Lattice Builder

A Software Tool for Constructing Lattice Rules

Pierre L'Ecuyer and David Munger

Informatique et Recherche Opérationnelle, Université de Montréal

Lattice Builder

A Software Tool for Constructing Lattice Rules

Pierre L'Ecuyer and David Munger

Informatique et Recherche Opérationnelle, Université de Montréal

Outline:

1. Setting: integration by (randomly-shifted) lattice rules.
2. Want a tool to search for good lattices adapted to problem at hand, for arbitrary dimension, size, figure of merit, ...
3. Selection criteria, construction methods, weights on projections, ...
4. Introducing **Lattice Builder**.

Monte Carlo integration

Want to estimate

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

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

Monte Carlo integration

Want to estimate

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

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

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

Central limit theorem: $\sqrt{n}(\hat{\mu}_n - \mu)/\sigma \;\Rightarrow\; N(0,1)$ when $n \to \infty$.

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.
That is, low discrepancy between empirical distribution of P_n and uniform distribution.

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.
That is, low discrepancy between empirical distribution of P_n and uniform distribution.

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

Randomized quasi-Monte Carlo (RQMC) estimator

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

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.

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 $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\} = L_s \cap [0,1)^s$.

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 $i = 0, \ldots, n-1$.
$n\mathbf{v}_1 = \mathbf{a} = (a_1, \ldots, a_s) \in \mathbb{Z}_n^s$.

Korobov rule: $\mathbf{a} = (1, a, a^2 \mod n, \ldots, a^{s-1} \mod n)$.

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 $i = 0, \ldots, n-1$.
$n\mathbf{v}_1 = \mathbf{a} = (a_1, \ldots, a_s) \in \mathbb{Z}_n^s$.

Korobov rule: $\mathbf{a} = (1, a, a^2 \mod n, \ldots, a^{s-1} \mod n)$.

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

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 $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\} = L_s \cap [0,1)^s$.

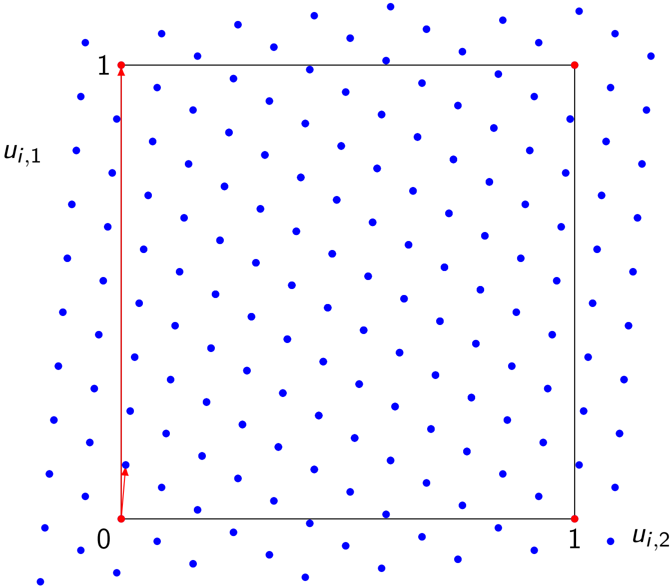
Lattice rule of rank 1: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \ldots, n-1$.
$n\mathbf{v}_1 = \mathbf{a} = (a_1, \ldots, a_s) \in \mathbb{Z}_n^s$.

Korobov rule: $\mathbf{a} = (1, a, a^2 \mod n, \ldots, a^{s-1} \mod n)$.

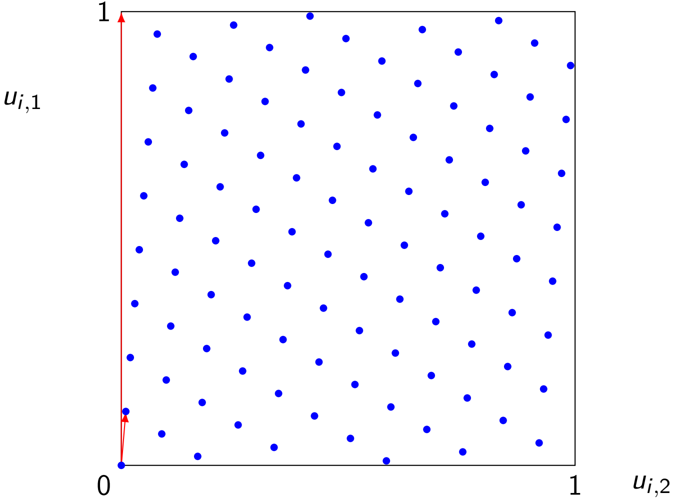
For any $\mathfrak{u} \subset \{1, \ldots, s\}$, the projection $L_s(\mathfrak{u})$ of L_s is also a lattice, with point set $P_n(\mathfrak{u})$.

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 $[0,1)^s$.

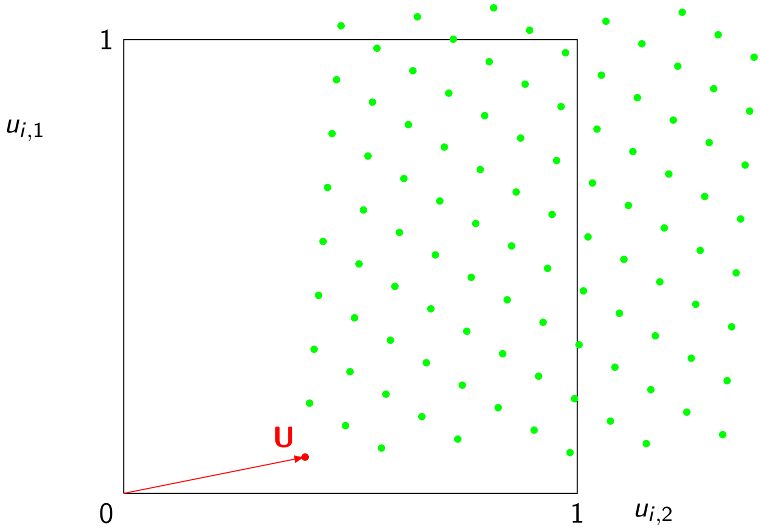
Randomly-shifted lattice: Two-dim. example



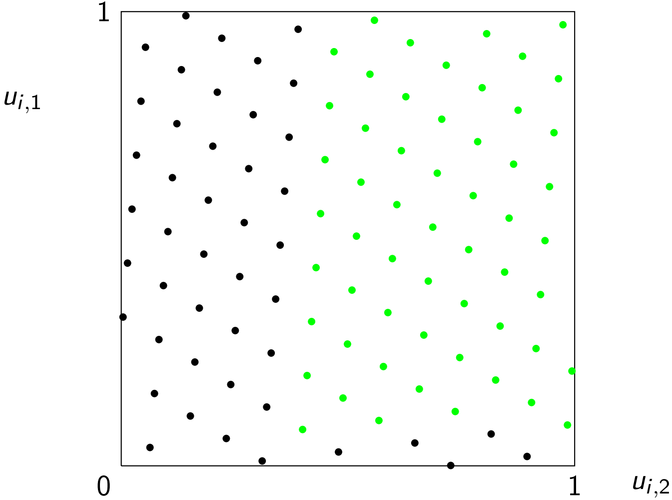
Randomly-shifted lattice: Two-dim. example



