Construction of (polynomial) lattice rules by smoothness-independent component-by-component digit-by-digit constructions

Talk at the Point Distributions Webinar

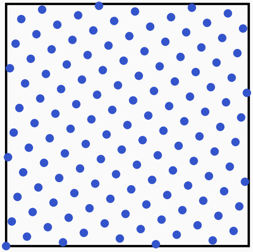
Adrian Ebert, RICAM, Linz

Joint research with D. Nuyens, P. Kritzer, O. Osisiogu and T. Stepaniuk

November 4, 2020

Table of contents

Lattice rules and periodic functions



Multivariate numerical integration

Approximate the integral of an s-variate function $f : [0,1]^s \to \mathbb{R}$

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

over the s-dimensional unit cube by a quasi-Monte Carlo (QMC) rule, i.e.,

$$I(f) = \int_{[0,1]^s} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \approx \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{x}_k) =: Q_N(f, \{\boldsymbol{x}_k\}_{k=0}^{N-1})$$

with deterministically chosen quadrature nodes $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\} \subset [0,1]^s$.

Multivariate numerical integration

Approximate the integral of an s-variate function $f : [0,1]^s \to \mathbb{R}$

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

over the s-dimensional unit cube by a quasi-Monte Carlo (QMC) rule, i.e.,

$$I(f) = \int_{[0,1]^s} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \approx \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{x}_k) =: Q_N(f, \{\boldsymbol{x}_k\}_{k=0}^{N-1})$$

with deterministically chosen quadrature nodes $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\} \subset [0,1]^s$.

Worst-case error

Let $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ be a Banach space and Q_N be a QMC rule with underlying point-set $P_N = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\} \subset [0,1]^s$. The *worst-case error* of Q_N w.r.t. $\mathcal{F}$ is defined as

$$e_{N,s}(Q_N, \mathcal{F}) := \sup_{\|f\|_{\mathcal{F}} \le 1} \left| \int_{[0,1]^s} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{x}_k) \right|.$$

Rank-1 lattice rule

A *rank-1 lattice rule* is a quasi-Monte Carlo rule with quadrature node set $P_N \subset [0,1]^s$ of the form

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;\middle|\; 0 \le k < N \right\} \subset [0,1]^s,$$

where $\boldsymbol{z} \in \mathbb{Z}^s$ is called the generating vector of the lattice rule.

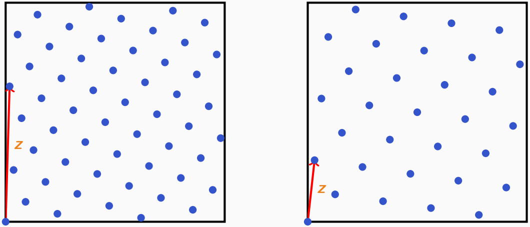


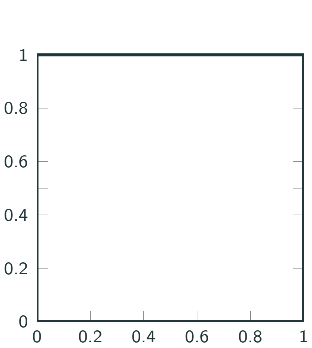
Figure 1: Fibonacci lattice with $N = 55$ and $\boldsymbol{z} = (1, 34)$ (left) and a rank-1 lattice with $N = 32$ and $\boldsymbol{z} = (1, 9)$ constructed by the CBC construction (right)

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



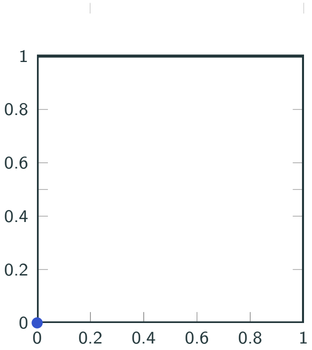
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



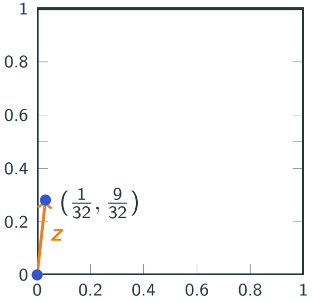
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \ : \ k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



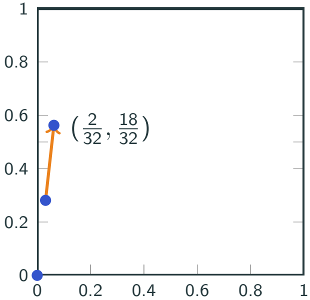
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



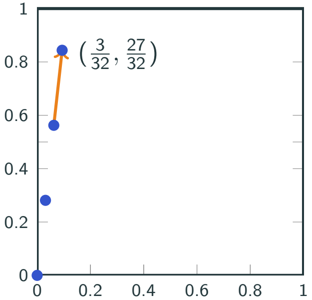
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



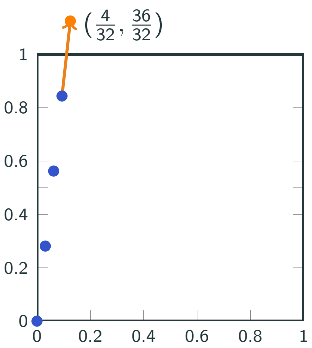
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



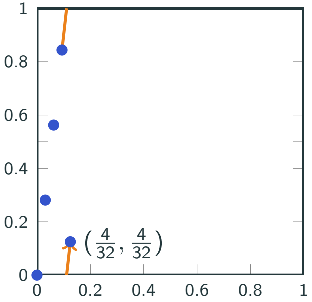
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



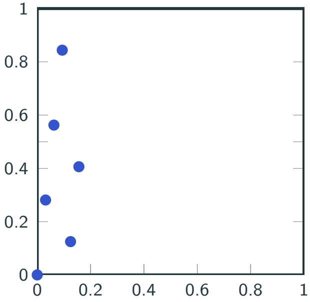
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



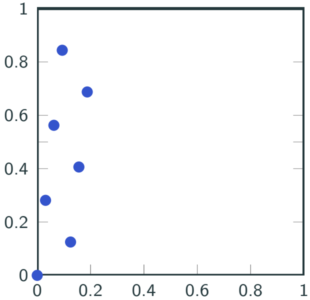
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \,:\, k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



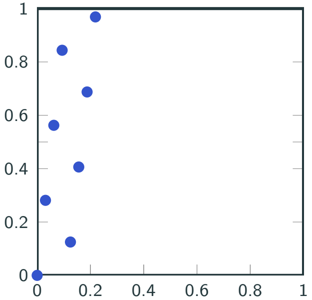
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \dots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



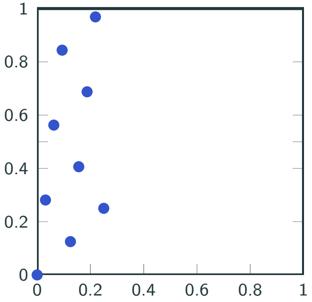
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



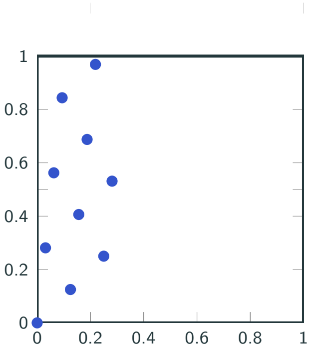
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



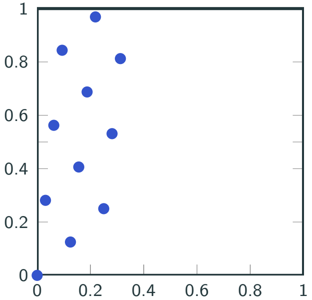
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



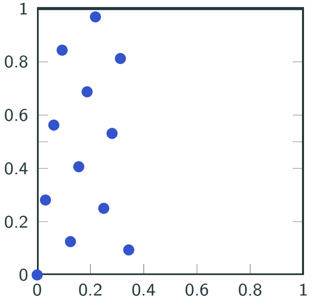
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



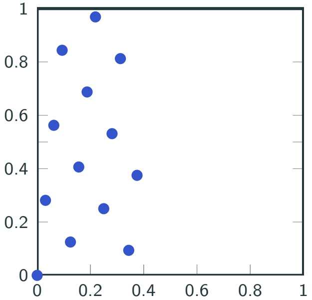
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



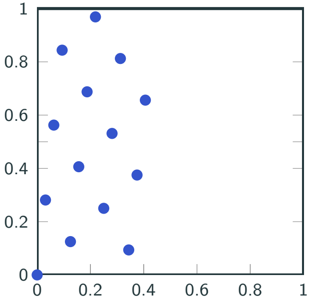
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



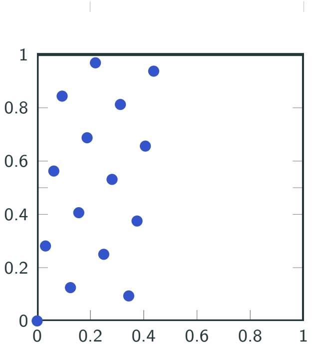
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



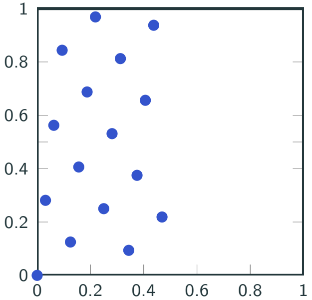
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



