Randomized lattice rules

Dirk Nuyens
NUMA, KU Leuven, Belgium

Joint work with Frances Kuo (UNSW Sydney)
and Laurence Wilkes (KU Leuven).

MCM
Sorbonne Université
Paris, France
June 2023

Lattice rules

Deterministic lattice rules

For $f \in \mathcal{H}_\alpha$ approximate the d-dimensional integral

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

by an n-point lattice rule with generating vector $\boldsymbol{z} \in \mathbb{Z}_n^d$

$$Q_{n,\boldsymbol{z}}(f) := \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f\left(\frac{\boldsymbol{z}k \bmod n}{n}\right).$$

Worst-case error for $f \in \mathcal{H}_\alpha$ for a given algorithm Q_n (e.g. $Q_{n,\boldsymbol{z}}$):

$$e^{\mathrm{det}}(Q_n, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \leq 1}} |I(f) - Q_n(f)|.$$

Deterministic lattice rules

For $f \in \mathcal{H}_\alpha$ approximate the d-dimensional integral

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

by an n-point lattice rule with generating vector $\boldsymbol{z} \in \mathbb{Z}_n^d$

$$Q_{n,\boldsymbol{z}}(f) := \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f\left(\frac{\boldsymbol{z} k \bmod n}{n}\right).$$

Worst-case error for $f \in \mathcal{H}_\alpha$ for a given algorithm Q_n (e.g. $Q_{n,\boldsymbol{z}}$):

$$e^{\mathrm{det}}(Q_n, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \leq 1}} |I(f) - Q_n(f)|.$$

$\rightsquigarrow$ For good lattice rule $Q_{n,\boldsymbol{z}}$ converges like $n^{-\alpha} \, \|f\|_\alpha$.
(Optimal. Bakhvalov.)

Randomized lattice rules

Consider a random family of deterministic rules $Q_n^* := \{Q_n^\omega\}_\omega$.

Randomized error or worst-case expected error for $f \in \mathcal{H}_\alpha$:

$$e^{\mathrm{ran}}(Q_n^*, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \leq 1}} \mathbb{E}_\omega\big[\,|I(f) - Q_n^\omega(f)|\,\big].$$

Randomized lattice rules

Consider a random family of deterministic rules $Q_n^* := \{Q_n^\omega\}_\omega$.

Randomized error or worst-case expected error for $f \in \mathcal{H}_\alpha$:

$$e^{\mathrm{ran}}(Q_n^*, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \leq 1}} \mathbb{E}_\omega[\,|I(f) - Q_n^\omega(f)|\,].$$

$\rightsquigarrow$ Possible to get $n^{-\alpha-1/2}\,\|f\|_\alpha$. (Optimal. Bakhvalov.)

Randomized lattice rules

Consider a random family of deterministic rules $Q_n^* := \{Q_n^\omega\}_\omega$.

Randomized error or worst-case expected error for $f \in \mathcal{H}_\alpha$:

$$e^{\mathrm{ran}}(Q_n^*, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \leq 1}} \mathbb{E}_\omega[\,|I(f) - Q_n^\omega(f)|\,].$$

$\rightsquigarrow$ Possible to get $n^{-\alpha-1/2}\,\|f\|_\alpha$. (Optimal. Bakhvalov.)

How?

Randomized lattice rules

Consider a random family of deterministic rules $Q_n^* := \{Q_n^\omega\}_\omega$.

Randomized error or worst-case expected error for $f \in \mathcal{H}_\alpha$:

$$e^{\mathrm{ran}}(Q_n^*, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \le 1}} \mathbb{E}_\omega[\,|I(f) - Q_n^\omega(f)|\,].$$

$\rightsquigarrow$ Possible to get $n^{-\alpha-1/2}\,\|f\|_\alpha$. (Optimal. Bakhvalov.)

How?

Random shifting? Random generating vector? Random n?

Randomized lattice rules

Consider a random family of deterministic rules $Q_n^* := \{Q_n^\omega\}_\omega$.

Randomized error or worst-case expected error for $f \in \mathcal{H}_\alpha$:

$$e^{\mathrm{ran}}(Q_n^*, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \leq 1}} \mathbb{E}_\omega[\,|I(f) - Q_n^\omega(f)|\,].$$

$\rightsquigarrow$ Possible to get $n^{-\alpha-1/2}\,\|f\|_\alpha$. (Optimal. Bakhvalov.)

How?
Random shifting? Random generating vector? Random n?
What is necessary?