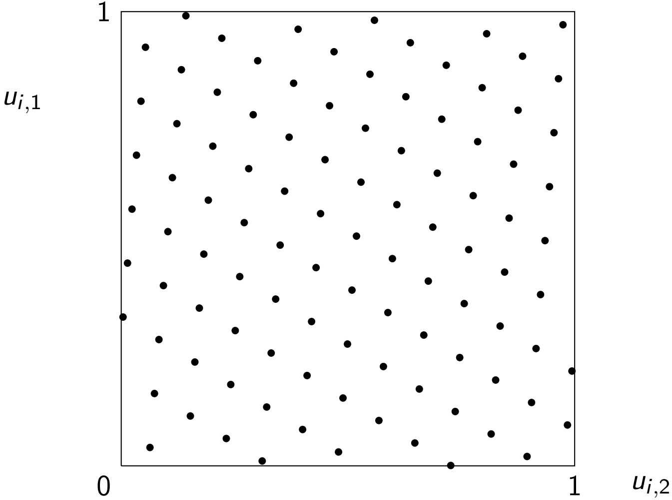
Randomly-shifted lattice: Two-dim. example



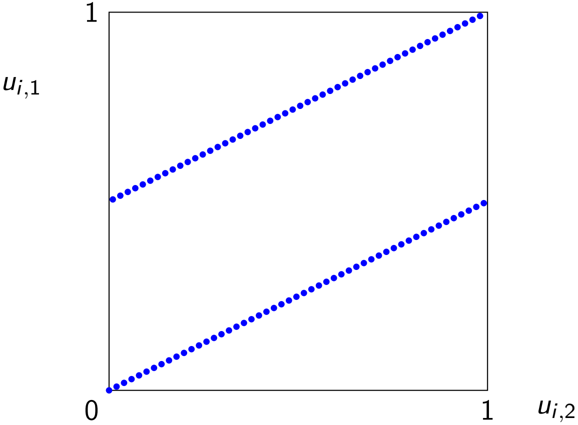
Randomly-shifted lattice: Two-dim. example



Randomly-shifted lattice: Two-dim. example



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



Good uniformity in one dimension, but not in two!

Variance expression

Suppose f has Fourier expansion

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

For a randomly shifted lattice, the exact variance is

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

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

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

But not a practical criterion...

Periodic smooth functions

If f has square-integrable mixed partial derivatives up to order $\alpha/2$, and the periodic continuations of its derivatives up to order $\alpha/2-1$ are 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

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

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

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

If $\alpha/2$ is a positive integer, 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 $\alpha/2$.

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!

[cf., tractability theory; Wózniakowski, Novak, Sloan, Dick, etc.]

A very general weighted $\mathcal{P}_\alpha$

$\mathcal{P}_\alpha$ can be generalized by giving different weights $w(\mathbf{h})$ to the vectors $\mathbf{h}$:

$$\tilde{\mathcal{P}}_\alpha \stackrel{\text{def}}{=} \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} w(\mathbf{h})(\max(1,|h_1|)\cdots\max(1,|h_s|))^{-\alpha}.$$

But how do we choose these weights? There are too many!

The optimal weights to minimize the variance are:

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

ANOVA decomposition

A cruder expansion of $f(\mathbf{u}) = f(u_1, \ldots, u_s)$:

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

where

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

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 $h_j \neq 0$.

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

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

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

(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

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

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

For f^*, we should take $\gamma_{\mathfrak{u}} = \rho^{|\mathfrak{u}|}\sigma_{\mathfrak{u}}^2$ for some constant ρ.

But there are still $2^s - 1$ subsets $\mathfrak{u}$ to consider!

One could define a simple parametric model for the square variations and then estimate the parameters by matching the ANOVA variances $\sigma_{\mathfrak{u}}^2$ (e.g., Wang and Sloan 2006, L. and Munger 2012).

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 $d = 2$.

Mixture: POD weights (Kuo et al. 2011).

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

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 $\mathcal{R}_\alpha$.

Can be computed for any $\alpha > 0$ (finite sum).

We compute it using 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 $\ell^*_{|\mathfrak{u}|}(n)$ (with Euclidean length):

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

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 $\ell^*_{|\mathfrak{u}|}(n)$ (with Euclidean length):

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

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

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

Computing time of $\ell_{\mathfrak{u}}(P_n)$ is almost independent of n, but exponential in $|\mathfrak{u}|$. Poor lattices can be eliminated quickly.

Can use a different norm, compute shortest vector in primal lattice, etc.

Search methods

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

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

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

Search methods

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

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

$P_{n_1} \subset P_{n_2} \subset \dots P_{n_m}$ 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 $(a_1, \dots, a_s)$ under consideration.
For $\mathcal{P}_{\gamma,\alpha}$, bounds by Sinescu and L. (2012) and Dick et al. (2008).
For CBC, we do this for each coordinate $j = 1, \dots, s$ (replace s by j).
Also used as filters.

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

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

Existing tools

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

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

Software for **using** (randomized) lattice rules in simulations is also available in many places (e.g., in 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.

Offers a 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

Examples of applications

Example: Playing with the Weights

To see the effect of weights selection on RQMC variance, when choosing a lattice rule, we shall integrate the worst-case function

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

whose RQMC variance is $\mathcal{P}_{\gamma,\alpha}$.

The ideal weights are $\gamma_\mathfrak{u} = v_\mathfrak{u}$. In our experiments, we measure the inflation factor of RQMC variance when we use a lattice rule constructed via fast CBC with different weights $\gamma_\mathfrak{u} \neq v_\mathfrak{u}$.

We start with $s = 10$ and $v_\mathfrak{u} = \Gamma^{|\mathfrak{u}|}$ for $|\mathfrak{u}| \le k$, for a given integer k and a given constant $\Gamma > 0$. We select the weights as $\gamma_\mathfrak{u} = \tilde{\Gamma}^{|\mathfrak{u}|}$ for $|\mathfrak{u}| \le \tilde{k}$, where $\tilde{\Gamma}$ and $\tilde{k}$ may differ from Γ and k.

