Construction Algorithms for (Polynomial) Lattice Points

Peter Kritzer

Johann Radon Institute for Computational and Applied Mathematics (RICAM)
Austrian Academy of Sciences

Joint work with

J. Dick, A. Ebert, G. Leobacher, D. Nuyens, O.O. Osisiogu, F. Pillichshammer

MCM 2019, Sydney

Research supported by the Austrian Science Fund, Project F5506-N26

Introduction and Motivation

Consider integration of functions on $[0,1]^d$,

$$I_d(f) = \int_{[0,1]^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \,,$$

where $f \in \mathcal{H}$, and $\mathcal{H}$ is some Banach space.

Approximate I_d by a quasi-Monte Carlo (QMC) rule,

$$I_d(f) \approx Q_{N,d}(f) = \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{x}_k),$$

where $\mathcal{P}_N = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\}$.

Both parameters d and N can be (very) large.

Worst case error in Banach space $\mathcal{H}$ with respect to $\mathcal{P}_N = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\}$:

$$e_{N,d}(\mathcal{H}, \mathcal{P}_N) := \sup_{f \in \mathcal{H}, \|f\| \le 1} |I_d(f) - Q_{N,d}(f)| .$$

Need $\mathcal{P}_N$ that makes $e_{N,d}(\mathcal{H}, \mathcal{P}_N)$ small: How can we find it?

Function space considered here:

Weighted Korobov space $(\mathcal{H}_{d,\alpha,\gamma}, \|\cdot\|_{d,\alpha,\gamma})$:

space of 1-periodic continuous functions f, where

$$\|f\|_{d,\alpha,\gamma}^2 = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \rho_{\alpha,\gamma}(\boldsymbol{h}) \, |\widehat{f}(\boldsymbol{h})|^2,$$

where $\widehat{f}(\boldsymbol{h})$is the $\boldsymbol{h}$-th Fourier coefficient of f.

The function $\rho_{\alpha,\gamma}(\boldsymbol{h})$ moderates the decay of the Fourier coefficients.

Set $\boldsymbol{\rho}_{\alpha,\boldsymbol{\gamma}}(\mathbf{0}) := 1$.

For $\boldsymbol{h} = (h_1, \ldots, h_d)$, let $\mathfrak{u} \subseteq \{1, \ldots, d\}$ be the set of the j with $h_j \neq 0$. Then

$$\boldsymbol{\rho}_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h}) = \gamma_{\mathfrak{u}}^{-1} \prod_{j \in \mathfrak{u}} |h_j|^{\alpha} .$$

- $\alpha > 1$: "smoothness parameter" (higher $\alpha \to$ smoother functions in $\mathcal{H}_{d,\alpha,\boldsymbol{\gamma}}$),
- $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1,\ldots,d\}}$: coordinate weights.

Weights?

- Sloan and Woźniakowski (1998): Assign weights to different groups of coordinates to model their different influence on a problem:

$$\gamma = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1,\ldots,d\}}$$

of positive reals: weights.

- larger weights $\simeq$ more influence of corresponding variables, smaller weights $\simeq$ less influence of corresponding variables.
- Suitable weights can help to reduce negative influence of the dimension.
- Important class of weights: product weights

$$\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$$

for positive reals γ_j. In this case, assume

$$1 = \gamma_1, \quad \text{and} \quad \gamma_j \geq \gamma_{j+1} \quad \text{for all } j.$$

Here:

$\mathcal{P}_N = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\}$ is a rank-1 lattice point set with generating vector

$$\boldsymbol{z} = (z_1, \ldots, z_d) \in \{1, \ldots, N-1\}^d.$$

Points of $\mathcal{P}_N$:

$$\boldsymbol{x}_n = (x_{n,1}, \ldots, x_{n,d})$$

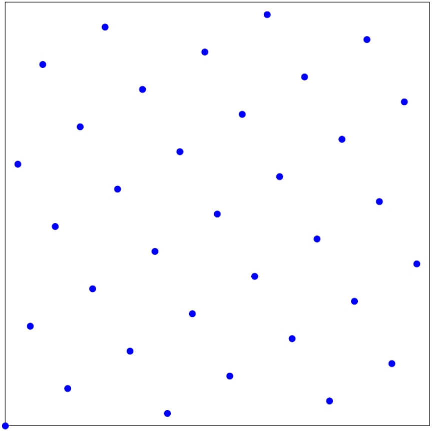
with

$$x_{n,j} = \left\{ \frac{n z_j}{N} \right\},$$

where $\{t\} = t - \lfloor t \rfloor$.

