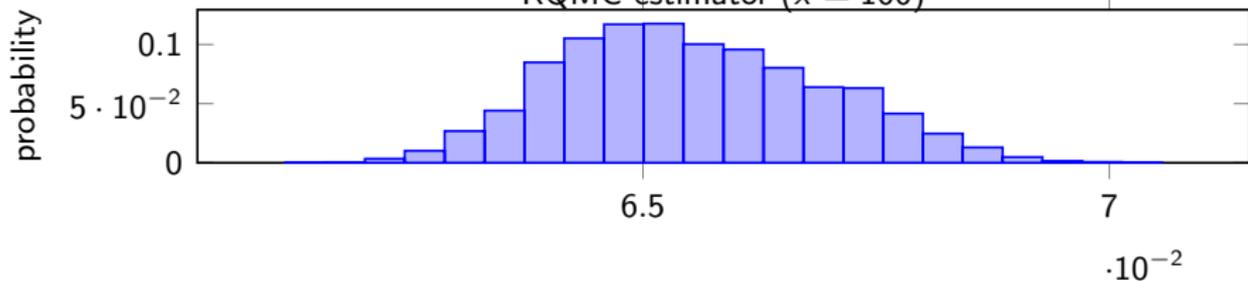
single MC draw ($x = 100$)



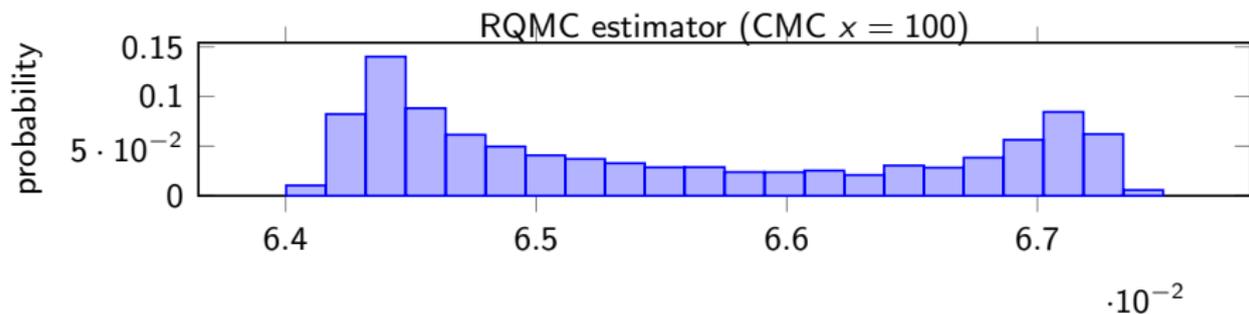
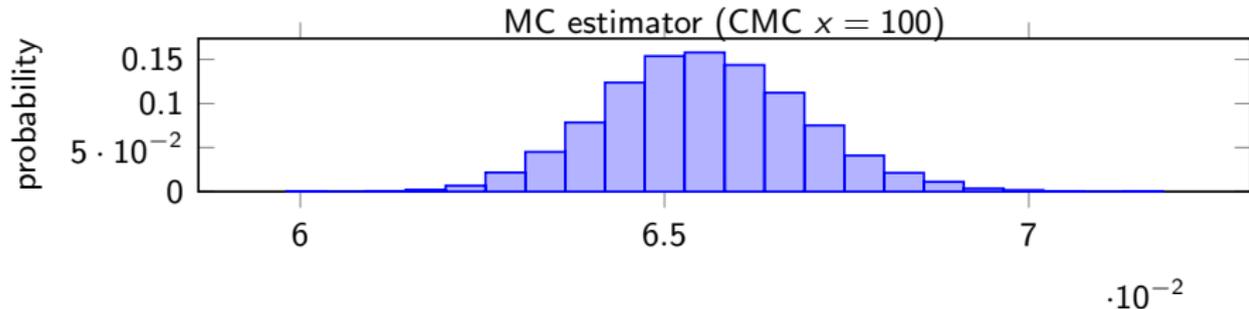
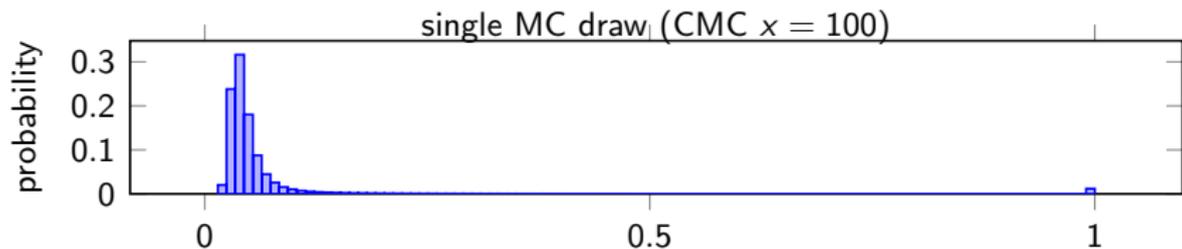
MC estimator ($x = 100$)



RQMC estimator ($x = 100$)



Histograms



Effective dimension

(Caflisch, Morokoff, and Owen 1997).

A function f has **effective dimension** d in proportion ρ in the **superposition sense** if

$$\sum_{|u| \leq d} \sigma_u^2 \geq \rho \sigma^2.$$

It has effective dimension d in the **truncation sense** if

$$\sum_{u \subseteq \{1, \dots, d\}} \sigma_u^2 \geq \rho \sigma^2.$$

High-dimensional functions with **low effective dimension** are frequent. One may **change** f to make this happen.

Example: Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \dots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

Example: Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \dots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c -dimensional geometric Brownian motion (GMB) at d observations times $0 < t_1 < \dots < t_d = T$, then we have $s = cd$.

Example: Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \dots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c -dimensional geometric Brownian motion (GMB) at d observations times $0 < t_1 < \dots < t_d = T$, then we have $s = cd$.

To generate \mathbf{Y} : Decompose $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^t$, generate $\mathbf{Z} = (Z_1, \dots, Z_s) \sim N(\mathbf{0}, \mathbf{I})$ where the (independent) Z_j 's are generated by inversion: $Z_j = \Phi^{-1}(U_j)$, and return $\mathbf{Y} = \mathbf{A}\mathbf{Z}$.

Example: Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \dots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c -dimensional geometric Brownian motion (GMB) at d observations times $0 < t_1 < \dots < t_d = T$, then we have $s = cd$.

To generate \mathbf{Y} : Decompose $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^t$, generate $\mathbf{Z} = (Z_1, \dots, Z_s) \sim N(\mathbf{0}, \mathbf{I})$ where the (independent) Z_j 's are generated by inversion: $Z_j = \Phi^{-1}(U_j)$, and return $\mathbf{Y} = \mathbf{A}\mathbf{Z}$.

Choice of \mathbf{A} ?

Example: Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \dots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c -dimensional geometric Brownian motion (GMB) at d observations times $0 < t_1 < \dots < t_d = T$, then we have $s = cd$.