Ratio of RQMC variances for modified vs ideal weights

n	A1	A2	B1	B2	C1	C2
2^8	1.11	1.21	1.13	4.08	3.82	6.80
2^9	1.21	1.10	1.42	10.5	2.93	7.25
2^{10}	1.36	1.38	2.04	4.64	2.86	5.94
2^{11}	1.24	1.43	2.40	6.18	2.15	5.14
2^{12}	1.42	1.66	3.79	13.2	2.47	5.94
2^{13}	1.30	2.38	5.51	9.09	2.66	5.97
2^{14}	1.51	2.54	**30.5**	8.66	**9.11**	**29.1**
2^{15}	1.46	1.93	25.6	**13.3**	3.52	9.71
2^{16}	**1.80**	**2.55**	3.13	12.9	2.73	10.2

A1: $k = \tilde{k} = s = 10$, $\Gamma^2 = 0.1$ and $\tilde{\Gamma}^2 = 0.001$. Not too bad.

A2: $k = \tilde{k} = s = 10$, $\Gamma^2 = 0.001$ and $\tilde{\Gamma}^2 = 0.1$. More impact.

B1: $\Gamma^2 = \tilde{\Gamma}^2 = 0.1$, $k = 4$ and $\tilde{k} = 2$. Criterion is blind on projections of order 3 and 4. This has unpredictable (sometimes dramatic) impact on RQMC variance.

B2: $\Gamma^2 = \tilde{\Gamma}^2 = 0.5$, $k = 2$ and $\tilde{k} = 4$. Gives weight to irrelevant projections. Stronger degradation on average.

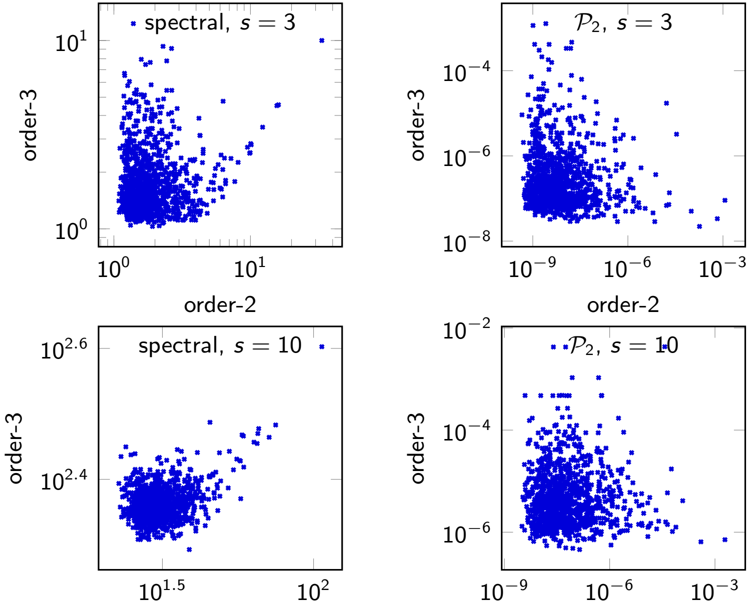
C1: $\Gamma^2 = \tilde{\Gamma}^2 = 0.1$ and $k = \tilde{k} = 4$, but increase the variation of f by replacing $v_{\mathfrak{u}}^2$ with $v_{\mathfrak{u}}^2 + \tilde{v}_{\mathfrak{u}}^2$, with
$\tilde{v}_{\mathfrak{u}}^2 = 1.0$ for $\mathfrak{u} = \{1,3\}, \{3,5\}, \{5,7\}, \{7,9\}$,
$\tilde{v}_{\mathfrak{u}}^2 = 0.5$ for $\mathfrak{u} = \{2,3,4\}, \{4,5,6\}, \{6,7,8\}, \{8,9,10\}$,
$\tilde{v}_{\mathfrak{u}}^2 = 0.25$ for $\mathfrak{u} = \{1,2,3,4\}, \{4,5,6,7\}, \{7,8,9,10\}$,
and $\tilde{v}_{\mathfrak{u}} = 0$ for all other $\mathfrak{u}$.
Important projections are not given enough weight relative to others.

C2: Like C1, but $v_{\mathfrak{u}}^2$ is replaced with only $\tilde{v}_{\mathfrak{u}}^2$, as defined above. Many irrelevant projections $\mathfrak{u}$, with $\tilde{v}_{\mathfrak{u}} = 0$, now have weights $\gamma_{\mathfrak{u}} > 0$.

In all cases, the variance ratio increases (non-monotonically) with n.

Competing Projections for spectral and $\mathcal{P}_2$ criteria

Plots of order 2 vs order 3 contributions, for 1000 random lattices with $n = 2^{20}$, with $s = 3$ (top) and 10 (bottom).

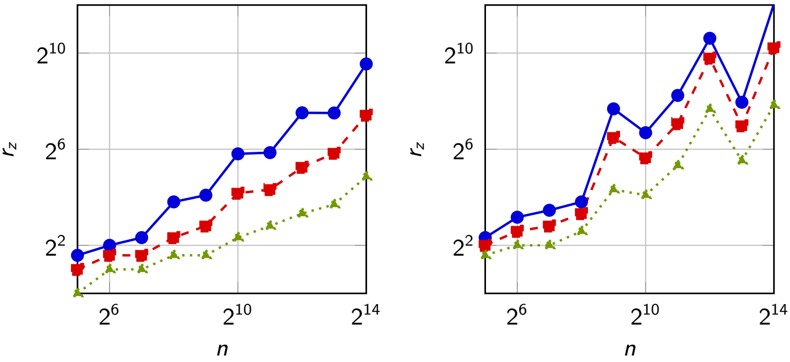


Construction via CBC vs random CBC

$\mathcal{P}_{\gamma,2}$ for product weights, $\gamma_j^2 = 0.1$, $s = 10$ (left) and $s = 5$ (right).

$$r_z = \min\{r : \mathbb{E}_{\text{random CBC}}[\mathcal{P}_{\gamma,2}] \leq (1 + z/100)\, \mathcal{P}_{\gamma,2}(\text{CBC})\}.$$

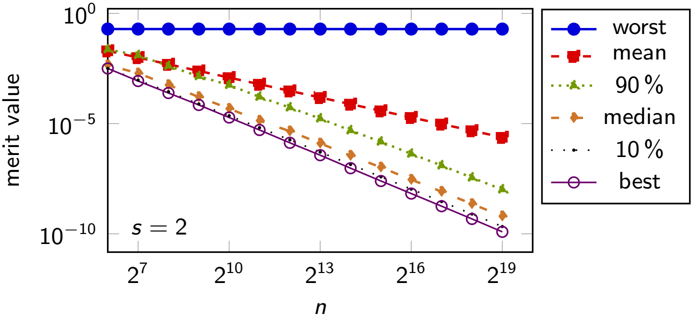
r_z as a function of n for $z = 5$ (—●—), $z = 10$ (- ■ -) and $z = 20$ (··▲··).