Note: Given N and d, $\boldsymbol{z}$ fully determines the lattice point set.

Lattice point set with $d = 2$, $N = 34$, and $\boldsymbol{z} = (1, 21)$:



Worst-case error of a lattice rule with generating vector $\boldsymbol{z}$:

$$e^2_{N,d,\alpha,\gamma}(\boldsymbol{z}) = \sum_{\substack{\boldsymbol{0}\neq\boldsymbol{h}\in\mathbb{Z}^d \\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \bmod N}} (\rho_{\alpha,\gamma}(\boldsymbol{h}))^{-1}.$$

Explicit formula for the (squared) worst-case error for product weights:

$$e^2_{N,d,\alpha,\gamma}(\boldsymbol{z}) = -1 + \frac{1}{N}\sum_{n=0}^{N-1}\prod_{j=1}^{d}\left(1+\gamma_j\varphi_\alpha\left(\left\{\frac{nz_j}{N}\right\}\right)\right),$$

where $\varphi_\alpha\left(\frac{k}{N}\right)$ can be computed for all values of $k = 0,\ldots,N-1$.

The error formula is easy to implement.

Question: is the Korobov space interesting?

- Yes, first reason: it is a model function space that makes it easier to understand how lattice rules work.
- Yes, second reason:

  Bounds on the worst-case error of lattice rules in the Korobov space with $\alpha = 2$ immediately yield bounds on the worst-case error of slightly modified lattice rules in certain unanchored Sobolev spaces.

- All that remains is to find “good” $\boldsymbol{z} \in \{1, \ldots, N-1\}^d$.
- Huge search space of size $(N-1)^d$. (e.g., $N = 10\,000$ and $d = 100$).
- Component by component (CBC) construction: greedy algorithm to construct z_j one at a time.
  Size of search space is $N-1$ per component.
  (Almost) optimal convergence of error bounds (Kuo, 2003).

Algorithm 1 (CBC construction)

Let N be given. Construct $\boldsymbol{z} = (z_1, \ldots, z_d)$ as follows.

- *Set $z_1 = 1$.*
- *For $s \in \{2, \ldots, d\}$ assume that $z_1, \ldots, z_{s-1}$ have already been found. Now choose $z_s \in \{1, \ldots, N-1\}$ such that*

$$e^2_{N,s,\alpha,\gamma}((z_1, \ldots, z_{s-1}, z_s))$$

  is minimized as a function of z_s.
- *Increase s and repeat the second step until $(z_1, \ldots, z_d)$ is found.*

- Can do fast CBC (Cools, Nuyens, 2006): computation cost of $\mathcal{O}(dN \log N)$.
- Computation cost of $\mathcal{O}(dN \log N)$ can still be demanding for big N, d.
- Might want to have big N, d simultaneously. $\rightarrow$ **Can we speed up the search?**

The reduced CBC construction
(joint work with J. Dick, G. Leobacher, F. Pillichshammer)

Idea: small weights for later components $\rightarrow$ make search space smaller for later components z_j.

- Let N be a prime power, $N = b^m$, b prime, $m \in \mathbb{N}$.
- Let $w_1, w_2, \ldots \in \mathbb{N}_0$ with $0 = w_1 \le w_2 \le \cdots$.
- Consider the sequence of reduced search spaces

$$\mathcal{Z}_{N,w_j} := \begin{cases} \{1 \le z < b^{m-w_j} : \gcd(z,N) = 1\} & \text{if} \quad w_j < m, \\ \{1\} & \text{if} \quad w_j \ge m. \end{cases}$$

- Note that

$$|\mathcal{Z}_{N,w_j}| := \begin{cases} b^{m-w_j-1}(b-1) & \text{if} \quad w_j < m, \\ 1 & \text{if} \quad w_j \ge m. \end{cases}$$

- write $Y_j := b^{w_j}$.

