What’s new in high-dimensional integration? – designing for applications

Ian H. Sloan

i.sloan@unsw.edu.au

The University of New South Wales – UNSW Australia

ANZIAM NSW/ACT, November, 2015

The theme

- High dimensional problems are important, but hard.
- Some interesting problems can now be tackled successfully.
- What’s new? What’s new is that we now know how to **design** high dimensional integration rules that are good for particular problems.

These are **Quasi-Monte Carlo** (or **QMC**) rules

High-dimensional integration

Consider

$$\int_0^1 \cdots \int_0^1 F(y_1, \ldots, y_{300}) \mathrm{d}y_1 \cdots \mathrm{d}y_{300}.$$

High-dimensional integration

Consider

$$\int_0^1 \cdots \int_0^1 F(y_1, \ldots, y_{300}) \mathrm{d}y_1 \cdots \mathrm{d}y_{300}.$$

Where might integrals with hundreds of dimensions occur?

High-dimensional integration

Consider

$$\int_0^1 \cdots \int_0^1 F(y_1, \ldots, y_{300}) \mathrm{d}y_1 \cdots \mathrm{d}y_{300}.$$

Where might integrals with hundreds of dimensions occur?

- finance
- statistics
- flow through a porous medium
- other stochastic pde – e.g. climate change done properly

How?

We cannot use a product rule! Even a 2-point Gauss rule in each of 100 dimensions would need 2^{100} function evaluations!

How?

We cannot use a product rule! Even a 2-point Gauss rule in each of 100 dimensions would need 2^{100} function evaluations!

In general we want to approximate (with maybe large s)

$$\begin{aligned} I_s(F) :=& \int_0^1 \cdots \int_0^1 F(y_1, \ldots, y_s) \mathrm{d}y_1 \ldots \mathrm{d}y_s \\ =& \int_{[0,1]^s} F(\mathbf{y}) \mathrm{d}\mathbf{y}. \end{aligned}$$

In practice this usually means that some transformation to the unit cube has already been carried out.

How?

We cannot use a product rule! Even a 2-point Gauss rule in each of 100 dimensions would need 2^{100} function evaluations!

In general we want to approximate (with maybe large s)

$$\begin{aligned} I_s(F) :=& \int_0^1 \cdots \int_0^1 F(y_1, \ldots, y_s) \mathrm{d}y_1 \ldots \mathrm{d}y_s \\ =& \int_{[0,1]^s} F(\mathbf{y}) \mathrm{d}\mathbf{y}. \end{aligned}$$

In practice this usually means that some transformation to the unit cube has already been carried out.

To design a good integration rule we need to be guided by **applications**.

PDE with random coefficients

PDE with random coefficients are now attracting great interest.

Example: flow through a porous medium

Darcy's law is $$\vec{q}(\mathbf{x}) \;=\; -a(\mathbf{x})\nabla p(\mathbf{x}),$$

where

$p(\mathbf{x})$ is pressure of the fluid

$\vec{q}(\mathbf{x})$ is velocity of the fluid

$a(\mathbf{x})$ is “permeability” of the medium

PDE with random coefficients

PDE with random coefficients are now attracting great interest.

Example: flow through a porous medium

Darcy’s law is $\vec{q}(\mathbf{x}) = -\boldsymbol{a}(\mathbf{x})\nabla \boldsymbol{p}(\mathbf{x}),$