Function space

Korobov space of dominating mixed smoothness $\alpha > 0$:

$$\mathcal{H}_\alpha := \left\{ f \in L_2([0,1]^d) : \|f\|_\alpha^2 := \sum_{\boldsymbol{h} \in \mathbb{Z}^d} r_\alpha^2(\boldsymbol{h}) \, |\hat{f}(\boldsymbol{h})|^2 < \infty \right\},$$

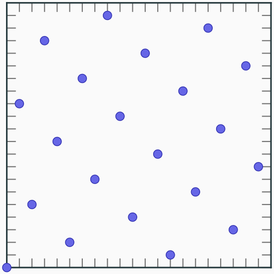
with

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

Weighted spaces... (Sloan, Woźniakowski...)

Example of a good lattice rule

Example: $n = 21$ and $\boldsymbol{z} = (1, 13)$ (Fibonacci lattice rule)



Constructive methods for deterministic error:

fast component-by-component (Nuyens & Cools 2006, ...)

$\rightarrow$ Fixed vector $\boldsymbol{z}$ for a given n.

(Or sequence of $n = p^m$, Cools, Kuo & Nuyens 2006).

Deterministic error

For $f \in \mathcal{H}_\alpha$, with $\alpha > 1/2$, and

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

Deterministic error

For $f \in \mathcal{H}_\alpha$, with $\alpha > 1/2$, and

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

we have

$$\frac{1}{n}\sum_{k\in\mathbb{Z}_n} f\left(\frac{\boldsymbol{z}k \bmod n}{n}\right) - \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \sum_{\substack{0\neq\boldsymbol{h}\in\mathbb{Z}^d \\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \pmod{n}}} \hat{f}(\boldsymbol{h}),$$

by the character sum for $\mathbb{Z}_n$, we have for $a = \boldsymbol{z}\cdot\boldsymbol{h} \in \mathbb{Z}$,

$$\frac{1}{n}\sum_{k\in\mathbb{Z}_n} \exp(2\pi\mathrm{i}\, k\, a/n) = \mathbb{1}\{a \equiv 0 \pmod{n}\}.$$

The good set

Define the "good set" of generating vectors for a prime p as

$$G^{(p)} := \left\{ \boldsymbol{z} \in \mathbb{Z}_p^d : e^{\det}(Q_{p,\boldsymbol{z}}) \leq \inf_{\lambda \in [1/2, \alpha)} \left(\frac{4}{p} \sum_{0 \neq \boldsymbol{h} \in \mathbb{Z}^d} r_\alpha^{-1/\lambda}(\boldsymbol{h}) \right)^\lambda \right\}.$$

The good set

Define the "good set" of generating vectors for a prime p as

$$G^{(p)} := \left\{ \boldsymbol{z} \in \mathbb{Z}_p^d : e^{\det}(Q_{p,\boldsymbol{z}}) \leq \inf_{\lambda \in [1/2, \alpha)} \left(\frac{4}{p} \sum_{0 \neq \boldsymbol{h} \in \mathbb{Z}^d} r_\alpha^{-1/\lambda}(\boldsymbol{h}) \right)^\lambda \right\}.$$

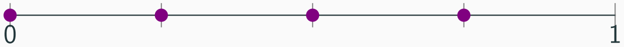
This set has more than $\left\lceil \frac{1}{2} p^d \right\rceil$ elements due to

$$\frac{1}{p^d} \sum_{\boldsymbol{z} \in \mathbb{Z}_p^d} \left[e^{\det}(Q_{p,\boldsymbol{z}}) \right]^{1/\lambda} \leq \frac{2}{p} \sum_{0 \neq \boldsymbol{h} \in \mathbb{Z}^d} r_\alpha^{-1/\lambda}(\boldsymbol{h}), \quad \forall \lambda \in [1/2, \alpha),$$

and Markov's inequality.