Algorithm 2 (Reduced CBC construction)

Let N, $w_1, \ldots, w_d$, and $Y_1, \ldots, Y_d$ be as above. Construct $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_d z_d)$ as follows.

- *Set $z_1 = 1$.*
- *For $s \in \{2, \ldots, d\}$ assume that $z_1, \ldots, z_{s-1}$ have already been found. Now choose $z_s \in \mathcal{Z}_{N,w_s}$ such that*

$$e^2_{N,s,\alpha,\gamma}((Y_1 z_1, \ldots, Y_{s-1} z_{s-1}, Y_s z_s))$$

  is minimized as a function of z_s.
- *Increase s and repeat the second step until $(Y_1 z_1, \ldots, Y_d z_d)$ is found.*

Usual CBC construction: $w_j = 0$ and $Y_j = 1$ for all j.

Theorem 3 (Dick/K./Leobacher/Pillichshammer, 2015)

Let $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_d z_d) \in \mathbb{Z}^d$ *be constructed according to the reduced CBC algorithm. Then,*

$$e_{N,d,\alpha,\gamma}((Y_1 z_1, \ldots, Y_d z_d)) \leq$$
$$\leq N^{-\alpha/2+\delta} \left(2 \prod_{j=1}^{d} \left(1 + \gamma_j^{\frac{1}{\alpha-2\delta}} 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right) b^{w_j} \right) \right)^{\alpha/2-\delta}$$

for all $\delta \in \left(0, \frac{\alpha-1}{2}\right]$*, where* ζ *is the Riemann zeta function.*

Theorem formulated for product weights, similar result holds for general weights.

If

$$B := \sum_{j=1}^{\infty} \gamma_j^{\frac{1}{\alpha - 2\delta}} b^{w_j} < \infty,$$

then the error bound is independent of the dimension.

- Fast CBC construction (Nuyens/Cools) for non-reduced case ($w_j = 0$) has computation cost of $\mathcal{O}(dN \log N)$.
- Idea also works for the reduced case (assuming product weights), yields reduced cost by exploiting additional structure of the case $w_j > 0$.
- Bonus: once $w_j \geq m$ the search space contains only one element, construction of additional components incurs no extra cost.
- Computational cost of the reduced fast CBC construction is

$$\mathcal{O}\left(N \log N + \min\{d, d^*\} N + \sum_{j=1}^{\min\{d,d^*\}} (m - w_j) N b^{-w_j} \right),$$

where $d^* := \max\{j \in \mathbb{N} : w_j < m\}$.

Computation times (in seconds) for $b = 2$, $\alpha = 2$, $\gamma_j = 0.7^j$:

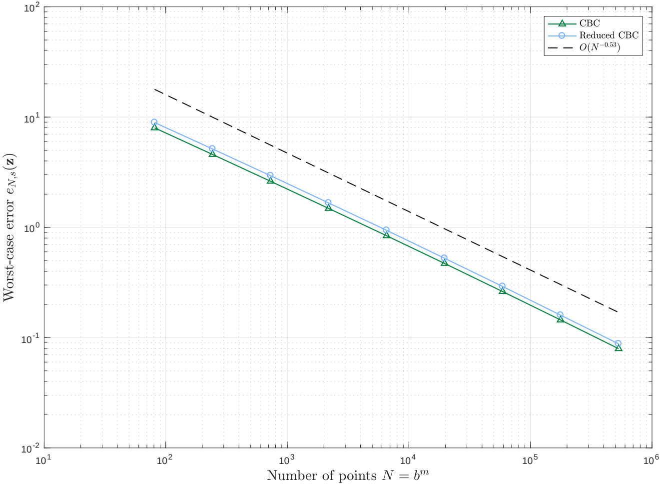
$w_j = 0$ (non-reduced)

$w_j = \lfloor 3 \log_b j \rfloor$ (reduced)

