Monte Carlo and Quasi-Monte Carlo

Want to estimate

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

where $f : [0,1)^s \to \mathbb{R}$ and $\mathbf{U}$ is a uniform r.v. over $[0,1)^s$.

Standard Monte Carlo:

- Generate n independent copies of $\mathbf{U}$, say $\mathbf{U}_1, \ldots, \mathbf{U}_n$;
- estimate μ by $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ where $X_i = f(\mathbf{U}_i)$.

Monte Carlo and Quasi-Monte Carlo

Want to estimate

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

where $f : [0,1)^s \to \mathbb{R}$ and $\mathbf{U}$ is a uniform r.v. over $[0,1)^s$.

Standard Monte Carlo:

- Generate n independent copies of $\mathbf{U}$, say $\mathbf{U}_1, \ldots, \mathbf{U}_n$;
- estimate μ by $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ where $X_i = f(\mathbf{U}_i)$.

Variance: $\mathrm{Var}[\hat{\mu}_n] = \sigma^2/n$ where $\sigma^2 = \mathrm{Var}[f(\mathbf{U})] = \int_{[0,1)^s} f^2(\mathbf{u})\, d\mathbf{u} - \mu$.

Central limit theorem: $\sqrt{n}(\hat{\mu}_n - \mu)/S_n \;\Rightarrow\; \sqrt{n}(\hat{\mu}_n - \mu)/\sigma \;\Rightarrow\; N(0,1)$ when $n \to \infty$, where S_n^2 is any consistent estimator of $\sigma^2 = \mathrm{Var}[f(\mathbf{U})]$.

Quasi-Monte Carlo (QMC)

Replace the random points $\mathbf{U}_i$ by a set of deterministic points $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ that cover $[0,1)^s$ more evenly.
This P_n is called a low-discrepancy point set if some measure of discrepancy between the empirical distribution of P_n and the uniform distribution $\to 0$ faster than for independent random points.

Quasi-Monte Carlo (QMC)

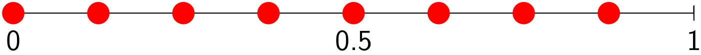
Replace the random points $\mathbf{U}_i$ by a set of deterministic points $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ that cover $[0,1)^s$ more evenly.
This P_n is called a low-discrepancy point set if some measure of discrepancy between the empirical distribution of P_n and the uniform distribution $\to 0$ faster than 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:

$P_n = \mathbb{Z}_n/n = \{0,\, 1/n,\, \ldots, (n-1)/n\}$:



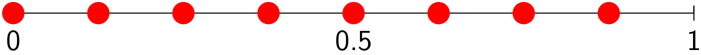
which gives

$$\bar{\mu}_n = \frac{1}{n}\sum_{i=0}^{n-1} f(i/n),$$

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

Obvious solutions:

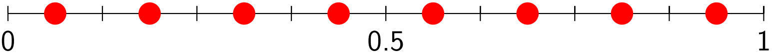
$P_n = \mathbb{Z}_n/n = \{0,\, 1/n,\, \ldots, (n-1)/n\}$:



which gives

$$\bar{\mu}_n = \frac{1}{n}\sum_{i=0}^{n-1} f(i/n),$$

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

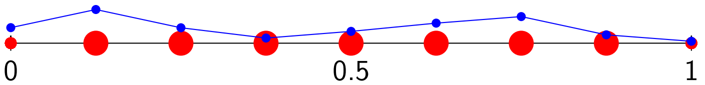


If we allow different weights on the $f(\mathbf{u}_i)$, we have the trapezoidal rule:

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

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

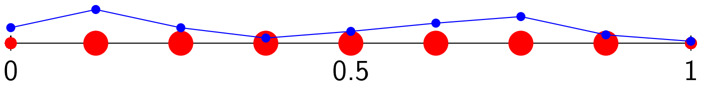
If we allow different weights on the $f(\mathbf{u}_i)$, we have the trapezoidal rule:



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

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

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



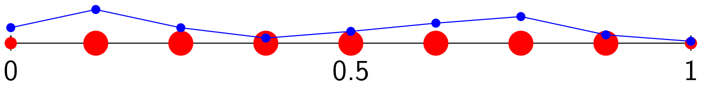
$$\frac{1}{n}\left[\frac{f(0)+f(1)}{2}+\sum_{i=1}^{n-1} f(i/n)\right],$$

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,

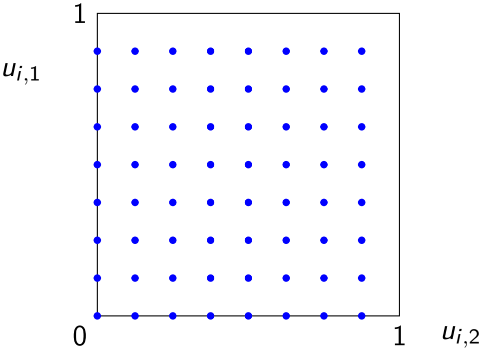
$$\frac{f(0)+4f(1/n)+2f(2/n)+\cdots+2f((n-2)/n)+4f((n-1)/n)+f(1)}{3n},$$

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.

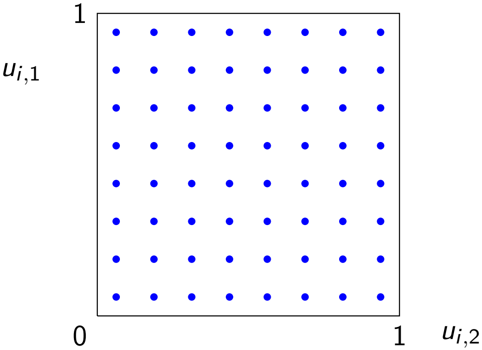
Simplistic solution for $s > 1$: rectangular grid

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



Simplistic solution for $s > 1$: rectangular grid

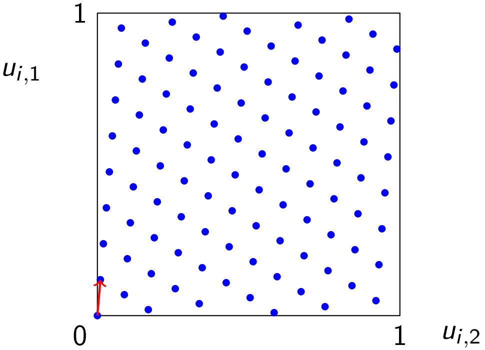
$P_n = \{(i_1/d, \ldots, i_s/d)$ such that $0 \le i_j < d \ \ \forall j\}$ where $n = d^s$.



Quickly becomes impractical when s increases.
And each one-dimensional projection has only d distinct points, each two-dimensional projections has only d^2 distinct points, etc.

Example: lattice with $s = 2$, $n = 101$, $a = 12$

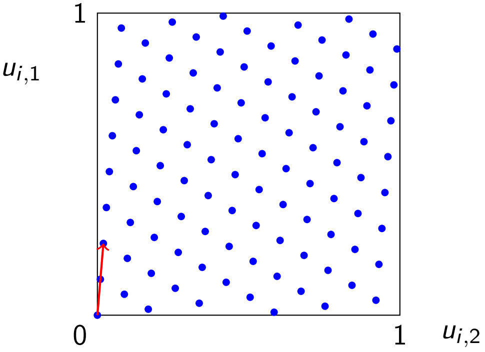
$$\begin{aligned} P_n &= \{(x/m,\, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\ &= \{(0,0),\, (1/101, 12/101), (2/101,\, 43/101), \ldots\}. \end{aligned}$$



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

Example: lattice with $s = 2$, $n = 101$, $a = 12$

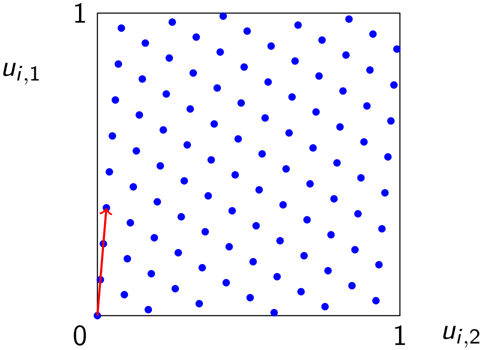
$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(0,0), (1/101, 12/101), (2/101, 43/101), \ldots\}.
\end{aligned}
$$



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

Example: lattice with $s = 2$, $n = 101$, $a = 12$

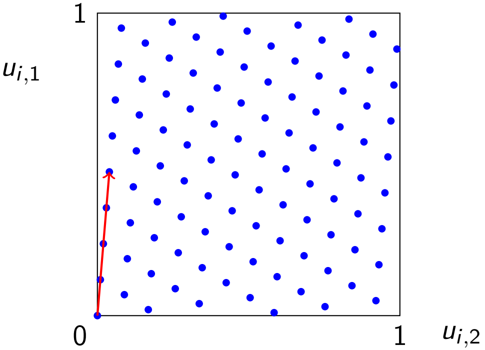
$$\begin{aligned} P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\ &= \{(0,0), (1/101, 12/101), (2/101, 43/101), \ldots\}. \end{aligned}$$



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

Example: lattice with $s = 2$, $n = 101$, $a = 12$

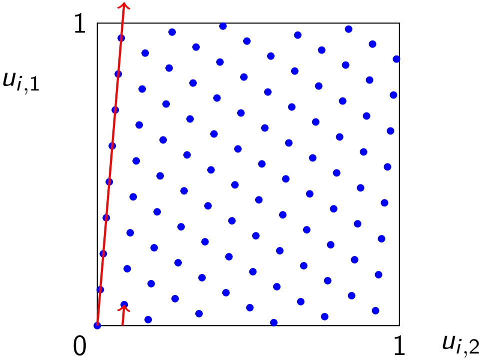
$$\begin{aligned} P_n &= \{(x/m,\ (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\ &= \{(0,0),\ (1/101, 12/101), (2/101,\ 43/101), \ldots\}. \end{aligned}$$



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

Example: lattice with $s = 2$, $n = 101$, $a = 12$

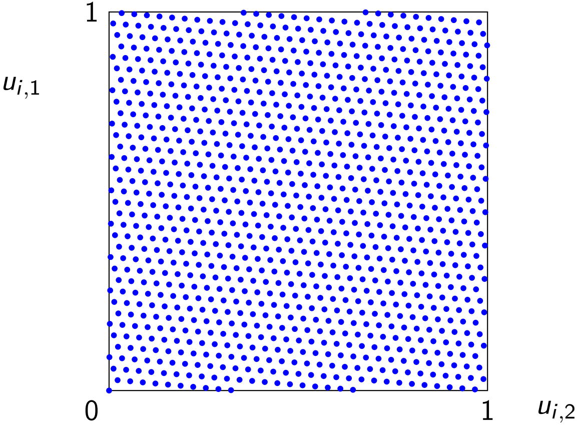
$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(0,0), (1/101, 12/101), (2/101, 43/101), \ldots\}.
\end{aligned}
$$



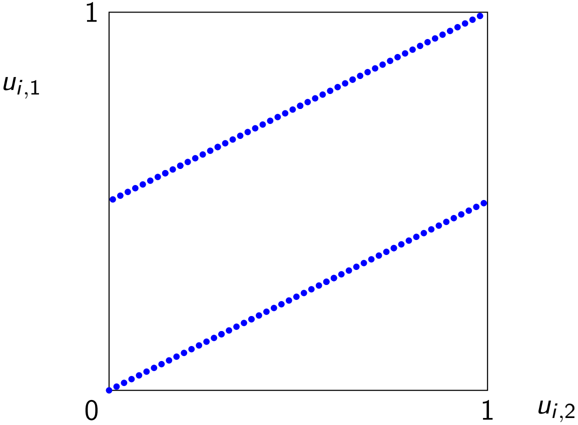
Here, each one-dimensional projection is $\{0,\, 1/n,\, \ldots, (n-1)/n\}$.

Another example: $s = 2$, $n = 1021$, $a = 90$

$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(x/1021, (90x/1021) \bmod 1) : x = 0, \ldots, 1020\}.
\end{aligned}
$$



Example of bad lattice: $s = 2$, $n = 101$, $a = 51$



Good uniformity in one dimension, but not in two!

Error and variance bounds

Koksma-Hlawka-type inequalities (worst-case error):

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

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.

Error and variance bounds

Koksma-Hlawka-type inequalities (worst-case error):

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

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.

"Classical" Koksma-Hlawka (worst-case) inequality for QMC: f must have finite variation in the sense of Hardy and Krause (implies no discontinuity not aligned with the axes), and several known constructions achieve $D(P_n) = O(n^{-1}(\ln n)^s) = O(n^{-1+\delta})$.

For certain Hilbert spaces of smooth functions f with square-integrable partial derivatives of order up to α: $\quad D(P_n) = O(n^{-\alpha+\delta})$.

Randomized quasi-Monte Carlo (RQMC)

An RQMC estimator of μ has the form

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

with $P_n = \{\mathbf{U}_0, \ldots, \mathbf{U}_{n-1}\} \subset (0,1)^s$ 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.

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

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

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

Randomized quasi-Monte Carlo (RQMC)

An RQMC estimator of μ has the form

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

with $P_n = \{\mathbf{U}_0, \ldots, \mathbf{U}_{n-1}\} \subset (0,1)^s$ 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.

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

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

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

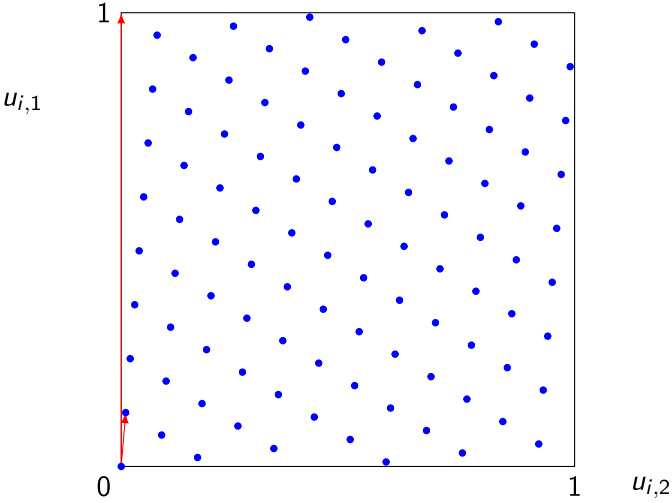
Special cases: antithetic variates ($n = 2$), Latin hypercube sampling (LHS), randomized quasi-Monte Carlo (RQMC).

Can compute m independent realizations $X_1, \ldots, X_m$ of $\hat{\mu}_{n,\mathrm{rqmc}}$, then estimate μ and $\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}]$ 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.

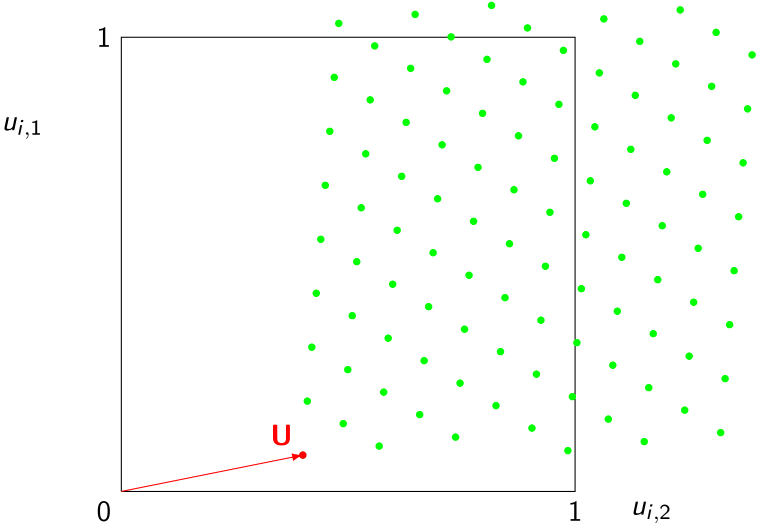
Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $a = 12$



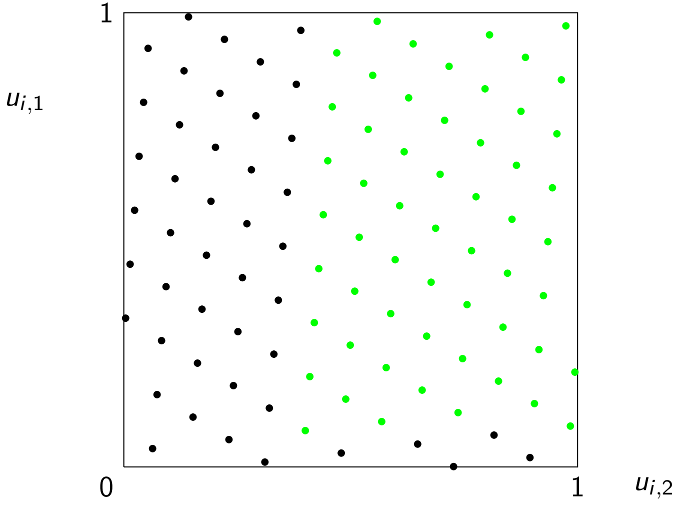
Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $a = 12$



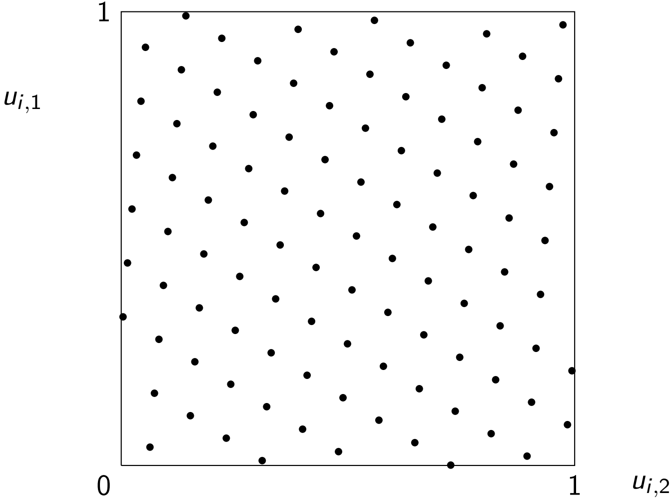
Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $a = 12$



Randomly-Shifted Lattice

Example: lattice with $s = 2$, $n = 101$, $a = 12$



Baby Example: Pricing An Asian Option

Price of a single asset evolves as $\{S(t),\ t \geq 0\}$, and is observed at times $t_1, \ldots, t_d = T$. Discounted payoff:

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

with discount rate r and strike price K. $S(t_1), \ldots, S(t_d)$ are simulated from a vector $\mathbf{U} \sim \text{Uniform}(0,1)^s$ (under risk-neutral measure).