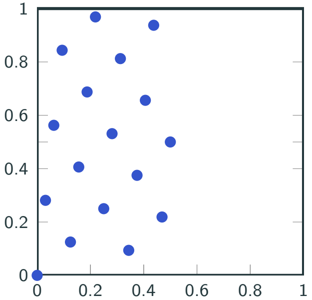
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



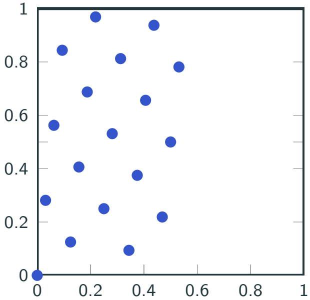
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \,:\, k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



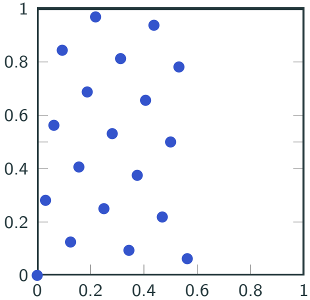
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \ : \ k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



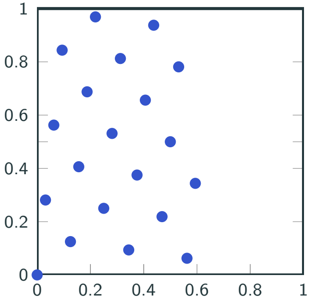
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



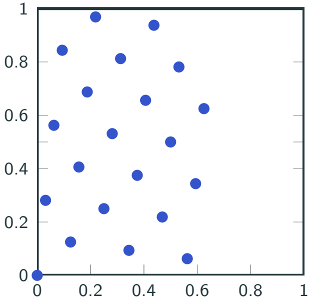
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



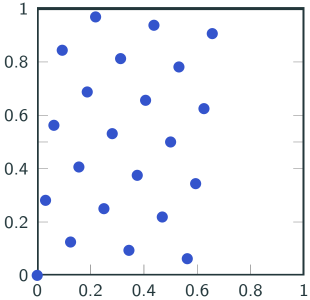
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \ : \ k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



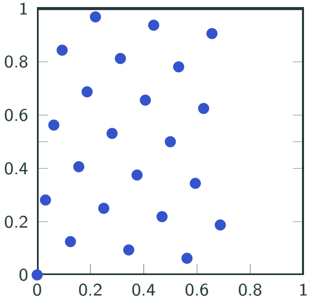
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \dots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



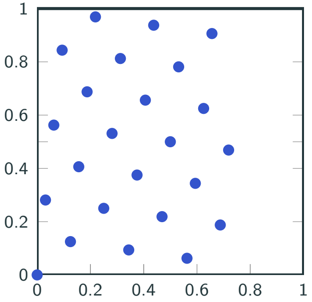
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



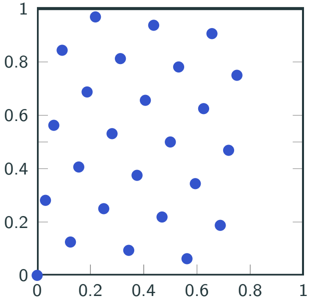
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \ : \ k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



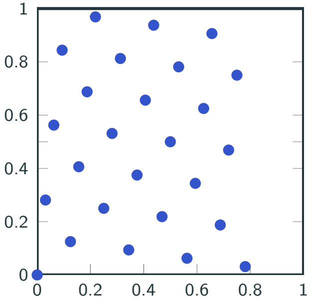
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \; : \; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



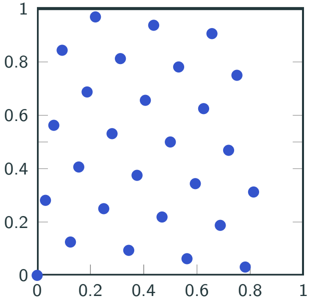
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



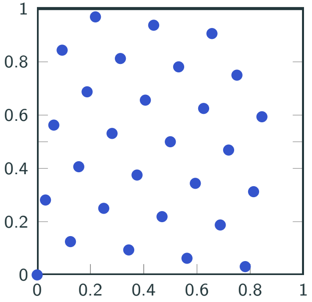
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



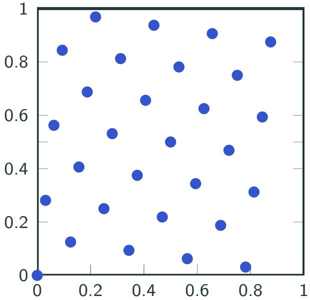
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



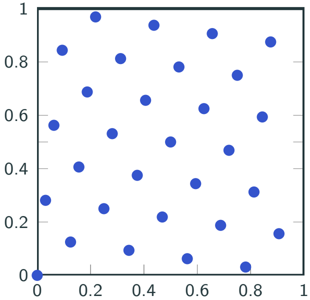
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



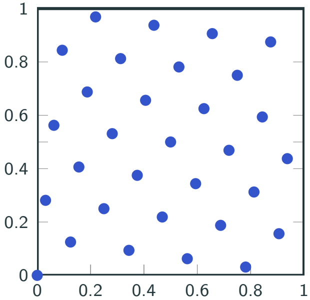
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \dots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



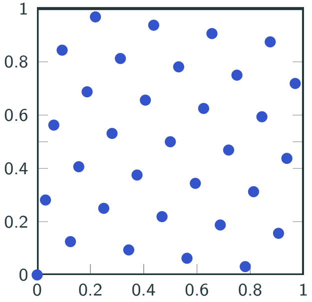
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Rank-1 lattice node set

The point set of a rank-1 lattice rule is given via

$$P_N = \left\{ \frac{k\boldsymbol{z} \bmod N}{N} \;:\; k = 0, 1, 2, \ldots, N-1 \right\}$$

with $\boldsymbol{z} \in \mathbb{Z}^s$ and can be constructed in the following way:



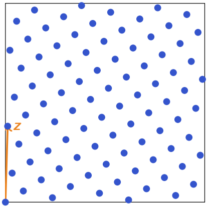
Lattice with $N = 32$ points and generating vector $\boldsymbol{z} = (1, 9)$

Good lattice, bad lattice

- Generating vector $\boldsymbol{z} \in \mathbb{Z}^s$ influences quality of lattice rule

Good lattice, bad lattice

- Generating vector $\boldsymbol{z} \in \mathbb{Z}^s$ influences quality of lattice rule

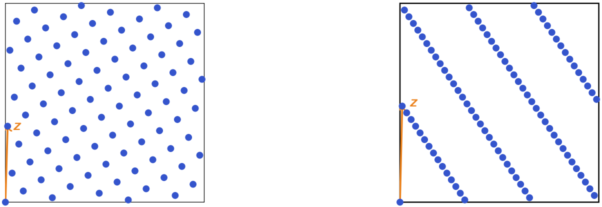


(a) well-distributed lattice

Illustration of rank-1 lattices with generating vectors $\boldsymbol{z} = (1, 34)$ and $N = 89$ points (left)

Good lattice, bad lattice

- Generating vector $\boldsymbol{z} \in \mathbb{Z}^s$ influences quality of lattice rule

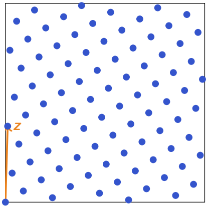


(a) well-distributed lattice (b) badly distributed lattice

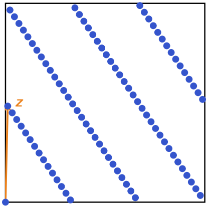
Illustration of rank-1 lattices with generating vectors $\boldsymbol{z} = (1, 34)$ and $N = 89$ points (left) and $\boldsymbol{z} = (1, 43)$ with $N = 89$ points (right)

Good lattice, bad lattice

- Generating vector $\boldsymbol{z} \in \mathbb{Z}^s$ influences quality of lattice rule



(a) well-distributed lattice



(b) badly distributed lattice

Illustration of rank-1 lattices with generating vectors $\boldsymbol{z} = (1, 34)$ and $N = 89$ points (left) and $\boldsymbol{z} = (1, 43)$ with $N = 89$ points (right)

- **Goal:** Find good generating vectors $\boldsymbol{z} \in \mathbb{Z}^s$ such that the obtained rank-1 lattice rules $Q_N(\cdot, \boldsymbol{z})$ are suited for numerical integration.

Lattice rule integration error

Consider 1-periodic, continuous functions $f : [0,1]^s \to \mathbb{R}$ with associated absolutely convergent Fourier series

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \hat{f}(\boldsymbol{h}) e^{2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{x}} \quad \text{with} \quad \hat{f}(\boldsymbol{h}) := \int_{[0,1]^s} f(\boldsymbol{x}) e^{-2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{x}} \,\mathrm{d}\boldsymbol{x}.$$

Lattice rule integration error

Consider 1-periodic, continuous functions $f : [0,1]^s \to \mathbb{R}$ with associated absolutely convergent Fourier series

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^s} \hat{f}(\boldsymbol{h}) e^{2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{x}} \quad \text{with} \quad \hat{f}(\boldsymbol{h}) := \int_{[0,1]^s} f(\boldsymbol{x}) e^{-2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{x}} \,\mathrm{d}\boldsymbol{x}.$$

The integration error of a lattice rule $Q_N(\cdot, \boldsymbol{z})$ then equals

$$Q_N(f, \boldsymbol{z}) - I(f) = \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \hat{f}(\boldsymbol{h}) \left[\frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi \mathrm{i} k \boldsymbol{h}\cdot\boldsymbol{z}/N} \right] = \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \hat{f}(\boldsymbol{h})\, \delta_N(\boldsymbol{h}\cdot\boldsymbol{z})$$

with indicator function for the dual lattice $\{\boldsymbol{h} \in \mathbb{Z}^s \mid \boldsymbol{h}\cdot\boldsymbol{z} \equiv 0 \pmod{N}\}$

$$\delta_N(m) := \begin{cases} 1, & \text{if } m \equiv 0 \pmod{N}, \\ 0, & \text{if } m \not\equiv 0 \pmod{N}. \end{cases}$$

Function space setting

It is then common to consider function spaces of periodic functions whose Fourier coefficients $\hat{f}(\boldsymbol{h})$ decay sufficiently fast.