where

$\boldsymbol{p}(\mathbf{x})$ is pressure of the fluid

$\vec{q}(\mathbf{x})$ is velocity of the fluid

$\boldsymbol{a}(\mathbf{x})$ is “permeability” of the medium

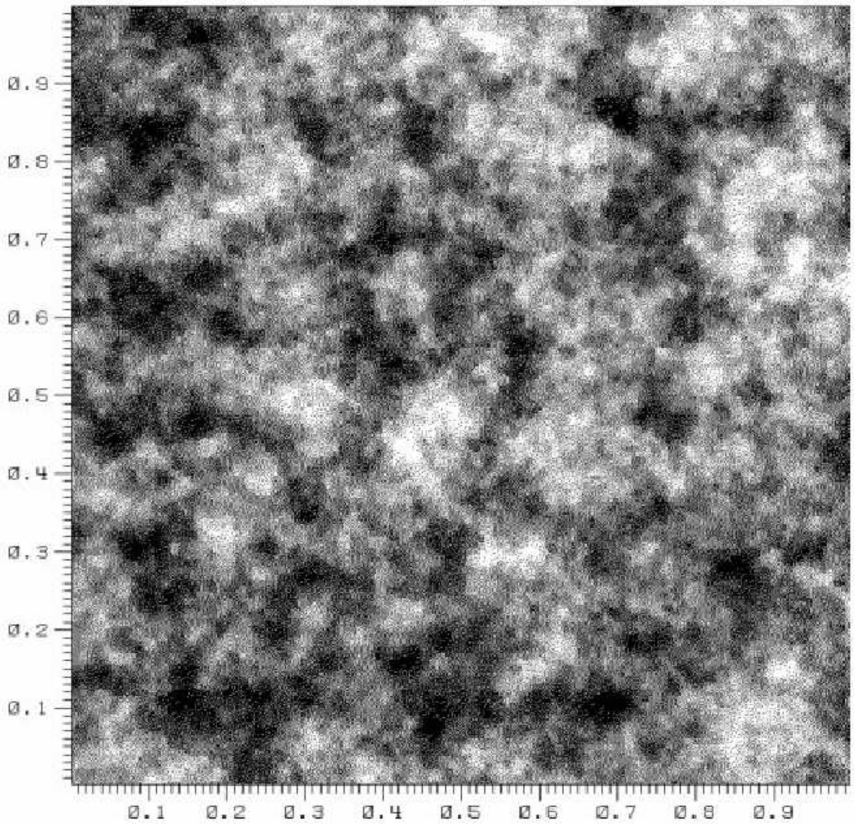
Incompressibility: $\nabla \cdot \vec{q} = 0$

Together these give a second order elliptic PDE:

$$\nabla \cdot (\boldsymbol{a}(\mathbf{x})\nabla \boldsymbol{p}(\mathbf{x})) = 0$$

Modelling the permeability

Describing in all the microscopic pores and channels in a real material is commonly considered much too hard. So it is common engineering practice to model the permeability as a random field:



A model problem – the “uniform” case

$$-\boldsymbol{\nabla}\cdot\left(a(\mathrm{x},\mathrm{y})\,\boldsymbol{\nabla} u(\mathrm{x},\mathrm{y})\right) \;=\; f(\mathrm{x}) \quad \text{in} \quad D\,,$$

$$u(\mathrm{x},\mathrm{y}) \;=\; 0 \quad \text{on} \quad \partial D, \qquad \mathrm{y}\in U := [0,1]^{\mathbb{N}},$$

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

A model problem – the “uniform” case

$$-\nabla \cdot (a(\mathrm{x},\mathrm{y})\,\nabla u(\mathrm{x},\mathrm{y})) \;=\; f(\mathrm{x}) \quad \text{in} \quad D\ ,$$

$$u(\mathrm{x},\mathrm{y}) \;=\; 0 \quad \text{on} \quad \partial D, \qquad \mathrm{y} \in U := [0,1]^{\mathbb{N}},$$

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

$$a(\mathrm{x},\mathrm{y}) \;=\; \overline{a} + \sum_{j=1}^{\infty} (y_j - \tfrac{1}{2})\,\psi_j(\mathrm{x}), \qquad \mathrm{x} \in D, \quad \mathrm{y} \in U\ ,$$