Baby Example: Pricing An Asian Option

Price of a single asset evolves as $\{S(t),\ t \geq 0\}$, and is observed at times $t_1, \ldots, t_d = T$. Discounted payoff:

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

with discount rate r and strike price K. $S(t_1), \ldots, S(t_d)$ are simulated from a vector $\mathbf{U} \sim \text{Uniform}(0,1)^s$ (under risk-neutral measure).

For a simple illustration, if S obeys a geometric Brownian motion, the expected discounted payoff (the option price) is

$$\begin{aligned}
\mu \quad = \quad & e^{-rT} \int_{[0,1)^d} \max\Bigg(0,\ \frac{1}{d}\sum_{i=1}^{d} S(0)\cdot \\
& \exp\left[(r - \sigma^2/2)t_i + \sigma \sum_{j=1}^{i} \sqrt{t_j - t_{j-1}}\,\Phi^{-1}(u_j)\right] - K\Bigg)\, du_1 \ldots du_d.
\end{aligned}$$

Numerical illustration: $s = d = 2$, $T = 1$ (year), $t_j = j/d$, $K = 100$, $s_0 = 100$, $r = 0.05$, $\sigma = 0.5$.

Numerical illustration: $s = d = 2$, $T = 1$ (year), $t_j = j/d$, $K = 100$, $s_0 = 100$, $r = 0.05$, $\sigma = 0.5$.

Exact value: $\mu \approx 17.0958$. Variance with MC: $\mathrm{Var}[f(\mathbf{U})] \approx 934.0$.

Numerical illustration: $s = d = 2$, $T = 1$ (year), $t_j = j/d$, $K = 100$, $s_0 = 100$, $r = 0.05$, $\sigma = 0.5$.

Exact value: $\mu \approx 17.0958$. Variance with MC: $\mathrm{Var}[f(\mathbf{U})] \approx 934.0$.

RQMC (lattice + random shift), n points, $m = 1000$ randomizations.

Asian option with $d = 2$. MC Variance: **934.0**.
For $n = 101$ and $a = 12$: $\bar{X}_m = 17.076$ and $nS_m^2 = \mathbf{77.9}$.
For $n = 65521$ and $a = 944$: $\bar{X}_m = 17.095$ and $nS_m^2 = \mathbf{4.03}$.
Variance reduction (or efficiency improvement) factors: 12 and 232.

Numerical illustration: $s = d = 2$, $T = 1$ (year), $t_j = j/d$, $K = 100$, $s_0 = 100$, $r = 0.05$, $\sigma = 0.5$.

Exact value: $\mu \approx 17.0958$. Variance with MC: $\mathrm{Var}[f(\mathbf{U})] \approx 934.0$.

RQMC (lattice + random shift), n points, $m = 1000$ randomizations.

Asian option with $d = 2$. MC Variance: **934.0**.
For $n = 101$ and $a = 12$: $\bar{X}_m = 17.076$ and $nS_m^2 = \mathbf{77.9}$.
For $n = 65521$ and $a = 944$: $\bar{X}_m = 17.095$ and $nS_m^2 = \mathbf{4.03}$.
Variance reduction (or efficiency improvement) factors: 12 and 232.

In s dimensions, we can take

$$P_n = \{\mathbf{u}_i = (i\mathbf{a} \bmod n)/n : \ i = 0, \dots, n-1\},$$

where $\mathbf{a} = (1,\ a,\ a^2 \bmod n,\ \dots,\ a^{s-1} \bmod n)$.

Numerical illustration: $s = d = 2$, $T = 1$ (year), $t_j = j/d$, $K = 100$, $s_0 = 100$, $r = 0.05$, $\sigma = 0.5$.

Exact value: $\mu \approx 17.0958$. Variance with MC: $\mathrm{Var}[f(\mathbf{U})] \approx 934.0$.

RQMC (lattice + random shift), n points, $m = 1000$ randomizations.

Asian option with $d = 2$. MC Variance: **934.0**.
For $n = 101$ and $a = 12$: $\bar{X}_m = 17.076$ and $nS_m^2 = \mathbf{77.9}$.
For $n = 65521$ and $a = 944$: $\bar{X}_m = 17.095$ and $nS_m^2 = \mathbf{4.03}$.
Variance reduction (or efficiency improvement) factors: 12 and 232.

In s dimensions, we can take

$$P_n = \{\mathbf{u}_i = (i\mathbf{a} \bmod n)/n : i = 0, \ldots, n-1\},$$

where $\mathbf{a} = (1,\ a,\ a^2 \bmod n,\ \ldots,\ a^{s-1} \bmod n)$.

Asian option with $d = 12$. $\mu \approx 13.122$. MC variance: **516.3**.
For $n = 101$: $\bar{X}_m = 13.089$ and $nS_m^2 = \mathbf{94.9}$.
For $n = 65521$: $\bar{X}_m = 13.122$ and $nS_m^2 = \mathbf{23.0}$.
Variance reduction factors: 5 and 22.

Lattice rules

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

Lattice rules

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

Lattice rule of rank 1: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \ldots, n-1$, where $n\mathbf{v}_1 = \mathbf{a} = (a_1, \ldots, a_s) \in \{0, 1, \ldots, n-1\}^s$.
Korobov rule: $\mathbf{a} = (1, a, a^2 \mod n, \ldots)$.

Lattice rules

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

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

Lattice rules

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

Lattice rule of rank 1: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \ldots, n-1$,
where $n\mathbf{v}_1 = \mathbf{a} = (a_1, \ldots, a_s) \in \{0, 1, \ldots, n-1\}^s$.
Korobov rule: $\mathbf{a} = (1, a, a^2 \mod n, \ldots)$.

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

Random shift modulo 1: generate a single point $\mathbf{U}$ uniformly over $(0,1)^s$ and add it to each point of P_n, modulo 1, coordinate-wise:
$\mathbf{U}_i = (\mathbf{u}_i + \mathbf{U}) \bmod 1$. Each $\mathbf{U}_i$ is uniformly distributed over $[0,1)^s$.

Variance expression

Suppose f has Fourier expansion

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

For a randomly shifted lattice, the exact variance is always (see L and Lemieux 2000)

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

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

From the viewpoint of variance reduction, an optimal lattice for f minimizes the square "discrepancy" $D^2(P_n) = \mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}]$.

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

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

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

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

$$\mathcal{P}_\alpha \stackrel{\mathrm{def}}{=} \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} (\max(1, h_1), \ldots, \max(1, h_s))^{-\alpha} = \mathcal{O}(n^{-\alpha+\delta}).$$

This $\mathcal{P}_\alpha$ 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.

This worst-case f is

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

where $B_{\alpha/2}$ is the Bernoulli polynomial of degree $\alpha/2$.
In particular, $B_1(u) = u - 1/2$ and $B_2(u) = u^2 - u + 1/6$.

This worst-case function is not necessarily representative of what happens in applications.

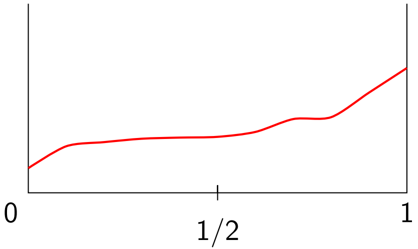
Moreover, the hidden factor in $\mathcal{O}$ increases quickly with s, so this result is not very useful for large s.

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 where RQMC is really effective!

Baker's transformation

To make the periodic continuation of f continuous.

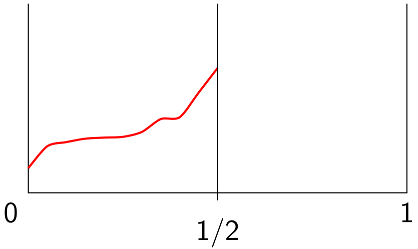
If $f(0) \neq f(1)$, define $\tilde{f}$ by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$.
This $\tilde{f}$ has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's transformation

To make the periodic continuation of f continuous.

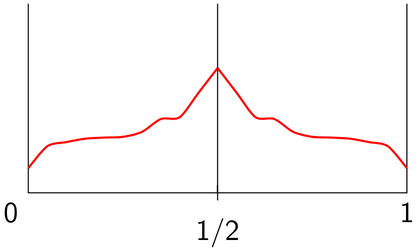
If $f(0) \neq f(1)$, define $\tilde{f}$ by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$.
This $\tilde{f}$ has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's transformation

To make the periodic continuation of f continuous.

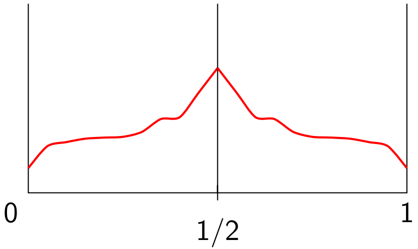
If $f(0) \neq f(1)$, define $\tilde{f}$ by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$.
This $\tilde{f}$ has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's 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 $\tilde{f}(0) = \tilde{f}(1)$.

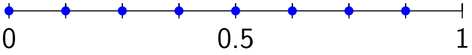


For smooth f, can reduce the variance to $O(n^{-4+\delta})$ (Hickernell 2002).
The resulting $\tilde{f}$ is symmetric with respect to $u = 1/2$.

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

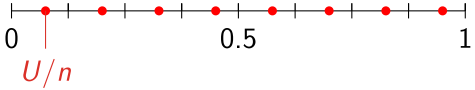
One-dimensional case

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 $\min[2U_j, 2(1-U_j)]$.



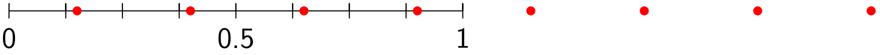
One-dimensional case

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 $\min[2U_j, 2(1 - U_j)]$.



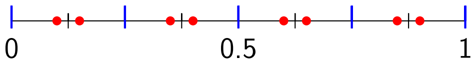
One-dimensional case

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.

Searching for a lattice that minimizes

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

is unpractical, because:

- the Fourier coefficients are usually unknown,
- there are infinitely many,
- must do it for each f.

ANOVA decomposition

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

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

where

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

and the Monte Carlo variance decomposes as

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

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

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

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

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

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

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

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

(finite sum) and the corresponding variation is

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

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

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

This $\mathcal{P}_{\gamma,\alpha}$ is the RQMC variance for the worst-case function

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

(with variation 1), whose square Fourier coefficients are

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

For this function, we have

$$\sigma_\mathfrak{u}^2 = \gamma_\mathfrak{u}\left[\mathrm{Var}[B_{\alpha/2}(U)]\frac{(4\pi^2)^{\alpha/2}}{((\alpha/2)!)^2}\right]^{|\mathfrak{u}|} = \gamma_\mathfrak{u}\left[|B_\alpha(0)|\frac{(4\pi^2)^{\alpha/2}}{(\alpha)!}\right]^{|\mathfrak{u}|}.$$

For $\alpha = 2$, this gives $\gamma_\mathfrak{u} = (3/\pi^2)^{|\mathfrak{u}|}\sigma_\mathfrak{u}^2 \approx (0.30396)^{|\mathfrak{u}|}\sigma_\mathfrak{u}^2$.
For $\alpha = 4$, this gives $\gamma_\mathfrak{u} = [45/\pi^4]^{|\mathfrak{u}|}\sigma_\mathfrak{u}^2 \approx (0.46197)^{|\mathfrak{u}|}\sigma_\mathfrak{u}^2$.
For $\alpha \to \infty$, we have $\gamma_\mathfrak{u} \to (0.5)^{|\mathfrak{u}|}\sigma_\mathfrak{u}^2$.

Note: The correct weights are not proportional to the variances $\sigma_\mathfrak{u}^2$.

Heuristics for choosing the weights

Would like to have $\gamma_{\mathfrak{u}}$ (approx.) proportional to $V^2(f_{\mathfrak{u}})$ for each $\mathfrak{u}$.
For f^*, this gives $\gamma_{\mathfrak{u}} = \rho^{|\mathfrak{u}|}\sigma_{\mathfrak{u}}^2$ for a constant ρ.

One could define a simple parametric model for the square variations and then estimate the parameters by matching the ANOVA variances (e.g., Wang and Sloan 2006, L and Munger 2010).

For example, product weights: $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$ for some constants $\gamma_j \geq 0$.

Order-dependent weights: $\gamma_{\mathfrak{u}}$ depends only on $|\mathfrak{u}|$.
Example: $\gamma_{\mathfrak{u}} = 1$ for $|\mathfrak{u}| \leq d$ and $\gamma_{\mathfrak{u}} = 0$ otherwise.
Wang (2007) suggests this with $d = 2$.

Note that all one-dimensional projections (before random shift) are the same. So the weights $\gamma_{\mathfrak{u}}$ for $|\mathfrak{u}| = 1$ are irrelevant.

More general weighted discrepancy

$$\mathcal{D}_q(P_n) = \left(\sum_{\emptyset \neq \mathfrak{u} \subseteq \{1,\ldots,s\}} [\gamma_{\mathfrak{u}} \, \mathcal{D}_{\mathfrak{u}}(P_n)]^q \right)^{1/q},$$

for any $q > 0$, where $\mathcal{D}_{\mathfrak{u}}(P_n)$ depends only on the projection $P_n(\mathfrak{u})$.

Example: to get $\mathcal{P}_{\gamma,\alpha}$, one would take $q = 2$ and

$$\mathcal{D}_{\mathfrak{u}}^2(P_n) = \sum_{\mathbf{h} \in L_s^* \,:\, \mathfrak{u}(\mathbf{h}) = \mathfrak{u}} (\max(1, |h_1|) \cdots \max(1, |h_s|))^{-\alpha}.$$

Usually, we take $q = 2$.

Weighted $\mathcal{R}_{\gamma,\alpha}$

Take

$$\mathcal{D}_{\mathfrak{u}}^2(P_n) = \frac{1}{n} \sum_{i=0}^{n-1} \prod_{j \in \mathfrak{u}} \left(\sum_{h=-\lfloor (n-1)/2 \rfloor}^{\lfloor n/2 \rfloor} \max(1, |h|)^{-\alpha} e^{2\pi \iota h u_{i,j}} - 1 \right).$$

Upper bounds on $\mathcal{P}_\alpha$ can be computed in terms of $\mathcal{R}_\alpha$.

In contrast to $\mathcal{P}_\alpha$, this one can be computed for any $\alpha > 0$, because the sum is truncated.
We compute it using fast Fourier transforms (FFT).

Figure of merit based on the spectral test

Compute the shortest vector $\ell_{\mathfrak{u}}(P_n)$ in dual lattice for each projection $\mathfrak{u}$ and normalize by an upper bound $\ell^*_{|\mathfrak{u}|}(n)$:

$$\mathcal{D}_{\mathfrak{u}}(P_n) = \frac{\ell^*_{|\mathfrak{u}|}(n)}{\ell_{\mathfrak{u}}(P_n)} \geq 1.$$

$1/\ell_{\mathfrak{u}}(P_n)$ is the distance between hyperplanes that contain all lattice points. We want $\ell_{\mathfrak{u}}(P_n)$ as large as possible.
Computing time of $\ell_{\mathfrak{u}}(P_n)$ is almost independent of n, but exponential in $|\mathfrak{u}|$.

L. and Lemieux (2000), etc., maximize

$$\min_{2\leq r\leq t_1} \frac{\ell_{\{1,\dots,r\}}(P_n)}{\ell^*_r(n)}$$

Figure of merit based on the spectral test

Compute the shortest vector $\ell_{\mathfrak{u}}(P_n)$ in dual lattice for each projection $\mathfrak{u}$ and normalize by an upper bound $\ell^*_{|\mathfrak{u}|}(n)$:

$$\mathcal{D}_{\mathfrak{u}}(P_n) = \frac{\ell^*_{|\mathfrak{u}|}(n)}{\ell_{\mathfrak{u}}(P_n)} \geq 1.$$

$1/\ell_{\mathfrak{u}}(P_n)$ is the distance between hyperplanes that contain all lattice points. We want $\ell_{\mathfrak{u}}(P_n)$ as large as possible.
Computing time of $\ell_{\mathfrak{u}}(P_n)$ is almost independent of n, but exponential in $|\mathfrak{u}|$.

L. and Lemieux (2000), etc., maximize

$$M_{t_1,\ldots,t_d} = \min\left[\min_{2\leq r\leq t_1} \frac{\ell_{\{1,\ldots,r\}}(P_n)}{\ell^*_r(n)},\ \min_{2\leq r\leq d}\ \min_{\substack{\mathfrak{u}=\{j_1,\ldots,j_r\}\subset\{1,\ldots,s\}\\ 1=j_1<\cdots<j_r\leq t_r}} \frac{\ell_{\mathfrak{u}}(P_n)}{\ell^*_r(n)}\right].$$

Search methods for good lattices

Korobov lattices. Search over all admissible a, for $\mathbf{a} = (1, a, a^2, \ldots, ...)$.
Random Korobov. Try r random values of a.

Search methods for good lattices

Korobov lattices. Search over all admissible a, for $\mathbf{a} = (1, a, a^2, \dots, ...)$.
Random Korobov. Try r random values of a.

Rank 1, exhaustive search.
Pure random search. Try admissible vectors $\mathbf{a}$ at random.

Search methods for good lattices

Korobov lattices. Search over all admissible a, for $\mathbf{a} = (1, a, a^2, \ldots, ...)$.
Random Korobov. Try r random values of a.

Rank 1, exhaustive search.
Pure random search. Try admissible vectors $\mathbf{a}$ at random.