To generate \mathbf{Y} : Decompose $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^t$, generate $\mathbf{Z} = (Z_1, \dots, Z_s) \sim N(\mathbf{0}, \mathbf{I})$ where the (independent) Z_j 's are generated by inversion: $Z_j = \Phi^{-1}(U_j)$, and return $\mathbf{Y} = \mathbf{A}\mathbf{Z}$.

Choice of \mathbf{A} ?

Cholesky factorization: \mathbf{A} is lower triangular.

Principal component decomposition (PCA) (Ackworth et al. 1998):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \text{diag}(\lambda_s, \dots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of \mathbf{P} are the corresponding unit-length eigenvectors.

Principal component decomposition (PCA) (Ackworth et al. 1998):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \text{diag}(\lambda_s, \dots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of \mathbf{P} are the corresponding unit-length eigenvectors. With this \mathbf{A} , Z_1 accounts for the max amount of variance of \mathbf{Y} , then Z_2 the max amount of variance cond. on Z_1 , etc.

Principal component decomposition (PCA) (Ackworth et al. 1998):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \text{diag}(\lambda_s, \dots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of \mathbf{P} are the corresponding unit-length eigenvectors. With this \mathbf{A} , Z_1 accounts for the **max amount of variance of \mathbf{Y}** , then Z_2 the max amount of variance cond. on Z_1 , etc.

Function of a Brownian motion (or other Lévy process):

Payoff depends on c -dimensional Brownian motion $\{\mathbf{X}(t), t \geq 0\}$ observed at times $0 = t_0 < t_1 < \dots < t_d = T$.

Principal component decomposition (PCA) (Ackworth et al. 1998):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \text{diag}(\lambda_s, \dots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of \mathbf{P} are the corresponding unit-length eigenvectors. With this \mathbf{A} , Z_1 accounts for the **max amount of variance of \mathbf{Y}** , then Z_2 the max amount of variance cond. on Z_1 , etc.

Function of a Brownian motion (or other Lévy process):

Payoff depends on c -dimensional Brownian motion $\{\mathbf{X}(t), t \geq 0\}$ observed at times $0 = t_0 < t_1 < \dots < t_d = T$.

Sequential (or **random walk**) method: generate $\mathbf{X}(t_1)$, then $\mathbf{X}(t_2) - \mathbf{X}(t_1)$, then $\mathbf{X}(t_3) - \mathbf{X}(t_2)$, etc.

Principal component decomposition (PCA) (Ackworth et al. 1998):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \text{diag}(\lambda_s, \dots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of \mathbf{P} are the corresponding unit-length eigenvectors. With this \mathbf{A} , Z_1 accounts for the **max amount of variance of \mathbf{Y}** , then Z_2 the max amount of variance cond. on Z_1 , etc.

Function of a Brownian motion (or other Lévy process):

Payoff depends on c -dimensional Brownian motion $\{\mathbf{X}(t), t \geq 0\}$ observed at times $0 = t_0 < t_1 < \dots < t_d = T$.

Sequential (or **random walk**) method: generate $\mathbf{X}(t_1)$, then $\mathbf{X}(t_2) - \mathbf{X}(t_1)$, then $\mathbf{X}(t_3) - \mathbf{X}(t_2)$, etc.

Bridge sampling (Moskowitz and Caflisch 1996). Suppose $d = 2^m$. generate $\mathbf{X}(t_d)$, then $\mathbf{X}(t_{d/2})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_d))$,

Principal component decomposition (PCA) (Ackworth et al. 1998):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \text{diag}(\lambda_s, \dots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of \mathbf{P} are the corresponding unit-length eigenvectors. With this \mathbf{A} , Z_1 accounts for the **max amount of variance of \mathbf{Y}** , then Z_2 the max amount of variance cond. on Z_1 , etc.

Function of a Brownian motion (or other Lévy process):

Payoff depends on c -dimensional Brownian motion $\{\mathbf{X}(t), t \geq 0\}$ observed at times $0 = t_0 < t_1 < \dots < t_d = T$.

Sequential (or **random walk**) method: generate $\mathbf{X}(t_1)$, then $\mathbf{X}(t_2) - \mathbf{X}(t_1)$, then $\mathbf{X}(t_3) - \mathbf{X}(t_2)$, etc.

Bridge sampling (Moskowitz and Caflisch 1996). Suppose $d = 2^m$.

generate $\mathbf{X}(t_d)$, then $\mathbf{X}(t_{d/2})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_d))$, then $\mathbf{X}(t_{d/4})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_{d/2}))$, and so on.

The first few $N(0, 1)$ r.v.'s already sketch the path trajectory.

Principal component decomposition (PCA) (Ackworth et al. 1998):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \text{diag}(\lambda_s, \dots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of \mathbf{P} are the corresponding unit-length eigenvectors. With this \mathbf{A} , Z_1 accounts for the **max amount of variance of \mathbf{Y}** , then Z_2 the max amount of variance cond. on Z_1 , etc.

Function of a Brownian motion (or other Lévy process):

Payoff depends on c -dimensional Brownian motion $\{\mathbf{X}(t), t \geq 0\}$ observed at times $0 = t_0 < t_1 < \dots < t_d = T$.

Sequential (or **random walk**) method: generate $\mathbf{X}(t_1)$, then $\mathbf{X}(t_2) - \mathbf{X}(t_1)$, then $\mathbf{X}(t_3) - \mathbf{X}(t_2)$, etc.

Bridge sampling (Moskowitz and Caflisch 1996). Suppose $d = 2^m$. generate $\mathbf{X}(t_d)$, then $\mathbf{X}(t_{d/2})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_d))$, then $\mathbf{X}(t_{d/4})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_{d/2}))$, and so on.

The first few $N(0, 1)$ r.v.'s already sketch the path trajectory.

Each of these methods corresponds to some matrix \mathbf{A} .

Choice has a large impact on the ANOVA decomposition of f .

Example: Pricing an Asian basket option

We have c assets, d observation times. Want to estimate $\mathbb{E}[f(\mathbf{U})]$, where

$$f(\mathbf{U}) = e^{-rT} \max \left[0, \frac{1}{cd} \sum_{i=1}^c \sum_{j=1}^d S_i(t_j) - K \right]$$

is the net discounted payoff and $S_i(t_j)$ is the price of asset i at time t_j .

Example: Pricing an Asian basket option

We have c assets, d observation times. Want to estimate $\mathbb{E}[f(\mathbf{U})]$, where

$$f(\mathbf{U}) = e^{-rT} \max \left[0, \frac{1}{cd} \sum_{i=1}^c \sum_{j=1}^d S_i(t_j) - K \right]$$

is the net discounted payoff and $S_i(t_j)$ is the price of asset i at time t_j .

Suppose $(S_1(t), \dots, S_c(t))$ obeys a geometric Brownian motion.

Then, $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \dots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$ and $s = cd$.