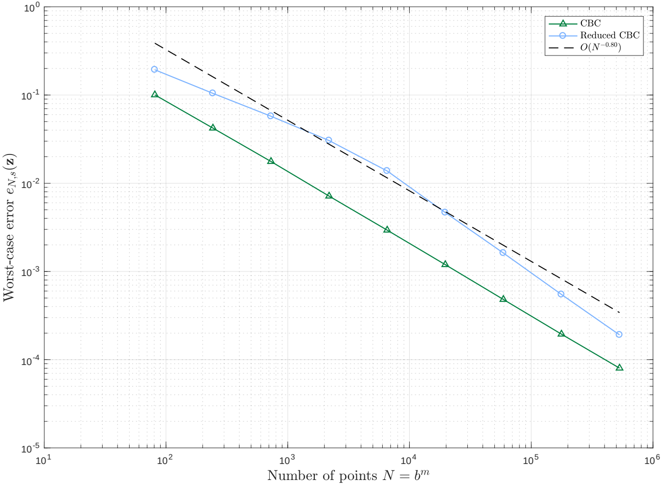
	$d = 100$	$d = 500$	$d = 1000$	d^*
$m = 10$	0.0329 0.0021	0.1600 0.0022	0.3230 0.0022	10
$m = 14$	0.0851 0.0076	0.4380 0.0105	0.8560 0.0071	25
$m = 18$	0.8320 0.0897	4.4400 0.0915	8.5400 0.0910	63
$m = 20$	4.1700 0.6230	21.5000 0.6430	42.4000 0.6360	101

fast CBC (top), reduced fast CBC (bottom).

Error convergence:
$b = 3$, $\alpha = 2$, $d = 100$, $\gamma_j = 0.8^j$:



Error convergence:
$b = 3$, $\alpha = 2$, $d = 100$, $\gamma_j = j^{-3}$:



The reduced fast CBC construction for POD weights (joint work with A. Ebert, D. Nuyens)

Recent result:

- Adaption of the reduced fast CBC construction to POD weights (motivated by a question by Ch. Schwab).
- POD weights: product and order-dependent weights, relevant for applications in PDEs with random coefficients:

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{j \in \mathfrak{u}} \gamma_j .$$

- Reduced fast CBC construction for this class of weights.
- Yields a slight improvement for the special case of product weights.

Theorem 4 (Ebert/K./Nuyens, 2019)

Given a sequence $0 \leq w_1 \leq w_2 \leq \cdots$ *and POD weights we can construct a lattice rule with* $N = b^m$ *points in* d *dimensions, whose worst-case error has almost optimal convergence, with construction cost of order*

$$\mathcal{O}\left(\sum_{j=1}^{\min\{d,d^*\}} (m - w_j + j)\, b^{m-w_j}\right),$$

The memory cost is $\mathcal{O}(\sum_{j=1}^{\min\{d,d^*\}} b^{m-w_j})$.

In case of product weights we need $\mathcal{O}(\sum_{j=1}^{\min\{d,d^*\}} (m - w_j)\, b^{m-w_j})$ *for the construction with memory* $\mathcal{O}(b^{m-w_1})$.

The successive coordinate search (SCS) construction

Question: can one improve on the quality of the output vector of the CBC construction?

Ebert/Leövey/Nuyens (2016): successive coordinate search (SCS) construction.

Basic idea:

- Assume product weights.
- Begin with a start vector $\boldsymbol{z}^{(0)} = (z_1^{(0)}, \ldots, z_d^{(0)})$.
- Update components one after the other by minimizing worst-case error.
- Returned vector is at least as good as initial vector.

Algorithm 5 (SCS construction)

Let N be a prime. Let $\boldsymbol{z}^{(0)} = (z_1^{(0)}, \ldots, z_d^{(0)}) \in \{0, 1, \ldots, N-1\}^d$ be given.

- *For $s \in \{1, \ldots, d\}$ assume that $z_1, \ldots, z_{s-1}$ have already been found. Now choose $z_s \in \{1, \ldots, N-1\}$ such that*

$$e^2_{N,d,\alpha,\gamma}((z_1, \ldots, z_{s-1}, z_s, z_{s+1}^{(0)}, \ldots, z_d^{(0)}))$$

  is minimized as a function of z_s.
- *Increase s and repeat the second step until $\boldsymbol{z} = (z_1, \ldots, z_d)$ is found.*

Theorem 6 (Ebert/Leövey/Nuyens, 2016)

Let $\boldsymbol{z}^{(1)}$ be constructed by the fast CBC algorithm.
Set $\boldsymbol{z}^{(0)} := \boldsymbol{z}^{(1)}$ in the SCS algorithm.
Let $\boldsymbol{z}$ be the vector returned by SCS. Then

$$e^2_{N,d,\alpha,\gamma}(\boldsymbol{z}) \le e^2_{N,d,\alpha,\gamma}(\boldsymbol{z}^{(1)}).$$

Theorem 7 (Ebert/Leövey/Nuyens, 2016)

Set $\boldsymbol{z}^{(0)} := (0, 0, \ldots, 0)$ in the SCS construction.
Then the SCS construction yields the same result as usual CBC construction. SCS is thus a generalization of CBC.