Component by component (CBC) construction. (Sloan, Kuo, etc.).
Let $a_1 = 1$;
For $j = 2, 3, \ldots, s$, find $z \in \{1, \ldots, n-1\}$, $\gcd(z, n) = 1$, such that $(a_1, a_2, \ldots, a_j = z)$ minimizes $\mathcal{D}_q(P_n(\{1, \ldots, j\}))$.

Fast CBC construction for $\mathcal{P}_{\gamma,\alpha}$: use FFT. (Nuyens, Cools).

Search methods for good lattices

Korobov lattices. Search over all admissible a, for $\mathbf{a} = (1, a, a^2, \ldots, ...)$.
Random Korobov. Try r random values of a.

Rank 1, exhaustive search.
Pure random search. Try admissible vectors $\mathbf{a}$ at random.

Component by component (CBC) construction. (Sloan, Kuo, etc.).
Let $a_1 = 1$;
For $j = 2, 3, \ldots, s$, find $z \in \{1, \ldots, n-1\}$, $\gcd(z, n) = 1$, such that $(a_1, a_2, \ldots, a_j = z)$ minimizes $\mathcal{D}_q(P_n(\{1, \ldots, j\}))$.

Fast CBC construction for $\mathcal{P}_{\gamma,\alpha}$: use FFT. (Nuyens, Cools).

Randomized CBC construction.
Let $a_1 = 1$;
For $j = 2, \ldots, s$, try r random $z \in \{1, \ldots, n-1\}$, $\gcd(z, n) = 1$, and retain $(a_1, a_2, \ldots, a_j = z)$ that minimizes $\mathcal{D}_q(P_n(\{1, \ldots, j\}))$.

Can add filters to eliminate poor lattices more quickly.

Embedded lattices

$P_{n_1} \subset P_{n_2} \subset \ldots P_{n_m}$ with $n_1 < n_2 < \cdots < n_m$, for some $m > 0$.
Usually: $n_k = b^{c+k}$ for integers $c \geq 0$ and $b \geq 2$, typically with $b = 2$,
$\mathbf{a}_k = \mathbf{a}_{k+1} \bmod n_k$ for all $k < m$, and the same random shift.

We need a measure that accounts for the quality of all m lattices.
We standardize the merit at all levels k so they have a comparable scale:

$$\mathcal{E}_q(P_n) = \mathcal{D}_q(P_n)/D_q(n),$$

where $D_q(n)$ is a normalization factor, e.g., a bound on $\mathcal{D}_q(P_n)$ or a bound on its average over all $(a_1, \ldots, a_s)$ under consideration.
For CBC, we do this for each coordinate $j = 1, \ldots, s$ (replace s by j).
Also used as filters.

Then we can take as a global measure (with sum or max):

$$\left[\bar{\mathcal{E}}_{q,m}(P_{n_1}, \ldots, P_{n_m})\right]^q = \sum_{k=1}^{m} w_k \left[\mathcal{E}_q(P_{n_k})\right]^q .$$

For $\mathcal{P}_{\gamma,\alpha}$, bounds by Sinescu and L'Ecuyer (2012) and Dick et al. (2008).

Existing tools

Construction: Precomputed tables for fixed criteria: Maisonneuve (1972), Sloan and Joe (1994), L. and Lemieux (2000), Kuo (2012), etc.

Nuyens (2012) provides Matlab code for fast-CBC construction of lattice rules based on $\mathcal{P}_{\gamma,\alpha}$, with product and order-dependent weights.

L and Munger [2012] propose Lattice Builder, a general software tool for constructing good lattices.

Use: Software for using (randomized) lattice rules in simulations is also available in many places, includng SSJ.

Lattice Builder

Implemented as C++ library, modular object-oriented design, accessible from a program via API.

Various choices of figures of merit, arbitrary weights, construction methods, etc. Easily extensible.

For better run-time efficiency, uses static polymorphism, via templates, rather than dynamic polymorphism.
Several other techniques to reduce computations and improve speed.

One pre-compiled program with Unix-like command line interface.
Also graphical interface.

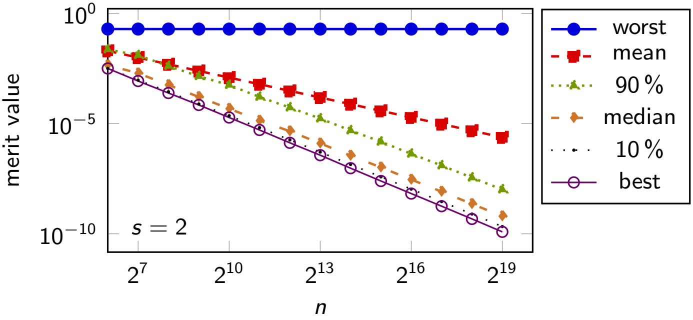
Available for download on GitHub, with source code, documentation, and precompiled executable codes for Linux or Windows, in 32-bit and 64-bit versions.

Show graphical interface

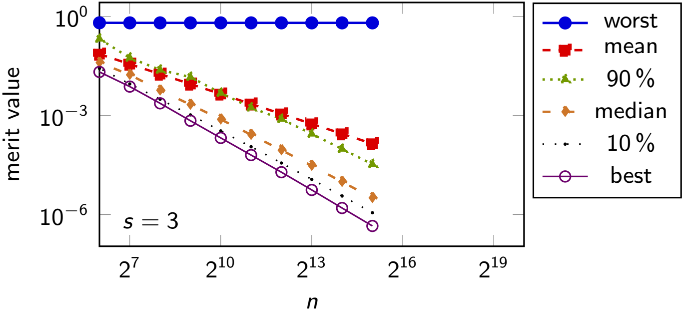
Quantiles of figure of merit

We computed $\mathcal{P}_{\gamma,2}$ with product weights with $\gamma_j^2 = 0.3$ for all j, for all admissible vectors $\mathbf{a} \in \{1\} \times U_n^{s-1}$, for $n = 2^e, \ldots, 2^{19}$.

For $s = 2$, a linear regression of $\log \mathcal{P}_{\gamma,2}$ vs $\log n$ for $2^{12} \le n \le 2^{19}$ gives decreasing rates of $n^{-1.92}$ for the best, and $n^{-1.87}$ and $n^{-1.77}$ for the 10 % and 90 % quantiles. The mean decreases as n^{-1} an the worst-case as n^0 (it is near 0.1948 for all n, obtained with $\mathbf{a} = (1, 1)$).

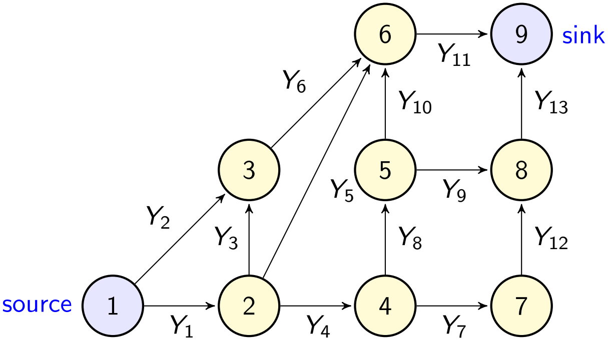


For $s = 3$, a linear regression $\log \mathcal{P}_{\gamma,2}$ vs $\log n$ for $2^{10} \leq n \leq 2^{15}$ gives decreasing rates of $n^{-1.76}$ for the best, and $n^{-1.64}$ and $n^{-1.39}$ for the 10 % and 90 % quantiles.
The mean decreases as n^{-1} an the worst-case as n^0 (it is near 0.6393).



Example: a stochastic activity network

Each arc j has random length $Y_j = F_j^{-1}(U_j)$.
Let $T = f(U_1, \ldots, U_{13})$ = length of longest path from node 1 to node 9.
Want to estimate $q(x) = \mathbb{P}[T > x]$ for a given constant x.



To estimate $q(x)$ by **MC**, we generate n independent realizations of T, say $T_1, \ldots, T_n$, and $(1/n)\sum_{i=1}^{n} \mathbb{I}[T_i > x]$.

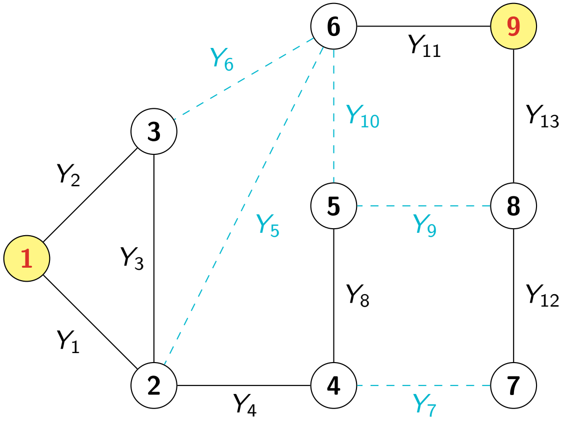
For **RQMC**, we replace the n realizations of $(U_1, \ldots, U_{13})$ by the n points of a randomly-shifted lattice.

To estimate $q(x)$ by **MC**, we generate n independent realizations of T, say $T_1, \ldots, T_n$, and $(1/n)\sum_{i=1}^{n} \mathbb{I}[T_i > x]$.

For **RQMC**, we replace the n realizations of $(U_1, \ldots, U_{13})$ by the n points of a randomly-shifted lattice.

Illustration: $Y_j \sim \text{Normal}(\mu_j, \sigma_j^2)$ for $j = 1, 2, 4, 11, 12$, and $Y_j \sim \text{Exponential}(1/\mu_j)$ otherwise.

The μ_j: 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.


6
9
Y_{11}
Y_6
Y_{10}
Y_{13}
3
Y_2
5
8
Y_5
Y_9
1
Y_3
Y_8
Y_{12}
Y_1
2
4
7
Y_4
Y_7

Conditional Monte Carlo estimator. Generate the Y_j's only for the 8 arcs that do not belong to the cut $\mathcal{L} = \{5, 6, 7, 9, 10\}$, and replace $\mathbb{I}[T > x]$ by its conditional expectation given those Y_j's,

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

This makes the integrand continuous in the U_j's.

Conditional Monte Carlo estimator. Generate the Y_j's only for the 8 arcs that do not belong to the cut $\mathcal{L} = \{5, 6, 7, 9, 10\}$, and replace $\mathbb{I}[T > x]$ by its conditional expectation given those Y_j's,

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

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 $\alpha_l + Y_l + \beta_l \le x$, which occurs with probability $\mathbb{P}[Y_l \le x - \alpha_l - \beta_l] = F_l[x - \alpha_l - \beta_l]$.

Conditional Monte Carlo estimator. Generate the Y_j's only for the 8 arcs that do not belong to the cut $\mathcal{L} = \{5, 6, 7, 9, 10\}$, and replace $\mathbb{I}[T > x]$ by its conditional expectation given those Y_j's,

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

This makes the integrand continuous in the U_j's.

To compute $X_{\rm e}$: 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.

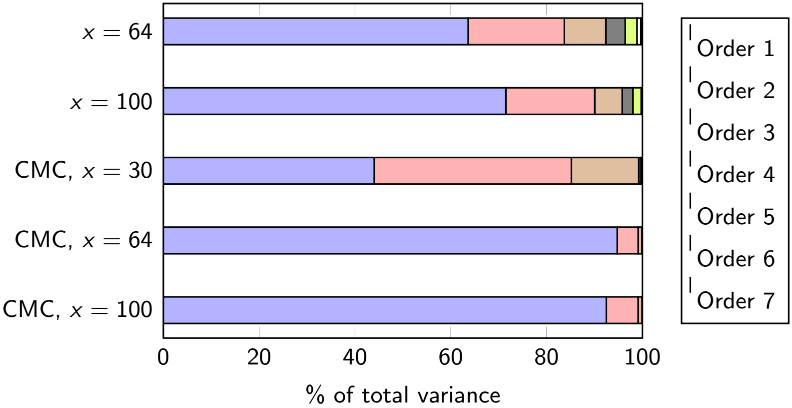
The longest path that passes through link l does not exceed x iff $\alpha_l + Y_l + \beta_l \le x$, which occurs with probability $\mathbb{P}[Y_l \le x - \alpha_l - \beta_l] = F_l[x - \alpha_l - \beta_l]$.

Since the Y_l are independent, we obtain

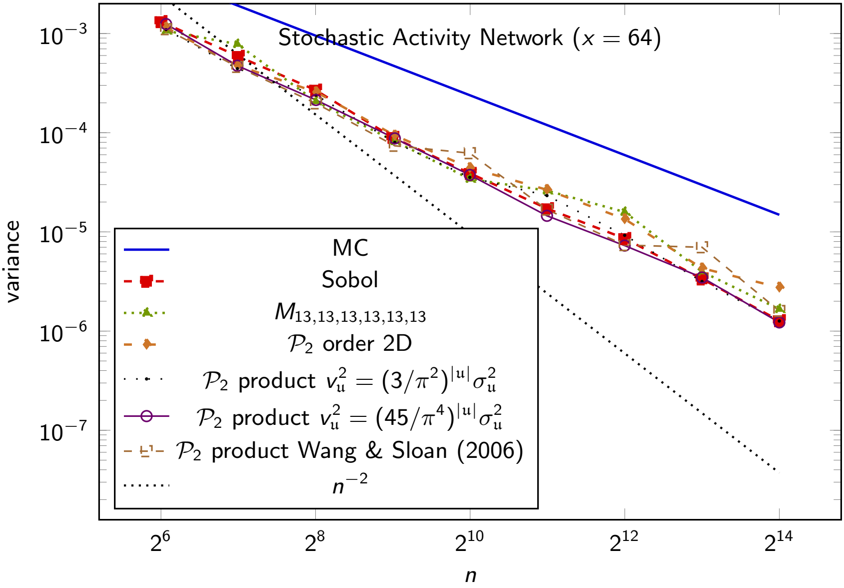
$$X_{\rm e} = 1 - \prod_{l \in \mathcal{L}} F_l[x - \alpha_l - \beta_l].$$

This $X_{\rm e}$ can be faster to compute than X, and it always has less variance.

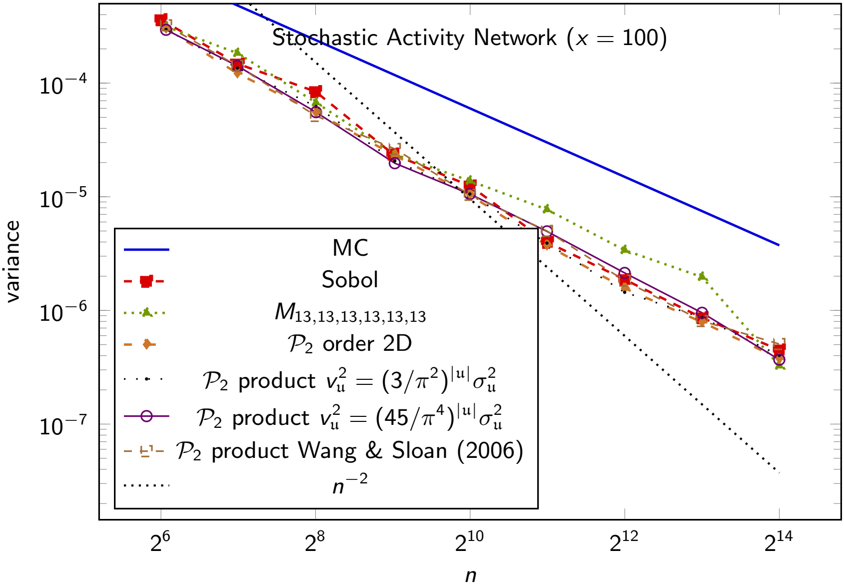
ANOVA Variances for the Stochastic Activity Network