One-dimensional intuition

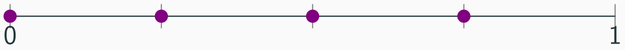
- "Lattice points": $x_k = k/n$, $k \in \mathbb{Z}_n$, for $n = 4, 5, 6$:



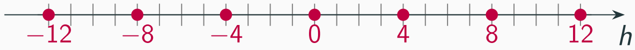
- "Dual lattice": $h \equiv 0 \pmod{n}$:

One-dimensional intuition

- “Lattice points”: $x_k = k/n$, $k \in \mathbb{Z}_n$, for $n = 4, 5, 6$:

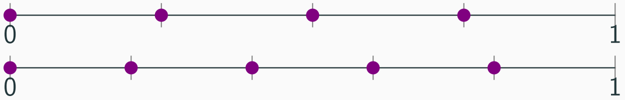


- “Dual lattice”: $h \equiv 0 \pmod{n}$:

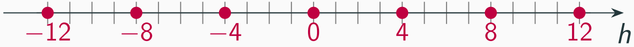


One-dimensional intuition

- “Lattice points”: $x_k = k/n$, $k \in \mathbb{Z}_n$, for $n = 4, 5, 6$:

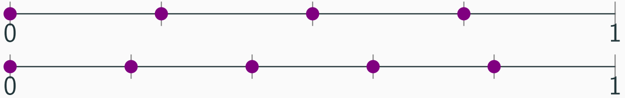


- “Dual lattice”: $h \equiv 0 \pmod{n}$:

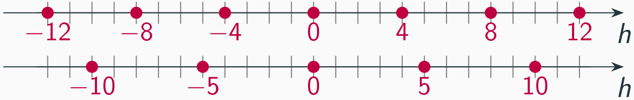


One-dimensional intuition

- "Lattice points": $x_k = k/n$, $k \in \mathbb{Z}_n$, for $n = 4, 5, 6$:

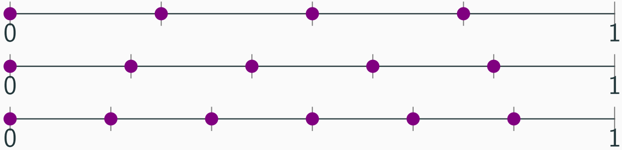


- "Dual lattice": $h \equiv 0 \pmod{n}$:

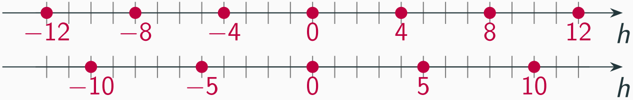


One-dimensional intuition

- "Lattice points": $x_k = k/n$, $k \in \mathbb{Z}_n$, for $n = 4, 5, 6$:

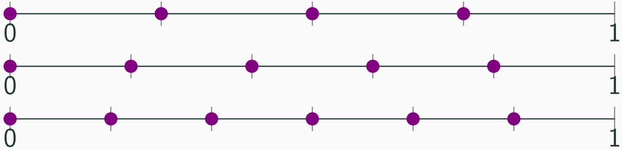


- "Dual lattice": $h \equiv 0 \pmod{n}$:

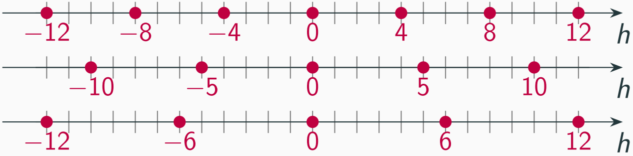


One-dimensional intuition

- "Lattice points": $x_k = k/n$, $k \in \mathbb{Z}_n$, for $n = 4, 5, 6$:

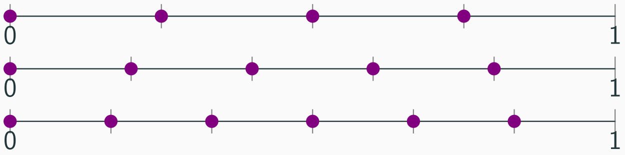


- "Dual lattice": $h \equiv 0 \pmod{n}$:

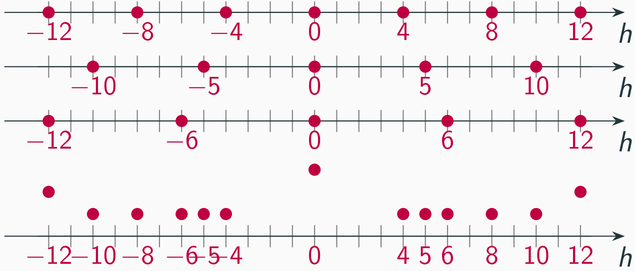


One-dimensional intuition

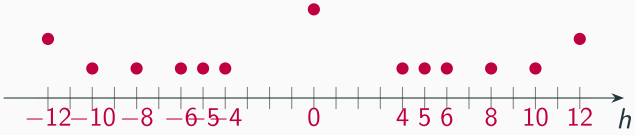
- "Lattice points": $x_k = k/n$, $k \in \mathbb{Z}_n$, for $n = 4, 5, 6$:



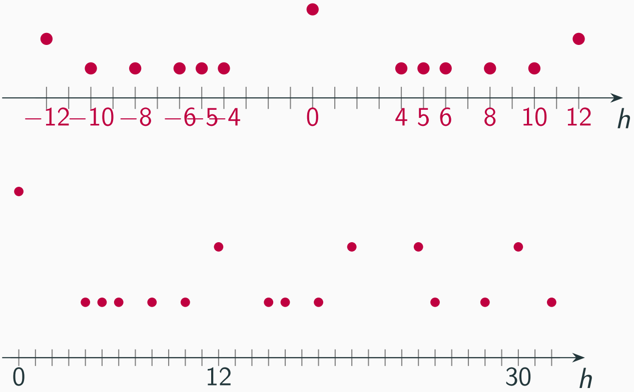
- "Dual lattice": $h \equiv 0 \pmod{n}$:



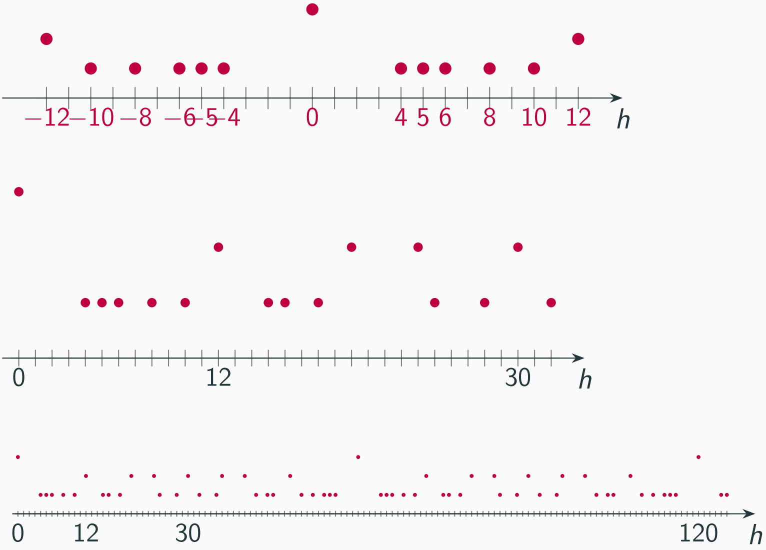
Zoom out a bit



Zoom out a bit



Zoom out a bit



Prior art

Prior art

- Bakhvalov (1961): lower and upper bounds using lattice rules for randomized error.
- Kritzer, Kuo, Nuyens, M. Ullrich (2019): randomised algorithm using lattice rules to achieve the near optimal rate.

Algorithm 1 [KKNU19]

Uniformly sample a prime $p \in P_n$.
Uniformly sample a generating vector $\boldsymbol{z} \in G^{(p)}$.
Use the lattice rule with generating vector $\boldsymbol{z}$ and p sample points.

Modifying the good set to allow for CBC construction

Define

$$\widetilde{G}_{d,\boldsymbol{z}'}^{(p)} := \left\{ z_d \in \mathbb{Z}_p : \theta_{d,\boldsymbol{z}'}^{(p)}(z_d) \leq \inf_{\lambda \in [1/2, \alpha)} \left(\frac{4}{p} \sum_{\substack{0 \neq \boldsymbol{h} \in \mathbb{Z}^d \\ h_d \neq 0}} r_\alpha^{-1/\lambda}(\boldsymbol{h}) \right)^{2\lambda} \right\}.$$

This has more than $\left\lceil \frac{1}{2} p \right\rceil$ elements by a similar method to before.

Depends on the previously fixed values of $\boldsymbol{z}' = (z_1, \ldots, z_{d-1})$.

CBC randomised algorithm

Dick, Goda and Suzuki (2022): component-by-component method.

Algorithm 2 [DGS22 / ...]

Uniformly sample a prime $p \in P_n$.

for $j = 1$ **to** d **do**

  Uniformly sample $z_j \in \widetilde{G}^{(p)}_{d,\boldsymbol{z}'}$.

end for

Use the lattice rule with generating vector $\boldsymbol{z}$ and p sample points.

Existence of a fixed vector method

Existence of a fixed vector method

Is it possible to fix the vector prior to the algorithm? Yes!

Existence of a fixed vector method

Is it possible to fix the vector prior to the algorithm? Yes!

And have the error converge like $n^{-\alpha-1/2}$? Yes!

Existence of a fixed vector method

Is it possible to fix the vector prior to the algorithm? Yes!