Example: Pricing an Asian basket option

We have c assets, d observation times. Want to estimate $\mathbb{E}[f(\mathbf{U})]$, where

$$f(\mathbf{U}) = e^{-rT} \max \left[0, \frac{1}{cd} \sum_{i=1}^c \sum_{j=1}^d S_i(t_j) - K \right]$$

is the net discounted payoff and $S_i(t_j)$ is the price of asset i at time t_j .

Suppose $(S_1(t), \dots, S_c(t))$ obeys a geometric Brownian motion.

Then, $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \dots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$ and $s = cd$.

Even with Cholesky decompositions of $\mathbf{\Sigma}$, the two-dimensional projections often account for more than 99% of the variance: low effective dimension in the superposition sense.

With PCA or bridge sampling, we get low effective dimension in the truncation sense. In realistic examples, the first two coordinates Z_1 and Z_2 often account for more than 99.99% of the variance!

Numerical experiment with $c = 10$ and $d = 25$

This gives a 250-dimensional integration problem.

Let $\rho_{i,j} = 0.4$ for all $i \neq j$, $T = 1$, $\sigma_i = 0.1 + 0.4(i - 1)/9$ for all i , $r = 0.04$, $S(0) = 100$, and $K = 100$. (Imai and Tan 2002).

Numerical experiment with $c = 10$ and $d = 25$

This gives a 250-dimensional integration problem.

Let $\rho_{i,j} = 0.4$ for all $i \neq j$, $T = 1$, $\sigma_i = 0.1 + 0.4(i - 1)/9$ for all i , $r = 0.04$, $S(0) = 100$, and $K = 100$. (Imai and Tan 2002).

Variance reduction factors for Cholesky (left) and PCA (right) (experiment from 2003):

Korobov Lattice Rules

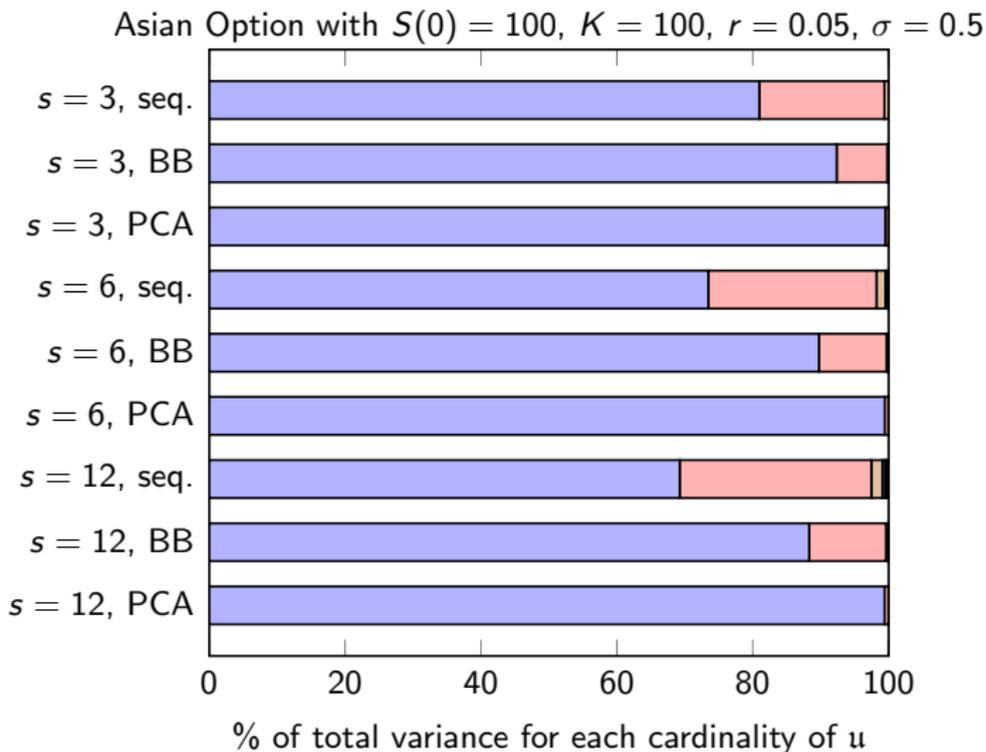
| | $n = 16381$ $a = 5693$ | $n = 65521$ $a = 944$ | $n = 262139$ $a = 21876$ |
|---------------------|---------------------------|--------------------------|-----------------------------|
| Lattice+shift | 18 878 | 18 1504 | 9 2643 |
| Lattice+shift+baker | 50 4553 | 46 3657 | 43 7553 |

Sobol' Nets

| | $n = 2^{14}$ | $n = 2^{16}$ | $n = 2^{18}$ |
|-----------------|--------------|--------------|--------------|
| Sobol+Shift | 10 1299 | 17 3184 | 32 6046 |
| Sobol+LMS+Shift | 6 4232 | 4 9219 | 35 16557 |

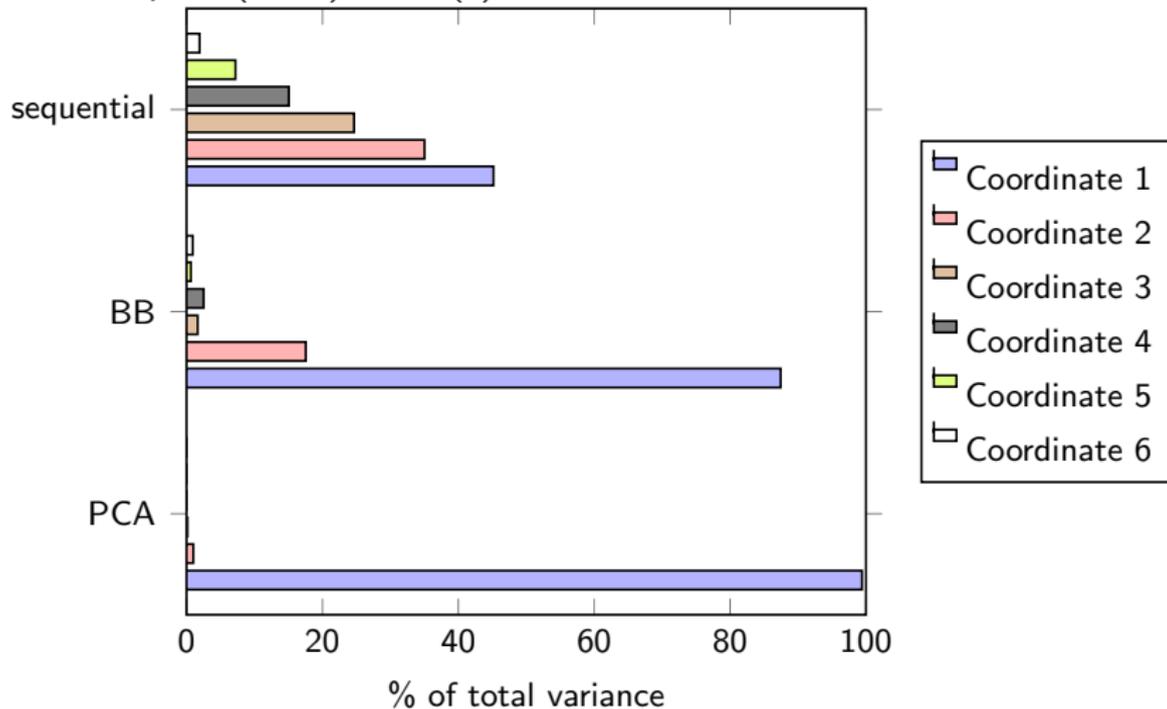
Note: The payoff function is not smooth and also unbounded!