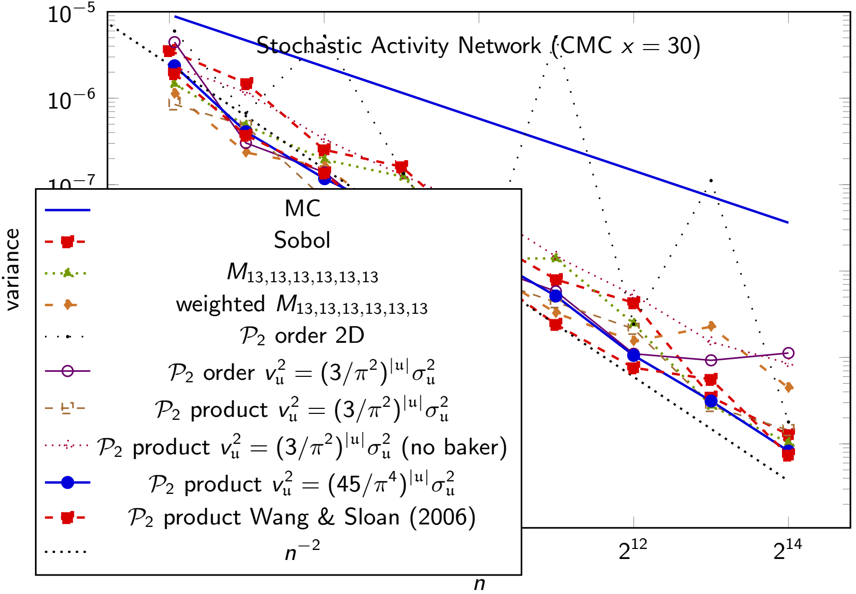
Lattices of Rank 1 with CBC



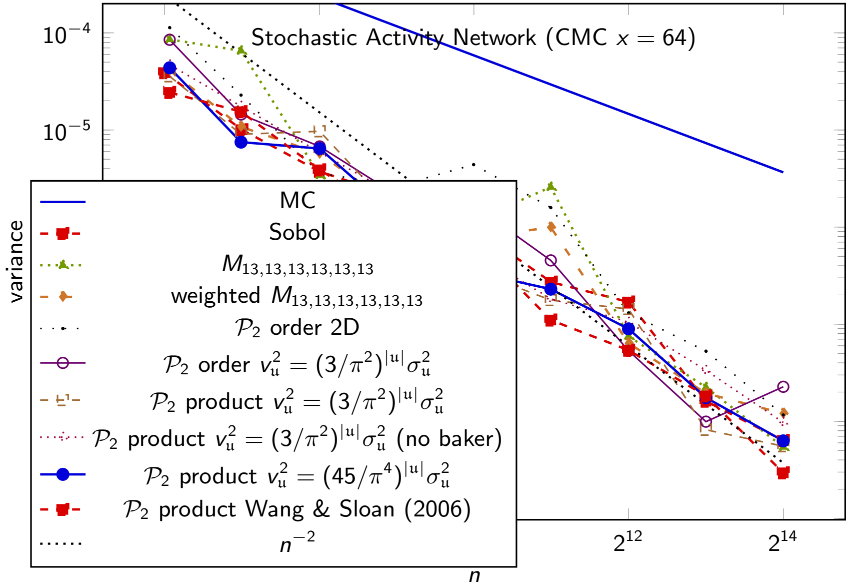
Lattices of Rank 1 with CBC



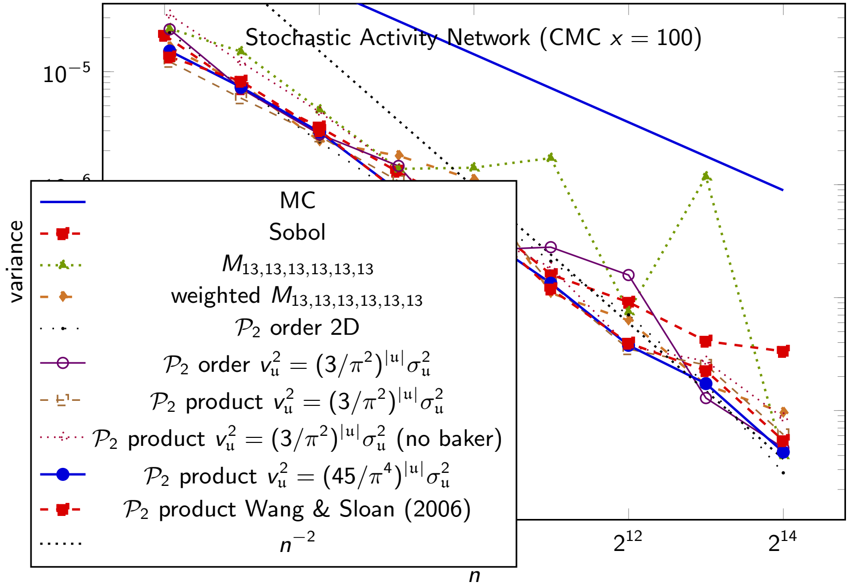
Lattices of Rank 1 with CBC



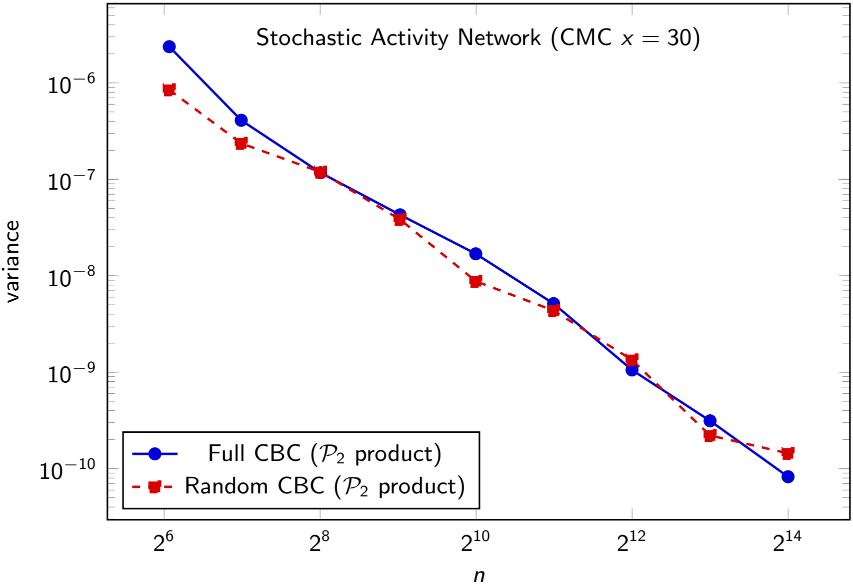
Lattices of Rank 1 with CBC



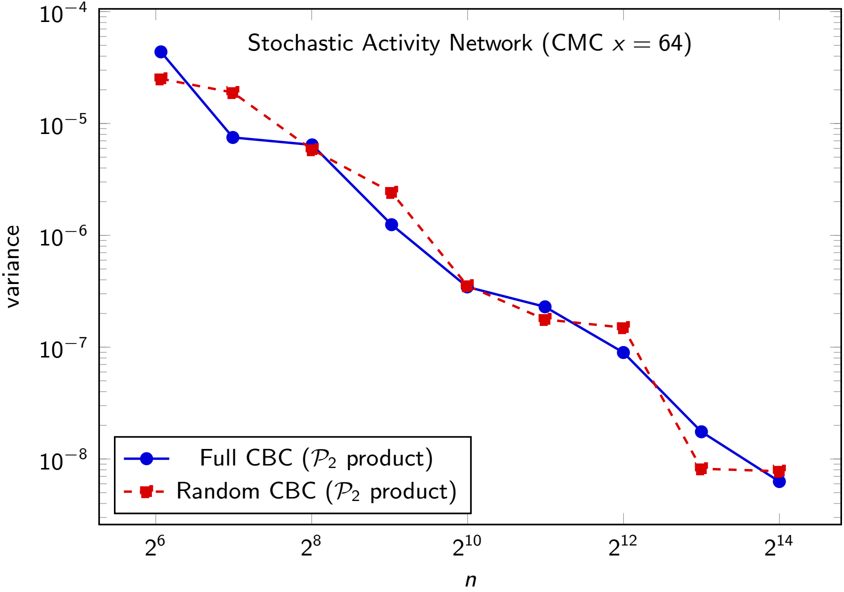
Lattices of Rank 1 with CBC



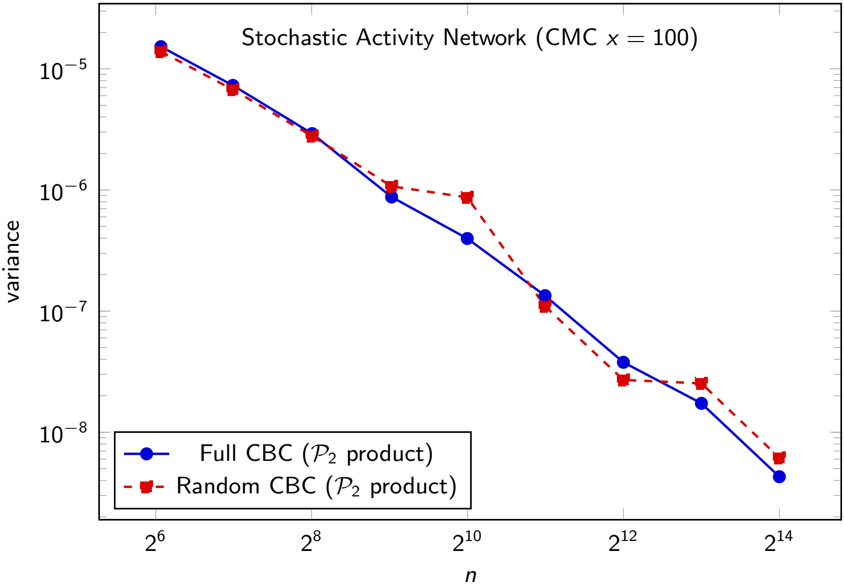
Random vs. Full CBC



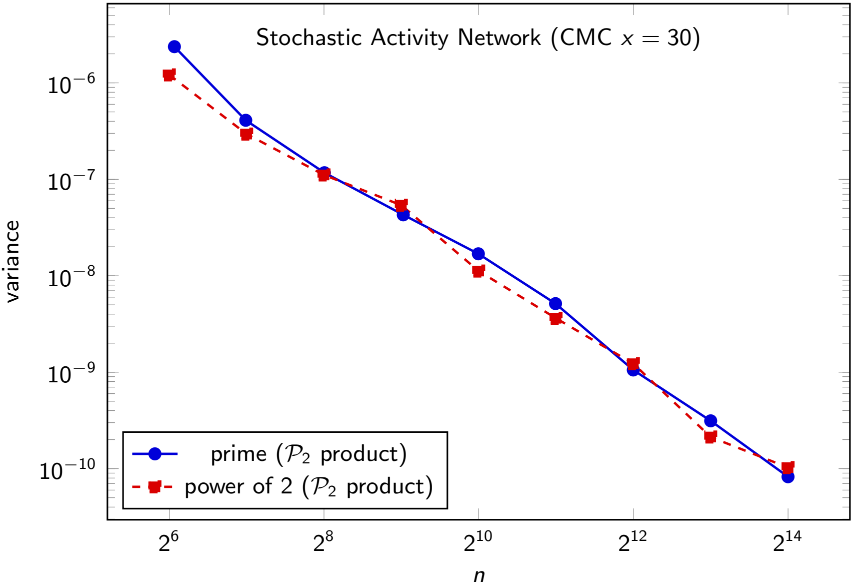
Random vs. Full CBC



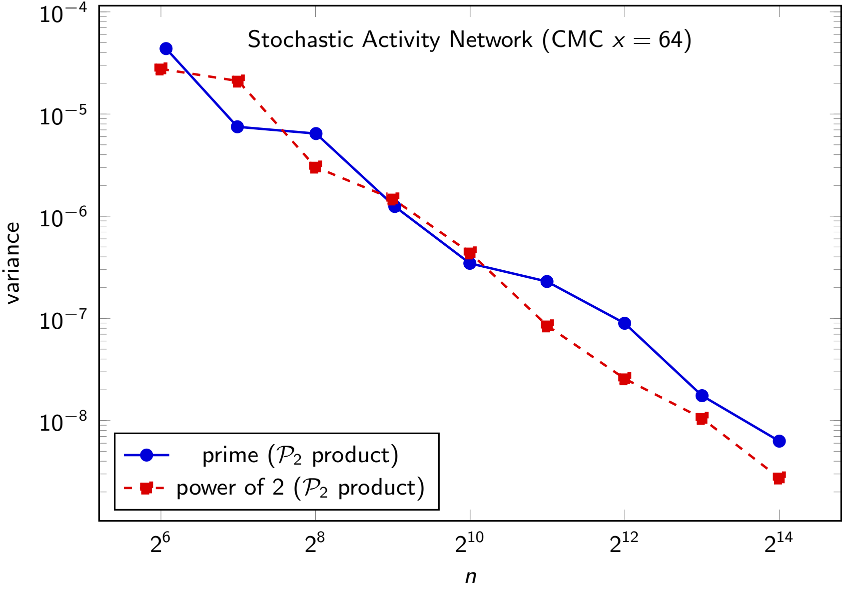
Random vs. Full CBC



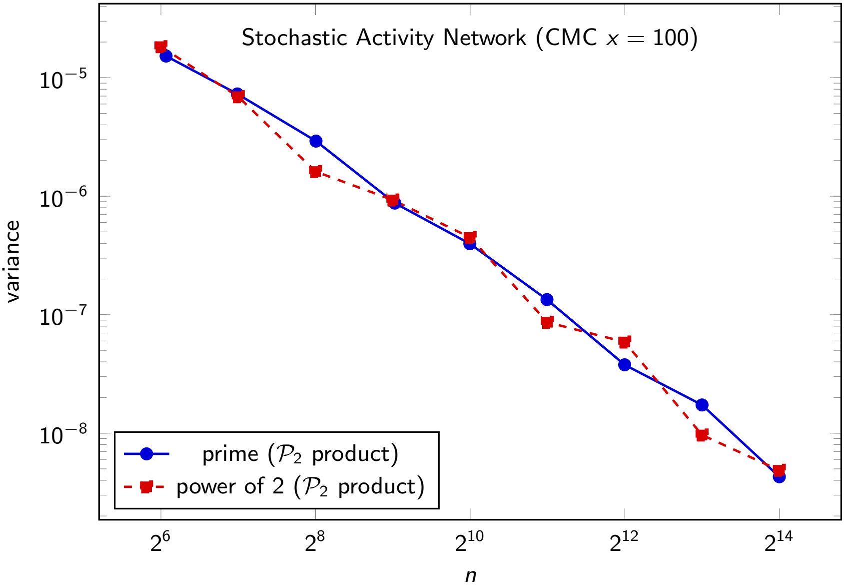
Prime vs. Power-of-2 Number of Points



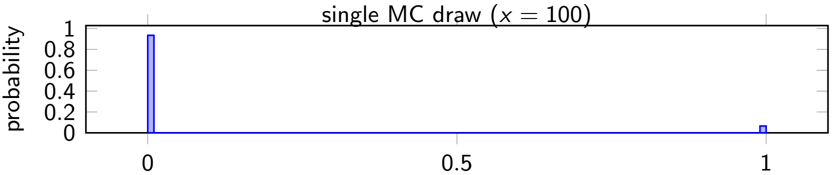
Prime vs. Power-of-2 Number of Points

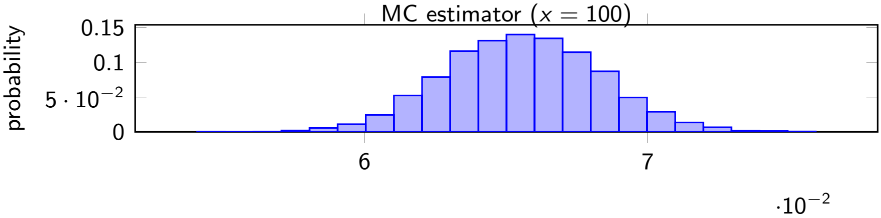


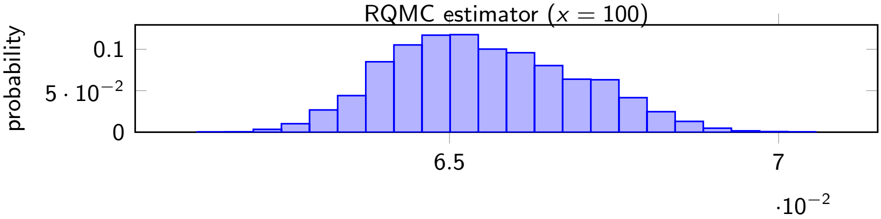
Prime vs. Power-of-2 Number of Points



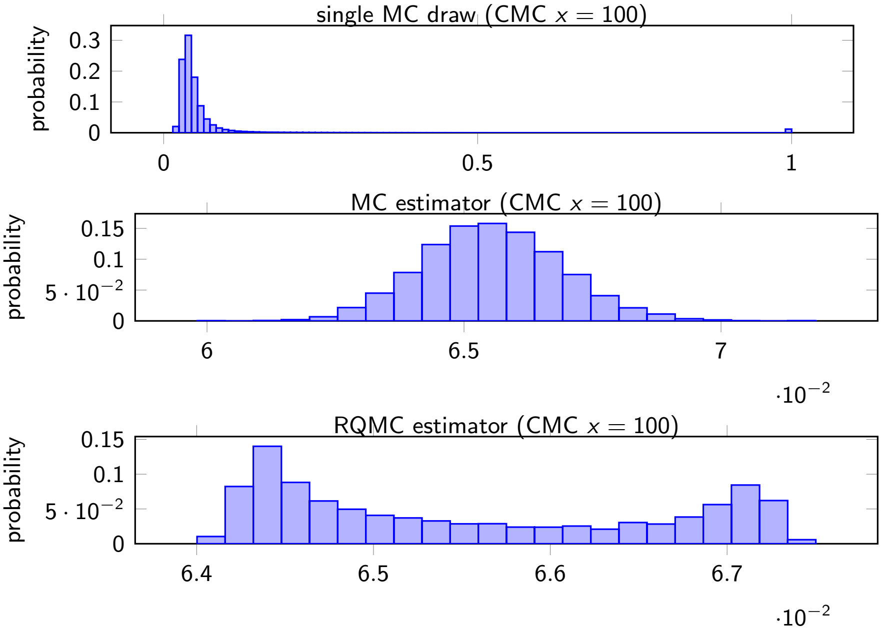
Histograms


single MC draw ($x = 100$)
probability
1
0.8
0.6
0.4
0.2
0
0
0.5
1


MC estimator ($x = 100$)
probability
0.15
0.1
$5 \cdot 10^{-2}$
0
6
7
$\cdot 10^{-2}$


RQMC estimator ($x = 100$)
probability
0.1
$5 \cdot 10^{-2}$
0
6.5
7
$\cdot 10^{-2}$

Histograms


single MC draw (CMC $x = 100$)
probability
0.3
0.2
0.1
0
0
0.5
1
MC estimator (CMC $x = 100$)
probability
0.15
0.1
$5 \cdot 10^{-2}$
0
6
6.5
7
$\cdot 10^{-2}$
RQMC estimator (CMC $x = 100$)
probability
0.15
0.1
$5 \cdot 10^{-2}$
0
6.4
6.5
6.6
6.7
$\cdot 10^{-2}$

Other classes of constructions

Digital nets (Sobol', Faure, Niederreiter, etc.).

Halton sequence, Hammersley point sets.

Etc.