The decay of the $\hat{f}(\boldsymbol{h})$ is measured by a decay function $r_\alpha(\boldsymbol{h})$ of the form

$$r_\alpha(h) := \begin{cases} 1, & \text{if } h = 0, \\ |h|^\alpha, & \text{if } h \neq 0 \end{cases} \qquad \text{and} \qquad r_\alpha(\boldsymbol{h}) := \prod_{j=1}^{s} r_\alpha(h_j)$$

with smoothness parameter $\alpha > 1$.

Function space setting

It is then common to consider function spaces of periodic functions whose Fourier coefficients $\hat{f}(\boldsymbol{h})$ decay sufficiently fast.

The decay of the $\hat{f}(\boldsymbol{h})$ is measured by a decay function $r_\alpha(\boldsymbol{h})$ of the form

$$r_\alpha(h) := \begin{cases} 1, & \text{if } h = 0, \\ |h|^\alpha, & \text{if } h \neq 0 \end{cases} \quad \text{and} \quad r_\alpha(\boldsymbol{h}) := \prod_{j=1}^{s} r_\alpha(h_j)$$

with smoothness parameter $\alpha > 1$.

In order to overcome the curse of dimensionality, we additionally introduce so-called weights $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$ which measure the importance of (groups of) variables $\boldsymbol{x}_{\mathfrak{u}} := (x_j)_{j \in \mathfrak{u}}$:

$$r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h}) := \gamma_{\mathrm{supp}(\boldsymbol{h})}^{-1} \prod_{j \in \mathrm{supp}(\boldsymbol{h})} |h_j|^\alpha .$$

Function space setting

It is then common to consider function spaces of periodic functions whose Fourier coefficients $\hat{f}(\boldsymbol{h})$ decay sufficiently fast.

The decay of the $\hat{f}(\boldsymbol{h})$ is measured by a decay function $r_\alpha(\boldsymbol{h})$ of the form

$$r_\alpha(h) := \begin{cases} 1, & \text{if } h = 0, \\ |h|^\alpha, & \text{if } h \neq 0 \end{cases} \quad \text{and} \quad r_\alpha(\boldsymbol{h}) := \prod_{j=1}^{s} r_\alpha(h_j)$$

with smoothness parameter $\alpha > 1$.

In order to overcome the curse of dimensionality, we additionally introduce so-called weights $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$ which measure the importance of (groups of) variables $\boldsymbol{x}_{\mathfrak{u}} := (x_j)_{j \in \mathfrak{u}}$:

$$r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h}) := \gamma_{\mathrm{supp}(\boldsymbol{h})}^{-1} \prod_{j \in \mathrm{supp}(\boldsymbol{h})} |h_j|^\alpha .$$

- Relation between $r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})$ and mixed partial derivatives $f^{(\boldsymbol{\tau})}, \boldsymbol{\tau} \in \mathbb{N}_0^s$

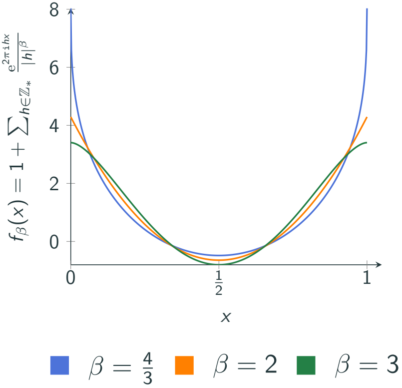
Function space setting

Define the norm of the Banach space $E^\alpha_{s,\boldsymbol{\gamma}}$ as

$$\|f\|_{E^\alpha_{s,\boldsymbol{\gamma}}} := \sup_{\boldsymbol{h}\in\mathbb{Z}^s} |\hat{f}(\boldsymbol{h})|\, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})$$

and for $\alpha > 1$ and positive weights define the weighted function space

$$E^\alpha_{s,\boldsymbol{\gamma}} := \{f \in L^2([0,1]^s) \mid \|f\|_{E^\alpha_{s,\boldsymbol{\gamma}}} < \infty\}.$$



Function space setting

Applying Hölder's inequality with $p = \infty$ and $q = 1$ to the integration error (see previous slide) yields

$$
\begin{aligned}
|Q_N(f, \boldsymbol{z}) - I(f)| &= \left| \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \hat{f}(\boldsymbol{h})\, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})\, r_{\alpha,\boldsymbol{\gamma}}^{-1}(\boldsymbol{h})\, \delta_N(\boldsymbol{h} \cdot \boldsymbol{z}) \right| \\
&\leq \underbrace{\left(\sup_{\boldsymbol{h} \in \mathbb{Z}^s} |\hat{f}(\boldsymbol{h})| r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h}) \right)}_{=: \|f\|_{E^{\alpha}_{s,\boldsymbol{\gamma}}}} \underbrace{\left(\sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \frac{\delta_N(\boldsymbol{h} \cdot \boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} \right)}_{= e_{N,s}(Q_N(\cdot,\boldsymbol{z}), E^{\alpha}_{s,\boldsymbol{\gamma}})} .
\end{aligned}
$$

Function space setting

Applying Hölder's inequality with $p = \infty$ and $q = 1$ to the integration error (see previous slide) yields

$$|Q_N(f, \boldsymbol{z}) - I(f)| = \left| \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \hat{f}(\boldsymbol{h})\, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})\, r_{\alpha,\boldsymbol{\gamma}}^{-1}(\boldsymbol{h})\, \delta_N(\boldsymbol{h} \cdot \boldsymbol{z}) \right|$$

$$\leq \underbrace{\left(\sup_{\boldsymbol{h} \in \mathbb{Z}^s} |\hat{f}(\boldsymbol{h})| r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h}) \right)}_{=: \|f\|_{E^\alpha_{s,\boldsymbol{\gamma}}}} \underbrace{\left(\sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \frac{\delta_N(\boldsymbol{h} \cdot \boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} \right)}_{= e_{N,s}(Q_N(\cdot, \boldsymbol{z}), E^\alpha_{s,\boldsymbol{\gamma}})}.$$

Function space setting

Applying Hölder's inequality with $p = \infty$ and $q = 1$ to the integration error (see previous slide) yields

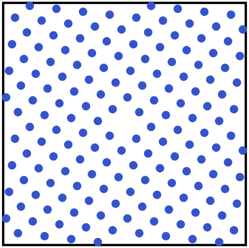
$$|Q_N(f,\boldsymbol{z}) - I(f)| = \left| \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \hat{f}(\boldsymbol{h})\, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})\, r_{\alpha,\boldsymbol{\gamma}}^{-1}(\boldsymbol{h})\, \delta_N(\boldsymbol{h} \cdot \boldsymbol{z}) \right|$$
$$\leq \underbrace{\left(\sup_{\boldsymbol{h} \in \mathbb{Z}^s} |\hat{f}(\boldsymbol{h})| r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h}) \right)}_{=: \|f\|_{E^\alpha_{s,\boldsymbol{\gamma}}}} \underbrace{\left(\sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \frac{\delta_N(\boldsymbol{h} \cdot \boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} \right)}_{= e_{N,s}(Q_N(\cdot,\boldsymbol{z}), E^\alpha_{s,\boldsymbol{\gamma}})}.$$

Theorem (Lattice rule worst-case error)

Let $N, s \in \mathbb{N}$, $\alpha > 1$ and a sequence of positive weights $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$ be given. Then the worst-case error $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ for the rank-1 lattice rule $Q_N(\cdot, \boldsymbol{z})$ in the space $E^\alpha_{s,\boldsymbol{\gamma}}$ satisfies

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) := e_{N,s}(Q_N(\cdot, \boldsymbol{z}), E^\alpha_{s,\boldsymbol{\gamma}}) = \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^s} \frac{\delta_N(\boldsymbol{h} \cdot \boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})}.$$

Quality measure and optimal coefficients

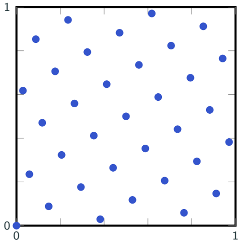


Quality measure $T_\alpha(N, z)$

For $\alpha \geq 1$ we introduce the quality measure

$$T_\alpha(N, \boldsymbol{z}) := \sum_{\substack{\boldsymbol{0} \neq \boldsymbol{h} \in M_{N,s} \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \pmod{N}}} \frac{1}{r_{\alpha,\gamma}(\boldsymbol{h})} = \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in M_{N,s}} \frac{\delta_N(\boldsymbol{h} \cdot \boldsymbol{z})}{r_{\alpha,\gamma}(\boldsymbol{h})}$$

with truncated index set $M_{N,s} = \{-(N-1), \ldots, N-1\}^s$.