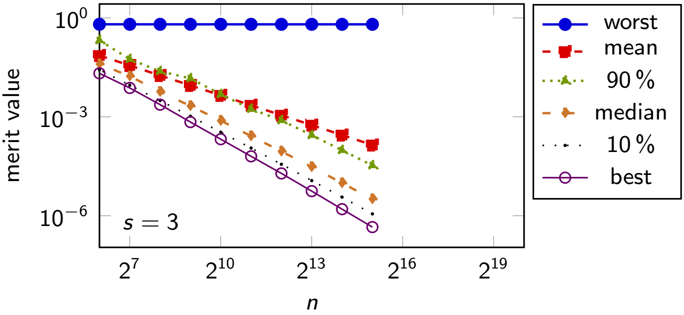
Quantiles of figure of merit

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 $n = 2^e, \ldots, 2^{19}$.

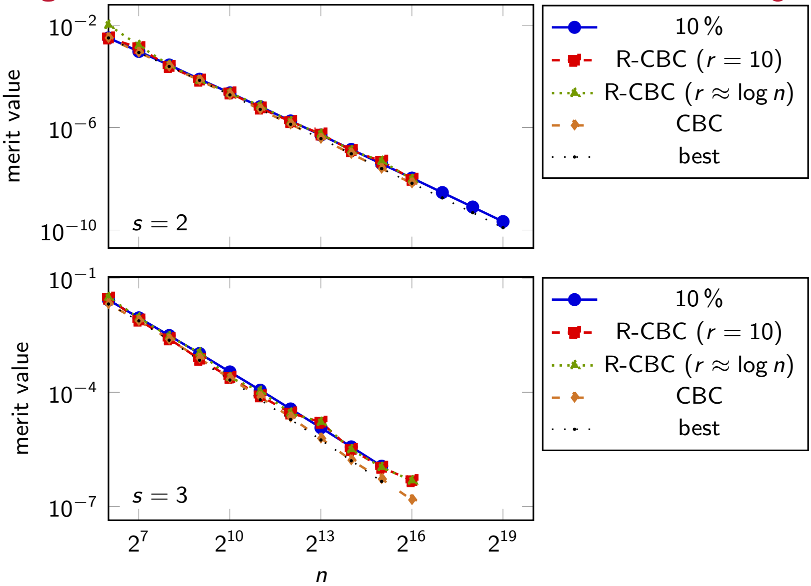
For $s = 2$, a linear regression of $\log \mathcal{P}_{\gamma,2}$ vs $\log n$ for $2^{12} \leq n \leq 2^{19}$ gives decreasing rates of $n^{-1.92}$ for the best, and $n^{-1.87}$ and $n^{-1.77}$ for the 10 % and 90 % quantiles. The mean decreases as n^{-1} an the worst-case as n^0 (it is near 0.1948 for 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} and the worst-case as n^0 (it is near 0.6393).

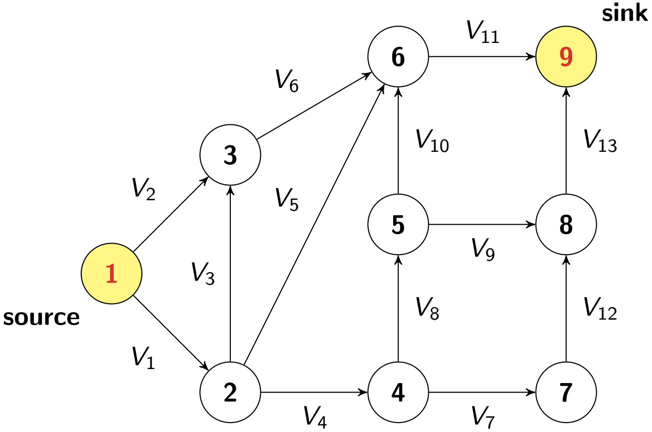


Convergence of CBC and random CBC with $r = 10$ and $r \approx \log n$



Example: a stochastic activity network

Each arc j has random length $V_j = F_j^{-1}(U_j)$.
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.

CMC estimator. Generate the V_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 V_j's, $\mathbb{P}[T > x \mid \{V_j, j \notin \mathcal{L}\}]$.
This makes the integrand continuous in the U_j's.

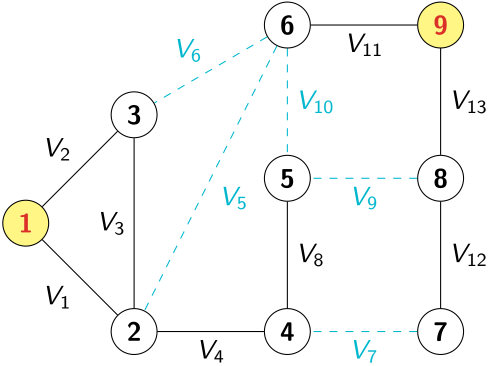
To estimate $q(x)$ by **MC**, we generate n independent realizations of T, say $T_1, \ldots, T_n$, and $(1/n)\sum_{i=1}^{n} \mathbb{I}[T_i > x]$.

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

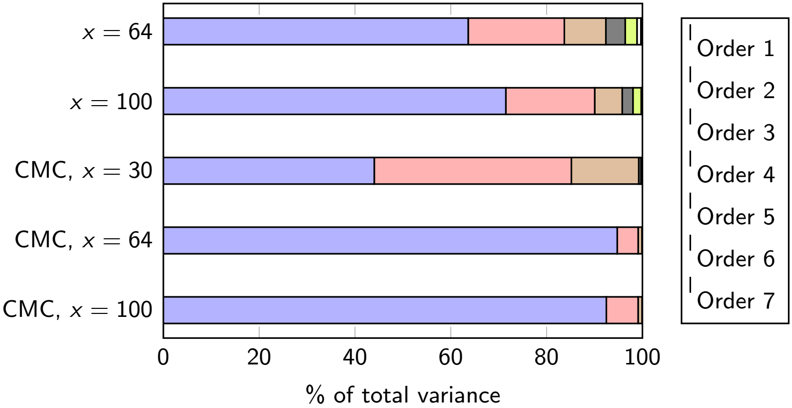
CMC estimator. Generate the V_j's only for the 8 arcs that do not belong to the cut $\mathcal{L} = \{5, 6, 7, 9, 10\}$, and replace $\mathbb{I}[T > x]$ by its conditional expectation given those V_j's, $\mathbb{P}[T > x \mid \{V_j, j \notin \mathcal{L}\}]$.
This makes the integrand continuous in the U_j's.