- SCS yields improvements over the CBC construction for pre-asymptotically moderately decreasing weights, e.g.

$$\gamma_j = 0.95^j.$$

- There is a fast implementation of the SCS construction:
  - Pre-computation with cost of $\mathcal{O}(dN)$,
  - matrix-vector multiplication of same speed as in CBC,
  - total cost of $\mathcal{O}(dN \log N)$.

The reduced fast SCS construction (joint work with A. Ebert)

Combine advantages of reduced fast CBC construction and fast SCS construction:
reduced fast SCS construction.

Same setting as for reduced fast CBC construction, brings similar results:

- Reduction in computation times if weights decay fast,
- similar error convergence behavior.
- Modify SCS construction to prime-power N,
- Generalization to arbitrary (not only product) weights.

The CBC-DBD construction
(ongoing joint work with A. Ebert, D. Nuyens, O.O. Osisiogu)

Slightly different function class (based on Korobov's original work):

$$E_{d,\gamma}^{\alpha} := \Big\{ f \in L^2([0,1]^d) \mid \exists\, C > 0 \text{ such that } |\hat{f}(\boldsymbol{h})| \le C\,(\rho_{\alpha,\gamma}(\boldsymbol{h}))^{-1}\ \forall\, \boldsymbol{h} \in \mathbb{Z}^d \Big\}.$$

$E_{d,\gamma}^{\alpha}$ is a Banach space with norm given by

$$\|f\|_{E_{d,\gamma}^{\alpha}} := \sup_{\boldsymbol{h} \in \mathbb{Z}^d} |\hat{f}(\boldsymbol{h})|\, \rho_{\alpha,\gamma}(\boldsymbol{h}).$$

Worst-case error of a lattice rule with generating vector $\boldsymbol{z}$:

$$\widetilde{e}_{N,d,\alpha,\gamma}(\boldsymbol{z}) = \sum_{\substack{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^d \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \bmod N}} (\rho_{\alpha,\gamma}(\boldsymbol{h}))^{-1}.$$

This yields a connection between $E_{d,\gamma}^{\alpha}$ and $(\mathcal{H}_{d,\alpha,\gamma}, \|\cdot\|_{d,\alpha,\gamma})$:

$$\widetilde{e}_{N,d,\alpha,\gamma}(\boldsymbol{z}) = e_{N,d,\alpha,\gamma}^{2}(\boldsymbol{z}).$$

Results on

$$\widetilde{e}_{N,d,\alpha,\gamma}(\boldsymbol{z}) \quad (\text{for } E_{d,\gamma}^{\alpha})$$

thus automatically imply results on

$$e_{N,d,\alpha,\gamma}^{2}(\boldsymbol{z}) \quad (\text{for } \mathcal{H}_{d,\alpha,\gamma}, \|\cdot\|_{d,\alpha,\gamma}).$$

Recall our goal: Efficiently construct a “good” generating vector

$$\boldsymbol{z} = (z_1, \ldots, z_d) \in \{0, \ldots, N-1\}^d$$

of a rank-1 lattice rule.

Assume $N = 2^n$ for some $n \in \mathbb{N}$.

For $s \in \{1, \ldots, d\}$ we write

$$z_s = z_{s,1} + z_{s,2}\, 2 + \cdots + z_{s,n}\, 2^{n-1}$$

with $z_{s,v} \in \{0, 1\}$.

- **Idea by Korobov:** Construct the digits $z_{s,v} \in \{0, 1\}$, $1 \leq v \leq n$, one after the other.
- This yields a digit-by-digit (DBD) construction.
- Go through the DBD-construction component-wise.
- This gives a CBC-DBD construction.
- Construction is extensible in d, but not in the number of digits (points).