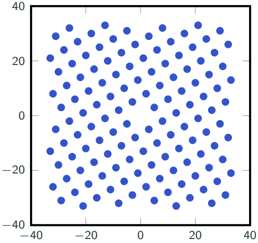


Figure 2: Fibonacci lattice with $N = 34$ and $\boldsymbol{z} = (1, 21)$ (left) with the corresponding set of $\boldsymbol{0} \neq \boldsymbol{h} \in M_{N,s}$ with $\boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \pmod{N}$ (right)

Connection with the worst-case error

The difference between $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ and its restriction to $M_{N,s}$ satisfies:

Lemma (Truncation error)

Let $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$ *be a sequence of positive weights and let* $\boldsymbol{z} \in \mathbb{Z}^s$ *with* $\gcd(z_j, N) = 1$ *for all* $j = 1, \ldots, s$. *Then, for* $\alpha > 1$, *we have that*

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) - T_\alpha(N, \boldsymbol{z}) \leq \frac{1}{N^\alpha} \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \, (4\zeta(\alpha))^{|\mathfrak{u}|}.$$

Connection with the worst-case error

The difference between $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ and its restriction to $M_{N,s}$ satisfies:

Lemma (Truncation error)

Let $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u}\subseteq\{1:s\}}$ *be a sequence of positive weights and let* $\boldsymbol{z} \in \mathbb{Z}^s$ *with* $\gcd(z_j, N) = 1$ *for all* $j = 1, \ldots, s$. *Then, for* $\alpha > 1$, *we have that*

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) - T_\alpha(N, \boldsymbol{z}) \leq \frac{1}{N^\alpha} \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \left(4\zeta(\alpha)\right)^{|\mathfrak{u}|}.$$

Under the same assumptions we obtain

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) = \sum_{\mathbf{0}\neq\boldsymbol{h}\in\mathbb{Z}^s} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} - \sum_{\mathbf{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} + \sum_{\mathbf{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})}$$

Connection with the worst-case error

The difference between $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ and its restriction to $M_{N,s}$ satisfies:

Lemma (Truncation error)

Let $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u}\subseteq\{1:s\}}$ *be a sequence of positive weights and let* $\boldsymbol{z} \in \mathbb{Z}^s$ *with* $\gcd(z_j, N) = 1$ *for all* $j = 1, \ldots, s$*. Then, for* $\alpha > 1$*, we have that*

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) - T_\alpha(N, \boldsymbol{z}) \leq \frac{1}{N^\alpha} \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \left(4\zeta(\alpha)\right)^{|\mathfrak{u}|}.$$

Under the same assumptions we obtain

$$\begin{aligned} e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) &= \sum_{\boldsymbol{0}\neq\boldsymbol{h}\in\mathbb{Z}^s} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} - \sum_{\boldsymbol{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} + \sum_{\boldsymbol{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} \\ &\leq \frac{1}{N^\alpha} \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \left(4\zeta(\alpha)\right)^{|\mathfrak{u}|} + \sum_{\boldsymbol{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} \end{aligned}$$

Connection with the worst-case error

The difference between $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ and its restriction to $M_{N,s}$ satisfies:

Lemma (Truncation error)

Let $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u}\subseteq\{1:s\}}$ be a sequence of positive weights and let $\boldsymbol{z} \in \mathbb{Z}^s$ with $\gcd(z_j, N) = 1$ for all $j = 1, \ldots, s$. Then, for $\alpha > 1$, we have that

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) - T_\alpha(N, \boldsymbol{z}) \leq \frac{1}{N^\alpha} \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \left(4\zeta(\alpha)\right)^{|\mathfrak{u}|}.$$

Under the same assumptions we obtain (using Jensen's inequality)

$$\begin{aligned} e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) &= \sum_{\mathbf{0}\neq\boldsymbol{h}\in\mathbb{Z}^s} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} - \sum_{\mathbf{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} + \sum_{\mathbf{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{h})} \\ &\leq \frac{1}{N^\alpha} \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \left(4\zeta(\alpha)\right)^{|\mathfrak{u}|} + \left(\sum_{\mathbf{0}\neq\boldsymbol{h}\in M_{N,s}} \frac{\delta_N(\boldsymbol{h}\cdot\boldsymbol{z})}{r_{1,\boldsymbol{\gamma}^{1/\alpha}}(\boldsymbol{h})} \right)^\alpha . \end{aligned}$$

Optimal coefficients modulo N

For the limiting case $\alpha = 1$, we analogously introduce the quality measure

$$T(N, \boldsymbol{z}) := \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in M_{N,s}} \frac{\delta_N(\boldsymbol{h} \cdot \boldsymbol{z})}{r_{1,\gamma}(\boldsymbol{h})}$$

as a quality criterion for good rank-1 lattice rules.

[1] N.Korobov. *Number-theoretic methods in approximate analysis*. Fizmatigiz, 1963.
N.Korobov. *On the computation of optimal coefficients*. Dokl. Akad. Nauk., 1982.

Optimal coefficients modulo N

For the limiting case $\alpha = 1$, we analogously introduce the quality measure

$$T(N, \boldsymbol{z}) := \sum_{\boldsymbol{0} \neq \boldsymbol{h} \in M_{N,s}} \frac{\delta_N(\boldsymbol{h} \cdot \boldsymbol{z})}{r_{1,\gamma}(\boldsymbol{h})}$$

as a quality criterion for good rank-1 lattice rules.

As in Korobov works[1], we introduce the concept of optimal coefficients.

Definition (Optimal coefficients modulo N)

For given $N \in \mathbb{N}$ and positive weights $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$, the components $z_1, \ldots, z_s$ of $\boldsymbol{z}$ are called optimal coefficients modulo N if for any $\delta > 0$ it holds that

$$T(N, \boldsymbol{z}) \leq C(\boldsymbol{\gamma}, \delta) N^{-1+\delta},$$

where $C(\boldsymbol{\gamma}, \delta)$ is a positive constant independent of s and N.

[1] N.Korobov. *Number-theoretic methods in approximate analysis*. Fizmatigiz, 1963.
N.Korobov. *On the computation of optimal coefficients*. Dokl. Akad. Nauk., 1982.

The componentwise digit-by-digit algorithm

The construction of good rank-1 lattice rules

An exhaustive search for good generating vectors $\boldsymbol{z} \in \{0, 1, \ldots, N-1\}^s$ such that the worst-case error $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ for our function space is small, is infeasible since the search space has size $\mathcal{O}(N^s)$.

Therefore, different search algorithms were introduced:

[2] N.Korobov. *On the computation of optimal coefficients*. Dokl. Akad. Nauk., 1982.

The construction of good rank-1 lattice rules

An exhaustive search for good generating vectors $\boldsymbol{z} \in \{0, 1, \ldots, N-1\}^s$ such that the worst-case error $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ for our function space is small, is infeasible since the search space has size $\mathcal{O}(N^s)$.

Therefore, different search algorithms were introduced:

- Korobov (1963) and later Sloan and Reztsov (2002) introduced a component-by-component (CBC) construction to find good generating vectors $\boldsymbol{z}$. (Greedy algorithm with complexity $\mathcal{O}(s\,N^2)$ and search space size reduced to $\mathcal{O}(s\,N)$)

[2] N.Korobov. *On the computation of optimal coefficients*. Dokl. Akad. Nauk., 1982.

The construction of good rank-1 lattice rules

An exhaustive search for good generating vectors $\boldsymbol{z} \in \{0, 1, \ldots, N-1\}^s$ such that the worst-case error $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ for our function space is small, is infeasible since the search space has size $\mathcal{O}(N^s)$.

Therefore, different search algorithms were introduced:

- Korobov (1963) and later Sloan and Reztsov (2002) introduced a component-by-component (CBC) construction to find good generating vectors $\boldsymbol{z}$. (Greedy algorithm with complexity $\mathcal{O}(s\,N^2)$ and search space size reduced to $\mathcal{O}(s\,N)$)
- The introduction of the fast CBC construction by Nuyens and Cools (2006) reduced the complexity of the algorithm to $\mathcal{O}(s\,N \ln N)$.

[2]N.Korobov. *On the computation of optimal coefficients*. Dokl. Akad. Nauk., 1982.

The construction of good rank-1 lattice rules

An exhaustive search for good generating vectors $\boldsymbol{z} \in \{0, 1, \ldots, N-1\}^s$ such that the worst-case error $e_{N,s,\alpha,\gamma}(\boldsymbol{z})$ for our function space is small, is infeasible since the search space has size $\mathcal{O}(N^s)$.

Therefore, different search algorithms were introduced:

- Korobov (1963) and later Sloan and Reztsov (2002) introduced a component-by-component (CBC) construction to find good generating vectors $\boldsymbol{z}$. (Greedy algorithm with complexity $\mathcal{O}(s\,N^2)$ and search space size reduced to $\mathcal{O}(s\,N)$)
- The introduction of the fast CBC construction by Nuyens and Cools (2006) reduced the complexity of the algorithm to $\mathcal{O}(s\,N\ln N)$.
- We will explore a different construction algorithm which originates from an article by Korobov[2] (1982, $3\frac{1}{2}$ pages long).

[2] N.Korobov. *On the computation of optimal coefficients*. Dokl. Akad. Nauk., 1982.

