Periodicity oils the wheels – periodicity, Quasi-Monte Carlo, and uncertainty quantification

Ian H. Sloan

UNSW Sydney, Australia

MCQMC, Linz, July 2022

July 22, 2022

This lecture takes a journey:

From **integration** of periodic functions on the unit cube $[0,1]^s$.

to high-dimensional **approximation** in realistic applications

Quasi-Monte Carlo integration

Quasi Monte Carlo integration (or QMC):

$$Q_{n,s}(f) = \frac{1}{n}\sum_{k=1}^{n} f(\mathbf{t}_k) \approx \int_{[0,1]^s} f(\mathbf{y})\mathrm{d}\mathbf{y} =: I_s(f),$$

with $\mathbf{t}_1, \ldots, \mathbf{t}_n \in [0,1]^s$, and cleverly chosen.

Quasi-Monte Carlo integration

Quasi Monte Carlo integration (or QMC):

$$Q_{n,s}(f) = \frac{1}{n}\sum_{k=1}^{n} f(\mathbf{t}_k) \approx \int_{[0,1]^s} f(\mathbf{y})\mathrm{d}\mathbf{y} =: I_s(f),$$

with $\mathbf{t}_1, \ldots, \mathbf{t}_n \in [0,1]^s$, and cleverly chosen.

How to choose $\mathbf{t_1}, \ldots, \mathbf{t_n}$ in a QMC rule?

- Non-periodic case: *Low discrepancy points* Koksma, Hlawka, Sobol, Faure, Niederreiter, Dick,...). But not for this lecture.
- Periodic case:

The number theorists and the periodic case

In the 1950s and 1960s number theorists (Korobov, Zaremba, Hua, ...) considered multiple integration for continuous periodic functions on $[0,1]^s$ for s arbitrary (but not necessarily large!).

Periodic means

$$f(y_1, y_2, \ldots, y_s) = f(y_1+1, y_2, \ldots, y_s) = f(y_1, y_2+1, \ldots, y_s) = \ldots.$$

The periodic setting

Let f be a continous periodic function on the unit s-cube with Fourier coefficients $\widehat{f}(\mathbf{h}) = \widehat{f}(h_1, \cdots, h_s)$ and absoutely convergent Fourier series

$$f(\mathbf{y}) = \sum_{\mathbf{h} \in \mathbb{Z}^s} \mathrm{e}^{\mathrm{i}\mathbf{h}\cdot\mathbf{y}} \widehat{f}(\mathbf{h}),$$

where

$$\widehat{f}(\mathbf{h}) = \int_{[0,1]^s} f(\mathbf{y}) \mathrm{e}^{-\mathrm{i}\mathbf{h}\cdot\mathbf{y}} \mathrm{d}\mathbf{y},$$

A typical periodic theorem of the 1950s and 60s

Let E_s be the set of s-variate continuous and *periodic functions* on $[0,1]^s$ such that

$$|\widehat{f}(h_1,\cdots,h_s)| \le \frac{1}{(\overline{h}_1\cdots\overline{h}_s)^2}, \quad h_1,\ldots,h_s \in \mathbb{Z},$$

where

$$\overline{h} = \begin{cases} 1 \text{ if } h = 0, \\ |h| \text{ if } h \neq 0. \end{cases}$$

Theorem. Assume n is prime. Then there exists $\mathbf{z} \in \mathbb{Z}^s$ such that

$$\left| I_s(f) - \frac{1}{n}\sum_{k=1}^{n} f\left(\left\{\frac{k\mathbf{z}}{n}\right\}\right)\right| \le \frac{C_s}{n^2} \qquad \forall\, f \in E_s,$$

with C_s independent of n.

(The braces tell us to take the *fractional part* of each component of the s-vector.)

That sum is a "Lattice Rule"

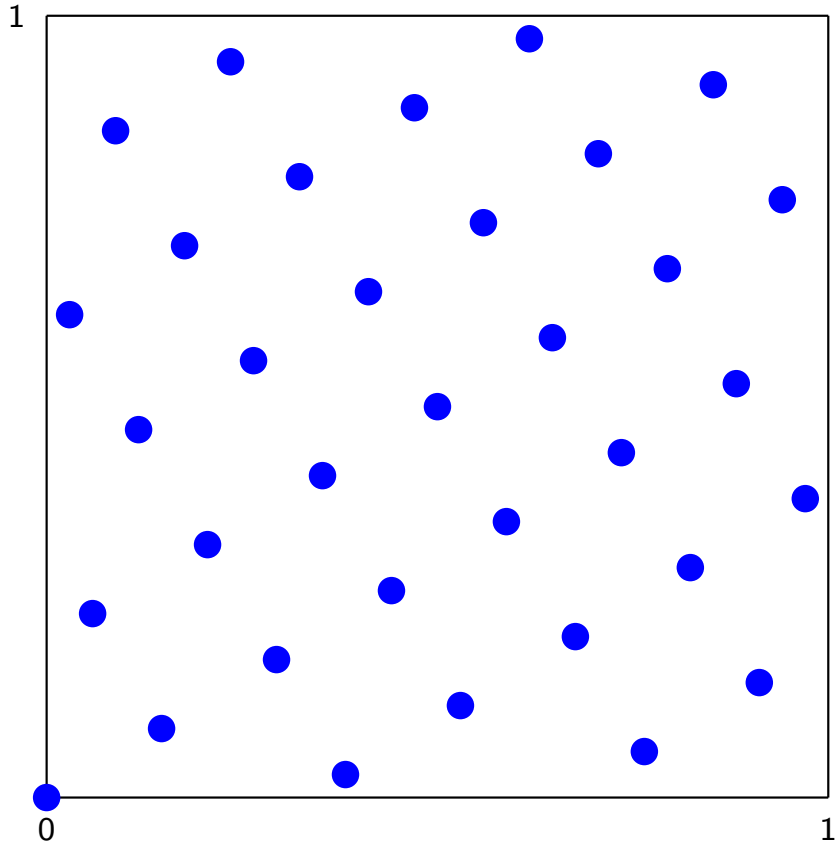
Lattice Rule (of rank 1)

$$Q_{n,s}(f) = \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{k\frac{\mathbf{z}}{n}\right\}\right),$$

$$\mathbf{z} \in \{1, \ldots, n-1\}^s$$

Example of a (good) lattice rule

$$s = 2,\ n = 34,\ \mathbf{z} = (1, 21)$$



$$\boldsymbol{t}_k := \left\{ \left(\frac{k}{34}, \frac{21k}{34} \right) \right\}, \qquad k = 1, \ldots, 34$$

Comments

- n^{-2} is an excellent result for large s: Product Simpson's rule: $n^{-2/s}$: Monte Carlo: $n^{-1/2}$; Koksma-Hlawka $n^{-1}(\log n)^{(s-1)/2}$

Comments

- n^{-2} is an excellent result for large s: Product Simpson's rule: $n^{-2/s}$: Monte Carlo: $n^{-1/2}$; Koksma-Hlawka $n^{-1}(\log n)^{(s-1)/2}$
- Proof rests on $\frac{1}{n}\sum_{k=1}^{n}\exp(2\pi \mathrm{i} k\mathbf{h}\cdot\mathbf{z}/n) = \begin{cases} 1 \text{ if } \mathbf{h}\cdot\mathbf{z} \equiv 0 \\ 0 \text{ if } \mathbf{h}\cdot\mathbf{z} \not\equiv 0 \end{cases}$

Comments

- n^{-2} is an excellent result for large s: Product Simpson's rule: $n^{-2/s}$: Monte Carlo: $n^{-1/2}$; Koksma-Hlawka $n^{-1}(\log n)^{(s-1)/2}$
- Proof rests on $\frac{1}{n}\sum_{k=1}^{n}\exp(2\pi \mathrm{i} k\mathbf{h}\cdot\mathbf{z}/n) = \begin{cases} 1 \text{ if } \mathbf{h}\cdot\mathbf{z} \equiv 0 \\ 0 \text{ if } \mathbf{h}\cdot\mathbf{z} \not\equiv 0 \end{cases}$
- The proof is not constructive! It depends on showing that the *average* over all $\mathbf{z} \in \{1, 2, \cdots, n-1\}^s$ satisfies the bound.

Comments

- n^{-2} is an excellent result for large s: Product Simpson's rule: $n^{-2/s}$: Monte Carlo: $n^{-1/2}$; Koksma-Hlawka $n^{-1}(\log n)^{(s-1)/2}$
- Proof rests on $\frac{1}{n}\sum_{k=1}^{n}\exp(2\pi \mathrm{i} k\mathbf{h}\cdot\mathbf{z}/n) = \begin{cases} 1 \text{ if } \mathbf{h}\cdot\mathbf{z} \equiv 0 \\ 0 \text{ if } \mathbf{h}\cdot\mathbf{z} \not\equiv 0 \end{cases}$
- The proof is not constructive! It depends on showing that the *average* over all $\mathbf{z} \in \{1, 2, \cdots, n-1\}^s$ satisfies the bound.
- Functions in E_s are VERY smooth: their **mixed** 2nd derivatives have bounded (in $\mathbf{h}$) Fourier coefficients.