ANOVA Variances for ordinary Asian Option

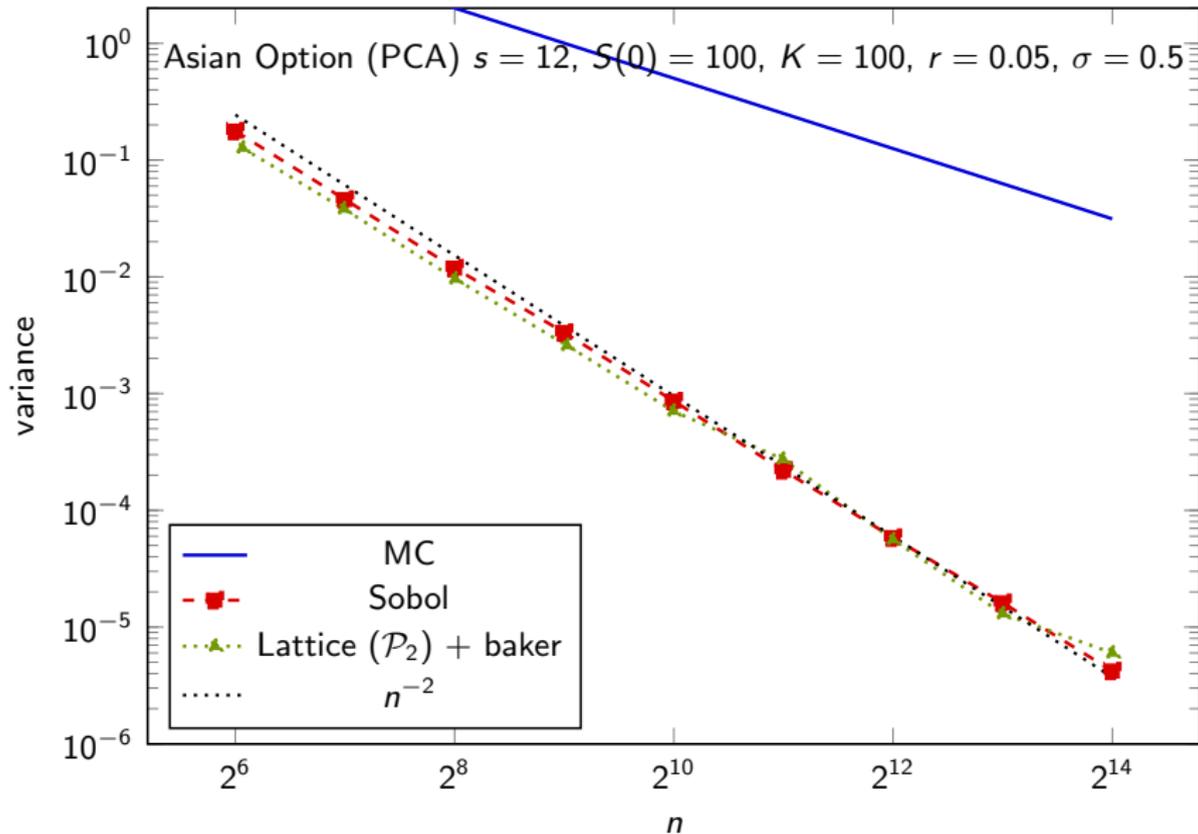


Total Variance per Coordinate for the Asian Option

Asian Option ($s = 6$) with $S(0) = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.5$



Variance with good lattices rules and Sobol points



Asian Option on a Single Asset, with control variate

Let $c = 1$, $S(0) = 100$, $r = \ln(1.09)$, $\sigma_i = 0.2$, $T = 120/365$,
 $t_j = D_1/365 + (T - D_1/365)(j - 1)/(d - 1)$ for $j = 1, \dots, d$,

Asian Option on a Single Asset, with control variate

Let $c = 1$, $S(0) = 100$, $r = \ln(1.09)$, $\sigma_i = 0.2$, $T = 120/365$,
 $t_j = D_1/365 + (T - D_1/365)(j - 1)/(d - 1)$ for $j = 1, \dots, d$,

We estimated the **optimal CV coefficient** by pilot runs for MC and for each combination of sampling scheme, RQMC method, and n .

Asian Option on a Single Asset, with control variate

Let $c = 1$, $S(0) = 100$, $r = \ln(1.09)$, $\sigma_i = 0.2$, $T = 120/365$,
 $t_j = D_1/365 + (T - D_1/365)(j - 1)/(d - 1)$ for $j = 1, \dots, d$,

We estimated the **optimal CV coefficient** by pilot runs for MC and for each combination of sampling scheme, RQMC method, and n .

| d | D_1 | K | μ | σ^2 | VRF of CV |
|-----|-------|-----|--------|------------|--------------------|
| 10 | 111 | 90 | 13.008 | 105 | 1.53×10^6 |
| 10 | 111 | 100 | 5.863 | 61 | 1.07×10^6 |
| 10 | 12 | 90 | 11.367 | 46 | 5400 |
| 10 | 12 | 100 | 3.617 | 23 | 3950 |
| 120 | 1 | 90 | 11.207 | 41 | 5050 |
| 120 | 1 | 100 | 3.367 | 20 | 4100 |

VRFs (per run) for RQMC vs MC, with $n \approx 2^{16}$.

Sequential sampling (left), bridge sampling (middle), and PCA (right).

| d | D_1 | K | P_n | without CV | | | with CV | | |
|-----|-------|-----|---------|------------|--------|--------|---------|-----|------|
| | | | | SEQ | BBS | PCA | SEQ | BBS | PCA |
| 10 | 111 | 90 | Kor+S | 5943 | 6014 | 13751 | 18 | 29 | 291 |
| 10 | 111 | 90 | Kor+S+B | 88927 | 256355 | 563665 | 90 | 177 | 668 |
| 10 | 111 | 90 | Sob+DS | 9572 | 12549 | 14279 | 63 | 183 | 4436 |
| 10 | 12 | 90 | Kor+S | 442 | 1720 | 13790 | 13 | 50 | 71 |
| 10 | 12 | 90 | Kor+S+B | 1394 | 26883 | 446423 | 31 | 66 | 200 |
| 10 | 12 | 90 | Sob+DS | 2205 | 9053 | 12175 | 27 | 67 | 434 |
| 120 | 1 | 90 | Kor+S | 192 | 2025 | 984 | 5 | 47 | 75 |
| 120 | 1 | 90 | Kor+S+B | 394 | 15575 | 474314 | 13 | 55 | 280 |
| 120 | 1 | 90 | Sob+DS | 325 | 7079 | 15101 | 3 | 48 | 483 |

For $d = 10$, Sobol' with PCA combined with CV reduces the variance approximately by a factor of 6.8×10^9 , without increasing the CPU time.