For $x \in (0,1)$ consider the Fourier series of the function $-2\ln(\sin(\pi x))$

$$-2\ln(\sin(\pi x)) = \ln(4) + \sum_{h \in \mathbb{Z}\setminus\{0\}} \frac{e^{2\pi \mathtt{i} h x}}{|h|}.$$

The relation to the error expression motivates us to define the quality function for our componentwise digit-by-digit (CBC-DBD) algorithm.

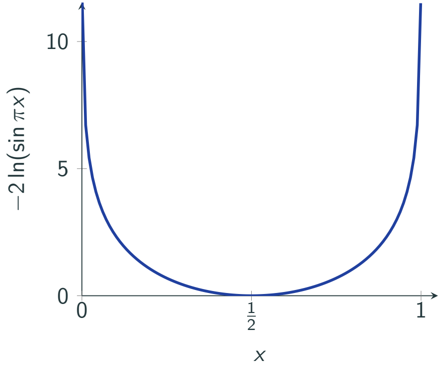


Figure 3: Behavior of the function $-2\ln(\sin \pi x)$ on the interval $[0,1]$.

Formulation of the CBC-DBD construction

Definition (Digit-wise quality function)

Let $x \in \mathbb{N}$ be an odd integer, $n, s \in \mathbb{N}$ be positive integers, and let $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$ be a sequence of positive weights. For $1 \leq v \leq n$ and $1 \leq r \leq s$ and positive integers $z_1, \ldots, z_{r-1}$, we define the quality function $h_{r,v,\boldsymbol{\gamma}} : \mathbb{Z} \to \mathbb{R}$ as

$$h_{r,v,\boldsymbol{\gamma}}(x) := \sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m \equiv 1 \pmod 2}}^{2^k} \Bigg[\sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:r-1\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \ln \frac{1}{\sin^2(\pi m z_j / 2^k)} \\ + \sum_{\mathfrak{w} \subseteq \{1:r-1\}} \gamma_{\mathfrak{w} \cup \{r\}} \left(\prod_{j \in \mathfrak{w}} \ln \frac{1}{\sin^2(\pi m z_j / 2^k)} \right) \ln \frac{1}{\sin^2(\pi m x / 2^v)} \Bigg]$$

Based on $h_{r,v,\boldsymbol{\gamma}}$ the component-wise digit-by-digit (CBC-DBD) algorithm can be formulated as follows.

Formulation of the CBC-DBD construction

Algorithm 1 Component-wise digit-by-digit construction

Input: Integer $n \in \mathbb{N}$, dimension s and positive weights $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$.

Set $z_{1,n} = 1$ and $z_{2,1} = \ldots = z_{s,1} = 1$.

for $r = 2$ **to** s **do**

for $v = 2$ **to** n **do**

$$z^* = \underset{z \in \{0,1\}}{\operatorname{argmin}}\, h_{r,v,\boldsymbol{\gamma}}(z_{r,v-1} + 2^{v-1} z)$$

$$z_{r,v} = z_{r,v-1} + 2^{v-1} z^*$$

end for

end for

Set $\boldsymbol{z} = (z_1, \ldots, z_s)$ with $z_r := z_{r,n}$ for $r = 1, \ldots, s$.

Return: Generating vector $\boldsymbol{z} = (z_1, \ldots, z_s)$ for $N = 2^n$.

The resulting vector $\boldsymbol{z} = (z_1, \ldots, z_s)$ is the generating vector of a lattice rule with $N = 2^n$ points in s dimensions.

Illustration of the CBC-DBD algorithm

- The generating vector $\boldsymbol{z}$ is constructed component-by-component, where each component is build up digit-by-digit.

$$\xleftarrow{\quad v=2,\dots,n \quad}$$

$$\text{r=2,...,s}\ \Big\downarrow \qquad \begin{array}{cccccc}
z_{1,n} = 1 & \cdots & z_{1,v} & \cdots & z_{1,2} & z_{1,1} = 1 \\
z_{2,n} & \cdots & z_{2,v} & \cdots & z_{2,2} & z_{2,1} = 1 \\
z_{3,n} & \cdots & z_{3,v} & \cdots & z_{3,2} & z_{3,1} = 1 \\
\vdots & & \vdots & & \vdots & \vdots \\
z_{r,n} & \cdots & z_{r,v} & \cdots & z_{r,2} & z_{r,1} = 1 \\
\vdots & & \vdots & & \vdots & \vdots \\
z_{s,n} & \cdots & z_{s,v} & \cdots & z_{s,2} & z_{s,1} = 1 \\
\vdots & & \vdots & & \vdots & \vdots
\end{array}$$

- The size of the search space is of order $\mathcal{O}(2ns) = \mathcal{O}(s \ln N)$.
- The construction is extensible in the dimension s.
- Naïve implementation has time complexity $\mathcal{O}(s^2 N \ln N)$.

Error convergence behavior (main result)

Theorem (A.E., P.Kritzer, D.Nuyens, O.Osisiogu)

Let $N = 2^n$ and $(\gamma_{\mathfrak{u}})_{\mathfrak{u}\subseteq\{1:s\}}$, with $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$ and $\gamma_j > 0$, be product weights. Then the corresponding generating vector $\boldsymbol{z}$, constructed by Algorithm 1, satisfies the following estimate:

$$T(N,\boldsymbol{z}) \le \frac{1}{N}\left[\prod_{j=1}^{s}\left(1+\gamma_j(\ln 4 + 2(1+\ln N))\right) + 2(1+\ln N)\prod_{j=1}^{s}\left(1+\gamma_j(2(1+2\ln N))\right)\right].$$

Error convergence behavior (main result)

Theorem (A.E., P.Kritzer, D.Nuyens, O.Osisiogu)

Let $N = 2^n$ and $(\gamma_{\mathfrak{u}})_{\mathfrak{u}\subseteq\{1:s\}}$, with $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$ and $\gamma_j > 0$, be product weights. Then the corresponding generating vector $\boldsymbol{z}$, constructed by Algorithm 1, satisfies the following estimate:

$$T(N,\boldsymbol{z}) \le \frac{1}{N}\left[\prod_{j=1}^{s}(1+\gamma_j(\ln 4 + 2(1+\ln N))) + 2(1+\ln N)\prod_{j=1}^{s}(1+\gamma_j(2(1+2\ln N)))\right].$$

Moreover, if the weights $(\gamma_j)_{j=1}^{s}$ satisfy the condition

$$\sum_{j=1}^{\infty}\gamma_j < \infty,$$

then $T(N,\boldsymbol{z})$ is bounded independently of the dimension s and $z_1,\dots,z_s$ are optimal coefficients modulo N.

Error convergence behavior (main result)

Theorem (A.E., P.Kritzer, D.Nuyens, O.Osisiogu)

Let $N = 2^n$ *and denote by* $\boldsymbol{z} = (z_1, \dots, z_s)$ *the generating vector constructed by Algorithm 1. If the weights* $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ *satisfy the condition*

$$\sum_{j=1}^{\infty} \gamma_j < \infty,$$

then for any $\delta > 0$ *and each* $\alpha > 1$ *the worst-case error* $e_{N,s,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z})$ *satisfies*

$$e_{N,s,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z}) \leq \frac{1}{N^\alpha} \left(\prod_{j=1}^{s} \left(1 + \gamma_j^\alpha (4\zeta(\alpha))\right) + C(\boldsymbol{\gamma}, \delta) N^{\alpha\delta} \right)$$

with weight sequence $\boldsymbol{\gamma}^\alpha = (\gamma_{\mathfrak{u}}^\alpha)_{\mathfrak{u} \subseteq \{1:s\}}$ *and positive constant* $C(\boldsymbol{\gamma}, \delta)$ *independent of* s *and* N*.*

Fast implementation of the algorithm

Cost analysis

For the implementation we consider the special case of product weights $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ for a sequence of positive reals $(\gamma_j)_{j \geq 1}$.

Cost analysis

For the implementation we consider the special case of product weights $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ for a sequence of positive reals $(\gamma_j)_{j \geq 1}$.

The digit-wise quality function $h_{r,v,\boldsymbol{\gamma}}(x)$ then equals

$$\sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m \equiv 1 \pmod 2}}^{2^k} \prod_{j=1}^{r-1} \left(1 + \gamma_j \ln \frac{1}{\sin^2(\pi m z_j / 2^k)}\right) \left(1 + \gamma_r \ln \frac{1}{\sin^2(\pi m x / 2^v)}\right).$$

Cost analysis

For the implementation we consider the special case of product weights $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ for a sequence of positive reals $(\gamma_j)_{j \geq 1}$.

The digit-wise quality function $h_{r,v,\boldsymbol{\gamma}}(x)$ then equals

$$\sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m \equiv 1 \pmod 2}}^{2^k} \prod_{j=1}^{r-1} \left(1 + \gamma_j \ln \frac{1}{\sin^2(\pi m z_j / 2^k)}\right)\left(1 + \gamma_r \ln \frac{1}{\sin^2(\pi m x / 2^v)}\right).$$

A single evaluation of $h_{r,v,\boldsymbol{\gamma}}(x)$ requires $\mathcal{O}(r \sum_{k=v}^{n} 2^{k-1})$ operations. The total cost of each inner loop over the $v = 2, \ldots, n$ is therefore

$$\mathcal{O}\left(r \sum_{v=2}^{n} 2 \sum_{k=v}^{n} 2^{k-1}\right) = \mathcal{O}\left(r \left(2^n n - 2(2^n - 1)\right)\right) = \mathcal{O}\left(r\, N \ln N\right).$$

Cost analysis

For the implementation we consider the special case of product weights $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ for a sequence of positive reals $(\gamma_j)_{j \geq 1}$.