And have the error converge like $n^{-\alpha-1/2}$? Yes!

Define the algorithm $K^*_{n,\boldsymbol{z}}$:

Algorithm 5 Fixed vector random algorithm (Kuo, Nuyens, Wilkes)

Uniformly sample $p \in P_n$.

Apply the lattice rule with the predefined $\boldsymbol{z}$ and p sample points.

Existence result

Theorem (Kuo, Nuyens, Wilkes)
There exists a vector $\boldsymbol{z} \in \mathbb{Z}^d$ which achieves the bound

$$e^{\mathrm{ran}}(K^*_{n,\boldsymbol{z}}) \leq \frac{C_\lambda \sqrt{\ln(n)}}{n^{\lambda+1/2}} \left(\sum_{0 \neq \boldsymbol{h} \in \mathbb{Z}^d} r_\alpha^{-1/\lambda}(\boldsymbol{h}) \right)^\lambda$$

for all $\frac{1}{2} \leq \lambda < \alpha$.

For the proof:
We take $\boldsymbol{z} \in \mathbb{Z}_N^d$ with $N := \prod_{p \in P_n} p$.
We average over all vectors which are good in the deterministic sense for all of the primes.

Caveats

1. The vectors involved have incredibly large components.

Caveats

1. The vectors involved have incredibly large components.
    - Use Chinese remainder theorem:

$$\mathbb{Z}_N \cong \bigotimes_{p \in P_n} \mathbb{Z}_p.$$

Caveats

1. The vectors involved have incredibly large components.
   - Use Chinese remainder theorem:

$$\mathbb{Z}_N \cong \bigotimes_{p \in P_n} \mathbb{Z}_p.$$

   - The generating vector is only ever considered modulo one of the primes in P_n. We break down the vector $\boldsymbol{z}$ into multiple vectors,

$$\boldsymbol{z} \cong (\boldsymbol{z}^{(p_1)}, \ldots, \boldsymbol{z}^{(p_L)}).$$

Caveats

1. The vectors involved have incredibly large components.
   - Use Chinese remainder theorem:

$$\mathbb{Z}_N \cong \bigotimes_{p \in P_n} \mathbb{Z}_p.$$

   - The generating vector is only ever considered modulo one of the primes in P_n. We break down the vector $\boldsymbol{z}$ into multiple vectors,

$$\boldsymbol{z} \cong (\boldsymbol{z}^{(p_1)}, \ldots, \boldsymbol{z}^{(p_L)}).$$

2. Existence, but what about construction?

CBC construction of the vector

What about the usual method?

We follow the standard CBC approach. If z_d is a component yet to be fixed, we can write

$$\left[e_d^{\text{ran}}(K_{n,\boldsymbol{z}}^*)\right]^2 = \left[e_{d-1}^{\text{ran}}(K_{n,\boldsymbol{z}'}^*)\right]^2 + \Theta(z_d).$$

What about the usual method?

We follow the standard CBC approach. If z_d is a component yet to be fixed, we can write

$$\left[e_d^{\text{ran}}(K_{n,\boldsymbol{z}}^*)\right]^2 = \left[e_{d-1}^{\text{ran}}(K_{n,\boldsymbol{z}'}^*)\right]^2 + \Theta(z_d).$$

If we were to try to minimise $\Theta(z_d)$ at each dimension, we would have to search all possibilities that z_d could take.

This would be an $O(dn^{n+3})$ algorithm!

A detour...

Instead, we define a quantity $T^{(p)}(z_d^{(p)})$ which satisfies

$$\Theta(z_d) = \frac{1}{|P_n|^2} \sum_{p \in P_n} T^{(p)}(z_d^{(p)}).$$

A detour...

Instead, we define a quantity $T^{(p)}(z_d^{(p)})$ which satisfies

$$\Theta(z_d) = \frac{1}{|P_n|^2} \sum_{p \in P_n} T^{(p)}(z_d^{(p)}).$$

The quantity $T^p(z_d^{(p)}) = T^{(p)}(\boldsymbol{z}', \{z_d^{(r)}\}_{r<p}; z_d^{(p)})$ is cleverly rewritten so as to not depend on the value of $z_d^{(q)}$ for any $q > p$.

A detour...

Instead, we define a quantity $T^{(p)}(z_d^{(p)})$ which satisfies

$$\Theta(z_d) = \frac{1}{|P_n|^2} \sum_{p \in P_n} T^{(p)}(z_d^{(p)}).$$

The quantity $T^p(z_d^{(p)}) = T^{(p)}(\boldsymbol{z}', \{z_d^{(r)}\}_{r<p}; z_d^{(p)})$ is cleverly rewritten so as to not depend on the value of $z_d^{(q)}$ for any $q > p$.