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

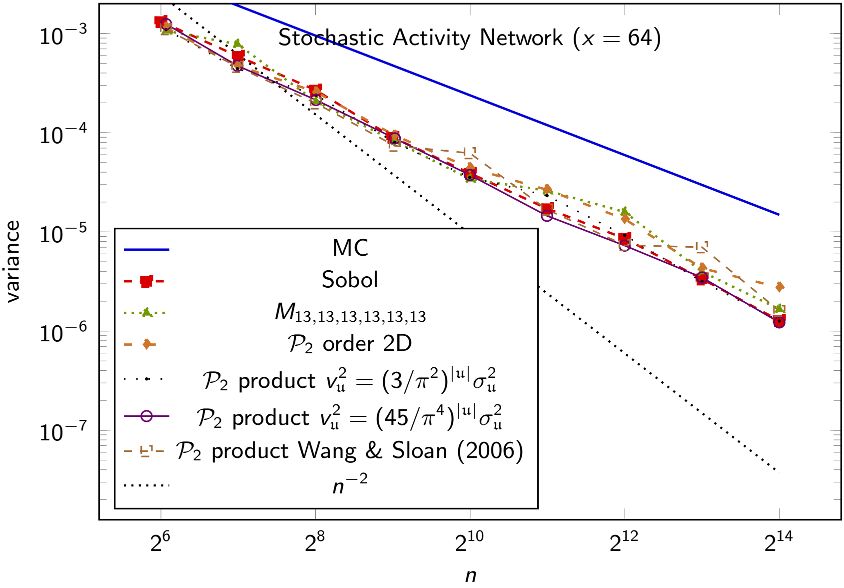
The μ_j: 13.0, 5.5, 7.0, 5.2, 16.5, 14.7, 10.3, 6.0, 4.0, 20.0, 3.2, 3.2, 16.5.


6
9
V_6
V_{11}
V_{10}
V_{13}
3
V_2
5
8
V_5
V_9
1
V_3
V_8
V_{12}
V_1
2
4
7
V_4
V_7

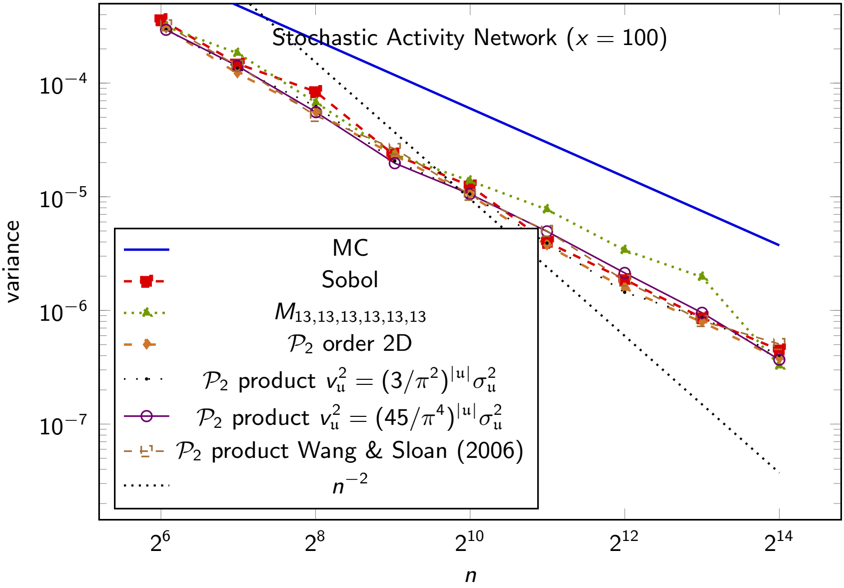
ANOVA Variances for the Stochastic Activity Network