The digit-wise quality function $h_{r,v,\boldsymbol{\gamma}}(x)$ then equals

$$\sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m \equiv 1 \pmod 2}}^{2^k} \prod_{j=1}^{r-1} \left(1 + \gamma_j \ln \frac{1}{\sin^2(\pi m z_j / 2^k)}\right) \left(1 + \gamma_r \ln \frac{1}{\sin^2(\pi m x / 2^v)}\right).$$

A single evaluation of $h_{r,v,\boldsymbol{\gamma}}(x)$ requires $\mathcal{O}(r \sum_{k=v}^{n} 2^{k-1})$ operations. The total cost of each inner loop over the $v = 2, \ldots, n$ is therefore

$$\mathcal{O}\left(r \sum_{v=2}^{n} 2 \sum_{k=v}^{n} 2^{k-1}\right) = \mathcal{O}\left(r \left(2^n n - 2(2^n - 1)\right)\right) = \mathcal{O}\left(r\, N \ln N\right).$$

Thus, a naïve implementation of the CBC-DBD algorithm has time complexity $\mathcal{O}\left(s^2\, N \ln N\right)$.

Fast implementation

A fast implementation can be obtained by evaluating $h_{r,v,\boldsymbol{\gamma}}(x) =$

$$\sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m \equiv 1 \pmod 2}}^{2^k} \prod_{j=1}^{r-1} \left(1 + \gamma_j \ln \frac{1}{\sin^2(\pi m z_j / 2^k)}\right) \left(1 + \gamma_r \ln \frac{1}{\sin^2(\pi m x / 2^v)}\right)$$

in a more efficient manner.

Fast implementation

A fast implementation can be obtained by evaluating $h_{r,v,\boldsymbol{\gamma}}(x) =$

$$\sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m \equiv 1 \pmod 2}}^{2^k} \prod_{j=1}^{r-1} \left(1 + \gamma_j \ln \frac{1}{\sin^2(\pi m z_j / 2^k)}\right)\left(1 + \gamma_r \ln \frac{1}{\sin^2(\pi m x / 2^v)}\right)$$

in a more efficient manner.

For $1 \leq r < s$ let $z_1, \ldots, z_r$ be constructed by Algorithm 1. For $k \in \{2, \ldots, n\}$ and odd $m \in \{1, \ldots, 2^k - 1\}$ define the term $q(r, k, m)$ by

$$q(r, k, m) = \prod_{j=1}^{r} \left(1 + \gamma_j \ln \frac{1}{\sin^2(\pi m z_j / 2^k)}\right).$$

Fast implementation

A fast implementation can be obtained by evaluating $h_{r,v,\boldsymbol{\gamma}}(x) =$

$$\sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m\equiv 1 \pmod 2}}^{2^k} \prod_{j=1}^{r-1} \left(1+\gamma_j \ln \frac{1}{\sin^2(\pi m z_j/2^k)}\right)\left(1+\gamma_r \ln \frac{1}{\sin^2(\pi m x/2^v)}\right)$$

in a more efficient manner.

For $1 \leq r < s$ let $z_1, \ldots, z_r$ be constructed by Algorithm 1. For $k \in \{2, \ldots, n\}$ and odd $m \in \{1, \ldots, 2^k - 1\}$ define the term $q(r, k, m)$ by

$$q(r,k,m) = \prod_{j=1}^{r} \left(1+\gamma_j \ln \frac{1}{\sin^2(\pi m z_j/2^k)}\right).$$

This way, the function $h_{r,v,\boldsymbol{\gamma}}(x)$ can be rewritten as

$$h_{r,v,\boldsymbol{\gamma}}(x) = \sum_{k=v}^{n} \frac{1}{2^{k-v}} \sum_{\substack{m=1 \\ m\equiv 1 \pmod 2}}^{2^k} q(r-1,k,m)\left(1+\gamma_r \ln \frac{1}{\sin^2(\pi m x/2^v)}\right).$$

Fast implementation

We can thus compute and store $q(r-1,k,m)$ for all values of k and m at cost $\mathcal{O}(N)$ and compute $q(r,k,m)$ via the recurrence relation

$$q(r,k,m) = q(r-1,k,m)\left(1+\gamma_r \ln \frac{1}{\sin^2(\pi m\, z_r/2^k)}\right).$$

This way, a single evaluation of $h_{r,v,\gamma}(x)$ requires only $\mathcal{O}(\sum_{k=v}^{n} 2^{k-1})$ operations, each inner loop $\mathcal{O}\left(N \ln N\right)$ operations.

Fast implementation

We can thus compute and store $q(r-1,k,m)$ for all values of k and m at cost $\mathcal{O}(N)$ and compute $q(r,k,m)$ via the recurrence relation

$$q(r,k,m) = q(r-1,k,m)\left(1+\gamma_r \ln \frac{1}{\sin^2(\pi m\, z_r/2^k)}\right).$$

This way, a single evaluation of $h_{r,v,\gamma}(x)$ requires only $\mathcal{O}(\sum_{k=v}^{n} 2^{k-1})$ operations, each inner loop $\mathcal{O}\left(N \ln N\right)$ operations.

Theorem (Fast implementation)

Let $n, s \in \mathbb{N}$ and $N = 2^n$. For a given positive weight sequence $\gamma = (\gamma_j)_{j=1}^{s}$, a generating vector $\boldsymbol{z} = (z_1, \ldots, z_s)$ can be computed via Algorithm 1 using $\mathcal{O}(s\, N \ln N)$ operations and requiring $\mathcal{O}(N)$ memory.

This algorithm has time complexity $\mathcal{O}(s\, N \ln N)$ and does not require the use of fast Fourier transforms (FFTs)!

Numerical results

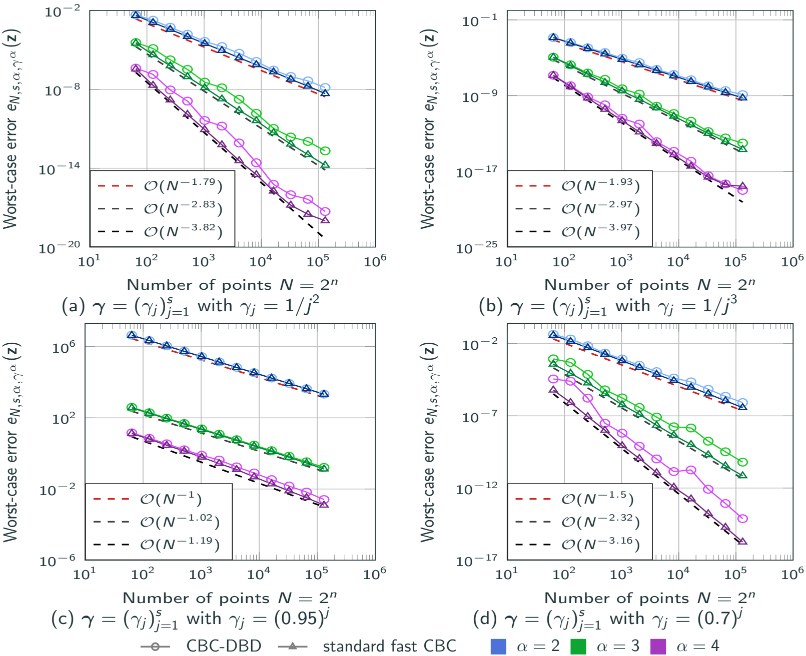
Error convergence behavior

Consider the convergence behavior of $e_{N,s,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z})$ for generating vectors constructed by the CBC-DBD algorithm and the fast CBC algorithm[3].

- Use product weights sequences $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$ with $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ and consider the worst-case error $e_{N,s,\alpha,\boldsymbol{\gamma}^\alpha}$ for $\alpha = 2, 3, 4$.
- The generators $\boldsymbol{z}_{\text{cbc-dbd}}$ are constructed by the CBC-DBD algorithm with n, s and weights $(\gamma_j)_{j=1}^s$ as input.
- The generators $\boldsymbol{z}_{\text{cbc}}$ are constructed by the fast CBC algorithm for $N = 2^n$ using the error $e_{N,s,\alpha,\boldsymbol{\gamma}^\alpha}$ as quality function.
- The error values of generators constructed by the standard fast CBC algorithm are used as a benchmark for our CBC-DBD construction.

[3] D. Nuyens, R. Cools. *Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points*. J. Complexity 22, 4–28, 2006.

Error convergence in the space $E^{\alpha}_{s,\boldsymbol{\gamma}}$ with $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$, $s = 100$, $\alpha = 2, 3, 4$.



Computation times

Table 1: Computation times (in seconds) for constructing generating vectors $\boldsymbol{z}$ of lattice rules with $N = 2^n$ points in s dimensions via the CBC-DBD algorithm (**bold font**) and the standard fast CBC construction (normal font). Constructed for weights of the form $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$. For the fast CBC construction the smoothness parameter $\alpha = 2$ was used.