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

A model problem – the “uniform” case

$$-\nabla\cdot(a(\mathrm{x},\mathrm{y})\,\nabla u(\mathrm{x},\mathrm{y})) \;=\; f(\mathrm{x}) \quad \text{in} \quad D\;,$$

$$u(\mathrm{x},\mathrm{y}) \;=\; 0 \quad \text{on} \quad \partial D, \qquad \mathrm{y}\in U := [0,1]^{\mathbb{N}},$$

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

$$a(\mathrm{x},\mathrm{y}) \;=\; \overline{a} + \sum_{j=1}^{\infty} (y_j - \tfrac{1}{2})\,\psi_j(\mathrm{x}), \qquad \mathrm{x}\in D, \quad \mathrm{y}\in U\;,$$

where $y_1, y_2, \ldots$ are independent random variables uniformly distributed on $[0,1]$; with $\overline{a}, \psi_j$ such that $\sum_j \|\psi_j\|_\infty < \infty$, and

$$a_{\max} \geq a(\mathrm{x},\mathrm{y}) \geq a_{\min} > 0,$$

making the PDE **strongly elliptic** for every y.

A model problem – the “uniform” case

$$-\nabla\cdot(a(\mathrm{x},\mathrm{y})\,\nabla u(\mathrm{x},\mathrm{y})) \;=\; f(\mathrm{x}) \quad \text{in} \quad D\,,$$

$$u(\mathrm{x},\mathrm{y}) \;=\; 0 \quad \text{on} \quad \partial D, \quad \mathrm{y}\in U := [0,1]^{\mathbb{N}},$$

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

$$a(\mathrm{x},\mathrm{y}) \;=\; \overline{a} + \sum_{j=1}^{s} (y_j - \tfrac{1}{2})\,\psi_j(\mathrm{x}), \qquad \mathrm{x}\in D, \quad \mathrm{y}\in U\,,$$

where $y_1, y_2, \ldots$ are independent random variables uniformly distributed on $[0,1]$; with $\overline{a}, \psi_j$ such that $\sum_j \|\psi_j\|_\infty < \infty$, and

$$a_{\text{max}} \ge a(\mathrm{x},\mathrm{y}) \ge a_{\text{min}} > 0,$$

making the PDE **strongly elliptic** for every y.

In practice truncate the sum after s terms.

Example

Take $d = 1$,

$$a(x, \mathrm{y}) \;=\; \overline{a} + \sum_{j=1}^{\infty} (y_j - \tfrac{1}{2})\, \psi_j(x), \qquad x \in [0, \pi], \quad \mathrm{y} \in U\ ,$$

and

$$\psi_j(x) := \frac{\sin(jx)}{j^\alpha}, \qquad \text{for some } \alpha > 1.$$

The bigger is α, the smoother the field $a(\cdot, \mathrm{y})$.

The lognormal case

In the lognormal case

$$\boldsymbol{a}(\mathrm{x},\mathrm{y}) = \overline{a}(\mathrm{x}) + \mathbf{exp}\left(\sum_{j=1}^{\infty} \boldsymbol{y_j} \sqrt{\mu_j}\xi_j(\mathrm{x})\right), \qquad \mathrm{x} \in D,$$

- $\boldsymbol{y_j}$ are i.i.d. standard normal random numbers
- $\boldsymbol{\mu_j}, \boldsymbol{\xi_j}$ are the eigenvalues and normalized eigenfunctions of the covariance operator for the Gaussian random field in the exponent

What might we want to compute?

- The mean pressure at a particular point or over a particular small region
- The effective permeability
- The mean “breakthrough time”
- . . .

What might we want to compute?

- The mean pressure at a particular point or over a particular small region
- The effective permeability
- The mean “breakthrough time”
- . . .

All are expected values – and expected values are integrals.

If there are many random variables, then the expected values are high-dimensional integrals.

Suppose the problem is to compute the expected value of

$$F(\mathrm{y}) := G(u(\cdot, \mathrm{y}))$$

for some linear functional G of the solution u of the PDE.