For $d = 120$, PCA is slower than SEQ by a factor of 2 or 3, but worth it.

Array-RQMC for Markov Chains

Setting: A Markov chain with state space $\mathcal{X} \subseteq \mathbb{R}^\ell$, evolves as

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

where the \mathbf{U}_j are i.i.d. uniform r.v.'s over $(0, 1)^d$. Want to estimate

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

Ordinary MC: n i.i.d. realizations of Y . Requires τs uniforms.

Array-RQMC: L., Lécot, Tuffin, et al. [2004, 2006, 2008, etc.]

Simulate an “array” (or population) of n chains in “parallel.”

Goal: Want **small discrepancy** between empirical distribution of states $S_{n,j} = \{X_{0,j}, \dots, X_{n-1,j}\}$ and theoretical distribution of X_j , at each step j .
At each step, use RQMC point set to advance all the chains by one step.

Some RQMC insight: To simplify, suppose $X_j \sim U(0, 1)^\ell$.

We estimate

$$\mu_j = \mathbb{E}[g_j(X_j)] = \mathbb{E}[g_j(\varphi_j(X_{j-1}, \mathbf{U}))] = \int_{[0,1]^{\ell+d}} g_j(\varphi_j(\mathbf{x}, \mathbf{u})) d\mathbf{x}d\mathbf{u}$$

by

$$\hat{\mu}_{\text{arqmc},j,n} = \frac{1}{n} \sum_{i=0}^{n-1} g_j(X_{i,j}) = \frac{1}{n} \sum_{i=0}^{n-1} g_j(\varphi_j(X_{i,j-1}, \mathbf{U}_{i,j})).$$

This is (roughly) RQMC with the point set $Q_n = \{(X_{i,j-1}, \mathbf{U}_{i,j}), 0 \leq i < n\}$.

We want Q_n to have low discrepancy (LD) over $[0, 1]^{\ell+d}$.

Some RQMC insight: To simplify, suppose $X_j \sim U(0, 1)^\ell$.

We estimate

$$\mu_j = \mathbb{E}[g_j(X_j)] = \mathbb{E}[g_j(\varphi_j(X_{j-1}, \mathbf{U}))] = \int_{[0,1]^{\ell+d}} g_j(\varphi_j(\mathbf{x}, \mathbf{u})) d\mathbf{x}d\mathbf{u}$$

by

$$\hat{\mu}_{\text{arqmc},j,n} = \frac{1}{n} \sum_{i=0}^{n-1} g_j(X_{i,j}) = \frac{1}{n} \sum_{i=0}^{n-1} g_j(\varphi_j(X_{i,j-1}, \mathbf{U}_{i,j})).$$

This is (roughly) RQMC with the point set $Q_n = \{(X_{i,j-1}, \mathbf{U}_{i,j}), 0 \leq i < n\}$.

We want Q_n to have low discrepancy (LD) over $[0, 1]^{\ell+d}$.

We do not choose the $X_{i,j-1}$'s in Q_n : they come from the simulation.

We select a LD point set

$$\tilde{Q}_n = \{(\mathbf{w}_0, \mathbf{U}_{0,j}), \dots, (\mathbf{w}_{n-1}, \mathbf{U}_{n-1,j})\},$$

where the $\mathbf{w}_i \in [0, 1)^\ell$ are fixed and each $\mathbf{U}_{i,j} \sim U(0, 1)^d$.

Permute the states $X_{i,j-1}$ so that $X_{\pi_j(i),j-1}$ is "close" to \mathbf{w}_i for each i (LD between the two sets), and compute $X_{i,j} = \varphi_j(X_{\pi_j(i),j-1}, \mathbf{U}_{i,j})$ for each i .

Example: If $\ell = 1$, can take $\mathbf{w}_i = (i + 0.5)/n$ and just sort the states.

For $\ell > 1$, there are various ways to define the matching (multivariate sort).

Array-RQMC algorithm

$X_{i,0} \leftarrow x_0$ (or $X_{i,0} \leftarrow x_{i,0}$) for $i = 0, \dots, n-1$;

for $j = 1, 2, \dots, \tau$ **do**

 Compute the permutation π_j of the states (for matching);

 Randomize afresh $\{\mathbf{U}_{0,j}, \dots, \mathbf{U}_{n-1,j}\}$ in \tilde{Q}_n ;

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

$\hat{\mu}_{\text{arqmc},j,n} = \bar{Y}_{n,j} = \frac{1}{n} \sum_{i=0}^{n-1} g(X_{i,j})$;

Estimate μ by the average $\bar{Y}_n = \hat{\mu}_{\text{arqmc},n} = \sum_{j=1}^{\tau} \hat{\mu}_{\text{arqmc},j,n}$.

Array-RQMC algorithm

$X_{i,0} \leftarrow x_0$ (or $X_{i,0} \leftarrow x_{i,0}$) for $i = 0, \dots, n-1$;

for $j = 1, 2, \dots, \tau$ **do**

 Compute the permutation π_j of the states (for matching);

 Randomize afresh $\{\mathbf{U}_{0,j}, \dots, \mathbf{U}_{n-1,j}\}$ in \tilde{Q}_n ;

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

$\hat{\mu}_{\text{arqmc},j,n} = \bar{Y}_{n,j} = \frac{1}{n} \sum_{i=0}^{n-1} g(X_{i,j})$;

Estimate μ by the average $\bar{Y}_n = \hat{\mu}_{\text{arqmc},n} = \sum_{j=1}^{\tau} \hat{\mu}_{\text{arqmc},j,n}$.

- Proposition:** (i) The average \bar{Y}_n is an **unbiased** estimator of μ .
 (ii) The **empirical variance** of m independent realizations gives an unbiased estimator of $\text{Var}[\bar{Y}_n]$.

Some generalizations

L., Lécot, and Tuffin [2008]: τ can be a random stopping time w.r.t. the filtration $\mathcal{F}\{(j, X_j), j \geq 0\}$.

L., Demers, and Tuffin [2006, 2007]: Combination with splitting techniques (multilevel and without levels), combination with importance sampling and weight windows. Covers particle filters.

L. and Sanvido [2010]: Combination with coupling from the past for exact sampling.

Dion and L. [2010]: Combination with approximate dynamic programming and for optimal stopping problems.

Gerber and Chopin [2015]: Sequential QMC.

Convergence results and applications

L., Lécot, and Tuffin [2006, 2008]: Special cases: convergence at MC rate, one-dimensional, stratification, etc. $\mathcal{O}(n^{-3/2})$ variance.

Lécot and Tuffin [2004]: Deterministic, one-dimension, discrete state.

El Haddad, Lécot, L. [2008, 2010]: Deterministic, multidimensional. $\mathcal{O}(n^{-1/(\ell+1)})$ worst-case error under some conditions.