	$s = 50$	$s = 100$	$s = 500$	$s = 1000$	$s = 2000$
$n = 10$	0.038 **0.061**	0.075 **0.119**	0.37 **0.595**	0.743 **1.184**	1.485 **2.371**
$n = 12$	0.047 **0.093**	0.096 **0.185**	0.476 **0.922**	0.951 **1.843**	1.897 **3.685**
$n = 14$	0.068 **0.155**	0.138 **0.31**	0.674 **1.547**	1.339 **3.081**	2.676 **6.166**
$n = 16$	0.165 **0.344**	0.304 **0.678**	1.423 **3.394**	2.845 **6.804**	5.626 **13.624**
$n = 18$	0.586 **1.145**	1.053 **2.293**	4.746 **11.63**	9.497 **23.1**	18.867 **46.184**
$n = 20$	3.357 **6.31**	6.203 **12.757**	28.935 **64.102**	57.438 **128.897**	114.284 **257.454**

Polynomial lattice rules

Walsh series representation

Consider functions $f : [0,1]^s \to \mathbb{R}$ given by their Walsh series

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{N}_0^s} \hat{f}(\boldsymbol{k})\,\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) \quad \text{with} \quad \hat{f}(\boldsymbol{k}) := \int_{[0,1]^s} f(\boldsymbol{x})\,\overline{\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}$$

with $\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^s \mathrm{wal}_{k_j}(x_j)$ and $\mathrm{wal}_k(x) = \mathrm{e}^{2\pi\mathtt{i}(\kappa_0\xi_1+\kappa_1\xi_2+\cdots+\kappa_{a-1}\xi_a)/b}$ for base b representations $k = \kappa_0 + \kappa_1 b + \cdots \kappa_{a-1}b^{a-1}$ and $x = \xi_1 b^{-1} + \xi_2 b^{-2} + \cdots$ with coefficients $\kappa_i, \xi_i \in \{0,1,\ldots,b-1\}$.

Walsh series representation

Consider functions $f : [0,1]^s \to \mathbb{R}$ given by their Walsh series

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{N}_0^s} \hat{f}(\boldsymbol{k}) \, \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) \quad \text{with} \quad \hat{f}(\boldsymbol{k}) := \int_{[0,1]^s} f(\boldsymbol{x}) \, \overline{\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}$$

with $\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^s \mathrm{wal}_{k_j}(x_j)$ and $\mathrm{wal}_k(x) = \mathrm{e}^{2\pi \mathtt{i}(\kappa_0 \xi_1 + \kappa_1 \xi_2 + \cdots + \kappa_{a-1}\xi_a)/b}$
for base b representations $k = \kappa_0 + \kappa_1 b + \cdots \kappa_{a-1} b^{a-1}$ and
$x = \xi_1 b^{-1} + \xi_2 b^{-2} + \cdots$ with coefficients $\kappa_i, \xi_i \in \{0, 1, \ldots, b-1\}$.

We introduce a function to measure the decay of the Walsh coefficients:

$$r_\alpha(\boldsymbol{k}) := \prod_{j=1}^s r_\alpha(k_j) \quad \text{and} \quad r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{k}) := \gamma_{\mathrm{supp}(\boldsymbol{k})}^{-1} \prod_{j \in \mathrm{supp}(\boldsymbol{k})} b^{\alpha \psi_b(k_j)}$$

with $\psi_b(k) = \lfloor \log_b(k) \rfloor$.

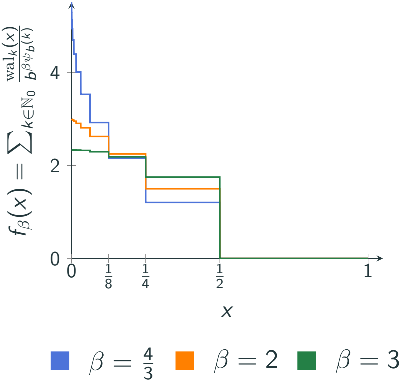
Weighted Walsh space

Define the norm of the Banach space $W^{\alpha}_{s,\boldsymbol{\gamma}}$ as

$$\|f\|_{W^{\alpha}_{s,\boldsymbol{\gamma}}} := \sup_{\boldsymbol{k}\in\mathbb{N}_0^s} |\hat{f}(\boldsymbol{k})|\, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{k})$$

and for $\alpha > 1$ and positive weights define the weighted function space

$$W^{\alpha}_{s,\boldsymbol{\gamma}} := \{f \in L^2([0,1]^s) \mid \|f\|_{W^{\alpha}_{s,\boldsymbol{\gamma}}} < \infty\}.$$



Polynomial lattice rules

Denote by $\mathbb{F}_b[x]$ the set of all polynomials over $\mathbb{F}_b$ and define the map $v_m : \mathbb{F}_b((x^{-1})) \to [0,1)$ by

$$v_m\left(\sum_{\ell=1}^{\infty} t_\ell\, x^{-\ell}\right) = \sum_{\ell=1}^{m} t_\ell\, b^{-\ell}.$$

For $n \in \mathbb{N}_0$ with base b expansion $n = n_0 + n_1 b + \cdots + n_a b^a$, we associate n with the polynomial $n(x) := \sum_{k=0}^{a} n_k\, x^k \in \mathbb{F}_b[x]$.

Polynomial lattice rules

Denote by $\mathbb{F}_b[x]$ the set of all polynomials over $\mathbb{F}_b$ and define the map $v_m : \mathbb{F}_b((x^{-1})) \to [0,1)$ by

$$v_m\left(\sum_{\ell=1}^{\infty} t_\ell\, x^{-\ell}\right) = \sum_{\ell=1}^{m} t_\ell\, b^{-\ell}.$$

For $n \in \mathbb{N}_0$ with base b expansion $n = n_0 + n_1 b + \cdots + n_a b^a$, we associate n with the polynomial $n(x) := \sum_{k=0}^{a} n_k\, x^k \in \mathbb{F}_b[x]$.

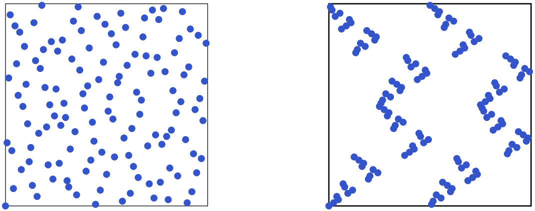
Polynomial lattice point set

Let b be prime and choose $p \in \mathbb{F}_b[x]$ with $\deg(p) = m$, and let $\boldsymbol{g} \in \mathbb{F}_b[x]$. Then the point set $P(\boldsymbol{g}, p)$, defined as the collection of the b^m points

$$\boldsymbol{x}_n := \left(v_m\left(\frac{n(x)\, g_1(x)}{p(x)}\right), \ldots, v_m\left(\frac{n(x)\, g_s(x)}{p(x)}\right)\right) \in [0,1)^s$$

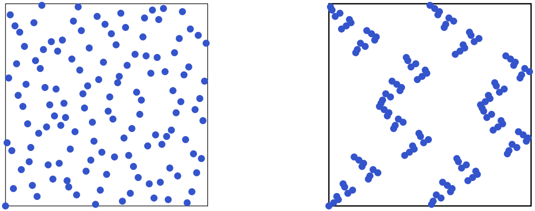
for $n \in \mathbb{F}_b[x]$ with $\deg(n) < m$, is called a polynomial lattice.

Integration error for PLR



Polynomial lattice node sets with 2^7 points in base $b = 2$ with irreducible polynomial $f = x^7 + x^3 + 1 \in \mathbb{F}_2[x]$ and the two generating vectors $\boldsymbol{g}_1 = (x^4 + x^2 + 1, x^2 + x)$ (left) and $\boldsymbol{g}_2 = (x^3 + 1, x^2 + x)$ (right).

Integration error for PLR



Polynomial lattice node sets with 2^7 points in base $b = 2$ with irreducible polynomial $f = x^7 + x^3 + 1 \in \mathbb{F}_2[x]$ and the two generating vectors $\boldsymbol{g}_1 = (x^4 + x^2 + 1, x^2 + x)$ (left) and $\boldsymbol{g}_2 = (x^3 + 1, x^2 + x)$ (right).

Also here the integration error can be represented in terms of the series coefficients, that is,

$$Q_{b^m}(f; P(\boldsymbol{g}, p)) - I(f) = \sum_{\mathbf{0} \neq \boldsymbol{k} \in \mathcal{D}(\boldsymbol{g}, p)} \hat{f}(\boldsymbol{k})$$

with dual net $\mathcal{D}(\boldsymbol{g}, p) = \{\boldsymbol{k} \in \mathbb{N}_0^s \mid \mathrm{tr}_m(\boldsymbol{k}) \cdot \boldsymbol{g} \equiv 0 \pmod{p}\}$.

Further strategy

- As for lattice rules, define the quantities

$$T(\boldsymbol{g},p) := \sum_{\mathbf{0}\neq\boldsymbol{k}\in A_p(\boldsymbol{g})} (r_{1,\gamma}(\boldsymbol{k}))^{-1}, \qquad T_\alpha(\boldsymbol{g},p) := \sum_{\mathbf{0}\neq\boldsymbol{k}\in A_p(\boldsymbol{g})} (r_{\alpha,\gamma}(\boldsymbol{k}))^{-1}$$

with index set given by $A_p(\boldsymbol{g}) = \{\boldsymbol{k} \in \{0,1,\ldots,b^m-1\}^s \mid \boldsymbol{k} \in \mathcal{D}(\boldsymbol{g},p)\}$.