To this end: Define a complicated quality function:

Let $x \in \mathbb{N}$ be odd, and $n, s \in \mathbb{N}$, and let $\gamma = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1,\ldots,d\}}$ be positive weights. For $1 \le v \le n$ and $1 \le s \le d$ let

$$h_{s,v,\gamma}(x) := \sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m \equiv 1 \bmod 2}}^{2^k} \left(\sum_{\emptyset \ne \mathfrak{u} \subseteq \{1,\ldots,s-1\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \ln \frac{1}{\sin^2(\pi m a_{j,n}/2^k)} \right.$$
$$\left. + \sum_{\mathfrak{w} \subseteq \{1,\ldots,j-1\}} \gamma_{\mathfrak{w} \cup \{s\}} \left(\prod_{j \in \mathfrak{w}} \ln \frac{1}{\sin^2(\pi m a_{j,n}/2^k)} \right) \ln \frac{1}{\sin^2(\pi m x/2^v)} \right).$$

Use $h_{s,v,\gamma}$ in the following CBC-DBD construction.

Algorithm 8 (CBC-DBD construction)

Given $N = 2^n$, $d \in \mathbb{N}$, *and* $\gamma = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1,\dots,d\}}$.

- *Set* $z_1 = 1$ *and* $a_{2,1} = \cdots = a_{d,1} = 1$.
- *Outer loop (CBC): for* $s \in \{2, \dots, d\}$:
- *Inner loop (DBD): for* $v \in \{2, \dots, n\}$:
- *Choose* $z_{s,v} = \operatorname{argmin}_{z \in \{0,1\}} h_{s,v,\gamma}(a_{s,v-1} + 2^{v-1} z)$.
- *Set* $a_{s,v} = a_{s,v-1} + 2^{v-1} z_{s,v}$.
- *Increase* v; *repeat inner loop until* $v = n$.
- *Increase* s; *repeat outer loop until* $s = d$.

Set $z_s := a_{s,n}$ *for* $2 \leq s \leq d$.

The CBC-DBD construction yields almost optimal results:

Adaption of proof idea by Korobov (1960s):

We write

$$M_d := \{-(N-1), \ldots, N-1\}^d \setminus \{\mathbf{0}\}$$

Note that

$$\begin{aligned} \widetilde{e}_{N,d,\alpha,\gamma}(\boldsymbol{z}) &= \sum_{\substack{\mathbf{0} \neq \boldsymbol{h} \in \mathbb{Z}^d \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \bmod N}} \frac{1}{\rho_{\alpha,\gamma}(\boldsymbol{h})} \\ &= \sum_{\substack{\boldsymbol{h} \in M_d \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \bmod N}} \frac{1}{\rho_{\alpha,\gamma}(\boldsymbol{h})} + \sum_{\substack{\boldsymbol{h} \notin M_d \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \bmod N}} \frac{1}{\rho_{\alpha,\gamma}(\boldsymbol{h})}. \end{aligned}$$

We can easily show:

Proposition 9

Let $\boldsymbol{z} = (z_1, \ldots, z_d) \in \mathbb{Z}^d$ *with* $\gcd(z_j, N) = 1$ *for all* $j = 1, \ldots, d$. *Then, for* $\alpha > 1$ *we have*

$$\sum_{\substack{\boldsymbol{h} \notin M_d \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \bmod N}} \frac{1}{\rho_{\alpha,\gamma}(\boldsymbol{h})} \leq \frac{1}{N^\alpha} \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1,\ldots,d\}} \gamma_{\mathfrak{u}}^{\alpha} (4\zeta(\alpha))^{|\mathfrak{u}|}.$$

The CBC-DBD algorithm constructs a generating vector $\boldsymbol{z} = (z_1, \ldots, z_d)$ such that there exist constants $C(\gamma, \delta)$ with

$$\sum_{\emptyset \neq \mathfrak{u} \subseteq \{1,\ldots,d\}} \gamma_{\mathfrak{u}} \sum_{\substack{\boldsymbol{h}_{\mathfrak{u}} \in M_{|\mathfrak{u}|} \\ \boldsymbol{h}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}} \equiv 0 \bmod N}} \frac{1}{\prod_{j \in \mathfrak{u}} |h_j|} \le C(\gamma, \delta) N^{-1+\delta}$$

for any $\delta > 0$, if the weights γ satisfy a suitable summability condition.