Fakherredine, El Haddad, Lécot [2012, 2013, 2014]: LHS, stratification, Sudoku sampling, ...

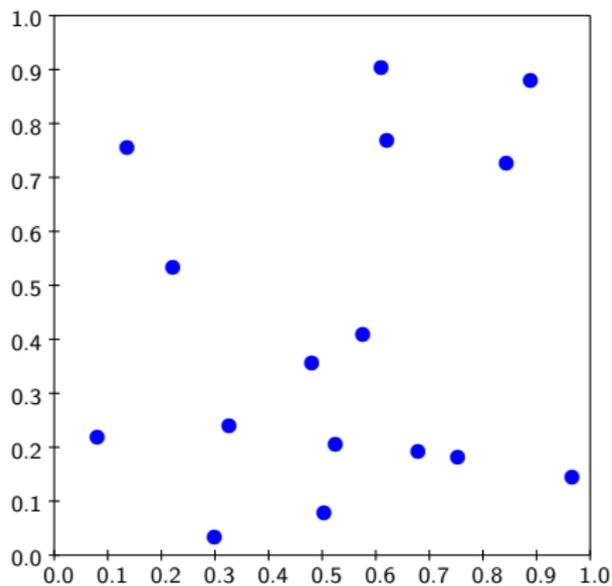
L., Lécot, Munger, and Tuffin [2016]: Survey, comparing sorts, and further examples, some with $\mathcal{O}(n^{-3})$ empirical variance.

Wächter and Keller [2008]: Applications in computer graphics.

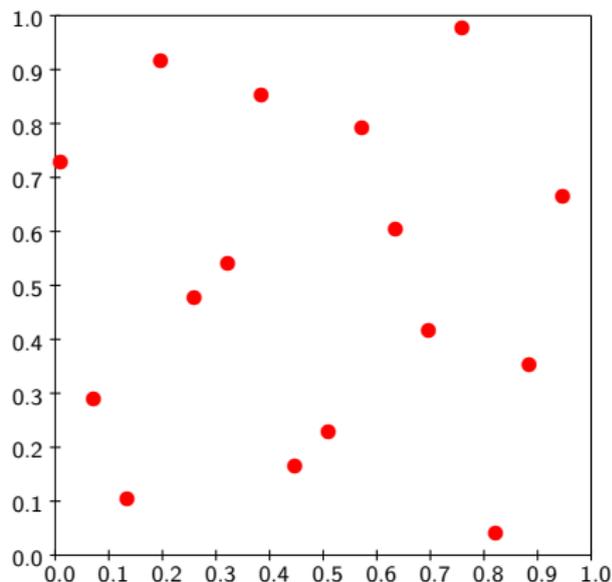
Gerber and Chopin [2015]: Sequential QMC (particle filters), Owen nested scrambling and Hilbert sort. $o(n^{-1})$ variance.