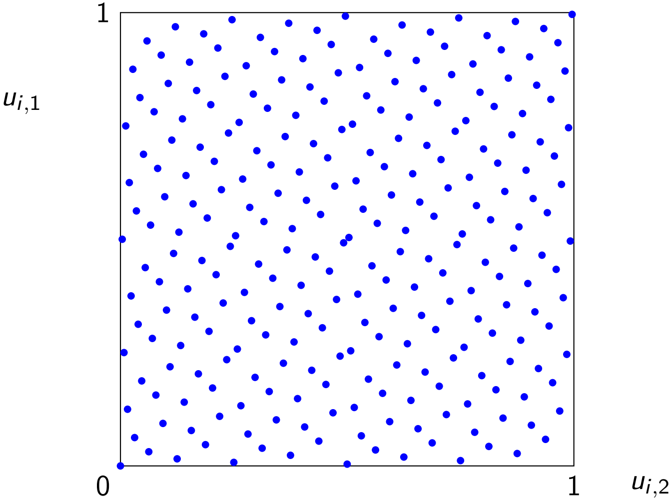
Baby example: Hammersley in two dimensions

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

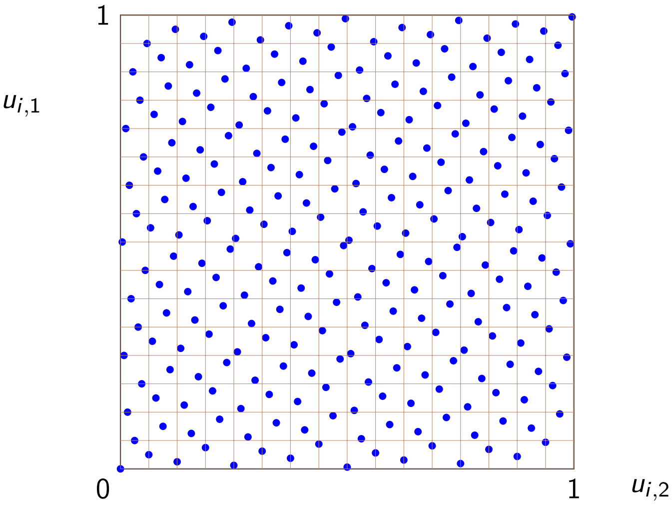
i	$u_{1,i}$	$u_{2,i}$
0	.00000000	.0
1	.00000001	.1
2	.00000010	.01
3	.00000011	.11
4	.00000100	.001
5	.00000101	.101
6	.00000110	.011
$\vdots$	$\vdots$	$\vdots$
254	.11111110	.01111111
255	.11111111	.11111111

Right side: van der Corput sequence in base 2.

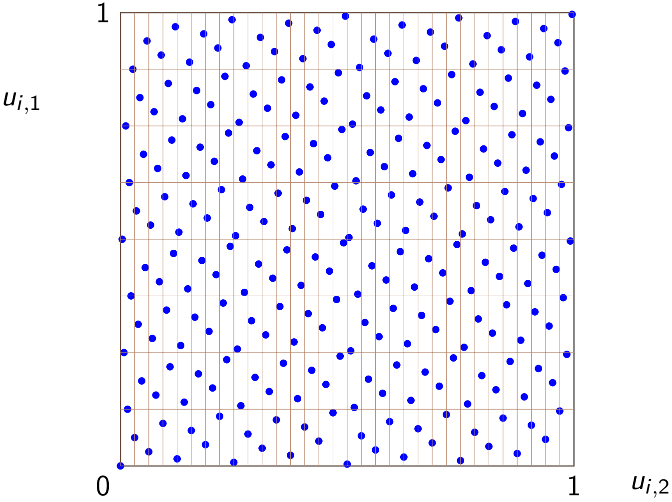
Example: $n = 2^8 = 256$ and $s = 2$.



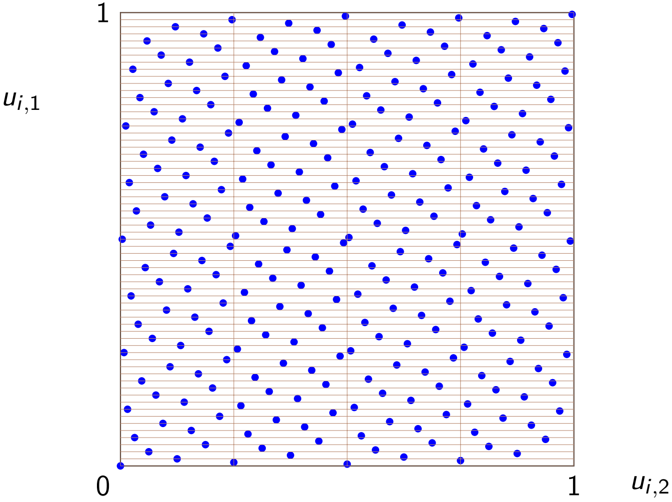
Example: $n = 2^8 = 256$ and $s = 2$.



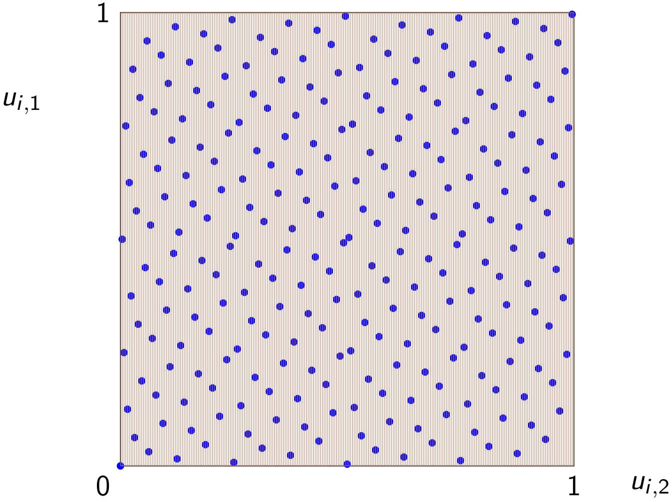
Example: $n = 2^8 = 256$ and $s = 2$.



Example: $n = 2^8 = 256$ and $s = 2$.



Example: $n = 2^8 = 256$ and $s = 2$.



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.

This is a special case of a digital net in base 2.

Generalizes to base $b > 2$.

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

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

Digital net in base b

Gives $n = b^k$ points. For $i = 0, \ldots, b^k - 1$, define:

$$
\begin{aligned}
i &= a_{i,0} + a_{i,1} b + \cdots + a_{i,k-1} b^{k-1}, \\
\begin{pmatrix} u_{i,j,1} \\ u_{i,j,2} \\ \vdots \end{pmatrix} &= \mathbf{C}_j \begin{pmatrix} a_{i,0} \\ \vdots \\ a_{i,k-1} \end{pmatrix} \bmod b, \\
u_{i,j} &= \sum_{\ell=1}^{\infty} u_{i,j,\ell} b^{-\ell}, \qquad \mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s}),
\end{aligned}
$$

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

Gives $n = b^k$ points. For $i = 0, \ldots, b^k - 1$, define:

$$\begin{aligned}
i &= a_{i,0} + a_{i,1} b + \cdots + a_{i,k-1} b^{k-1}, \\
\begin{pmatrix} u_{i,j,1} \\ u_{i,j,2} \\ \vdots \end{pmatrix} &= \mathbf{C}_j \begin{pmatrix} a_{i,0} \\ \vdots \\ a_{i,k-1} \end{pmatrix} \bmod b, \\
u_{i,j} &= \sum_{\ell=1}^{\infty} u_{i,j,\ell} b^{-\ell}, \qquad \mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s}),
\end{aligned}$$

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

Suppose we divide axis j in b^{q_j} equal parts, for each j. This determines a partition of $[0,1)^s$ into $2^{q_1+\cdots+q_s}$ rectangles of equal sizes. If each rectangle contains exactly the same number of points, we say that the point set P_n is $(q_1,\ldots,q_s)$-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 $\mathbf{C}_2$, ..., 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 (q,k,s)-net iff it is $(q_1,\ldots,q_s)$-equidistributed whenever $q_1+\cdots+q_s=k-q$. This is possible only if $b \geq s-1$.

Suppose we divide axis j in b^{q_j} equal parts, for each j. This determines a partition of $[0,1)^s$ into $2^{q_1+\cdots+q_s}$ rectangles of equal sizes. If each rectangle contains exactly the same number of points, we say that the point set P_n is $(q_1,\ldots,q_s)$-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 $\mathbf{C}_2$, ..., 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 (q,k,s)-net iff it is $(q_1,\ldots,q_s)$-equidistributed whenever $q_1+\cdots+q_s = k-q$. This is possible only if $b \geq s-1$.

An infinite sequence $\{\mathbf{u}_0,\mathbf{u}_1,\ldots,\}$ in $[0,1)^s$ is a (q,s)-sequence in base b if for all $k>0$ and $\nu \geq 0$, $Q(k,\nu) = \{\mathbf{u}_i : i = \nu b^k, \ldots, (\nu+1)b^k - 1\}$, is a (q,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 $2^{q_1+\cdots+q_s}$ rectangles of equal sizes. If each rectangle contains exactly the same number of points, we say that the point set P_n is $(q_1,\ldots,q_s)$-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 $\mathbf{C}_2$, ..., 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 (q,k,s)-net iff it is $(q_1,\ldots,q_s)$-equidistributed whenever $q_1+\cdots+q_s = k-q$. This is possible only if $b \geq s-1$.

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

Faure nets and sequences in base b

Faure (1982) proposed the matrices

$$\mathbf{C}_j = \mathbf{P}^j \bmod b = \mathbf{P}\mathbf{C}_{j-1} \bmod b$$

with $\mathbf{C}_0 = \mathbf{I}$ and $\mathbf{P} = (p_{l,c})$ upper triangular where

$$p_{l,c} = \binom{c}{l} = \frac{c!}{l!(c-l)!}$$

for $l \le c$.

Faure nets and sequences in base b

Faure (1982) proposed the matrices

$$\mathbf{C}_j = \mathbf{P}^j \bmod b = \mathbf{P}\mathbf{C}_{j-1} \bmod b$$

with $\mathbf{C}_0 = \mathbf{I}$ and $\mathbf{P} = (p_{l,c})$ upper triangular where

$$p_{l,c} = \binom{c}{l} = \frac{c!}{l!(c-l)!}$$

for $l \le c$.

Faure proved that if b is prime and $b \ge s$, this gives a $(0, s)$-sequence in base b.

Thus, for all $k > 0$ and $\nu \ge 0$, $Q(k, \nu) = \{\mathbf{u}_i : i = \nu b^k, \ldots, (\nu+1)b^k - 1\}$ (which contains $n = b^k$ points) is a $(0, k, s)$-net in base b.

In this set, each coordinate j visits all values in $\{0, 1/n, \ldots, (n-1)/n\}$ once and only once.

If we fix $n = b^k$, we can gain one dimension: $\mathbf{C}_j$ becomes $\mathbf{C}_{j+1}$ for all $j \geq 0$ and we take the reflected identity for $\mathbf{C}_0$ (the first coordinate of each point i is i/n). This point set in $s+1$ dimensions is still a $(0, k, s)$-net in base b.

Sobol' nets and sequences

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

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

Sobol' nets and sequences

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

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

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

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

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 $m_{j,0}, \ldots, m_{j,d_j-1}$ (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 $m_{j,0}, \ldots, m_{j,d_j-1}$ (the first d_j columns).

Then, $m_{j,d_j}, m_{j,d_j+1}, \ldots$ 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 (q, s)-sequence with $q \leq d_0 + \cdots + d_{s-1} + 1 - s$.

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 $m_{j,0}, \ldots, m_{j,d_j-1}$ (the first d_j columns).

Then, $m_{j,d_j}, m_{j,d_j+1}, \ldots$ 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 (q, s)-sequence with $q \leq d_0 + \cdots + d_{s-1} + 1 - s$.

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.

Random digital shift

Equidistribution in digital boxes is lost with random shift modulo 1, but can be kept with a random digital shift in base 2:
Generate one point $\mathbf{U} \sim U(0,1)^s$ and XOR it bitwise with each $\mathbf{u}_i$.

Example for $s = 2$:

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

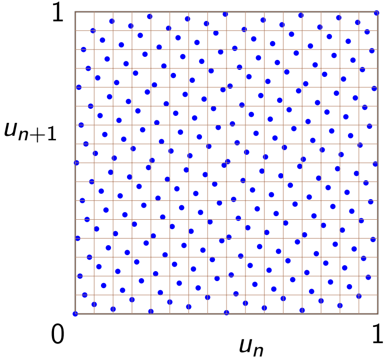
Preservation of the equidistribution:

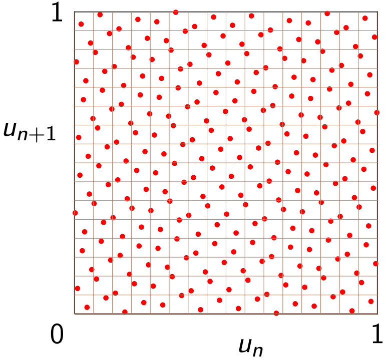
$$\begin{aligned}
\mathbf{u}_i &= (\mathtt{0.***},\ \mathtt{0.*****}) \\
\mathbf{U} &= (\mathtt{0.010},\ \mathtt{0.11101}) \\
\mathbf{u}_i \oplus \mathbf{U} &= (\mathtt{0.***},\ \mathtt{0.*****})
\end{aligned}$$

Example with

$$\begin{aligned}\mathbf{U} &= (0.1270111220,\ 0.3185275653) \\ &= (0.0010000010000011100,\ 0.0101000110001011000)_2.\end{aligned}$$

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





Random digital shift in base b

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

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

For $b = 2$, this is a bitwise XOR.

Proposition. $\tilde{P}_n$ is $(q_1, \ldots, q_s)$-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 of this.

Then we apply a random digital shift in base b.

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 of this.

Then we apply a random digital shift in base b.

Nested uniform scrambling (Owen 1995).
More costly. But provably reduces the variance to $O(n^{-3}(\log n)^s)$ when f is sufficiently smooth!

Example: Same experiment as for the lattice, but with Hammersley points in base 2, with random matrix scramble + random digital shift.

Example: Same experiment as for the lattice, but with Hammersley points in base 2, with random matrix scramble + random digital shift.

Asian option with $s = 2$.
For $n = 2^{10} = 1024$: $\bar{X}_m = 17.096$ and $nS_m^2 = 1.815$.
For $n = 2^{16} = 65536$: $\bar{X}_m = 17.096$ and $nS_m^2 = 0.034$.

MC Variance: 934.0.
With RQMC, work-normalized variance is divided by 515 and 27,120.

Example: Same experiment as for the lattice, but with Hammersley points in base 2, with random matrix scramble + random digital shift.

Asian option with $s = 2$.
For $n = 2^{10} = 1024$: $\bar{X}_m = 17.096$ and $nS_m^2 = 1.815$.
For $n = 2^{16} = 65536$: $\bar{X}_m = 17.096$ and $nS_m^2 = 0.034$.

MC Variance: 934.0.
With RQMC, work-normalized variance is divided by 515 and 27,120.

Asian option with $s = 12$. Sobol' points in 12 dimensions.
For $n = 2^{10} = 1024$: $\bar{X}_r = 13.122$ and $nS_r^2 = 6.2$.
For $n = 2^{16} = 65536$: $\bar{X}_r = 13.122$ and $nS_r^2 = 1.7$.

MC variance: 516.3.
RQMC divides the variance by: 84 and 304.

Effective dimension

A function f has effective dimension d in proportion ρ in the superposition sense (Owen 1998) if

$$\sum_{|\mathfrak{u}|\le d} \sigma_{\mathfrak{u}}^2 \ge \rho\sigma^2.$$

It has effective dimension d in the truncation sense (Caflisch, Morokoff, and Owen 1997) if

$$\sum_{\mathfrak{u}\subseteq\{1,\ldots,d\}} \sigma_{\mathfrak{u}}^2 \ge \rho\sigma^2.$$

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 $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

Example: Function of a Multinormal vector

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

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

Example: Function of a Multinormal vector

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

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

To generate $\mathbf{Y}$: Decompose $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^{\mathrm{t}}$, generate $\mathbf{Z} = (Z_1, \ldots, Z_s) \sim N(\mathbf{0}, \mathbf{I})$ 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 $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \boldsymbol{\Sigma})$.

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

To generate $\mathbf{Y}$: Decompose $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^{\mathrm{t}}$, generate $\mathbf{Z} = (Z_1, \ldots, Z_s) \sim N(\mathbf{0}, \mathbf{I})$ 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 $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

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

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

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 $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (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 $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\boldsymbol{\Sigma}$ 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 $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\boldsymbol{\Sigma}$ 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:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\ t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d = T$.

Principal component decomposition (PCA) (Ackworth et al. 1998):
$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\boldsymbol{\Sigma}$ 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:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\ t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d = T$.

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 $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\boldsymbol{\Sigma}$ 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:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\ t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d = T$.

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 $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (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:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\ t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d = T$.

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 $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\boldsymbol{\Sigma}$ 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:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\ t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d = T$.

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 $(S_1(t), \ldots, S_c(t))$ obeys a geometric Brownian motion.
Then, $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$ 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 $(S_1(t), \ldots, S_c(t))$ obeys a geometric Brownian motion.
Then, $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ and $s = cd$.

Even with Cholesky decompositions of $\boldsymbol{\Sigma}$, 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 $K = 100$.

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 $K = 100$.

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

Korobov Lattice Rules

	$n = 16381$, $a = 5693$		$n = 65521$, $a = 944$		$n = 262139$, $a = 21876$	
Kor+S	18	878	18	1504	9	2643
Kor+S+B	50	4553	46	3657	43	7553

Sobol' Nets

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

An Asian Option on a Single Asset

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 $j = 1, \ldots, d$,

An Asian Option on a Single Asset

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 $j = 1, \ldots, d$,

Kemna and Vorst (1990) suggested using the payoff based on the geometric average as a control variate (CV).

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

An Asian Option on a Single Asset

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 $j = 1, \ldots, d$,

Kemna and Vorst (1990) suggested using the payoff based on the geometric average as a control variate (CV).

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

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.

Asian Option Under a Variance Gamma Process

$S(t)$ = value of a given asset at time t.
VG model (e.g., Madan, Carr, and Chang 1998):

$$S(t) = S(0)\exp\{rt + B(G(t;1,\nu),\theta,\sigma) + \omega t\},$$

where $\omega = \ln(1 - \theta\nu - \sigma^2\nu/2)/\nu$,
B is a Brownian process with drift and variance parameters θ and σ,
G is a gamma process with mean and variance parameters 1 and ν,
B and G are independent.