Further strategy

- As for lattice rules, define the quantities

$$T(\boldsymbol{g},p) := \sum_{\mathbf{0}\neq\boldsymbol{k}\in A_p(\boldsymbol{g})} (r_{1,\boldsymbol{\gamma}}(\boldsymbol{k}))^{-1}, \qquad T_\alpha(\boldsymbol{g},p) := \sum_{\mathbf{0}\neq\boldsymbol{k}\in A_p(\boldsymbol{g})} (r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{k}))^{-1}$$

with index set given by $A_p(\boldsymbol{g}) = \{\boldsymbol{k} \in \{0,1,\ldots,b^m-1\}^s \mid \boldsymbol{k} \in \mathcal{D}(\boldsymbol{g},p)\}$.

- Relate the quality measure $T(\boldsymbol{g},p)$ to the worst-case error expression for polynomial lattice rules in the space $W_{s,\boldsymbol{\gamma}}^{\alpha}$.

Further strategy

- As for lattice rules, define the quantities

$$T(\boldsymbol{g},p) := \sum_{\mathbf{0}\neq\boldsymbol{k}\in A_p(\boldsymbol{g})} (r_{1,\gamma}(\boldsymbol{k}))^{-1}, \qquad T_\alpha(\boldsymbol{g},p) := \sum_{\mathbf{0}\neq\boldsymbol{k}\in A_p(\boldsymbol{g})} (r_{\alpha,\gamma}(\boldsymbol{k}))^{-1}$$

with index set given by $A_p(\boldsymbol{g}) = \{\boldsymbol{k} \in \{0,1,\ldots,b^m-1\}^s \mid \boldsymbol{k} \in \mathcal{D}(\boldsymbol{g},p)\}$.

- Relate the quality measure $T(\boldsymbol{g},p)$ to the worst-case error expression for polynomial lattice rules in the space $W_{s,\gamma}^\alpha$.

- Introduce the digit-wise quality function and formulate a component-by-component digit-by-digit construction algorithm.

Formulation of the CBC-DBD construction for PLRs

Definition (Digit-wise quality function)

Let $q \in \mathbb{F}_b[x]$, $m, s \in \mathbb{N}$, and let $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$ with $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ be product weights. For integers $w \in \{1 : m\}$, $r \in \{1 : s\}$, and polynomials $g_1, \ldots, g_{r-1} \in \mathbb{F}_b[x]$ with $\gcd(g_j, x) = 1$, we define the quality function $h_{r,w,m,\boldsymbol{\gamma}} : \mathbb{F}_b[x] \to \mathbb{R}$ as

$$
\begin{aligned}
&h_{r,w,m,\boldsymbol{\gamma}}(q) \\
&:= \sum_{t=w}^{m} \frac{1}{b^{t-w}} \sum_{\substack{\ell=1 \\ \ell \not\equiv 0 \pmod{b}}}^{b^t-1} \left(1 + \gamma_r (1-b) \left(\left\lfloor \log_b \left(v_w \left(\frac{\ell(x)\, q(x)}{x^w} \right) \right) \right\rfloor + 1 \right) \right) \times \\
&\quad \times \prod_{j=1}^{r-1} \left(1 + \gamma_j (1-b) \left(\left\lfloor \log_b \left(v_t \left(\frac{\ell(x)\, g_j(x)}{x^t} \right) \right) \right\rfloor + 1 \right) \right).
\end{aligned}
$$

Based on $h_{r,w,m,\boldsymbol{\gamma}}$ the component-wise digit-by-digit (CBC-DBD) algorithm for polynomial lattice rules can be formulated as follows.

Formulation of the CBC-DBD construction for PLRs

Algorithm 2 Component-wise digit-by-digit construction

Input: Integer $n \in \mathbb{N}$, dimension s and positive weights $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq \{1:s\}}$.

Set $g_{1,m} = 1$ and $g_{2,1} = \ldots = g_{s,1} = 1$.

for $r = 2$ **to** s **do**

  for $w = 2$ **to** m **do**

    $g^* = \underset{g \in \mathbb{F}_b}{\operatorname{argmin}}\ h_{r,w,m,\boldsymbol{\gamma}}(g_{r,w-1} + x^{w-1} g)$

    $g_{r,w} = g_{r,w-1} + g^* x^{w-1}$

  end for

end for

Set $\boldsymbol{g} = (g_1, \ldots, g_s)$ with $g_r := g_{r,m}$ for $r = 1, \ldots, s$.

Return: Generating vector $\boldsymbol{g} = (g_1, \ldots, g_s) \in (\mathbb{F}_b[x])^s$ with $\deg(g_j) < m$.

- For ease of computations, we fix $b = 2$ in the numerical experiments.

Error convergence behavior (main result)

Theorem (A.E., P.Kritzer, O.Osisiogu, T.Stepaniuk)

Let b be prime, let $m, s \in \mathbb{N}$ with $m \geq 4$, let $N = b^m$, and let $(\gamma_j)_{j\geq 1}$ be positive product weights satisfying

$$\sum_{j\geq 1} \gamma_j < \infty.$$

Also, denote by $\boldsymbol{g}$ the generating vector obtained by Algorithm 2, run for the weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$. Then, for any $\delta > 0$ and each $\alpha > 1$, the generating vector $\boldsymbol{g}$ satisfies

$$e_{b^m,s,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{g}) \leq \frac{1}{N^\alpha} \left(C(\boldsymbol{\gamma}^\alpha) + \bar{C}\,(\boldsymbol{\gamma}, \delta)\; N^{\alpha\delta} \right),$$

with positive constants $C(\boldsymbol{\gamma}^\alpha)$ and $\bar{C}\,(\boldsymbol{\gamma}, \delta)$, which are independent of the dimension s and the number of points N.

Error convergence behavior (main result)

Theorem (A.E., P.Kritzer, O.Osisiogu, T.Stepaniuk)

Let b be prime, let $m, s \in \mathbb{N}$ with $m \geq 4$, let $N = b^m$, and let $(\gamma_j)_{j\geq 1}$ be positive product weights satisfying

$$\sum_{j\geq 1} \gamma_j < \infty.$$

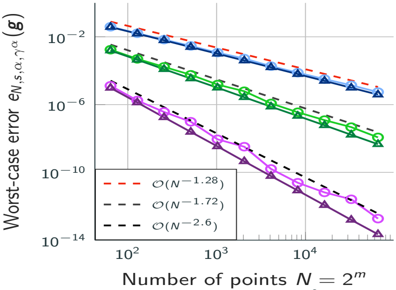
Also, denote by $\boldsymbol{g}$ the generating vector obtained by Algorithm 2, run for the weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$. Then, for any $\delta > 0$ and each $\alpha > 1$, the generating vector $\boldsymbol{g}$ satisfies

$$e_{b^m,s,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{g}) \leq \frac{1}{N^\alpha}\left(C(\boldsymbol{\gamma}^\alpha) + \bar{C}(\boldsymbol{\gamma},\delta)\, N^{\alpha\delta}\right),$$

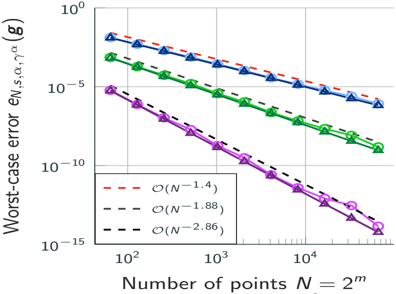
with positive constants $C(\boldsymbol{\gamma}^\alpha)$ and $\bar{C}(\boldsymbol{\gamma},\delta)$, which are independent of the dimension s and the number of points N.

- Fast construction using only $\mathcal{O}(s\, m\, 2^m)$ operations available

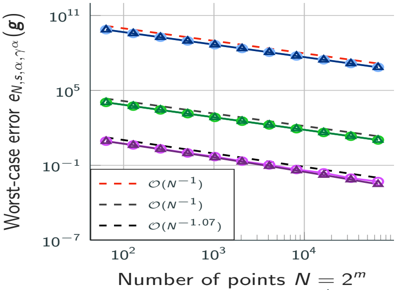
Error convergence in the space $W^{\alpha}_{s,\boldsymbol{\gamma}}$ with $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$, $s = 100$, $\alpha = 1.5, 2, 3$.



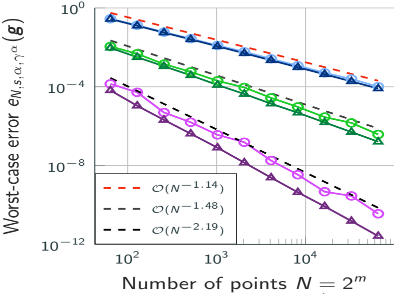
(a) $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^{s}$ with $\gamma_j = 1/j^2$



(b) $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^{s}$ with $\gamma_j = 1/j^3$



(c) $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^{s}$ with $\gamma_j = (0.95)^j$



(d) $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^{s}$ with $\gamma_j = (0.7)^j$

CBC-DBD — standard fast CBC — $\alpha = 1.5$ — $\alpha = 2$ — $\alpha = 3$

Thank you for your attention!

Lattice rules Quality measure CBC-DBD construction

Fast implementation Numerical results

Polynomial lattice rules

References

- A. Ebert, P. Kritzer, D. Nuyens, O.Osisiogu. *Digit-by-digit and component-by-component constructions of lattice rules for periodic functions with unknown smoothness*. Available on arXiv

- A. Ebert, P. Kritzer, O.Osisiogu, T.Stepaniuk. *Component-by-component digit-by-digit construction of good polynomial lattice rules in weighted Walsh spaces*. Available on arXiv