A (4,4) mapping

States of the chains

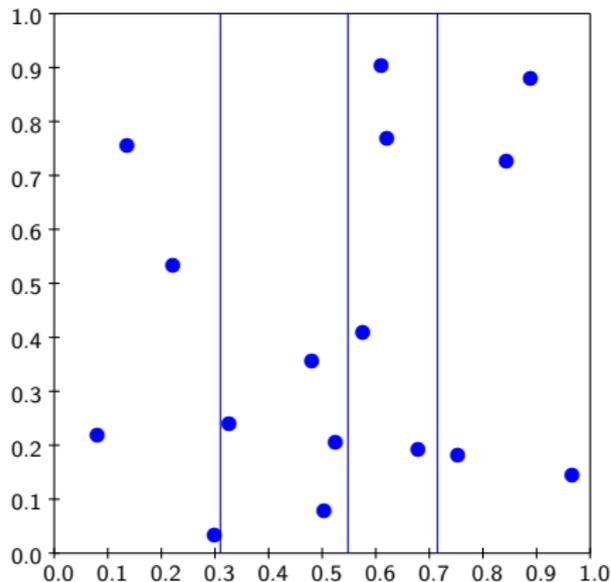


Sobol' net in 2 dimensions after random digital shift

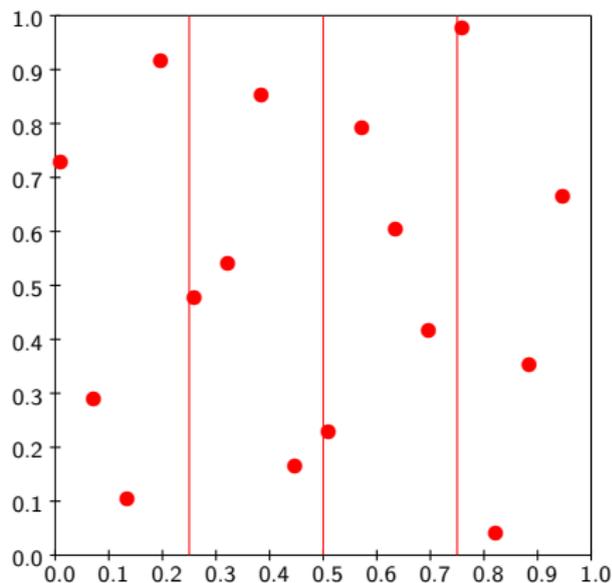


A (4,4) mapping

States of the chains

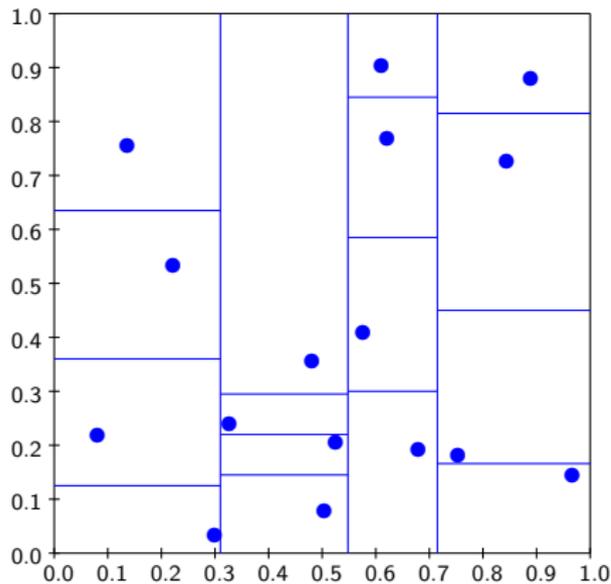


Sobol' net in 2 dimensions after random digital shift

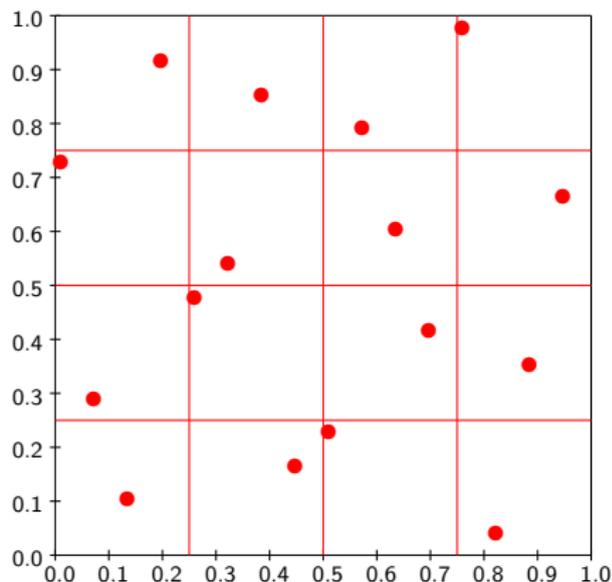


A (4,4) mapping

States of the chains

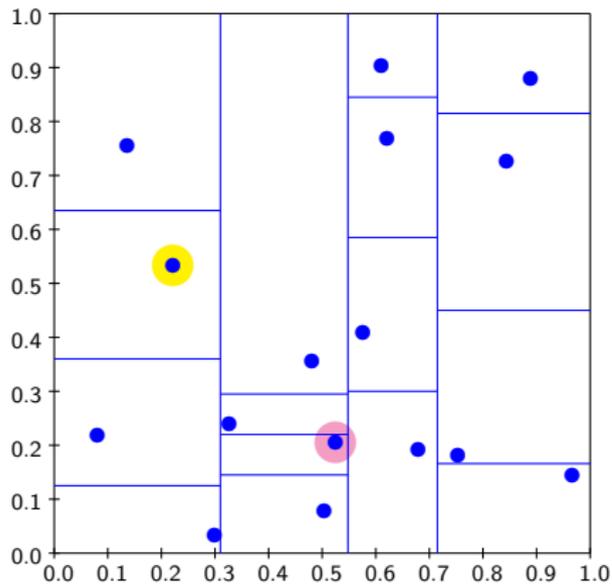


Sobol' net in 2 dimensions after random digital shift

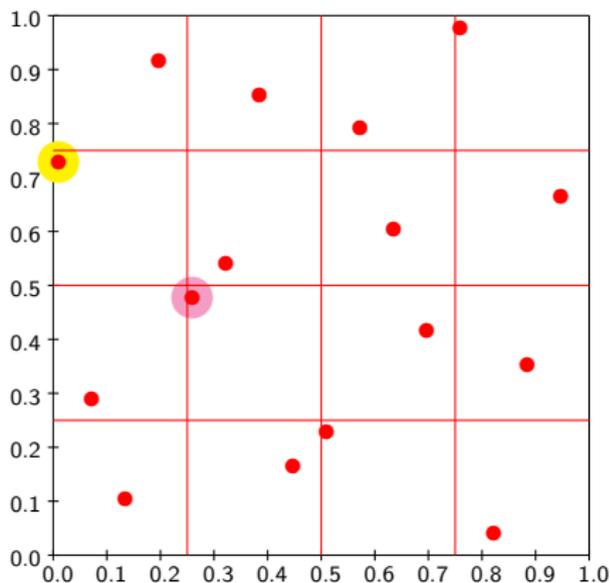


A (4,4) mapping

States of the chains



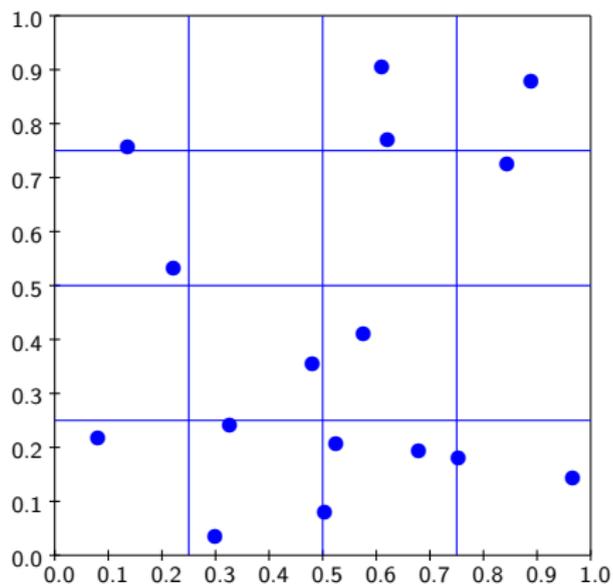
Sobol' net in 2 dimensions after random digital shift



Hilbert curve sort

Map the state to $[0, 1]$, then sort.

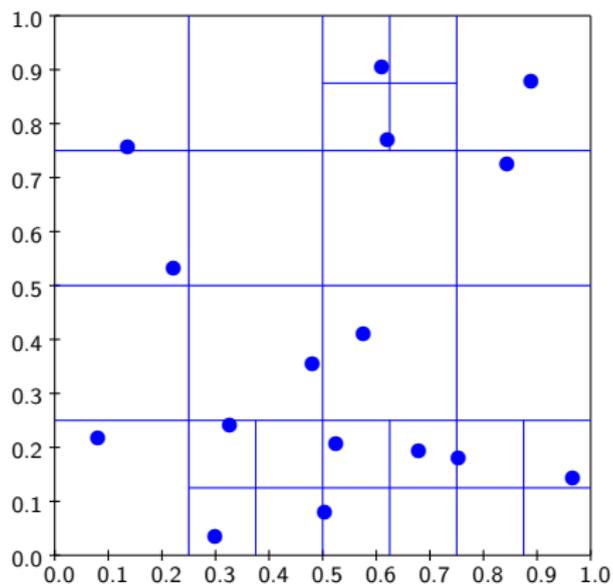
States of the chains



Hilbert curve sort

Map the state to $[0, 1]$, then sort.

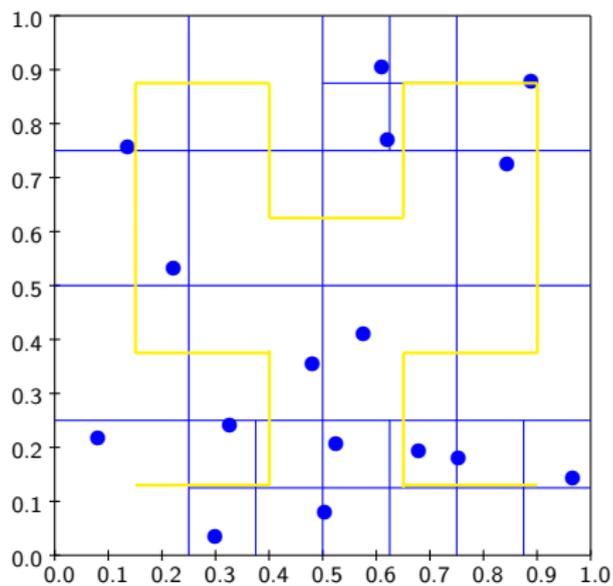
States of the chains



Hilbert curve sort

Map the state to $[0, 1]$, then sort.