Asian Option Under a Variance Gamma Process

$S(t)$ = value of a given asset at time t.
VG model (e.g., Madan, Carr, and Chang 1998):

$$S(t) = S(0)\exp\{rt + B(G(t;1,\nu),\theta,\sigma) + \omega t\},$$

where $\omega = \ln(1 - \theta\nu - \sigma^2\nu/2)/\nu$,
B is a Brownian process with drift and variance parameters θ and σ,
G is a gamma process with mean and variance parameters 1 and ν,
B and G are independent.

Asian call option has discounted payoff:

$$e^{-rt}\max\left(0,\ \frac{1}{d}\sum_{j=1}^{d} S(t_j) - K\right).$$

This is an integration problem in $s = 2d$ dimensions.

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1)$,

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),$

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),\ \tau_2 = G(t_2),$

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),\ \tau_2 = G(t_2),\ B(\tau_2),\ \ldots.$

Can be done with either MC or QMC.

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),\ \tau_2 = G(t_2),\ B(\tau_2),\ \ldots.$

Can be done with either MC or QMC.

Bridge sampling (BGBS) (Avramidis, L'Ecuyer, Tremblay 2003):
Generate $\tau_d = G(t_d)$,

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),\ \tau_2 = G(t_2),\ B(\tau_2),\ \ldots.$

Can be done with either MC or QMC.

Bridge sampling (BGBS) (Avramidis, L'Ecuyer, Tremblay 2003):
Generate $\tau_d = G(t_d)$, $B(\tau_d)$,

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),\ \tau_2 = G(t_2),\ B(\tau_2),\ \ldots.$

Can be done with either MC or QMC.

Bridge sampling (BGBS) (Avramidis, L'Ecuyer, Tremblay 2003):
Generate $\tau_d = G(t_d)$, $B(\tau_d)$, $\tau_{d/2} = G(t_{d/2}),\ B(\tau_{d/2})$,

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),\ \tau_2 = G(t_2),\ B(\tau_2),\ \ldots.$

Can be done with either MC or QMC.

Bridge sampling (BGBS) (Avramidis, L'Ecuyer, Tremblay 2003):
Generate $\tau_d = G(t_d)$, $B(\tau_d)$, $\tau_{d/2} = G(t_{d/2})$, $B(\tau_{d/2})$, $\tau_{d/4} = G(t_{d/4})$, $B(\tau_{d/4})$, and so on.

For $t_a < t < t_b$ and $\tau_a < \tau < \tau_b$, the distribution of $G(t)$ conditional on $G(t_a), G(t_b)$ is known (beta) and the distribution of $B(\tau)$ conditional on $B(\tau_a), B(\tau_b)$ is known (normal).

Sequential sampling (BGSS): Generate
$\tau_1 = G(t_1),\ B(\tau_1),\ \tau_2 = G(t_2),\ B(\tau_2),\ \ldots.$

Can be done with either MC or QMC.

Bridge sampling (BGBS) (Avramidis, L'Ecuyer, Tremblay 2003):
Generate $\tau_d = G(t_d)$, $B(\tau_d)$, $\tau_{d/2} = G(t_{d/2})$, $B(\tau_{d/2})$, $\tau_{d/4} = G(t_{d/4})$, $B(\tau_{d/4})$, and so on.

For $t_a < t < t_b$ and $\tau_a < \tau < \tau_b$, the distribution of $G(t)$ conditional on $G(t_a), G(t_b)$ is known (beta) and the distribution of $B(\tau)$ conditional on $B(\tau_a), B(\tau_b)$ is known (normal).

Difference of gammas bridge sampling (DGBS) (Avramidis, L'Ecuyer, Tremblay 2003): Write S as a difference of two gamma processes and use bridge sampling for each.

Let $\theta = -0.1436$, $\sigma = 0.12136$, $\nu = 0.3$, $r = 0.1$, $T = 1$, $d = 32$, $K = 101$, and $S(0) = 100$.

Let $\theta = -0.1436$, $\sigma = 0.12136$, $\nu = 0.3$, $r = 0.1$, $T = 1$, $d = 32$, $K = 101$, and $S(0) = 100$.

Variance reduction factors with:
BGSS (left), BGBS (middle), and DGBS (right):

Korobov Lattice Rules

	$n = 16381$ $a = 5693$			$n = 65521$ $a = 944$			$n = 262139$ $a = 21876$		
Kor+S	17	54	119	24	138	263	22	285	557
Kor+S+B	52	53	57	44	44	433	92	93	1688

Sobol' Nets

	$n = 2^{14}$			$n = 2^{16}$			$n = 2^{18}$		
Sob+S	37	359	585	41	421	1077	75	510	1154
Sob+LMS+S	29	530	557	49	565	995	77	735	1642

Example: Pricing an Asian option

Single asset, s observation times $t_1, \ldots, t_s$. Want to estimate $\mathbb{E}[f(\mathbf{U})]$, where

$$f(\mathbf{U}) = e^{-rt_s} \max\left[0,\ \frac{1}{s}\sum_{j=1}^{s} S(t_j) - K\right]$$

and $\{S(t),\ t \geq 0\}$ is a geometric Brownian motion.
We have $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

Example: Pricing an Asian option

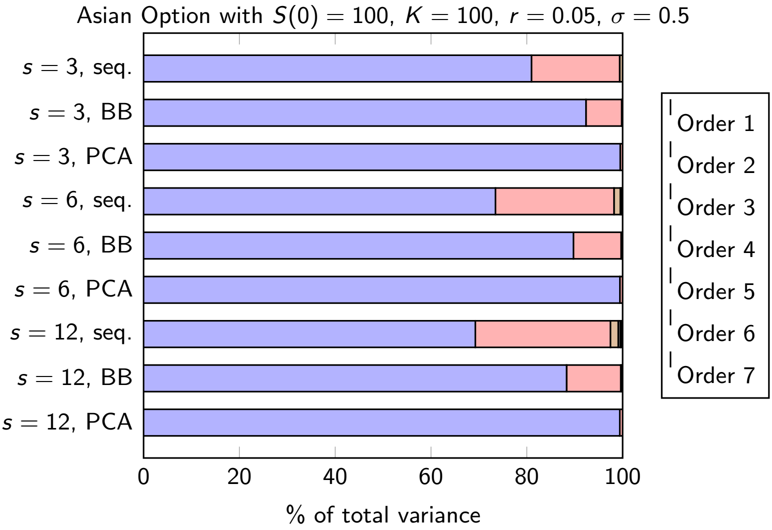
Single asset, s observation times $t_1, \ldots, t_s$. Want to estimate $\mathbb{E}[f(\mathbf{U})]$, where

$$f(\mathbf{U}) = e^{-rt_s} \max\left[0,\ \frac{1}{s}\sum_{j=1}^{s} S(t_j) - K\right]$$

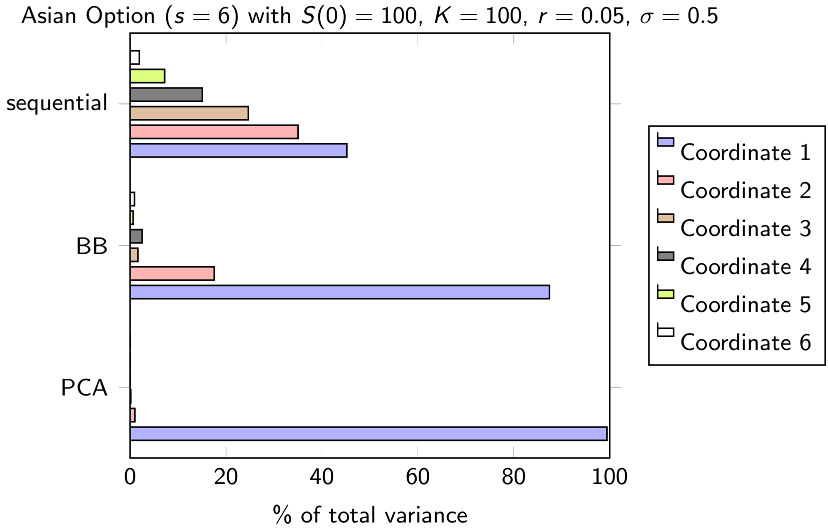
and $\{S(t),\ t \geq 0\}$ is a geometric Brownian motion.
We have $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

Let $S(0) = 100$, $K = 100$, $r = 0.05$, $t_s = 1$, and $t_j = jT/s$ for $1 \leq j \leq s$.
We consider $\sigma = 0.2$, 0.5 and $s = 3, 6, 12$.

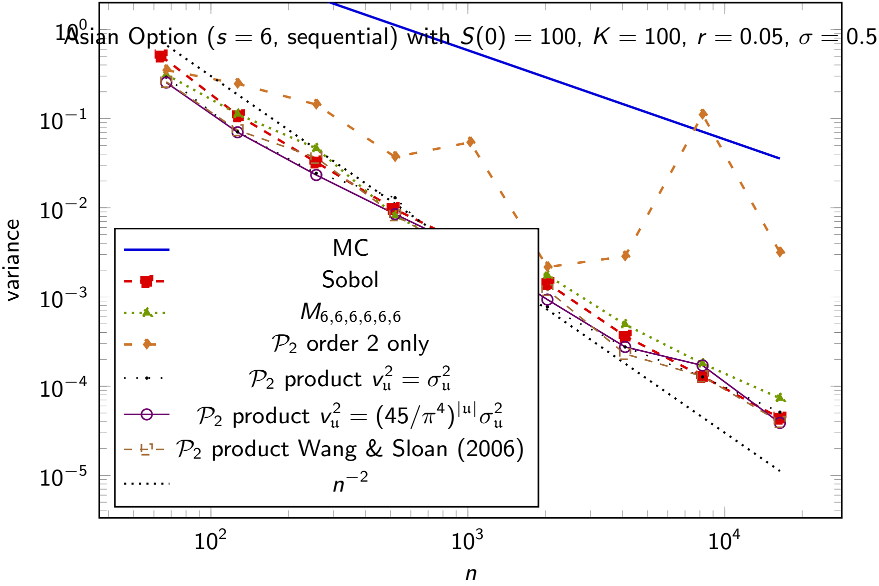
ANOVA Variances for the Asian Option



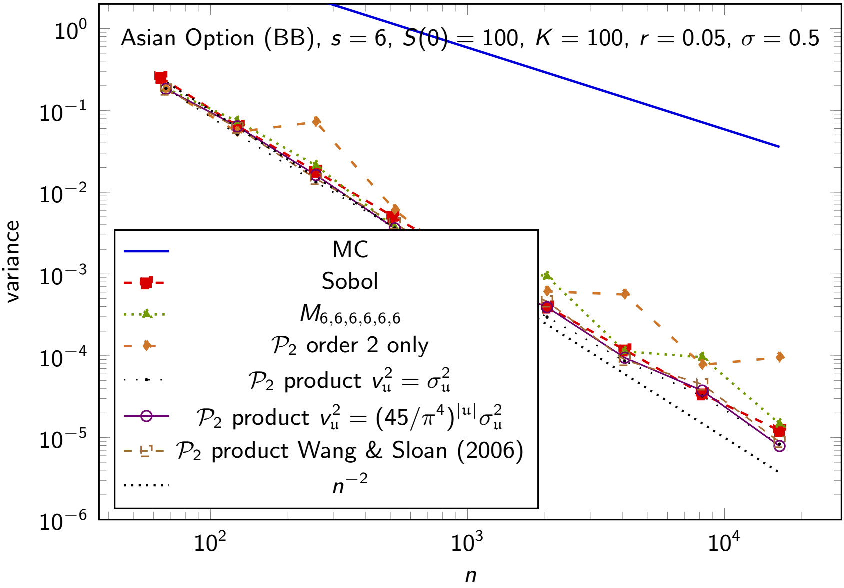
Total Variance per Coordinate for the Asian Option