Comments

- n^{-2} is an excellent result for large s: Product Simpson's rule: $n^{-2/s}$: Monte Carlo: $n^{-1/2}$; Koksma-Hlawka $n^{-1}(\log n)^{(s-1)/2}$
- Proof rests on $\frac{1}{n}\sum_{k=1}^{n}\exp(2\pi \mathrm{i} k\mathbf{h}\cdot\mathbf{z}/n) = \begin{cases} 1 \text{ if } \mathbf{h}\cdot\mathbf{z} \equiv 0 \\ 0 \text{ if } \mathbf{h}\cdot\mathbf{z} \not\equiv 0 \end{cases}$
- The proof is not constructive! It depends on showing that the *average* over all $\mathbf{z} \in \{1, 2, \cdots, n-1\}^s$ satisfies the bound.
- Functions in E_s are VERY smooth: their **mixed** 2nd derivatives have bounded (in $\mathbf{h}$) Fourier coefficients.
- What about the "constant" C_s? The dependence on dimension was largely ignored in the early works.

How big is C_s?

Recall **Theorem**. Assume n is prime and $f \in E_s$. Then there exists $\mathbf{z} \in \mathbb{Z}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \leq \frac{C_s}{n^2} \qquad \forall f \in E_s,$$

with C_s independent of n.

Theorem (IHS and H Woźniakowski, Math Comp 1997) For all $\mathbf{z} \in \{1, \cdots, n-1\}^s$

$$C_s \geq (2^s - 1)^2.$$

How big is C_s?

Recall **Theorem**. Assume n is prime and $f \in E_s$. Then there exists $\mathbf{z} \in \mathbb{Z}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \leq \frac{C_s}{n^2} \qquad \forall f \in E_s,$$

with C_s independent of n.

Theorem (IHS and H Woźniakowski, Math Comp 1997) For all $\mathbf{z} \in \{1, \cdots, n-1\}^s$

$$C_s \geq (2^s - 1)^2.$$

We have the "CURSE OF DIMENSIONALITY".

How big is C_s?

Recall **Theorem**. Assume n is prime and $f \in E_s$. Then there exists $\mathbf{z} \in \mathbb{Z}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \leq \frac{C_s}{n^2} \qquad \forall f \in E_s,$$

with C_s independent of n.

Theorem (IHS and H Woźniakowski, Math Comp 1997) For all $\mathbf{z} \in \{1, \cdots, n-1\}^s$

$$C_s \geq (2^s - 1)^2.$$

We have the "CURSE OF DIMENSIONALITY".

Proof idea: If $n < 2^s$ then for every $\mathbf{z}$ there is an $f \in E_s$ that vanishes at all cubature points.

But the integration problem CAN be tractable!

It is convenenient now to change to a **Hilbert space** setting.

In that setting it is now known that we need to introduce "weights", to capture the fact that often some variables are more important than others.

A periodic Hilbert space setting with weights

$$\|f\|_H := \left[\sum_{\mathbf{h}\in\mathbb{Z}} \left(\prod_{j,h_j\neq 0} \frac{|h_j|^\alpha}{\gamma_j}\right) |\widehat{f}(\mathbf{h})|^2\right]^{1/2},$$

- $\alpha > 1$ is a smoothness paraemter
- $\gamma_1, \gamma_2, \ldots, \gamma_s$ are *weights*: the weight γ_j is small if the variable y_j is unimportant.

A periodic Hilbert space setting with weights

$$\|f\|_H := \left[\sum_{\mathbf{h}\in\mathbb{Z}} \left(\prod_{j,h_j\neq 0} \frac{|h_j|^\alpha}{\gamma_j}\right) |\widehat{f}(\mathbf{h})|^2\right]^{1/2},$$

- $\alpha > 1$ is a smoothness paraemter
- $\gamma_1, \gamma_2, \ldots, \gamma_s$ are *weights*: the weight γ_j is small if the variable y_j is unimportant.

WHY choose this particular function space? Because H is a **reproducing kernel Hilbert space** with a somewhat simple reproducing kernel! (coming later).

Lifting the curse of dimensionality

Theorem (IHS and Woźniakowski 2001)
Assume n is prime. Then $\exists\, \mathbf{z} \in \{1, \ldots, n-1\}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \leq \frac{C_s}{\sqrt{n}} \|f\|_H \qquad \forall\, f \in H,$$

where C_s is independent of s iff $\sum_{j=1}^{\infty} \gamma_j < \infty$.

Lifting the curse of dimensionality

Theorem (IHS and Woźniakowski 2001)
Assume n is prime. Then $\exists\, \mathbf{z} \in \{1, \ldots, n-1\}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \le \frac{C_s}{\sqrt{n}} \|f\|_H \qquad \forall\, f \in H,$$

where C_s is independent of s iff $\sum_{j=1}^{\infty} \gamma_j < \infty$.

And if $\sum_{j=1}^{\infty} \gamma_j^{1/\alpha} < \infty$ then $\exists\, \mathbf{z} \in \{1, \ldots, n-1\}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \le \frac{C_{s,\alpha}}{n^{\alpha/2}} \|f\|_H \qquad \forall\, f \in H$$

Lifting the curse of dimensionality

Theorem (IHS and Woźniakowski 2001)
Assume n is prime. Then $\exists\, \mathbf{z} \in \{1, \ldots, n-1\}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \le \frac{C_s}{\sqrt{n}} \|f\|_H \qquad \forall\, f \in H,$$

where C_s is independent of s iff $\sum_{j=1}^{\infty} \gamma_j < \infty$.

And if $\sum_{j=1}^{\infty} \gamma_j^{1/\alpha} < \infty$ then $\exists\, \mathbf{z} \in \{1, \ldots, n-1\}^s$ such that

$$\left| I_s(f) - \frac{1}{n} \sum_{k=1}^{n} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \right| \le \frac{C_{s,\alpha}}{n^{\alpha/2}} \|f\|_H \qquad \forall\, f \in H$$

The proofs were again not constructive. But good (CBC) constructions of $\mathbf{z}$ now exist.

Another expression for the H norm with general weights

With α an even integer, H is an "unanchored" weighted Sobolev space of dominating mixed smoothness of order $\alpha/2$:

$$\|f\|_H^2 = \sum_{\mathbf{u}\subseteq\{1:s\}} \frac{1}{(2\pi)^{\alpha|\mathbf{u}|}\gamma_{\mathbf{u}}} \int_{[0,1]^{|\mathbf{u}|}} \times \left| \int_{[0,1]^{s-|\mathbf{u}|}} \left(\prod_{j\in\mathbf{u}} \frac{\partial^{\alpha/2}}{\partial y_j^{\alpha/2}} \right) f(\mathbf{y}_{\mathbf{u}}, \mathbf{y}_{-\mathbf{u}}) \mathrm{d}\mathbf{y}_{-\mathbf{u}} \right|^2 \mathrm{d}\mathbf{y}_{\mathbf{u}},$$

- $\gamma_{\mathbf{u}} > 0$ is a generalization of $\prod_{j=1}^{s} \gamma_j$
- $\gamma_{\mathbf{u}}$ is associated with the subspace $\mathbf{u}$ of $\{1:s\}$,
- $\mathbf{y}_{\mathbf{u}}$ denotes the components of $\mathbf{y}$ that belong to the subset $\mathbf{u}$,
- $\mathbf{y}_{-\mathbf{u}}$ denotes the components that do not belong to $\mathbf{u}$.

From integration (easier) to approximation (harder):

For $f \in H$, how to **approximate** $f(y_1, y_2, \ldots, y_s) = f(\mathbf{y})$, for s very large?

(Large s $\Longrightarrow$ 'The curse of dimensionality'.)

$$f(\mathbf{y}) \approx f_n(\mathbf{y}) = \sum_{k=1}^{n} \ldots f(\mathbf{t}_k),$$

We need these ingredients:

- A set $\mathbf{t}_1, \ldots, \mathbf{t}_n$ at which to compute the n function values.
- A finite-dimensional function space for the approximation.
- A way of constructing the approximating function $f_n(\mathbf{y})$.

Reference: Kaarnioja, Kazashi, Kuo, Nobile, Sloan, Numer. Math. 2022.

There's a **lot** of interest in high-dimensional approximation.

Some buzz words:

- Polynomial chaos (Wiener)
- Generalized polynomial chaos
- Stochastic Galerkin,
- Stochastic collocation
- Sparse grid
- Monte Carlo
- Multilevel Monte Carlo
- Best N-term approximation
- Reduced order modelling
- Sparse approximation, other nonlinear methods
- Uncertainty quantification

Our approximatiuon is linear, a **kernel** approximation

Given a function f and a set of points $\mathbf{t}_1, \mathbf{t}_2, \ldots \mathbf{t}_n$ in $[0,1)^s$, a **kernel approximation** based on the points $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$ takes the form

$$f_n(\mathbf{y}) = \sum_{k=1}^{n} a_k K(\mathbf{t}_k, \mathbf{y}), \qquad \mathbf{y} \in [0,1)^s.$$

$K(\cdot,\cdot)$ is a kernel if $K(\mathbf{y}, \mathbf{y}') = K(\mathbf{y}', \mathbf{y})$ and if K is positive definite, i.e. if

$$\sum_{k=1}^{N} \sum_{k'=1}^{N} b_k K(\mathbf{y}_k, \mathbf{y}_{k'}) b_{k'} \geq 0 \ \ \forall \ \ \mathbf{y}_k, b_k, N,$$

with equality only if $b_k = 0$ for all k.