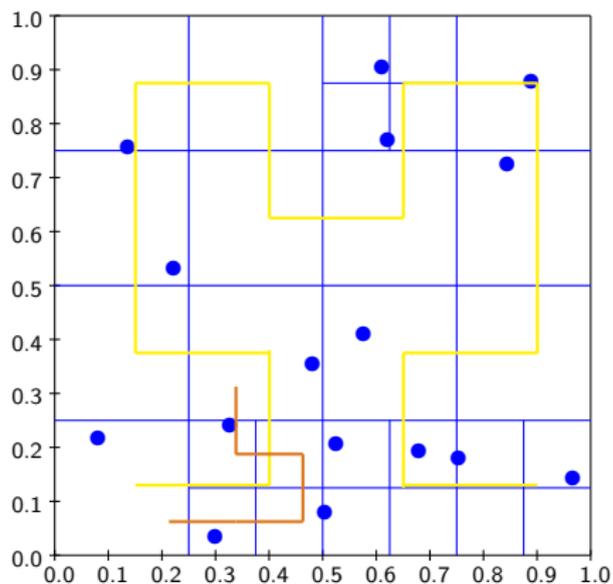
States of the chains



Hilbert curve sort

Map the state to $[0, 1]$, then sort.

States of the chains

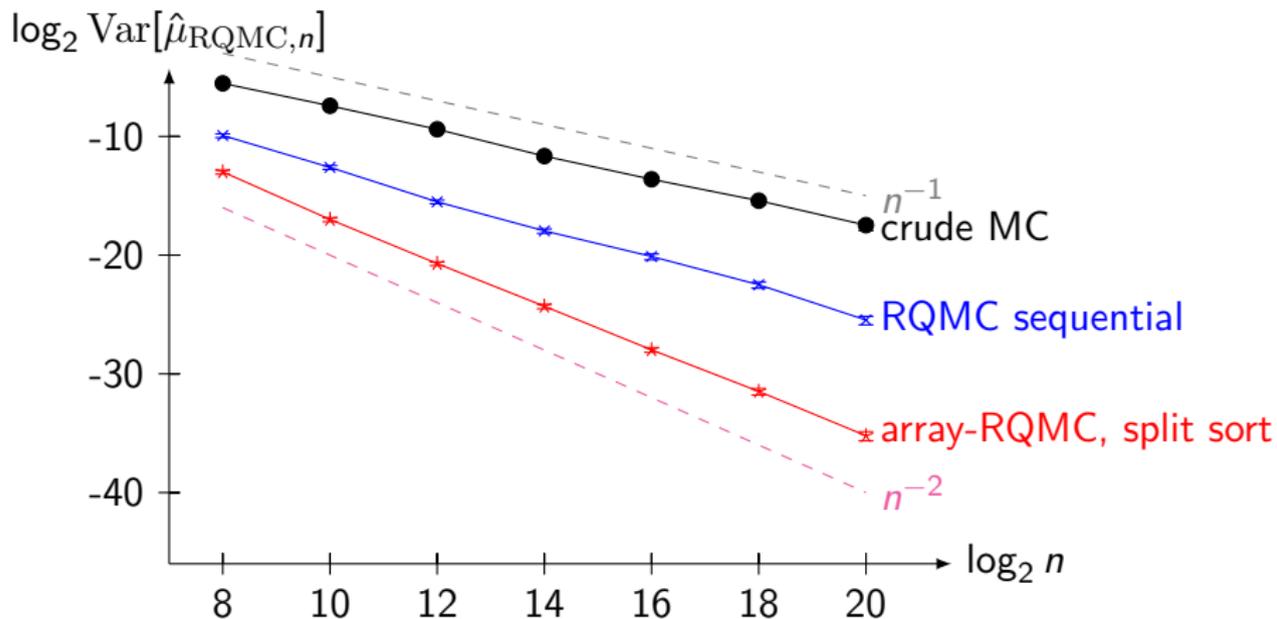


Example: Asian Call Option

$S(0) = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.15$, $t_j = j/52$, $j = 0, \dots, \tau = 13$.

RQMC: Sobol' points with linear scrambling + random digital shift.

Similar results for randomly-shifted lattice + baker's transform.



Example: Asian Call Option

| Sort | RQMC points | $\frac{\log_2 \text{Var}[\bar{Y}_{n,j}]}{\log_2 n}$ | VRF | CPU (sec) |
|--------------------------------|---------------|---|-------------------|-----------|
| Batch sort ($n_1 = n_2$) | SS | -1.38 | 2.0×10^2 | 744 |
| | Sobol | -2.03 | 4.2×10^6 | 532 |
| | Sobol+NUS | -2.03 | 2.8×10^6 | 1035 |
| | Korobov+baker | -2.04 | 4.4×10^6 | 482 |
| Hilbert sort (logistic map) | SS | -1.55 | 2.4×10^3 | 840 |
| | Sobol | -2.03 | 2.6×10^6 | 534 |
| | Sobol+NUS | -2.02 | 2.8×10^6 | 724 |
| | Korobov+baker | -2.01 | 3.3×10^6 | 567 |

VRF for $n = 2^{20}$. CPU time for $m = 100$ replications.

Conclusion, discussion, etc.

- ▶ RQMC can improve the accuracy of estimators considerably in some applications.
- ▶ Cleverly modifying the function f can often bring huge statistical efficiency improvements in simulations with RQMC.
- ▶ There are often many possibilities for how to change f to make it smoother, periodic, and reduce its effective dimension.
- ▶ Point set constructions should be based on discrepancies that take that into account. Can take a weighted average (or worst-case) of uniformity measures over a selected set of projections.
- ▶ Nonlinear functions of expectations: RQMC also reduces the bias.
- ▶ RQMC for density estimation.
- ▶ RQMC for optimization.
- ▶ Array-RQMC for Markov chains. Sequential RQMC. Other QMC methods for Markov chains.
- ▶ Still a lot to learn and do ...

- ▶ *Monte Carlo and Quasi-Monte Carlo Methods 2014, 2012, 2010, ...* Springer-Verlag, Berlin, 2016, 2014, 2012, ...
- ▶ J. Dick and F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, Cambridge, U.K., 2010.
- ▶ P. L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 13(3):307–349, 2009.
- ▶ C. Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer-Verlag, New York, NY, 2009.
- ▶ 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.
- ▶ I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford, 1994.

- ▶ M. Gerber and N. Chopin. Sequential quasi-Monte Carlo. *Journal of the Royal Statistical Society, Series B*, 77(Part 3):509–579, 2015.
- ▶ 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.
- ▶ P. L'Ecuyer, C. Lécot, and A. L'Archevêque-Gaudet. On array-RQMC for Markov chains: Mapping alternatives and convergence rates. *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 485–500, Berlin, 2009. Springer-Verlag.
- ▶ 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.
- ▶ 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*, 2016.
<http://dx.doi.org/10.1016/j.matcom.2016.07.010>.

- ▶ 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.
- ▶ C. Wächter and A. Keller. Efficient simultaneous simulation of Markov chains. *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 669–684, Berlin, 2008. Springer-Verlag.