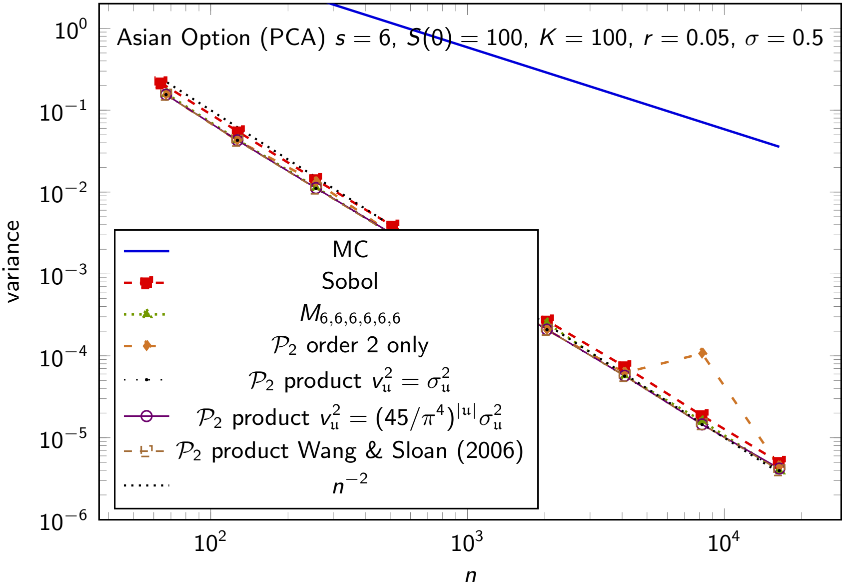
Lattices of Rank 1 with CBC



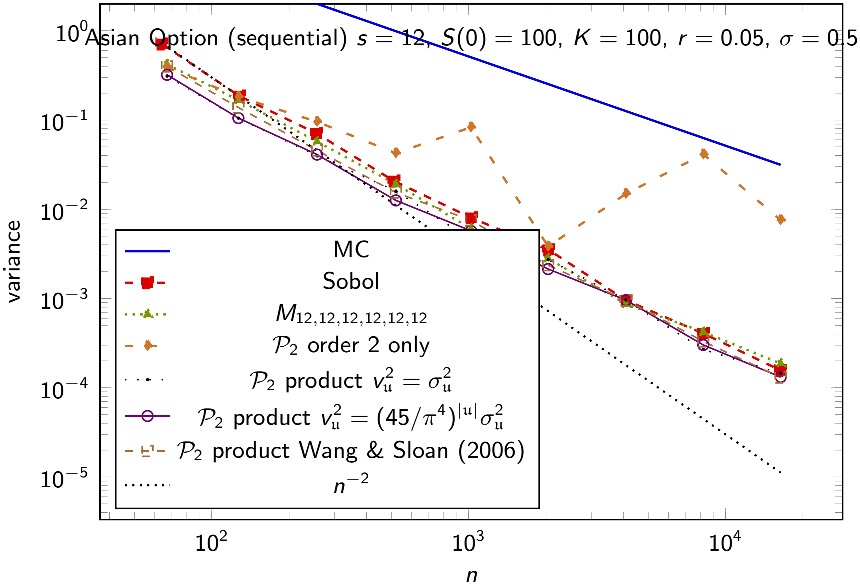
Lattices of Rank 1 with CBC



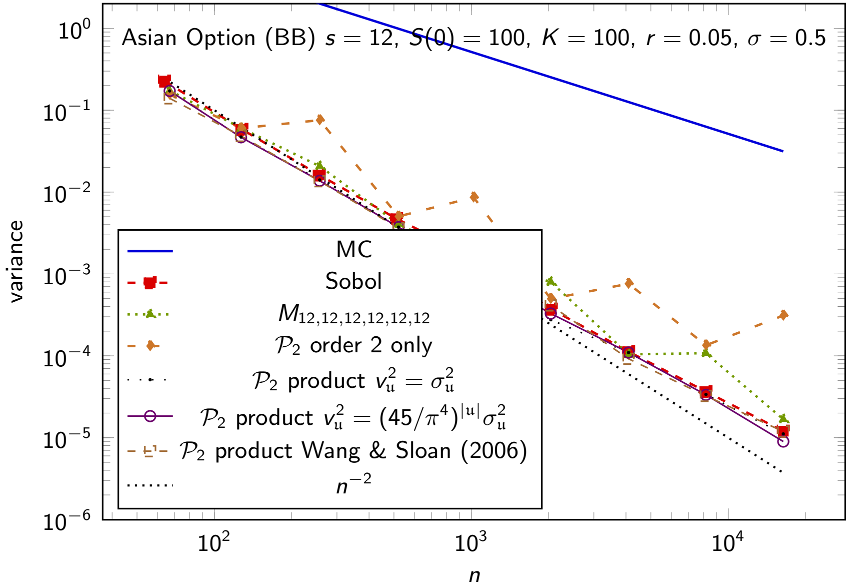
Lattices of Rank 1 with CBC



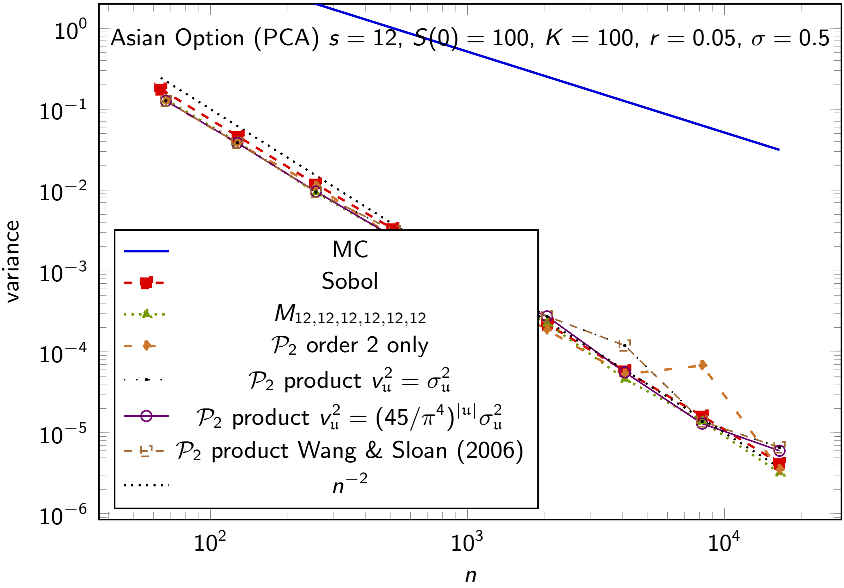
Lattices of Rank 1 with CBC



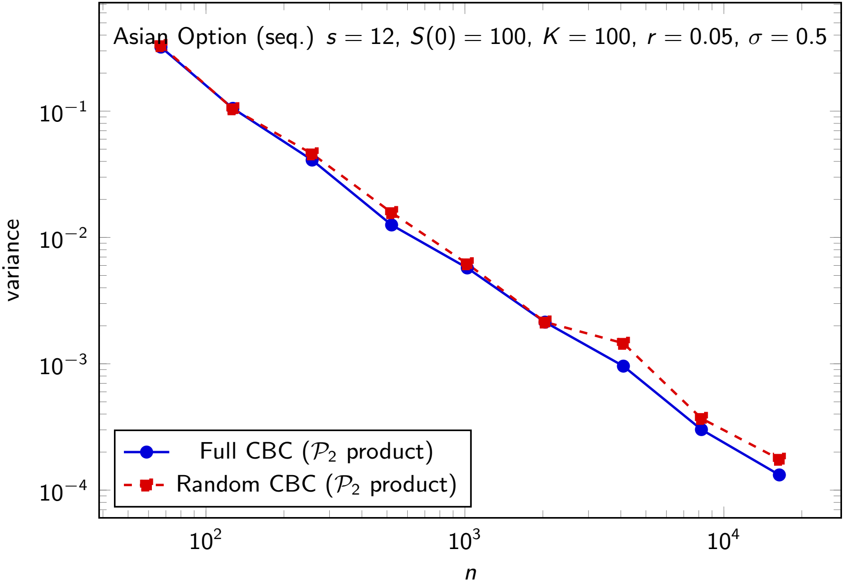
Lattices of Rank 1 with CBC



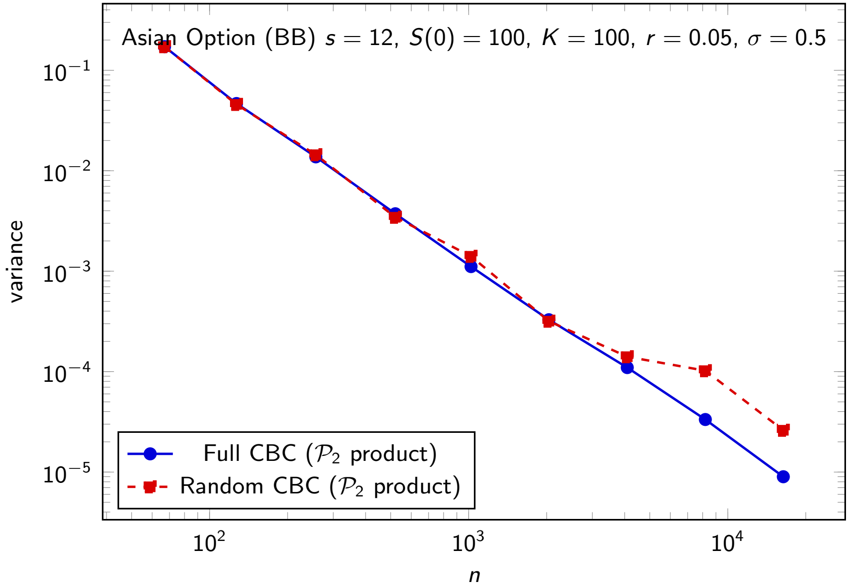
Lattices of Rank 1 with CBC



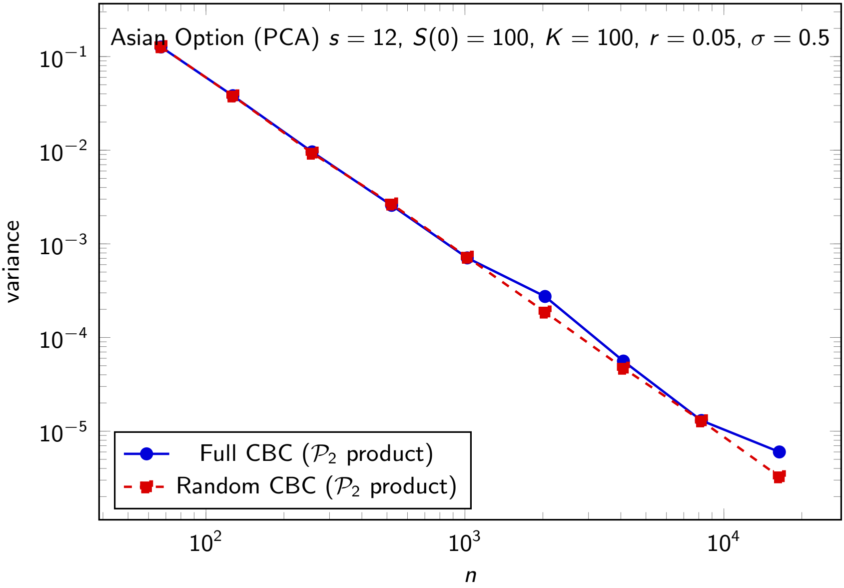
Random vs. Full CBC



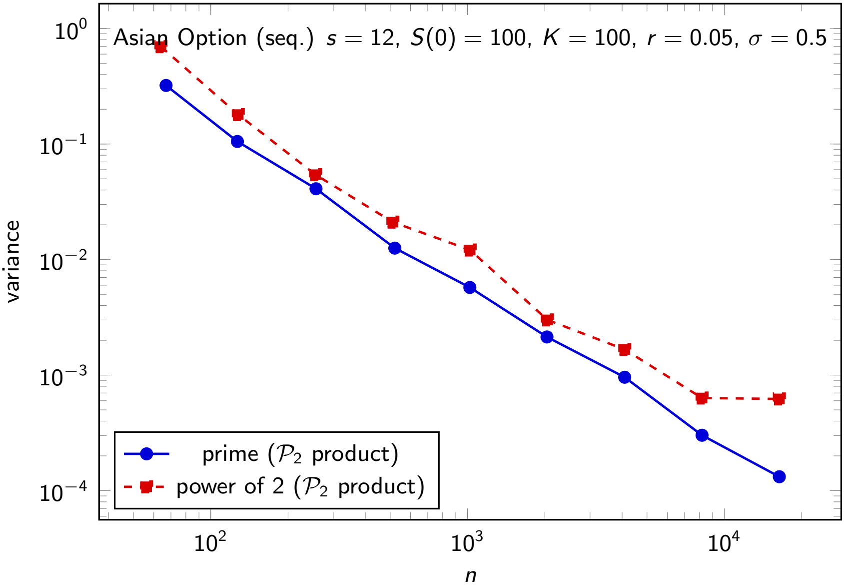
Random vs. Full CBC



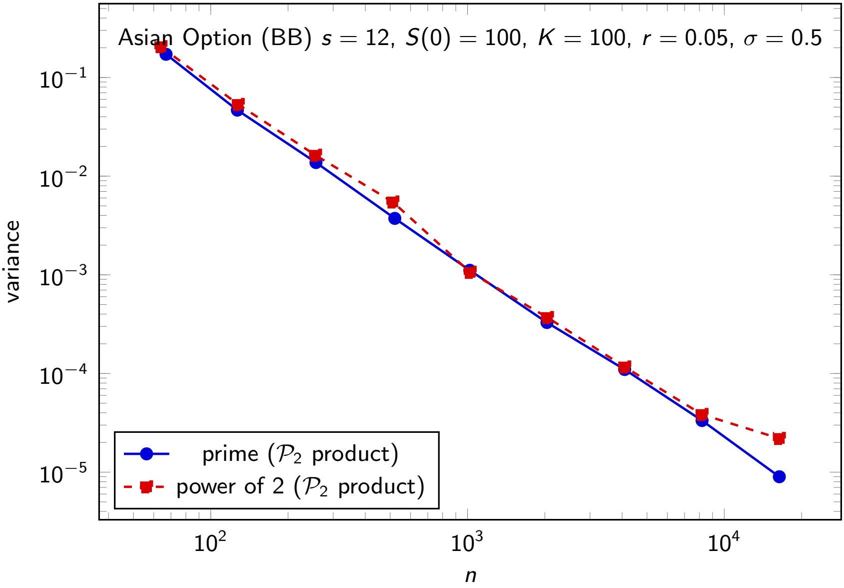
Random vs. Full CBC



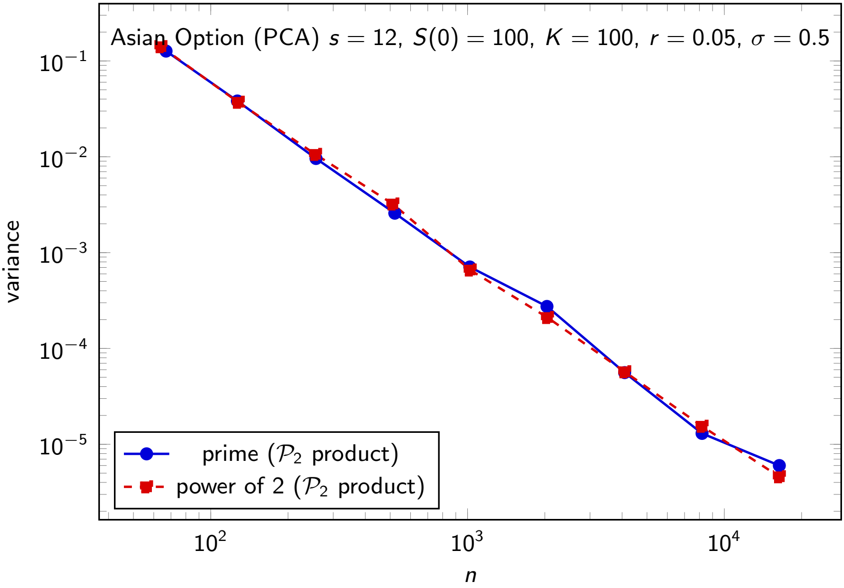
Prime vs. Power-of-2 Number of Points



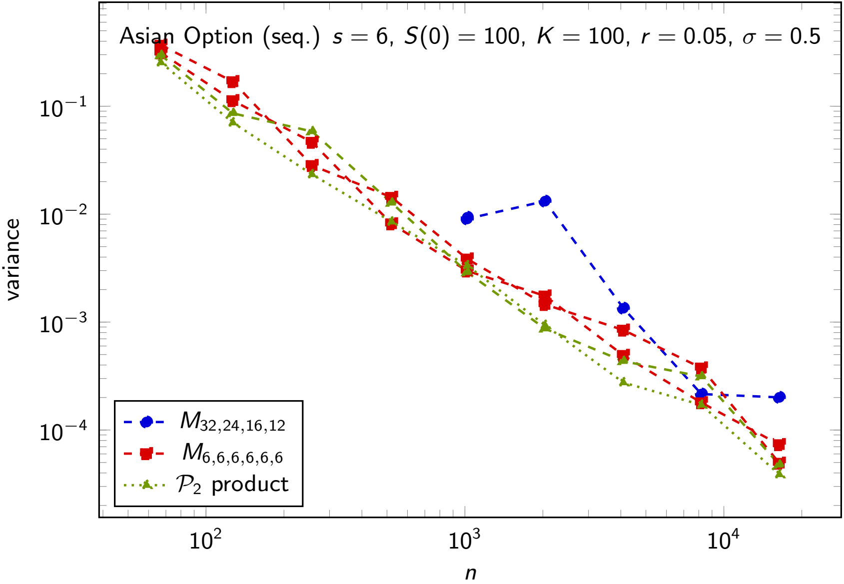
Prime vs. Power-of-2 Number of Points



Prime vs. Power-of-2 Number of Points

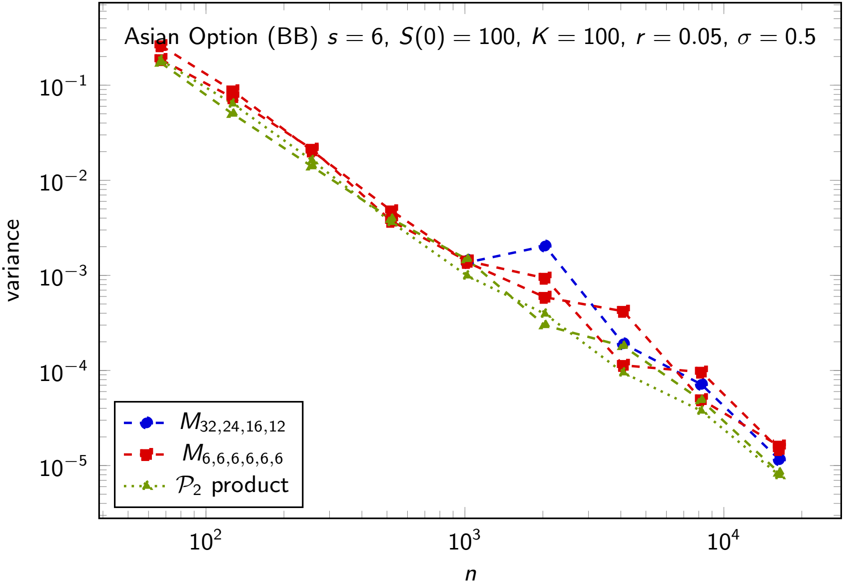


Korobov vs. CBC



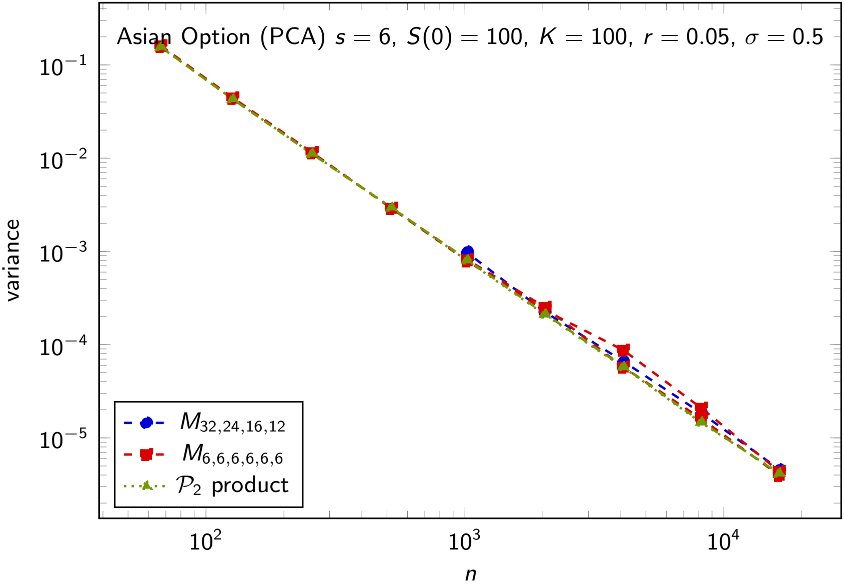
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



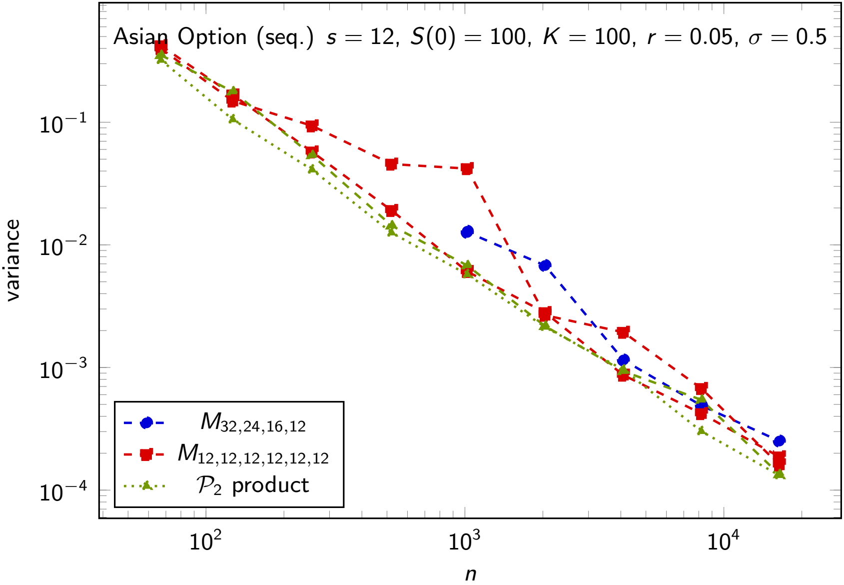
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



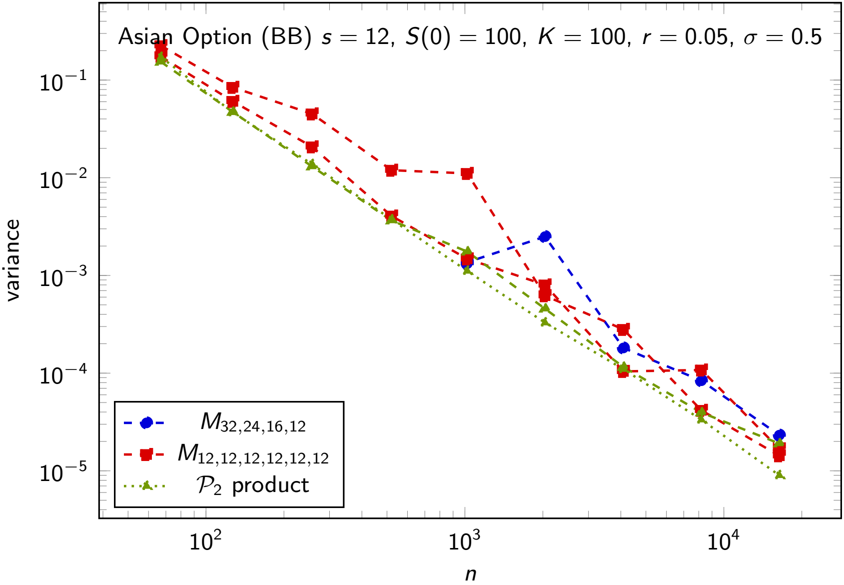
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