Kernel interpolation

A kernel **interpolant** based on the points $\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_n}$ is given by

$$f_n(\mathbf{y}) = \sum_{k=1}^{n} a_k K(\mathbf{t}_k, \mathbf{y}), \qquad \mathbf{y} \in [0,1)^s,$$

together with

$$f_n\Big(\mathbf{t}_k\Big) = f(\mathbf{t}_k), \text{ for } k = 1, \cdots, n.$$

Kernel interpolation

A kernel **interpolant** based on the points $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$ is given by

$$f_n(\mathbf{y}) = \sum_{k=1}^{n} a_k K(\mathbf{t}_k, \mathbf{y}), \qquad \mathbf{y} \in [0,1)^s,$$

together with

$$f_n\left(\mathbf{t}_k\right) = f(\mathbf{t}_k), \text{ for } k = 1, \cdots, n.$$

$$\Longrightarrow \sum_{k=1}^{n} K(\mathbf{t}_k, \mathbf{t}_{k'})\, a_k = f\left(\mathbf{t}_{k'}\right), \qquad k' = 1, \ldots, n\,.$$

The matrix $K(\mathbf{t}_k, \mathbf{t}_{k'})$ is typically dense. Solving the linear system (always possible) can be a major cost when n is large. But not for us.

Good choices for K and $\{\mathbf{t}_1, \ldots, \mathbf{t}_s\}$

Advocated by Zeng, Leung, Hickernell 2004, and Zeng, Kritzer, Hickernell 2009.

We take $K(\mathbf{y}, \mathbf{y}')$ to be the "reproducing kernel" of H: i.e.

- $K(\mathbf{y}, \mathbf{y}') = K(\mathbf{y}', \mathbf{y})$,
- $K(\cdot, \mathbf{y}) \in H \quad \forall \mathbf{y} \in H$, and
- $\langle f, K(\cdot, \mathbf{y}) \rangle_H = f(\mathbf{y}) \quad \forall\, f \in H$;

and take the interpolation points to be lattice points:

$$\mathbf{t}_k = \left\{ \frac{k\mathbf{z}}{n} \right\}, \quad k = 1, \ldots, n,$$

But how to choose $\mathbf{z}$? Later!

Here's our kernel

$$K(\mathbf{y}, \mathbf{y}') = \sum_{\mathbf{u} \subseteq \{1:s\}} \gamma_{\mathbf{u}} \prod_{j \in \mathbf{u}} \eta_\alpha(y_j - y'_j),$$

where η_α is a known periodic function of a single variable, given by

$$\eta_\alpha(y) = \sum_{h \neq 0} \frac{e^{2\pi i h y}}{|h|^\alpha},$$

and the $\gamma_{\mathbf{u}}$ are our weights.

Here's our kernel

$$K(\mathbf{y}, \mathbf{y}') = \sum_{\mathbf{u} \subseteq \{1:s\}} \gamma_{\mathbf{u}} \prod_{j \in \mathbf{u}} \eta_\alpha(y_j - y'_j),$$

where η_α is a known periodic function of a single variable, given by

$$\eta_\alpha(y) = \sum_{h \neq 0} \frac{e^{2\pi i h y}}{|h|^\alpha},$$

and the $\gamma_{\mathbf{u}}$ are our weights.

For α an even integer, η_α is a simple piecewise polynomial:

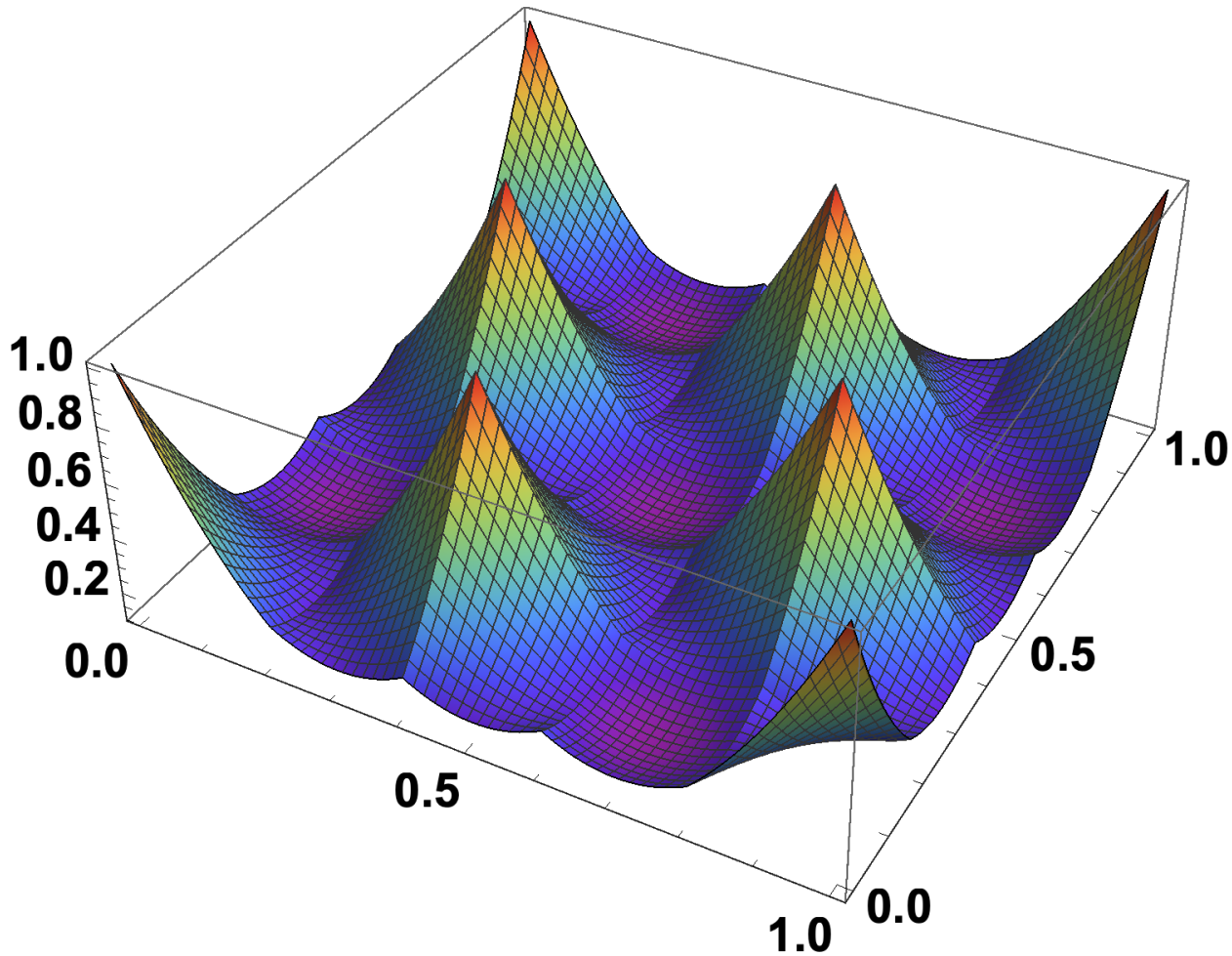
$$\eta_\alpha(y) = \frac{(2\pi)^\alpha}{(-1)^{\alpha/2-1}\alpha!} B_\alpha(\{y\})$$

with

$$B_2(y) = y^2 - y + \frac{1}{6}, \qquad B_4(y) = y^4 - 2y^3 + y^2 - \frac{1}{30}, \qquad \ldots$$

What does a kernel interpolant look like?

Here's the kernel interpolant of the function $f(y_1, y_2) = 1$.
Here $s = 2, n = 5$.



Lattice points give a major simplification:

The matrix elements become

$$K(\mathbf{y}_k, \mathbf{y}_{k'}) = \kappa(\mathbf{y}_k - \mathbf{y}_{k'}) = \kappa\left(\left\{\frac{(k-k')\mathbf{z}}{n}\right\}\right).$$

Lattice points give a major simplification:

The matrix elements become

$$K(\mathbf{y}_k, \mathbf{y}_{k'}) = \kappa(\mathbf{y}_k - \mathbf{y}_{k'}) = \kappa\left(\left\{\frac{(k-k')\mathbf{z}}{n}\right\}\right).$$

So the interpolation matrix is a **circulant** matrix, making it easy to invert by FFT – needs only $\boldsymbol{O(n \ln n)}$ flops.

But can we **compute** the kernel?