This allows us to fix the residues of the component z_d modulo each of the primes in P_n in increasing order. This uniquely sets the value of $z_d \in \mathbb{Z}_N$.

Constructing the vector

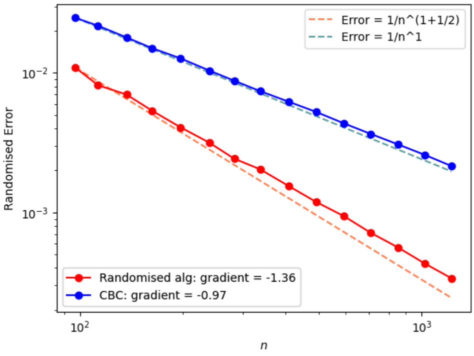
Algorithm 6 Optimal vector construction at n (Kuo, Nuyens, Wilkes)

for $j = 1$ to d **do**
  for $p \in P_n$ in increasing order **do**
    Compute $\theta_j^{(p)}(z_j^{(p)})$ for all $z_j^{(p)} \in \mathbb{Z}_p$.
    Compute $T_j^{(p)}(z_j^{(p)})$ for all $z_j^{(p)} \in \mathbb{Z}_p$.
    Choose from the $\lceil \tau p \rceil$ best choices for $\theta_j^{(p)}$ to minimize $T_j^{(p)}$.
  end for
end for

- Calculating the randomised error of an arbitrary vector takes $O(dn^4 \ln(n)^{-2})$.
- The complexity of this construction algorithm is only $O(dn^4)$ for product weights.

Randomised error vs deterministic error for $\alpha = 1$, $d = 30$

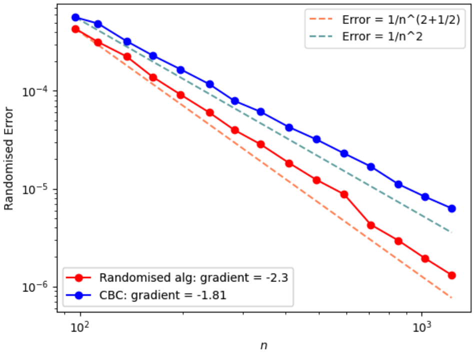
We use product weights $\gamma = \{j^{-3}\}_{j=1}^{d}$.



- The deterministic algorithm is $Q_{n,\boldsymbol{z}}$ for $\boldsymbol{z}$ chosen by CBC.
- The randomised algorithm is $K^*_{n,\boldsymbol{z}}$ with $\boldsymbol{z}$ chosen via the described method.

Randomised error vs deterministic error for $\alpha = 2$, $d = 30$

We use product weights $\gamma = \{j^{-3}\}_{j=1}^{d}$.



- The deterministic algorithm is $Q_{n,\boldsymbol{z}}$ for $\boldsymbol{z}$ chosen by CBC.
- The randomised algorithm is $K^{*}_{n,\boldsymbol{z}}$ with $\boldsymbol{z}$ chosen via the described method.

Conclusions

- Fixed vector algorithm:

Theorem
For $\alpha > 1/2$ *and all* $\lambda \in [1/2, \alpha)$*:*

$$e^{\mathrm{ran}}(K^*_{n,\boldsymbol{z}}) \leq \frac{(C_{\tau,\lambda} \ln n)^{1/2}}{n^{\lambda+1/2}} (\mu_{d,\alpha,\boldsymbol{\gamma}}(\lambda))^{\lambda}.$$

- For $\alpha \in (0, 1/2]$: the usual trick does not work since we want a fixed vector $\boldsymbol{z}$.
- Solved by relaxing the sup in the error bound:

Theorem
For $\alpha > 0$ *and all* $\lambda \in (0, \alpha)$*,* $r \in \mathbb{N}$ *and* $r \geq 1/(2\lambda)$*:*

$$e^{\mathrm{rms}}(K^{**}_{n,\boldsymbol{z}}) \leq \frac{1}{n^{\lambda+(r-1)/(2r)}} \left(\frac{C_3 \, r \ln(n)}{C_1 \, \ln(r+1)} \right)^{1/2} (\mu_{d,\alpha,\boldsymbol{\gamma}}(\lambda))^{\lambda}.$$

Thanks for listening!