This result then yields

Theorem 10 (Ebert/K./Osisiogu/Nuyens, 2019)

Let $N = 2^n$, and let $\gamma = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1,\ldots,d\}}$ be a sequence of positive weights satisfying

$$\sum_{\substack{\mathfrak{u} \subseteq \mathbb{N} \\ |\mathfrak{u}| < \infty}} |\mathfrak{u}| \, (\gamma_{\mathfrak{u}})^{1/|\mathfrak{u}|} < \infty$$

Let $\boldsymbol{z} = (z_1, \ldots, z_d)$ be the generating vector constructed by the CBC-DBD algorithm. Then for any $\delta > 0$ and $\alpha > 1$ we have

$$\widetilde{e}_{N,d,\gamma^\alpha,\alpha}(\boldsymbol{z}) \leq \frac{1}{N^\alpha} \left(\sum_{\emptyset \neq \mathfrak{u} \subseteq \{1,\ldots,d\}} \gamma_u^\alpha (4\zeta(\alpha))^{|\mathfrak{u}|} + C(\gamma, \delta) N^{\alpha\delta} \right)$$

with $\gamma^\alpha = (\gamma_{\mathfrak{u}}^\alpha)_{\mathfrak{u} \subseteq \{1,\ldots,d\}}$ and some positive constant $C(\gamma, \delta)$ independent of d and N.

Note that:

- The proof technique is such that the CBC-DBD algorithm is independent of α.
- Rather elementary methods can be applied to ensure a run-time of $\mathcal{O}(dN \log N)$ of the CBC-DBD algorithm.

Niederreiter & Pillichshammer, 2009:

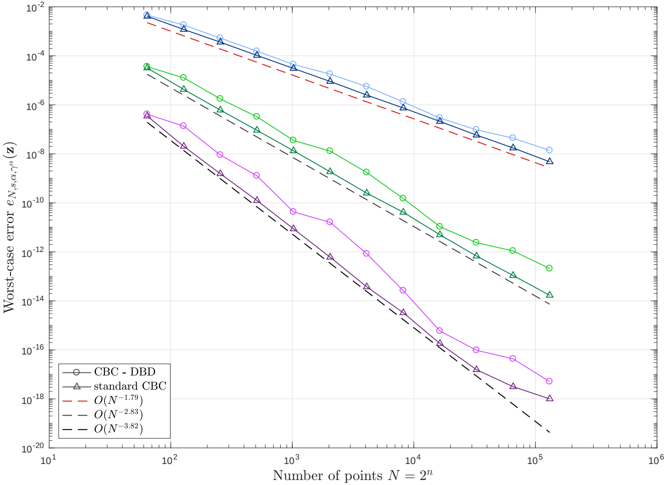
- Idea of DBD construction for extensible (polynomial) lattice rules.
- Rules extensible in the number of points N, but not in dimension d (CBC-DBD construction: extensible in d, but not in N).
- Convergence rate not optimal (CBC-DBD-construction: almost optimal convergence rate possible).

Future research:

Can we obtain better convergence rate for extensible rules using Korobov's original method?

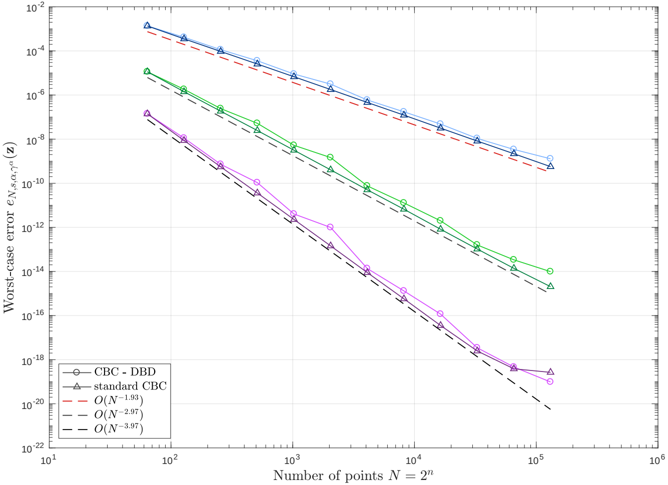
Can we obtain an algorithm extensible in N *and* d?