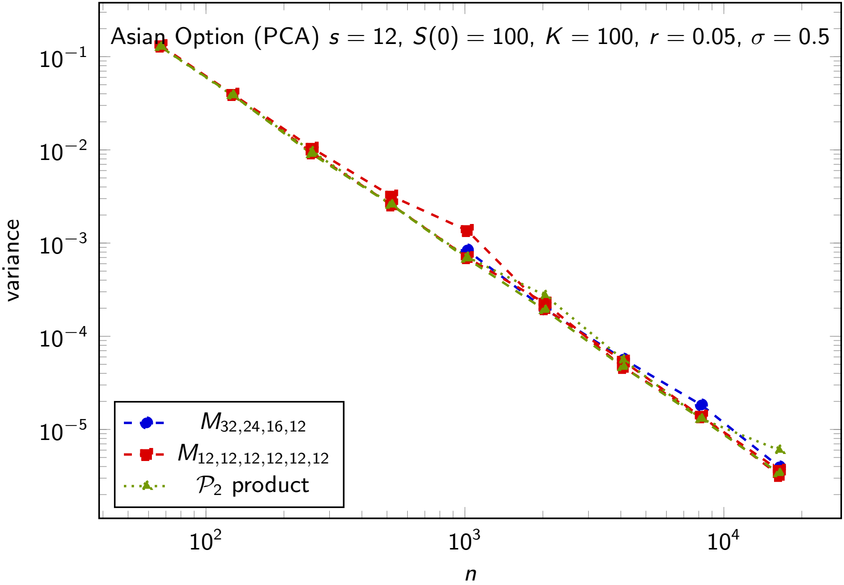
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



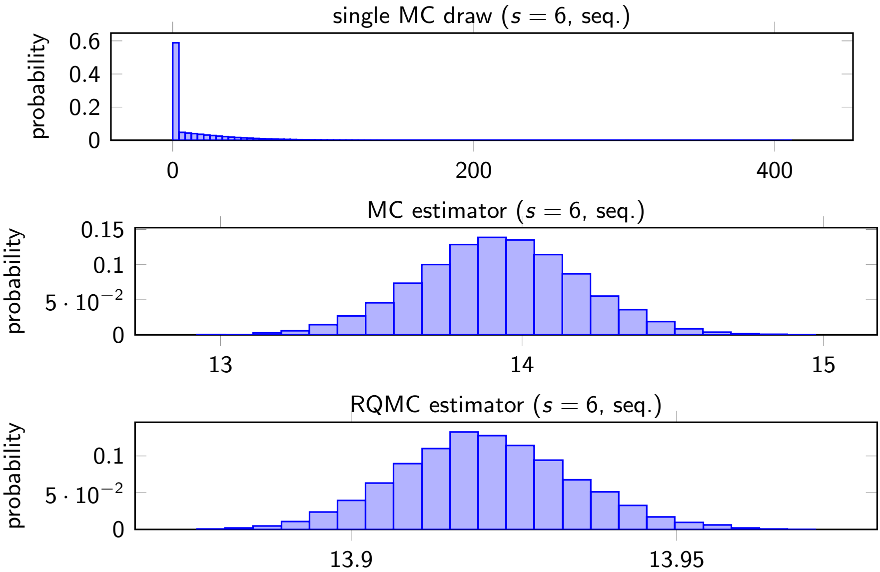
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC

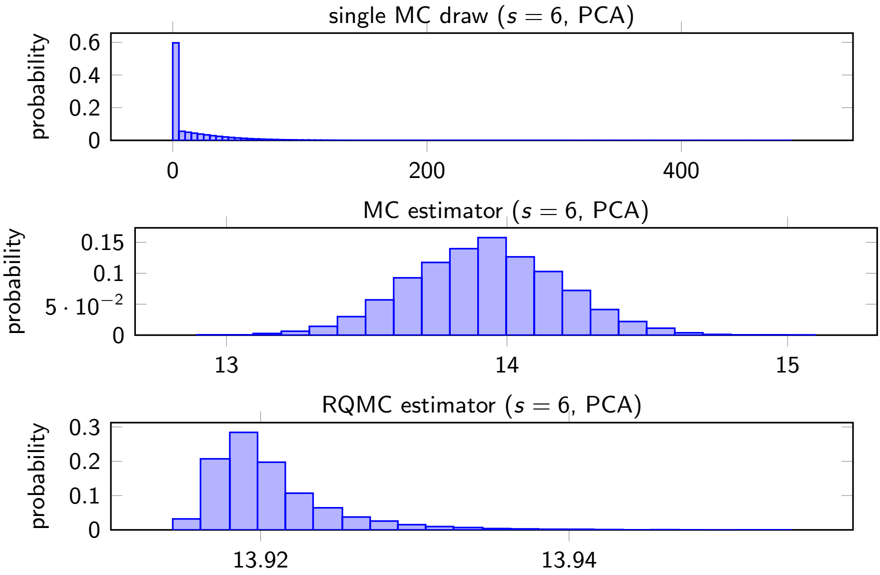


Solid: CBC. Dashed: Korobov.

Histograms for the Asian Option, $s = 6$, sequential



Histograms for the Asian option, $s = 6$, PCA

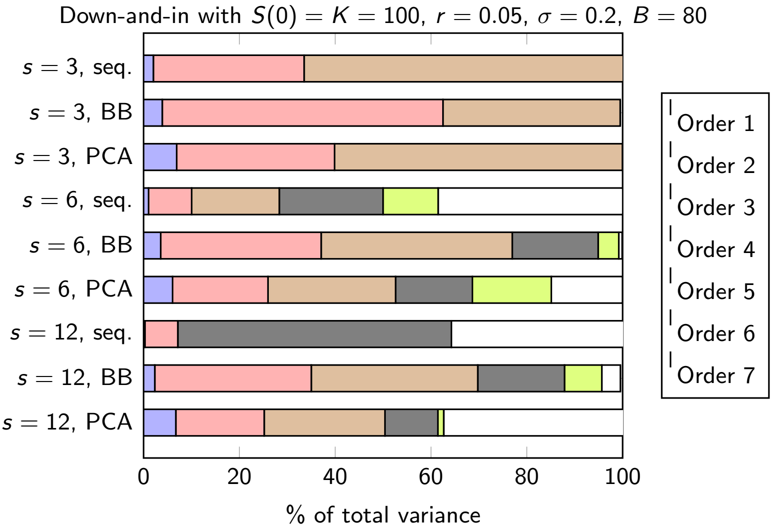


A down-and-in Asian option with barrier B

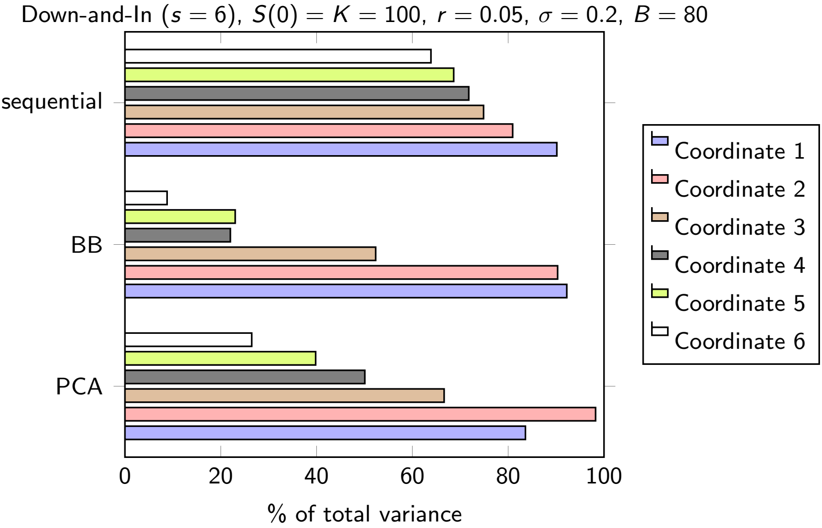
Same as for Asian option, except that payoff is zero unless

$$\min_{1 \leq j \leq s} S(t_j) \leq 80.$$

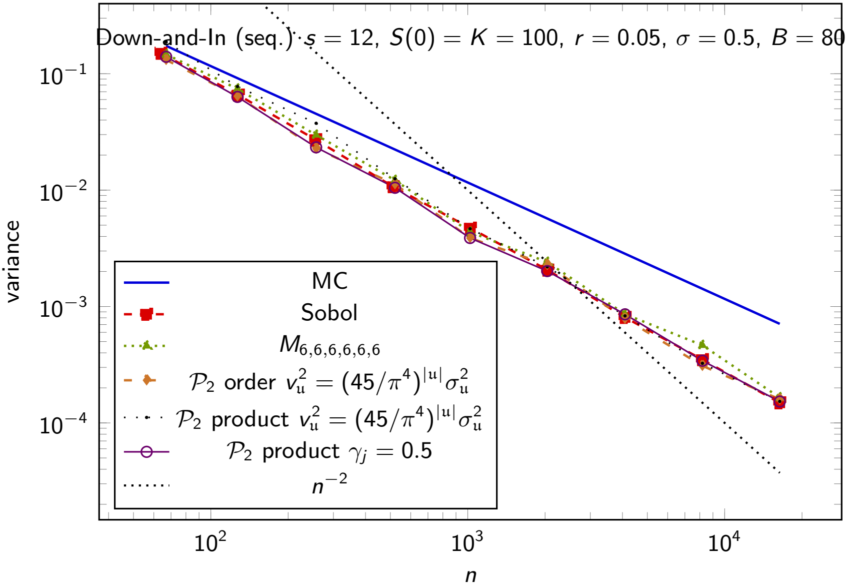
ANOVA Variances for the down-and-in Asian Option



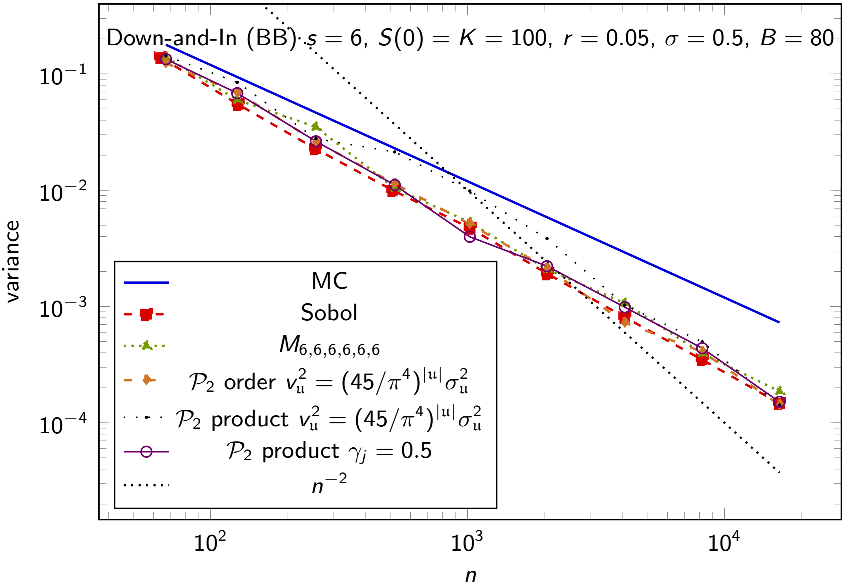
Total Variance per Coordinate for the down-and-in Asian Option



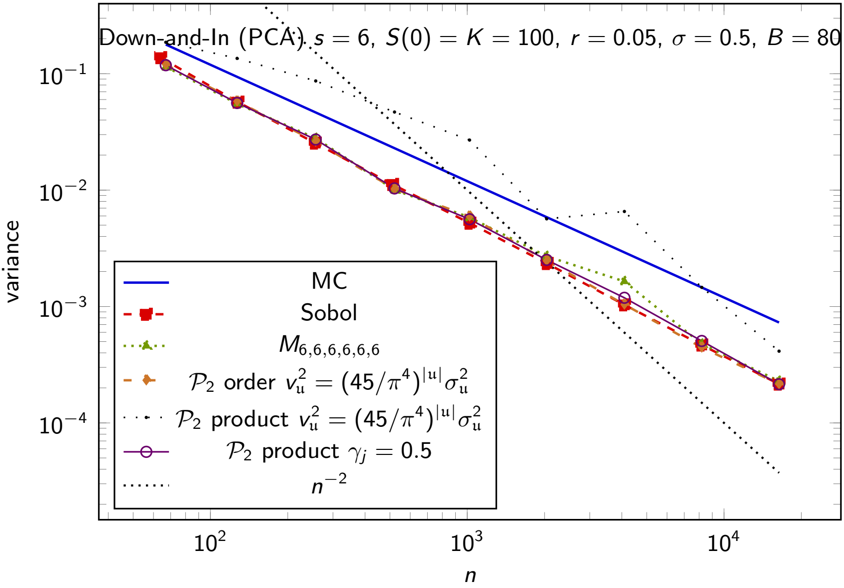
Lattices of Rank 1 with CBC



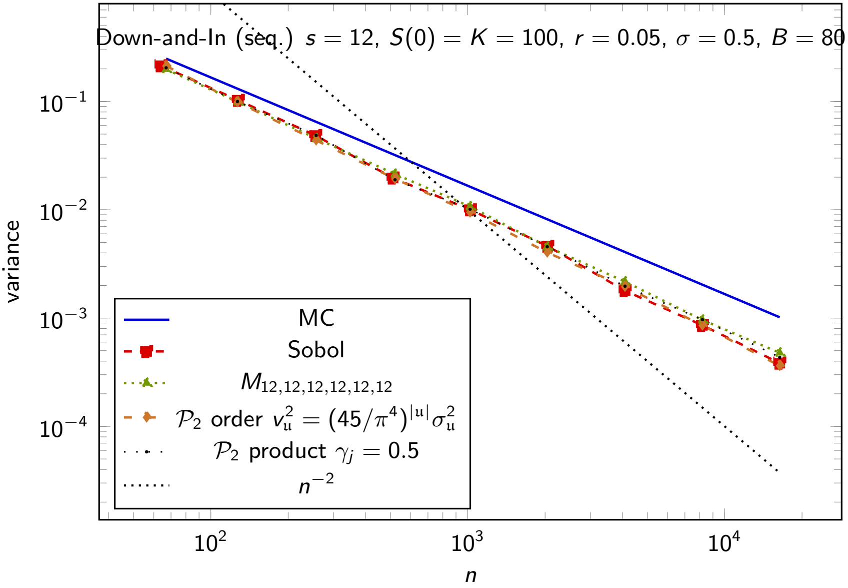
Lattices of Rank 1 with CBC



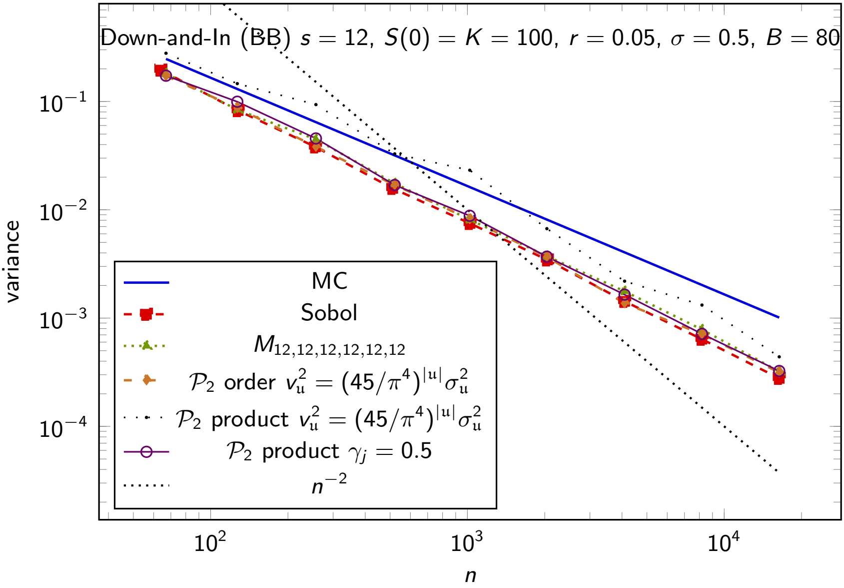
Lattices of Rank 1 with CBC



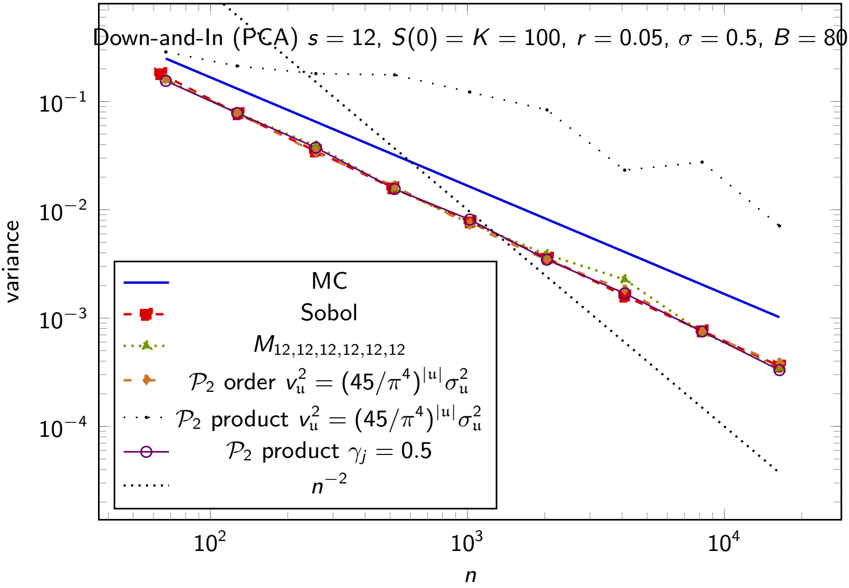
Lattices of Rank 1 with CBC



Lattices of Rank 1 with CBC



Lattices of Rank 1 with CBC



Conclusion and Comments on RQMC

- 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 optimization.
- Array-RQMC for Markov chains.
- Still a lot to learn in that area...