Computing the kernel is **not** feasible for general weights $\gamma_{\mathbf{u}}$: recall

$$\kappa(\mathbf{y} - \mathbf{y}') = \sum_{\mathbf{u} \subseteq \{1:s\}} \gamma_{\mathbf{u}} \prod_{j \in \mathbf{u}} \eta_\alpha(y_j - y_j').$$

There are 2^s subsets of $\{1, \ldots, s\}$!

But can we **compute** the kernel?

Computing the kernel is **not** feasible for general weights $\gamma_{\mathbf{u}}$: recall

$$\kappa(\mathbf{y}-\mathbf{y}') = \sum_{\mathbf{u}\subseteq\{1:s\}} \gamma_{\mathbf{u}} \prod_{j\in\mathbf{u}} \eta_\alpha(y_j - y_j').$$

There are 2^s subsets of $\{1,\ldots,s\}$!

But for **product weights** the computation is easy:

Recall: product weights take the form

$$\gamma_{\mathbf{u}} = \prod_{j\in\mathbf{u}} \gamma_j.$$

$$\implies \kappa(\mathbf{y}-\mathbf{y}') = \prod_{j\in\{1:s\}} \left(1+\gamma_j \eta(y_j - y_j')\right).$$

With product weights the method is fast

For product weights the method can compute $f_n(\mathbf{y})$ at m arbitrary points $\mathbf{y}$ in a time of order

$$O(\underbrace{Tn}_{f(t_k)\forall k} + \underbrace{n \ln n}_{\text{lin. sys.}} + \underbrace{snm}_{\text{kernel evals.}}),$$

Here T is the time for a single function evaluation.

There is also an off-line cost of order $O(sn \ln n)$ for computing $\mathbf{z}$.

For product weights the method is fast!

But how to choose $\gamma_{\mathbf{u}}$? How to choose $\mathbf{z}$?

It all starts with theoretical error bounds: the idea is to make choices of $\gamma_{\mathbf{u}}$ and $\mathbf{z}$ that give us good L_2 error bounds:

Defn.: The **worst-case L_2 error** for $f \in H$ and an algorithm A for approximating f is

$$e(A, L_2, H) := \sup\{||f - A(f)||_{L_2} : f \in H, \quad \|f\|_H \leq 1\} \quad \text{for } f \in H.$$

But how to choose $\gamma_{\mathbf{u}}$? How to choose $\mathbf{z}$?

It all starts with theoretical error bounds: the idea is to make choices of $\gamma_{\mathbf{u}}$ and $\mathbf{z}$ that give us good L_2 error bounds:

Defn.: The **worst-case L_2 error** for $f \in H$ and an algorithm A for approximating f is

$$e(A, L_2, H) := \sup\{\|f - A(f)\|_{L_2} : f \in H, \quad \|f\|_H \leq 1\} \quad \text{for } f \in H.$$

If A is linear this implies

$$\boxed{\|f - A(f)\|_{L_2} \leq e(A, L_2, H) \times \|f\|_H}.$$

An upper bound on the wce

Recall $\boxed{\|f - A(f)\|_{L_2} \leq e(A, L_2, H) \times \|f\|_H.}$

Let A^* denote our kernel interpolant algorithm.

Note that $e(A^*, L_2, H)$ depends on both $\gamma_{\mathbf{u}}$ and $\mathbf{z}$.

Step 1 is to find an easily computed upper bound on $e(A^*, L_2, H)$ that uses the same function values.

Now A^* is known (see eg H Wendland *Scattered Data Approximation* CUP) to be'best' among all algorithms A that use $f(\boldsymbol{t}_1), \ldots, f(\boldsymbol{t}_N)$ as the information, in the sense that

$$e(A^*, L_2, H) \leq e(A, L_2, H).$$

So for an upper bound we can use any other algorithm that uses the same information.

For bounding $e(A^*, L_2, H)$ we use:

For upper bound we use a trigonometric polynomial algorithm $A^\dagger$

$$
\begin{aligned}
f(\mathbf{y}) &= \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \left(\int_{[0,1]^s} f(\mathbf{y}')\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}'}\,\mathrm{d}\mathbf{y}' \right) \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}} \\
&\approx \sum_{\boldsymbol{h}\in\mathcal{A}_n} \left(\int_{[0,1]^s} f(\mathbf{y}')\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}'}\,\mathrm{d}\mathbf{y}' \right) \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}} \\
&\approx \sum_{\boldsymbol{h}\in\mathcal{A}_n} \left(\frac{1}{n}\sum_{k=1}^{N} f\left(\left\{ \frac{k\mathbf{z}}{n} \right\} \right) \mathrm{e}^{-2\pi\mathrm{i}k\boldsymbol{h}\cdot\mathbf{z}/n} \right) \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}} =: A^\dagger(f),
\end{aligned}
$$

where $\mathcal{A}_n := \{\boldsymbol{h} \in \mathbb{Z}^s : r(\boldsymbol{h}) \le M\}$ for some $M = M(n)$

For bounding $e(A^*, L_2, H)$ we use:

For upper bound we use a trigonometric polynomial algorithm $A^\dagger$

$$
\begin{aligned}
f(\mathbf{y}) &= \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \left(\int_{[0,1]^s} f(\mathbf{y}')\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}'}\,\mathrm{d}\mathbf{y}' \right) \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}} \\
&\approx \sum_{\boldsymbol{h}\in\mathcal{A}_n} \left(\int_{[0,1]^s} f(\mathbf{y}')\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}'}\,\mathrm{d}\mathbf{y}' \right) \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}} \\
&\approx \sum_{\boldsymbol{h}\in\mathcal{A}_n} \left(\frac{1}{n}\sum_{k=1}^{N} f\left(\left\{\frac{k\mathbf{z}}{n}\right\}\right)\mathrm{e}^{-2\pi\mathrm{i}k\boldsymbol{h}\cdot\mathbf{z}/n} \right) \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\mathbf{y}} =: A^\dagger(f),
\end{aligned}
$$

where $\mathcal{A}_n := \{\boldsymbol{h} \in \mathbb{Z}^s : r(\boldsymbol{h}) \le M\}$ for some $M = M(n)$

$$
\implies \|f - A^*(f)\|_{L_2} \le e(A^*, L_2)\|f\|_H \le e(A^\dagger, L_2)\|f\|_H,
$$

Now use

Theorem (Cools, Kuo, Nuyens, IHS, Contemporary Math 2020) Assume n is prime. Given weights $\gamma_{\mathbf{u}} > 0$, for every $\lambda \in (\frac{1}{\alpha}, 1]$, there exists a choice of $\mathbf{z} \in \{1 : s\}^s$ such that

$$e(A^\dagger, L_2, H) \le \frac{\kappa}{n^{1/4\lambda}} \left(1 + \sum_{\emptyset \ne \mathbf{u} \subseteq \{1:s\}} |\mathbf{u}| \gamma_u^\lambda [2\zeta(\alpha\lambda)]^{|\mathbf{u}|} \right)^{1/2\lambda},$$

where $\kappa = \ldots$.

Now use

Theorem (Cools, Kuo, Nuyens, IHS, Contemporary Math 2020) Assume n is prime. Given weights $\gamma_{\mathbf{u}} > 0$, for every $\lambda \in (\frac{1}{\alpha}, 1]$, there exists a choice of $\mathbf{z} \in \{1:s\}^s$ such that

$$e(A^\dagger, L_2, H) \le \frac{\kappa}{n^{1/4\lambda}} \left(1 + \sum_{\emptyset \ne \mathbf{u} \subseteq \{1:s\}} |\mathbf{u}| \gamma_u^\lambda [2\zeta(\alpha\lambda)]^{|\mathbf{u}|} \right)^{1/2\lambda},$$

where $\kappa = \ldots$.

The proof is constructive! The components $z_1, z_2, \ldots$ are constructed component by component (CBC). The proof is inductive: at each step it is shown that the best choice satisfies the bound for dimensions $(1, 2, \ldots, s)$.

For this $\mathbf{z}$ we have, for all $\lambda \in (\frac{1}{\alpha}, 1]$

$$e(A^*, L_2, H) \leq \frac{\kappa}{n^{1/4\lambda}} \left(1 + \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:s\}} |\mathbf{u}| \gamma_u^\lambda [2\zeta(\alpha\lambda)]^{|\mathbf{u}|} \right)^{1/2\lambda},$$

- To get the best convergence rate one might take $\lambda = 1/\alpha$, but then the constant becomes infinite. Take λ CLOSE to $1/\alpha$. Then rate is close to $n^{-\alpha/4}$.
- That rate is best possible, given that we use lattice points, (Byrenheid, Kämmerer, Ulrich and Volkmer, 2017).
- However, the optimal rate in the space H is $n^{-\alpha/2}$.