Error convergence:
$N = 2^n$, $d = 100$, $\gamma_j = j^{-2}$, $\alpha = 2$, $\alpha = 3$, $\alpha = 4$:

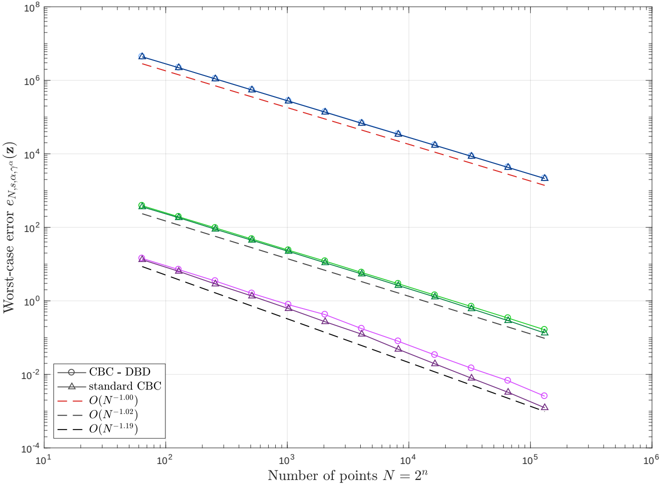


Error convergence:

$N = 2^n$, $d = 100$, $\gamma_j = j^{-3}$, $\alpha = 2, \alpha = 3, \alpha = 4$:

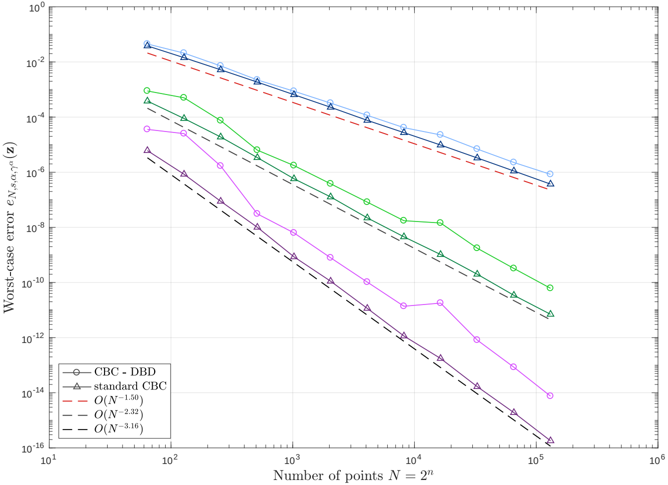


Error convergence:
$N = 2^n$, $d = 100$, $\gamma_j = 0.95^j$, $\alpha = 2$, $\alpha = 3$, $\alpha = 4$:



Error convergence:

$N = 2^n$, $d = 100$, $\gamma_j = 0.7^j$, $\alpha = 2$, $\alpha = 3$, $\alpha = 4$:



Polynomial lattice rules:

- Polynomial lattice rules: similar to lattice rules, but integer arithmetic replaced by polynomial arithmetic over finite fields.
- Special cases of Niederreiter's (t, m, d)-nets, powerful QMC methods.
- CBC, fast CBC, reduced fast CBC, SCS, fast SCS, reduced fast SCS work analogously for polynomial lattice rules.
- CBC-DBD construction for polynomial lattice rules: in progress.

Conclusion

- Component-by-component (CBC) construction of (polynomial) lattice points is a standard method in modern QMC theory.
- Fast CBC construction by Nuyens/Cools needs $\mathcal{O}(dN \log N)$ operations.
- We can reduce the construction cost depending on the coordinate weights, sometimes obtain independence of the dimension.
- The (fast) SCS construction can improve on the quality of generating vectors, as compared to the CBC construction. We can reduce the run-time of the SCS algorithm, similarly to that of the CBC construction.
- CBC-DBD construction seems to be a useful alternative to previous algorithms.

Thanks for your attention.