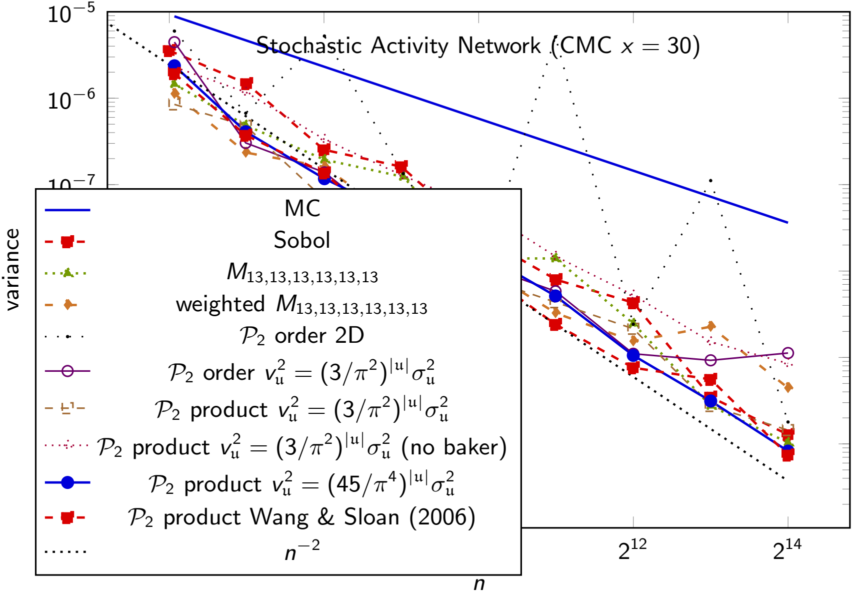
Lattices of Rank 1 with CBC



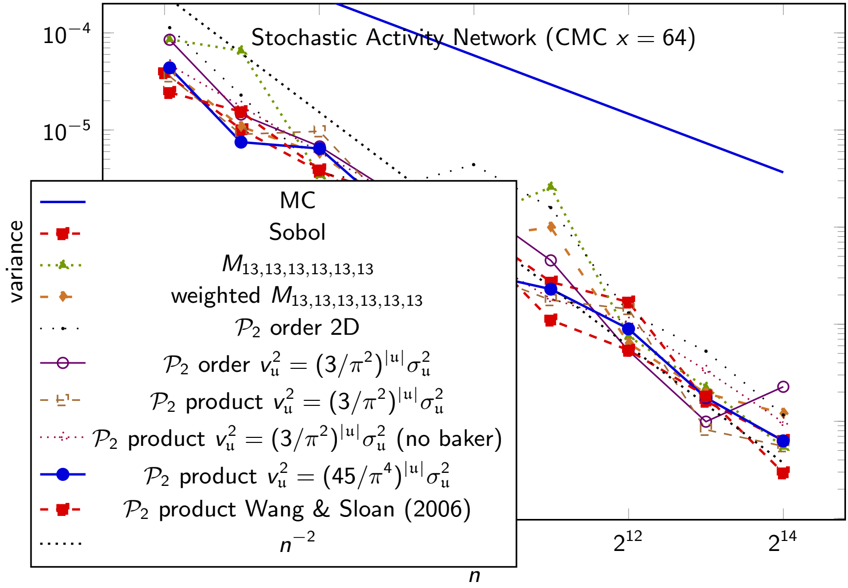
Lattices of Rank 1 with CBC



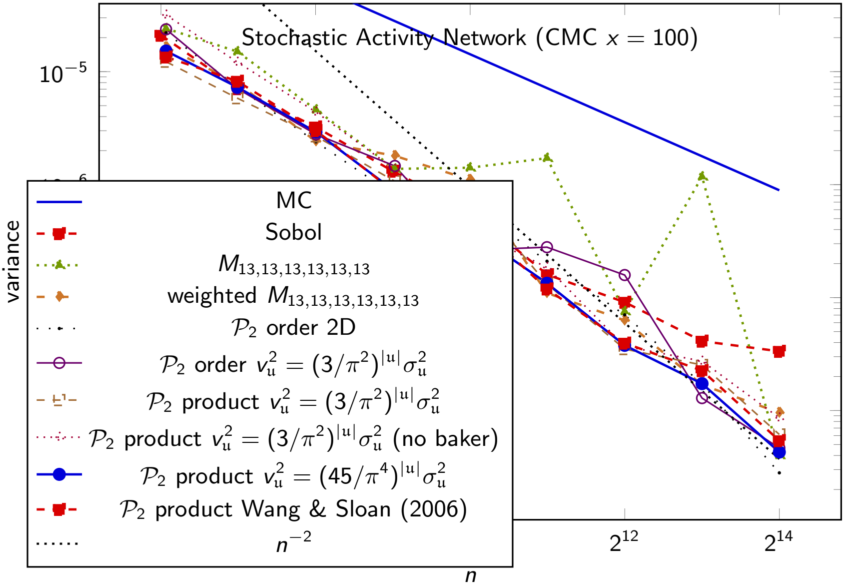
Lattices of Rank 1 with CBC



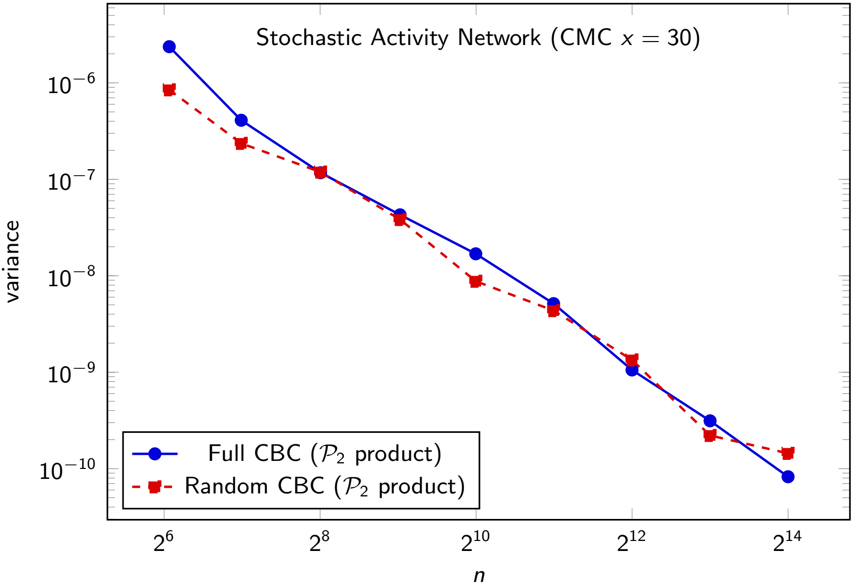
Lattices of Rank 1 with CBC



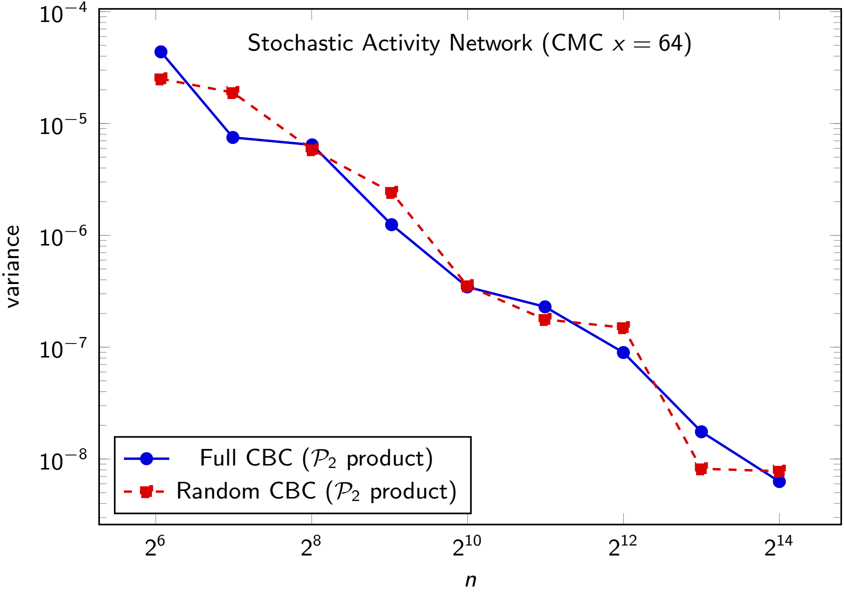
Lattices of Rank 1 with CBC



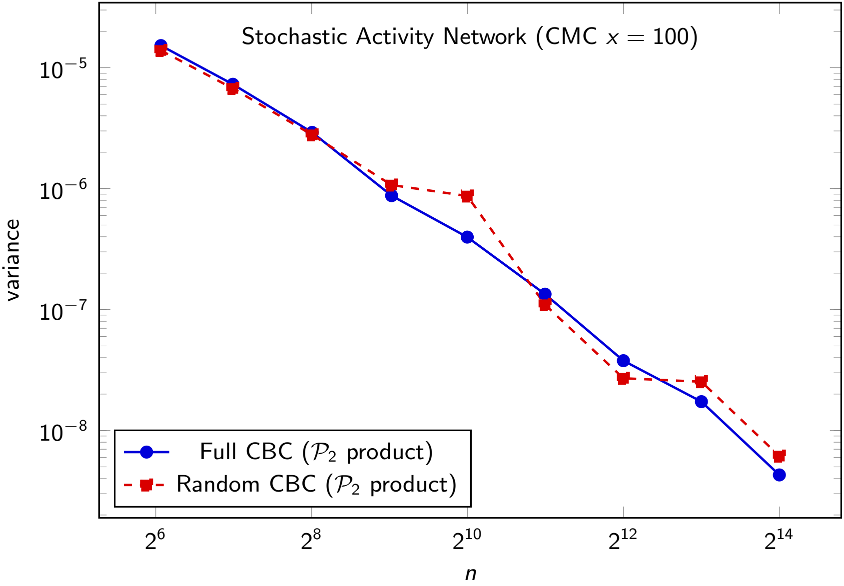
Random vs. Full CBC



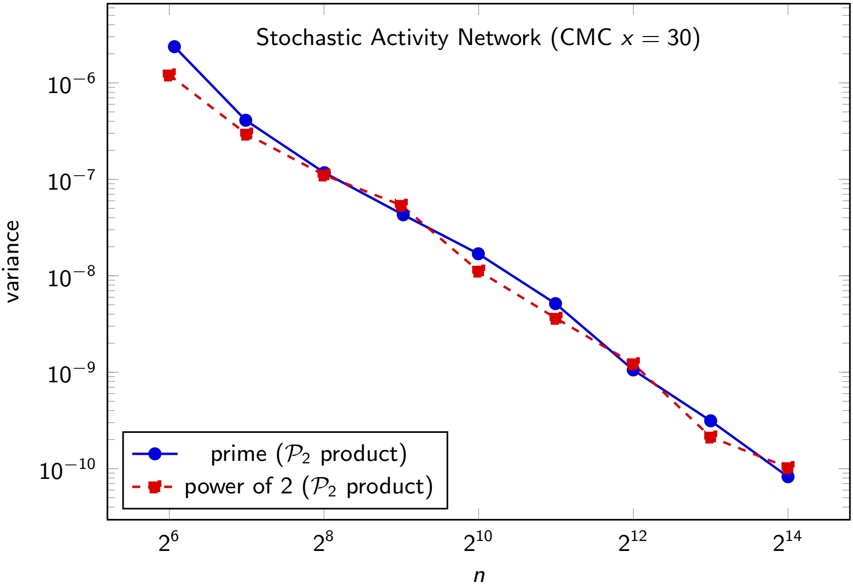
Random vs. Full CBC