So now we have, with the $\mathbf{z}$ constructed by CBC,

$$\|f - A^*(f)\|_{L_2} \le \frac{\kappa}{n^{1/4\lambda}} \left(1 + \sum_{\emptyset \ne \mathbf{u} \subseteq \{1:s\}} |\mathbf{u}| \gamma_u^\lambda [2\zeta(\alpha\lambda)]^{|\mathbf{u}|}\right)^{1/2\lambda} \times \|f\|_H.$$

So now we have, with the $\mathbf{z}$ constructed by CBC,

$$\|f - A^*(f)\|_{L_2} \le \frac{\kappa}{n^{1/4\lambda}} \left(1 + \sum_{\emptyset \ne \mathbf{u} \subseteq \{1:s\}} |\mathbf{u}| \gamma_u^\lambda [2\zeta(\alpha\lambda)]^{|\mathbf{u}|} \right)^{1/2\lambda} \times \|f\|_H.$$

How now to choose the weights $\gamma_{\mathbf{u}}$'?

That depends on the application, through $\|f\|_H$:

In principle, **we seek weights that minimise the product**.

Application: An elliptic PDE with random coefficient

A much-studied model PDE is:

$$-\nabla \cdot (a(\mathbf{x},\mathbf{y})\,\nabla u(\mathbf{x},\mathbf{y})) = g(\mathbf{x}) \quad \mathbf{x} \in D\,,$$
$$u(\mathbf{x},\mathbf{y}) = 0 \quad \text{on} \quad \partial D, \quad \mathbf{y} \in U := [-\tfrac{1}{2},\tfrac{1}{2}]^{\mathbb{N}},$$

with D a bounded Lipschitz domain in $\mathbb{R}^d$, $d = 2, 3$, and

$$a(\mathbf{x},\mathbf{y}) = \overline{a} + \sum_{j=1}^{\infty} y_j\,\psi_j(\mathbf{x}), \qquad \mathbf{x} \in D, \quad \mathbf{y} = (y_1, y_2, \ldots) \in U\,,$$

where $y_1, y_2, \ldots$ are parameters representing independent random variables uniformly distributed on $[-\frac{1}{2},\frac{1}{2}]$.

And $\overline{a}$ large enough to ensure strong ellipticity for all $\mathbf{x}, \mathbf{y}$.

The same PDE with periodic random coefficient

$$-\nabla \cdot (a(\mathbf{x}, \mathbf{y}) \nabla u(\mathbf{x}, \mathbf{y})) = g(\mathbf{x}) \quad \mathbf{x} \in D\,,$$
$$u(\mathbf{x}, \mathbf{y}) = 0 \quad \text{on} \quad \partial D, \quad \mathbf{y} \in U := [-\tfrac{1}{2}, \tfrac{1}{2}]^{\mathbb{N}},$$

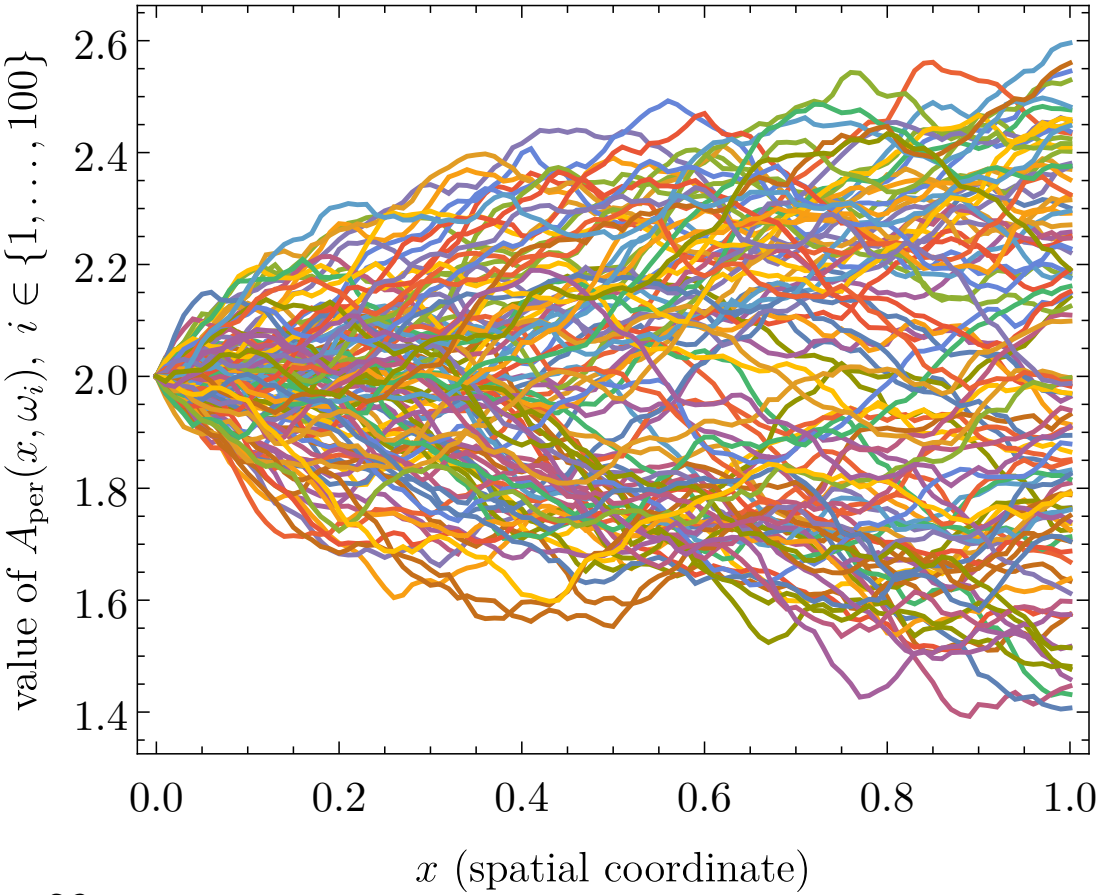
with D a bounded Lipschitz domain in $\mathbb{R}^d$, $d = 2, 3$, and

$$a(\mathbf{x}, \mathbf{y}) = \overline{a} + \sum_{j=1}^{\infty} \frac{1}{\sqrt{6}} \sin(2\pi y_j)\, \psi_j(\mathbf{x}), \quad \mathbf{x} \in D, \quad \mathbf{y} \in U\,,$$

where $y_1, y_2, \ldots$ are parameters representing independent random variables uniformly distributed on $[-\frac{1}{2}, \frac{1}{2}]$, or equivalently on $[0, 1]$.

The factor $\frac{1}{\sqrt{6}}$ ensures the (co)variance of $a(\mathbf{x}, \cdot)$ is unchanged.
See Kaarnioja, Kuo and IHS, SINUM 2020

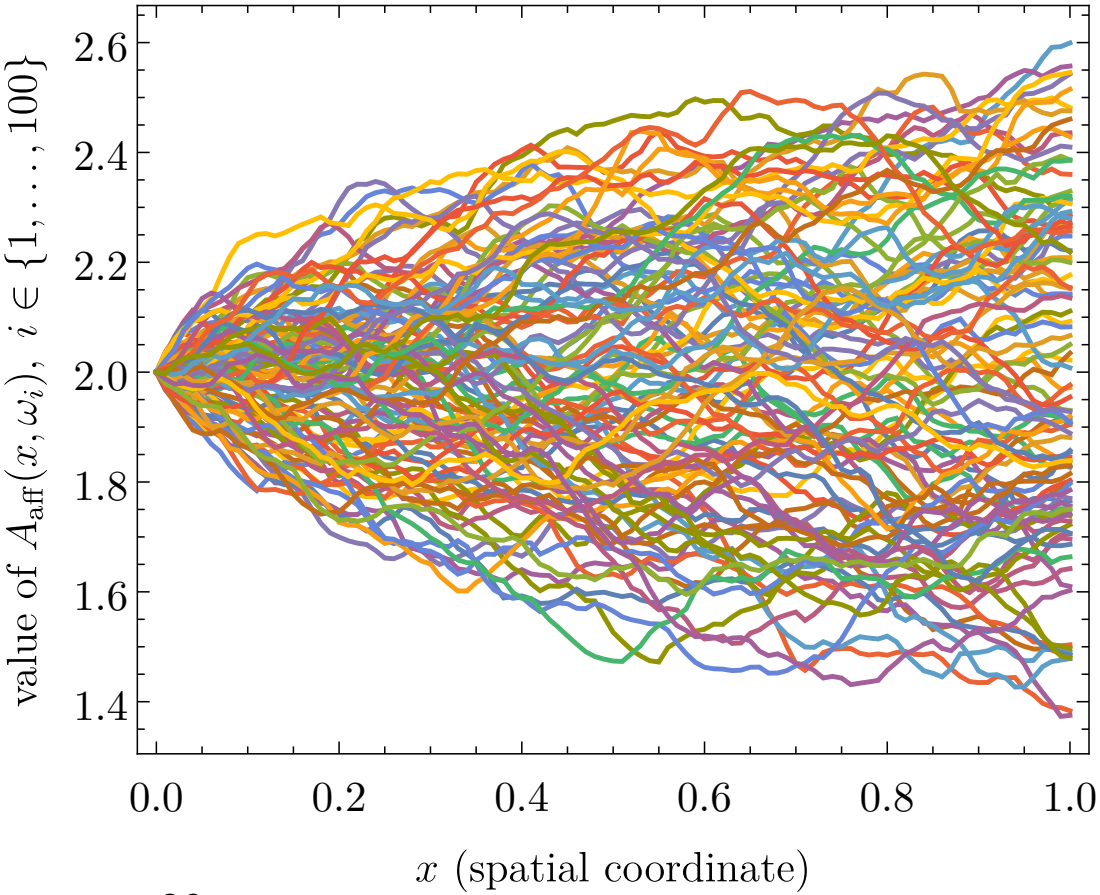
100 realisations of a univariate **periodic** random field



Here

$$a(x, \mathbf{y}) = 2 + \sum_{j=1}^{\infty} \frac{1}{\sqrt{6}} \sin(2\pi y_j)\, \psi_j(x), \qquad \psi_j(x) = j^{-\frac{3}{2}} \sin\left((j - \frac{1}{2})\pi x\right).$$

100 realisations of a univariate **affine** random field



Here

$$a(x, \mathbf{y}) = 2 + \sum_{j=1}^{\infty} y_j \, \psi_j(x), \qquad \psi_j(x) = j^{-\frac{3}{2}} \sin\left((j - \frac{1}{2}) \pi x \right).$$

Is it allowed to change the random field model?

Generalized polynomial chaos (GPC) in any of its its versions also changes the model!

(GPC takes as the approximation, products of univariate orthogonal polynomials with chosen integrable weight functions.

The model [but not the approximation] is the same as ours if the weight function is Chebyshev of the first kind.)

Truncate to a finite number of dimensions:

$$-\nabla \cdot (a(\mathbf{x},\mathbf{y})\,\nabla u(\mathbf{x},\mathbf{y})) = g(\mathbf{x}) \quad \mathbf{x} \in D\ ,$$
$$u(\mathbf{x},\mathbf{y}) = 0 \quad \text{on} \quad \partial D, \quad \mathbf{y} \in U_s = [0,1]^s,$$

with D a bounded Lipschitz domain in $\mathbb{R}^d$, $d = 2, 3$, and

$$a(\mathbf{x},\mathbf{y}) = \overline{a} + \sum_{j=1}^{s} \frac{1}{\sqrt{6}} \sin(2\pi y_j)\,\psi_j(\mathbf{x}), \qquad \mathbf{x} \in D, \quad \mathbf{y} \in U_s,$$

where $y_1, y_2, \ldots$ are parameters representing independent random variables uniformly distributed on $[0, 1]$.

Truncation introduces a manageable error, not discussed here.

How to handle $\mathbf{x}$ and $\mathbf{y}$ dependence?

For $\mathbf{x} \in D$: use Finite Element Method. (Another manageable source of error, not discussed.)

For $\mathbf{y} \in U_s = [0,1]^s$, use kernel interpolation, that is

$$f(\mathbf{y}) := f_{\mathbf{x}}(\mathbf{y}) := u(\mathbf{x}, \mathbf{y}), \quad \mathbf{x} \in D, \mathbf{y} \in U_s$$
$$f_n(\mathbf{y}) := f_{\mathbf{x},n}(\mathbf{y}) := A^*(u(\mathbf{x}, \mathbf{y})), \quad \mathbf{x} \in D, \mathbf{y} \in U_s,$$

$$\mathrm{Error}_n := \left[\int_D \int_{U_s} |f - f_n|^2 \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{x} \right]^{1/2}.$$

Weights $\gamma_{\mathbf{u}}$?

It can shown that

$$\|(\|f\|_{H(U_s)})\|_{L_2(D)} \leq \cdots,$$

leading to weights $\gamma_{\mathbf{u}} = \ldots$ that give an error bound

$$\text{Error}_n \leq \frac{C(\lambda)}{n^{1/4\lambda}}, \quad \forall\, \lambda \in (1/\alpha, 1],$$

with $C(\lambda)$ an explicit constant that is independent of s.

Numerical example

$$-\nabla\Big(a(\mathbf{x},\mathbf{y})\nabla u(\mathbf{x},\mathbf{y})\Big) = x_2, \quad \mathbf{x}\in D,\ \mathbf{y}\in U,$$
$$u(\mathbf{x},\mathbf{y}) = 0, \quad \mathbf{x}\in \partial D,\ \mathbf{y}\in U,$$

in the domain $D = (0,1)^2$, with

$$a(\mathbf{x},\mathbf{y}) := 1 + \sum_{j=1}^{s} \frac{\sin(2\pi y_j)}{\sqrt{6}}\psi_j(\mathbf{x}), \quad \mathbf{x}\in D,\ \mathbf{y}\in U,$$

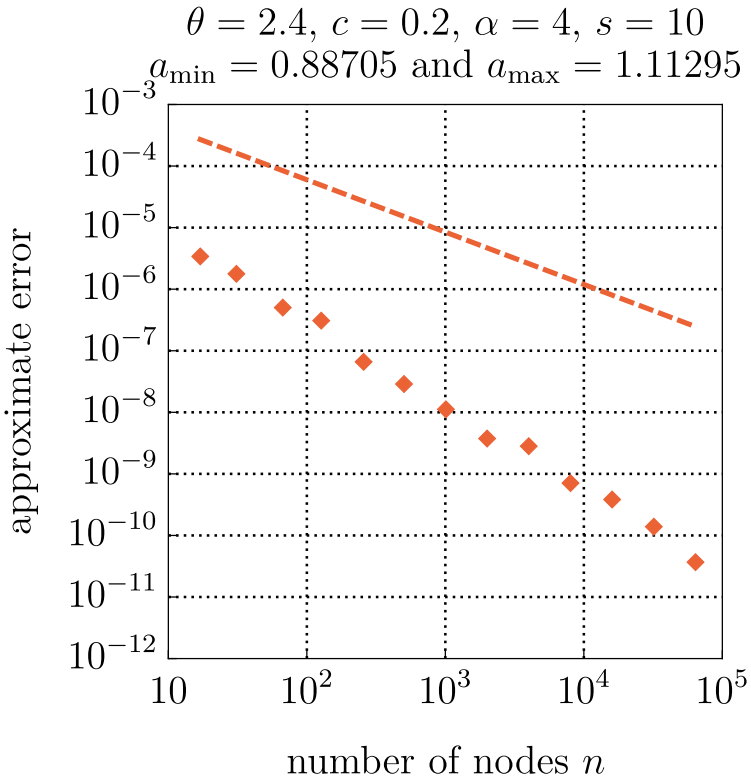
where

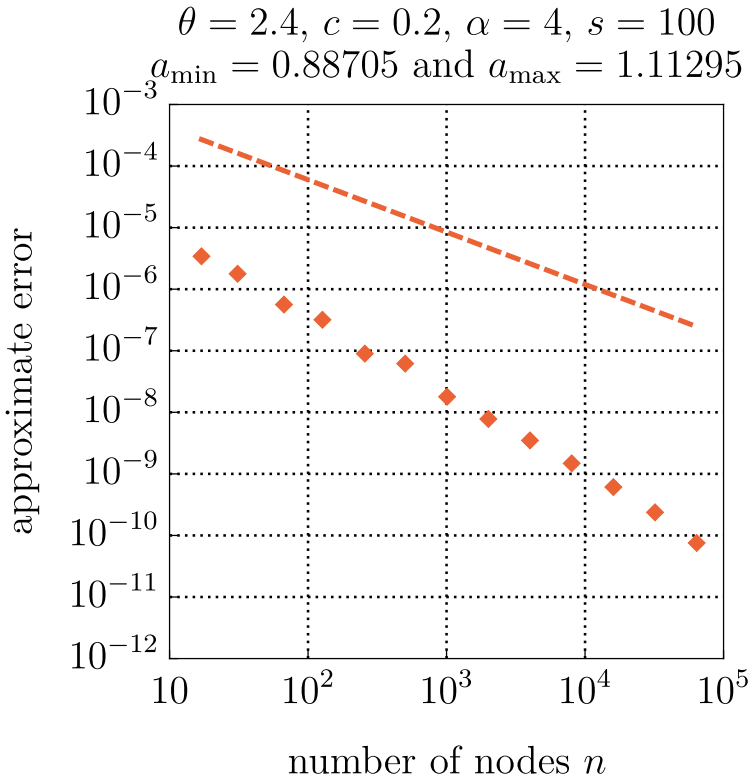
$$\psi_j(\mathbf{x}) = c j^{-\theta} \sin\left(j\pi x_1\right)\sin\left(j\pi x_2\right),$$

with $\theta > 1$ and $c > 0$ variable,
and with $s = 10$ or 100.