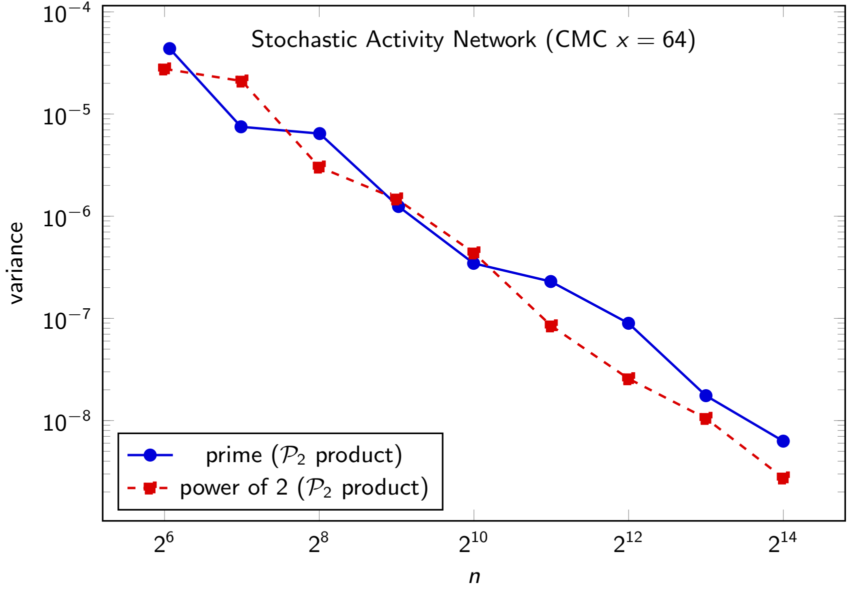
Random vs. Full CBC



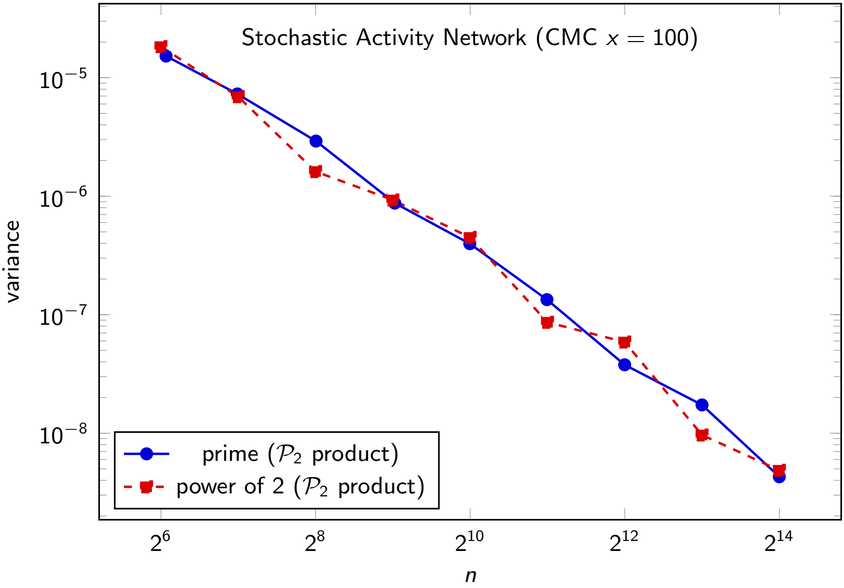
Prime vs. Power-of-2 Number of Points



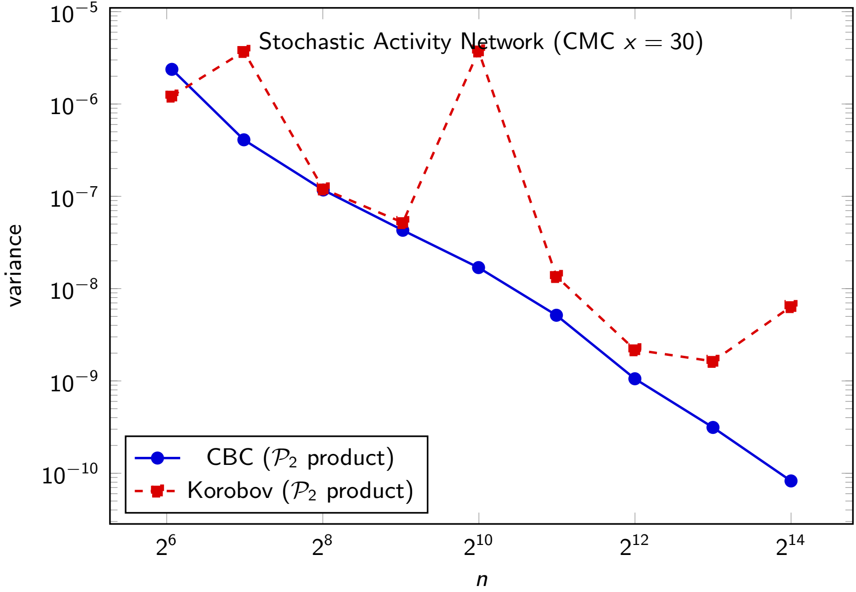
Prime vs. Power-of-2 Number of Points



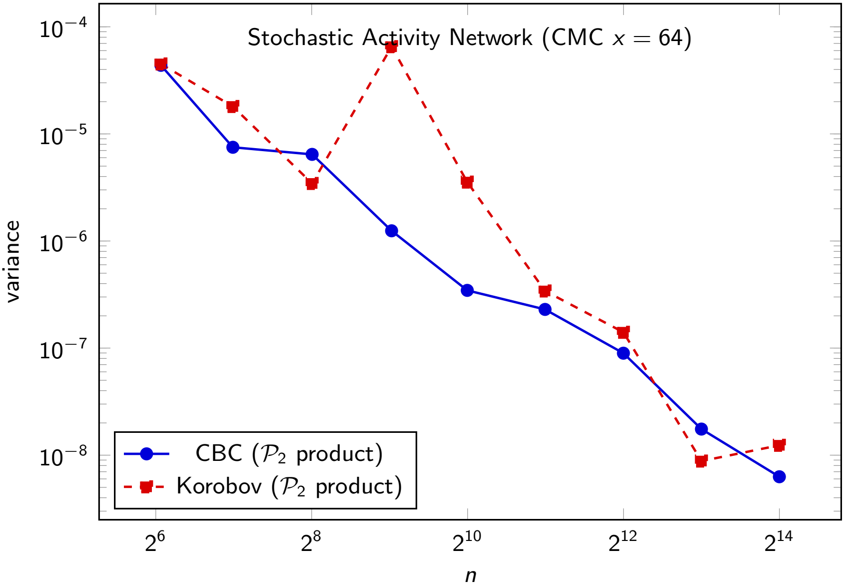
Prime vs. Power-of-2 Number of Points



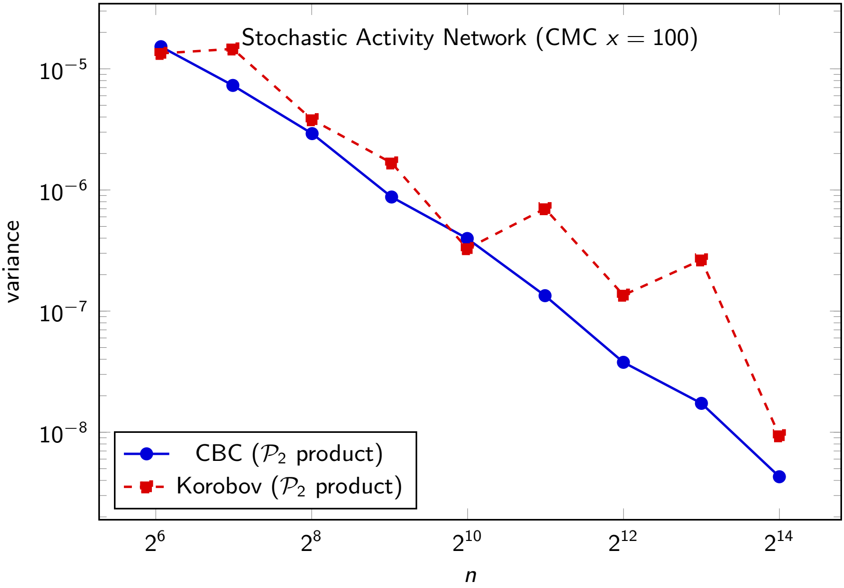
Korobov vs. CBC



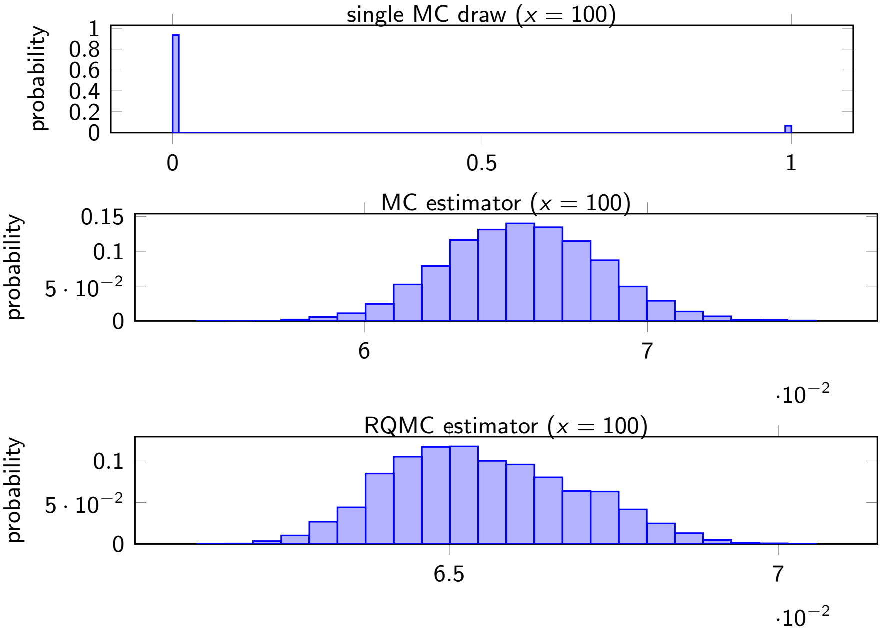
Korobov vs. CBC



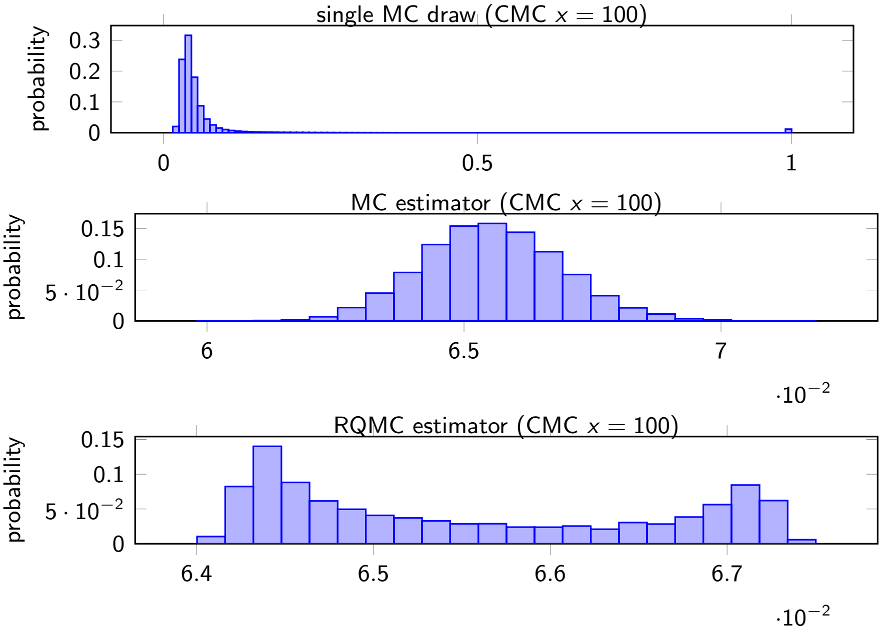
Korobov vs. CBC



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
$\cdot 10^{-2}$

Conclusion and future work

– We (and you!) have a flexible software tool to search for good lattice rules based on various criteria, arbitrary weights, with several choices of method, arbitrary dimension and number of points.

– Can be handy for applications, as well as for empirical investigations on lattice rules behavior.

Future:

– Consider other figures of merit, e.g., based on other function spaces.

– Design and implement a similar software tool for digital nets.