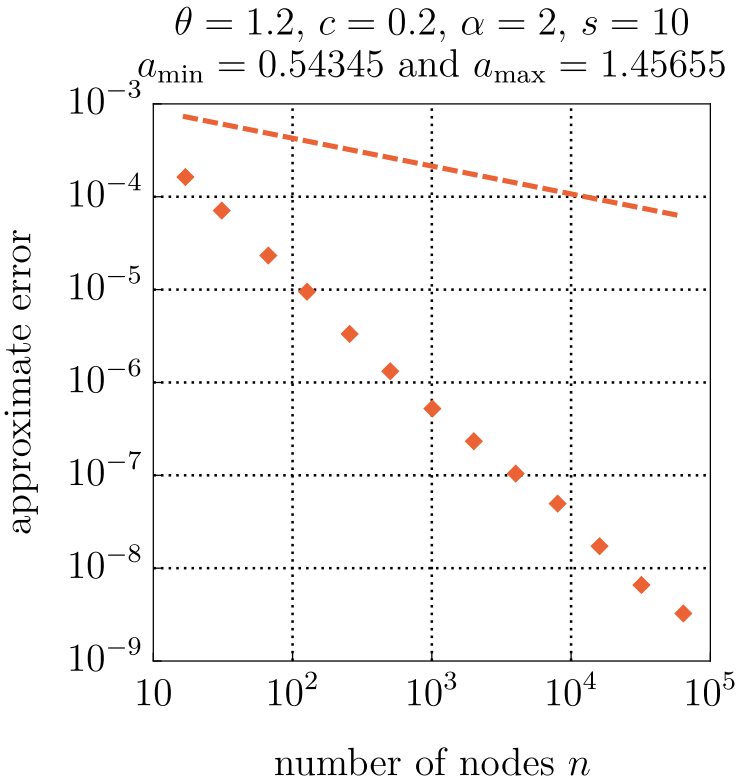
V Kaarnioja, Y Kazashi, F Kuo, F Nobile, IHS, Numer. Math., Jan, '22

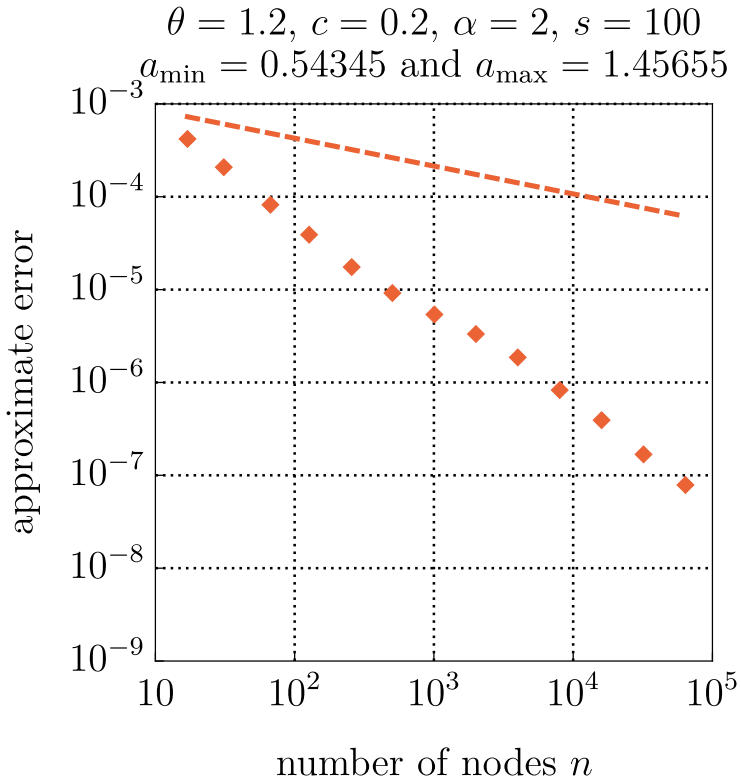
Numerical results for $\theta = 2.4, c = 0.2, \alpha = 4$



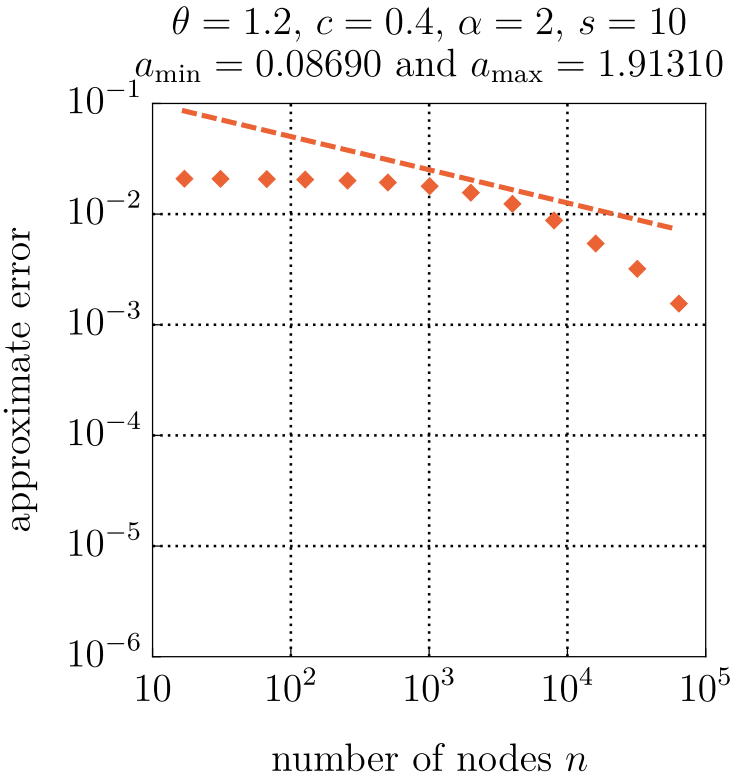


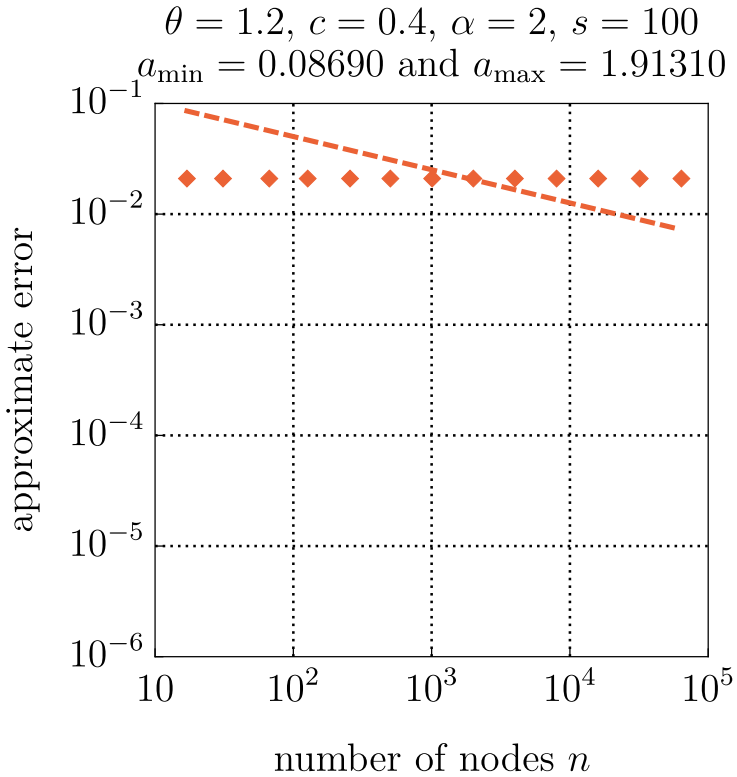
Numerical results for $\theta = 1.2, c = 0.2, \alpha = 2$





Numerical results for $\theta = 1.2, c = 0.4, \alpha = 2$





But the weights used above were not product weights!

They were more complicated ("SPOD") weights:

$$\gamma_{\mathbf{u}} := \sum_{\mathbf{m}_{\mathbf{u}} \in \{1,2\}^{|\mathbf{u}|}} (|\mathbf{m}_{\mathbf{u}}|!)^{17/11} \prod_{j \in \mathbf{u}} \left(\frac{(cj^{-2.4}/\sqrt{6})^{m_j}}{\sqrt{2e^{1/e}\zeta(20/17)}} \right)^{17/11}.$$

But the weights used above were not product weights!

They were more complicated ("SPOD") weights:

$$\gamma_{\mathbf{u}} := \sum_{\mathbf{m}_{\mathbf{u}} \in \{1,2\}^{|\mathbf{u}|}} (|\mathbf{m}_{\mathbf{u}}|!)^{17/11} \prod_{j \in \mathbf{u}} \left(\frac{(cj^{-2.4}/\sqrt{6})^{m_j}}{\sqrt{2e^{1/e}\zeta(20/17)}} \right)^{17/11}.$$

For these weights the cost (Cools, Kuo, Nuyens, IHS 2019(2)) is

$$O(Tn + n \ln n + s^2 nm);$$

and the cost for computing $\mathbf{z}$ is $O(sn \ln n + s^3 n)$.

Recall: product weights

$$O(\underbrace{Tn}_{f(t_k)\forall k} + \underbrace{n \ln n}_{\text{lin. sys.}} + \underbrace{snm}_{\text{kernel evals.}}),$$

and the cost for computing $\mathbf{z}$ is $O(sn \ln n)$.

Can we get good results with product weights?

YES WE CAN! We use **serendipitous weights**.
V Kaarnioja, F Kuo, IHS, in progress

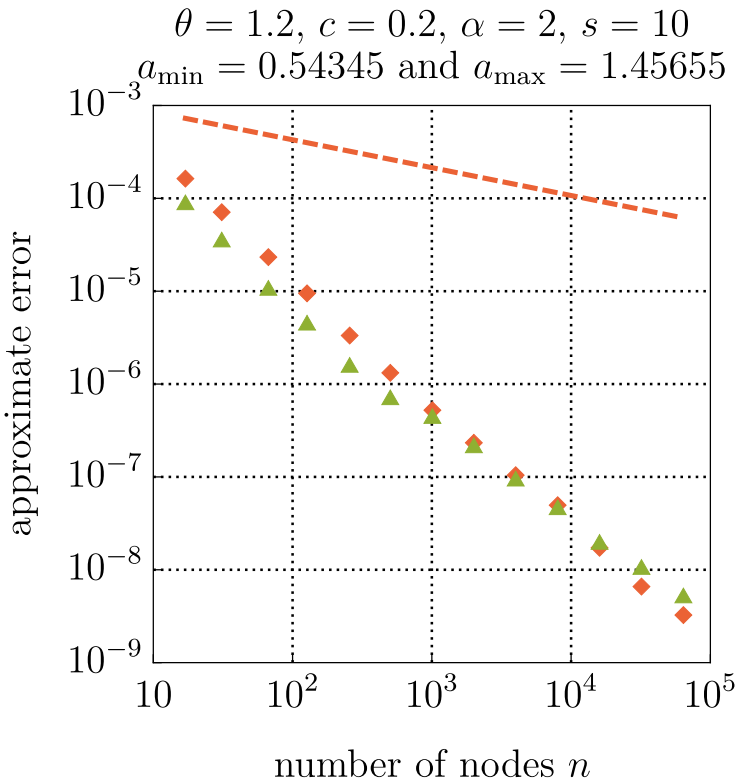
["Serendipity" – happy discovery by accident]

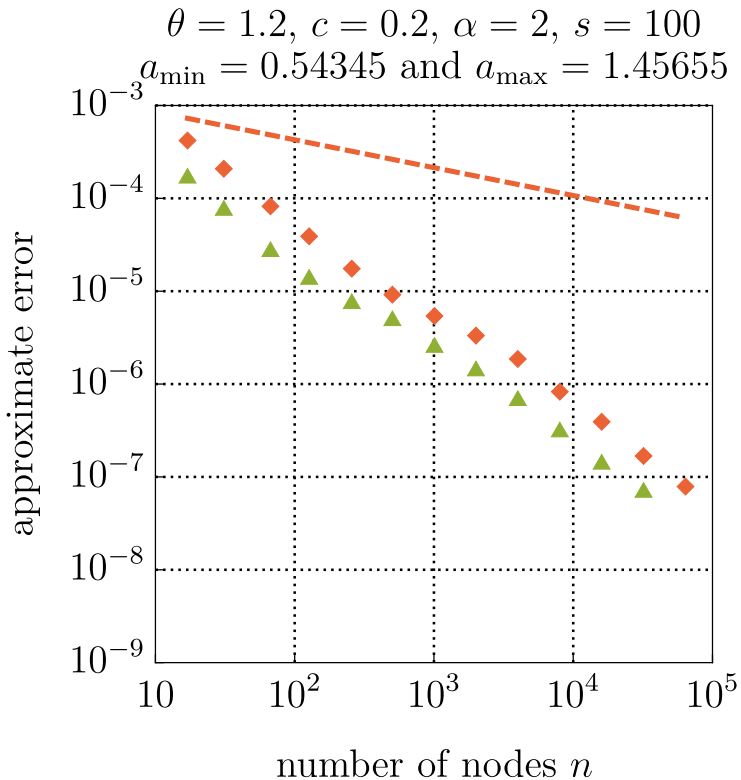
$$\gamma_{\mathbf{u}} := \sum_{\mathbf{m}_{\mathbf{u}} \in \{1,2\}^{|\mathbf{u}|}} \underbrace{(|\mathbf{m}_{\mathbf{u}}|!)^{17/11}}_{\text{OMIT!}} \prod_{j \in \mathbf{u}} \left(\frac{(cj^{-2.4}/\sqrt{6})^{m_j}}{\sqrt{2e^{1/e}\zeta(20/17)}} \right)^{17/11}.$$

$$\implies \gamma_{\mathbf{u}} = \prod_{j \in \mathbf{u}} \sum_{m_j=1}^{2} \left(\frac{(cj^{-2.4}/\sqrt{6})^{m_j}}{\sqrt{2e^{1/e}\zeta(20/17)}} \right)^{17/11},$$

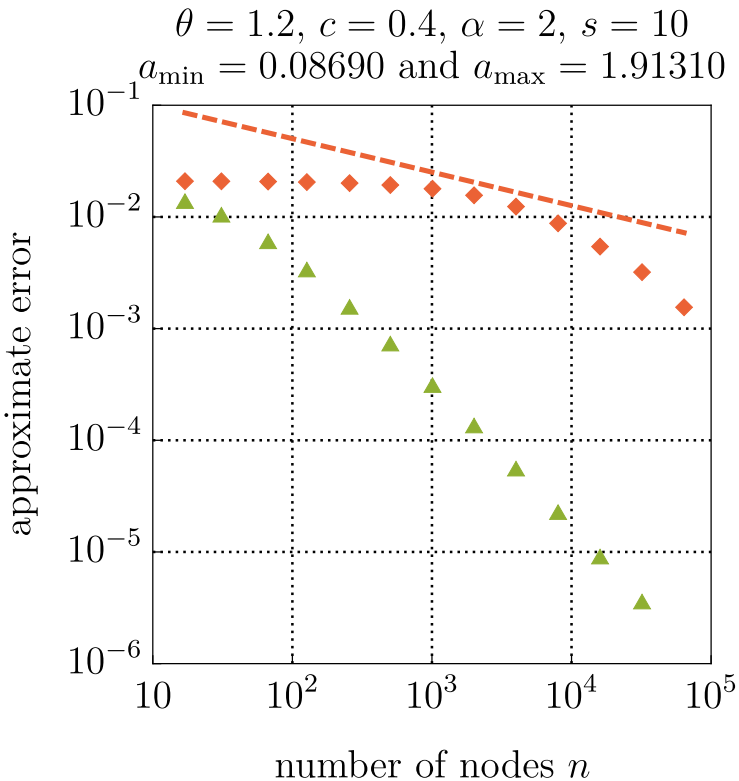
leading to the same error bound, but a different (bigger!) constant.

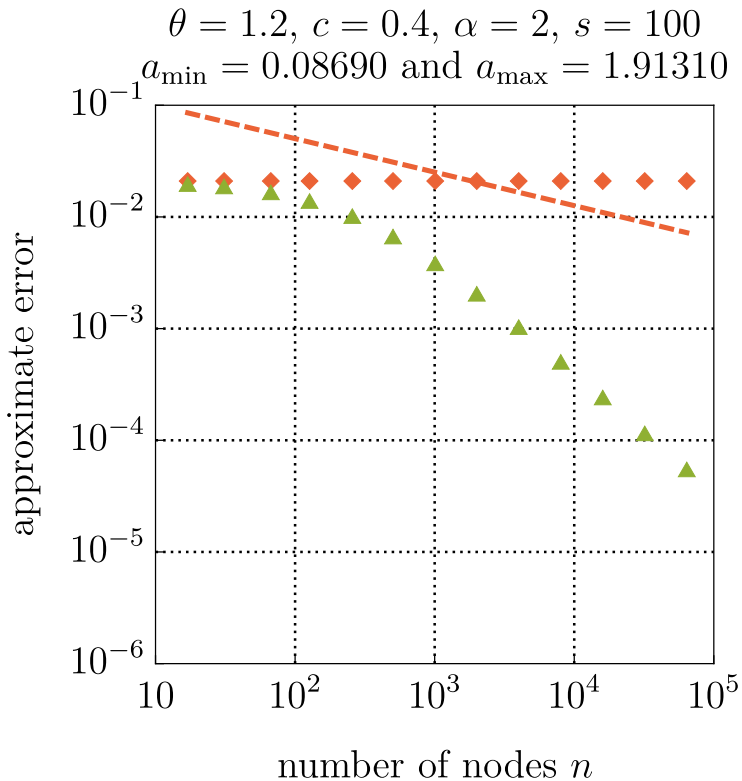
Numerical results for $\theta = 1.2, c = 0.2, \alpha = 2$





NUmerical results for $\theta = 1.2, c = 0.4, \alpha = 2$





Summary

From **integration** of periodic functions on the unit cube $[0,1]^s$.

to high-dimensional **approximation** in a realistic application

- Approximation can be both mathematically rigorous and feasible for large dimensionality and large number of points.
- The cost is merely linear in s IF we use periodic random variables and serendipitous weights.
- Serendipitpous weights are seen to give excellent results even for **hard** high-dimensional probelms – better than we presently understand.