Suppose the problem is to compute the expected value of

$$F(\mathbf{y}) := G(u(\cdot, \mathbf{y}))$$

for some linear functional G of the solution u of the PDE.

The expected value is in principle an infinite-dimensional **integral**, where the meaning is:

$$I[F] := \int_{[0,1]^{\mathbb{N}}} F(\mathbf{y})\,\mathrm{d}\mathbf{y}$$

$$:= \lim_{s\to\infty} \int_{[0,1]^s} F(y_1, \ldots, y_s, \tfrac{1}{2}, \tfrac{1}{2}, \ldots)\,\mathrm{d}y_1 \ldots \mathrm{d}y_s.$$

Suppose the problem is to compute the expected value of

$$F(\mathbf{y}) := G(u(\cdot,\mathbf{y}))$$

for some linear functional G of the solution u of the PDE.

The expected value is in principle an infinite-dimensional **integral**, where the meaning is:

$$\begin{aligned} I[F] &:= \int_{[0,1]^{\mathbb{N}}} F(\mathbf{y})\mathrm{d}\mathbf{y} \\ &:= \lim_{s\to\infty} \int_{[0,1]^s} F(y_1,\ldots,y_s,\tfrac{1}{2},\tfrac{1}{2},\ldots)\mathrm{d}y_1\ldots\mathrm{d}y_s. \end{aligned}$$

Note that replacing $y_{s+1}, y_{s+2}, \ldots$ by $\frac{1}{2}$ is equivalent to replacing $a(\mathbf{x},\mathbf{y})$ by

$a_s(\mathbf{x},\mathbf{y}) \;:=\; \overline{a} + \sum_{j=1}^{s}(y_j - \frac{1}{2})\,\psi_j(\mathbf{x}).$

Many approaches:

- polynomial chaos,
- generalized polynomial chaos,
- stochastic Galerkin,
- stochastic collocation
- Monte Carlo
- multilevel Monte Carlo
- Quasi-Monte Carlo (QMC)
- multilevel Quasi-Monte Carlo

Many approaches:

- polynomial chaos,
- generalized polynomial chaos,
- stochastic Galerkin,
- stochastic collocation
- Monte Carlo
- multilevel Monte Carlo
- Quasi-Monte Carlo (QMC)
- multilevel Quasi-Monte Carlo

All methods face serious challenges when the effective dimensionality is high.

Many approaches:

- polynomial chaos,
- generalized polynomial chaos,
- stochastic Galerkin,
- stochastic collocation
- Monte Carlo
- multilevel Monte Carlo
- Quasi-Monte Carlo (QMC)
- multilevel Quasi-Monte Carlo

All methods face serious challenges when the effective dimensionality is high. And when all else fails, people turn to Monte Carlo methods. QMC aims to beat Monte Carlo.

Many contributors:

Norbert Wiener, "The Homogeneous Chaos", 1938

Babuska/Nobile/Tempone, Babuska/Tempone/Zouraris, Barth/Schwab/Zollinger,

Charrier, Charrier/Scheichl/Teckentrup, Cliffe, Giles/Scheichl/Teckentrup, Cliffe,

Graham/Scheichl/Stals, Cohen/ Chkifa/Schwab, Cohen/De Vore/Schwab,

Graham/Scheichl/Ullmann, Hansen/Schwab, Harbrecht/Peters/Siebenmorgen,

Hoang/Schwab, Karniadakis/ Xiu, Kunoth/Schwab, Nobile/Tempone/Webster,

Schwab/Todor, Schillings/Schwab, Teckentrup/Scheichl/Giles/Ullmann, Webster

MATHEMATICS & STATISTICS
THE UNIVERSITY OF NEW SOUTH WALES

And for QMC applied to PDE with random coefficients:

Graham/Kuo/Nuyens/Scheichl/Sloan 2011 (lognormal case, no error analysis, circulant embedding), Kuo/Schwab/Sloan (uniform case with error analysis), Kuo/Schwab/Sloan (multi-level for the uniform case), Schwab (uniform case, general operator equations), Le Gia (uniform case for sphere), Graham/Kuo/Nichols/Scheichl/Schwab/Sloan (lognormal case, analysis and numerics), Graham/Kuo/Scheichl/Schwab/Sloan/Ullmann), (multi-level lognormal case), Harbrecht/Peters/Siebenmorgen

Monte Carlo (MC)

$$Q^{\mathrm{MC}}_{N,s}(F) := \frac{1}{N}\sum_{k=1}^{N} F(\mathbf{t}_k),$$

with $\mathbf{t}_1,\ldots,\mathbf{t}_N$ chosen **randomly** and independently from a uniform distribution on $[0,1]^s$.

Quasi-Monte Carlo

For QMC we take

$$I_s(F) \approx Q_{N,s}(F) := \frac{1}{N} \sum_{k=1}^{N} F(\mathrm{t}_k) \ ,$$

with $\mathrm{t}_1, \ldots, \mathrm{t}_N$ **deterministic** (and cleverly chosen).

Quasi-Monte Carlo

For QMC we take

$$I_s(F) \approx Q_{N,s}(F) := \frac{1}{N} \sum_{k=1}^{N} F(\mathbf{t}_k) \ ,$$

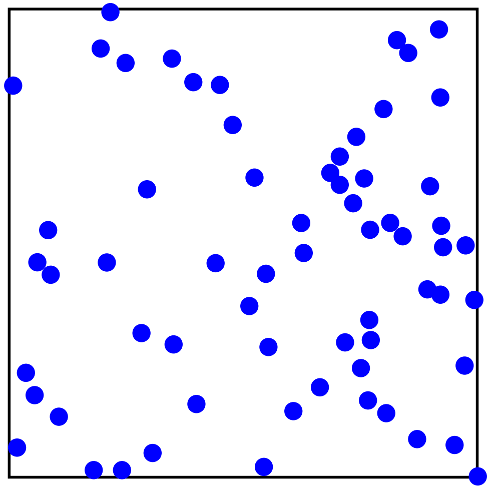
with $\mathbf{t}_1, \ldots, \mathbf{t}_N$ **deterministic** (and cleverly chosen).

How to choose $\mathbf{t}_1, \ldots, \mathbf{t}_N$?

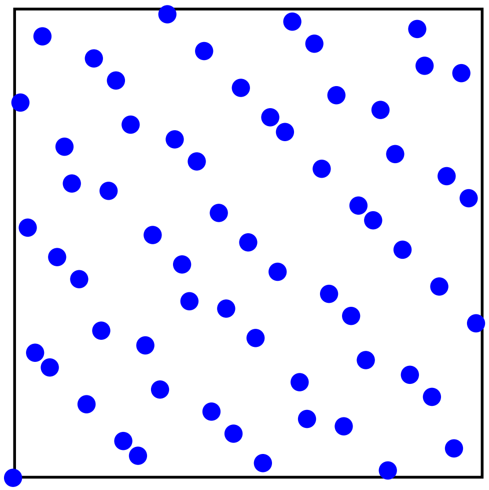
MC and QMC points for 2 dimensions

$$I_s(F) \approx \frac{1}{N}\sum_{k=1}^{N} F(\mathbf{t}_k), \quad \mathbf{t}_k \in [0,1]^s$$

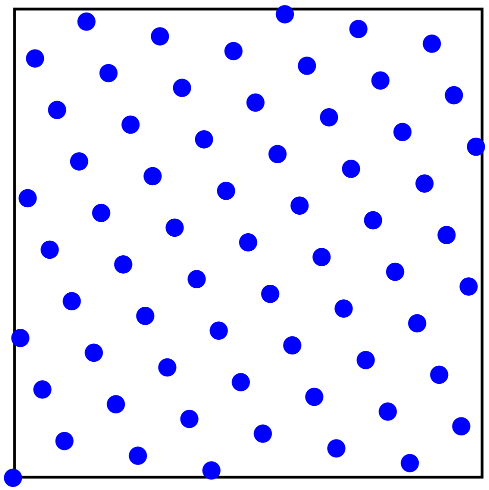
The points:



Monte Carlo method
with 64 “random” points



First 64 points of
2D Sobol′ sequence



A lattice rule
with 64 points

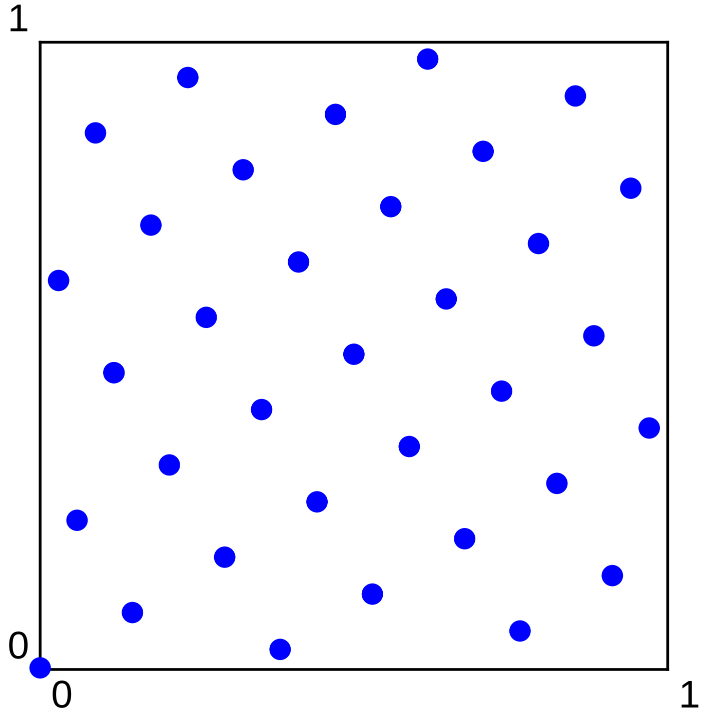
We consider only the simplest kind of lattice rule, given by

$$Q_{N,s}(\mathbf{z};F) = \frac{1}{N}\sum_{k=0}^{N-1} F\left(\left\{k\frac{\mathbf{z}}{N}\right\}\right),$$

where $\mathbf{z} \in \{1,\ldots,N-1\}^s$, and the braces mean that each component of the s-vector in the braces is to be replaced by its fractional part.

Example of lattice rule

$$N = 34, \mathbf{z} = (1, 21)$$



The QMC approach to PDE with random coefficients

- We approximate the s-dimensional integral by an N-point QMC rule,

The QMC approach to PDE with random coefficients

- We approximate the s-dimensional integral by an N-point QMC rule,
- for each QMC point $\mathbf{y}_k, k = 1, \ldots, N$ compute the field $a_s(\mathbf{x}, \mathbf{y}_k)$, and then find an approximate solution of the flow problem by the finite-element method;

The QMC approach to PDE with random coefficients

- We approximate the s-dimensional integral by an N-point QMC rule,
- for each QMC point $\mathbf{y}_k, k = 1, \ldots, N$ compute the field $a_s(\mathbf{x}, \mathbf{y}_k)$, and then find an approximate solution of the flow problem by the finite-element method;
- then compute any desired physical quantity by averaging the N values so obtained.

The QMC approach to PDE with random coefficients

- We approximate the s-dimensional integral by an N-point QMC rule,
- for each QMC point $\mathbf{y}_k, k = 1, \ldots, N$ compute the field $a_s(\mathbf{x}, \mathbf{y}_k)$, and then find an approximate solution of the flow problem by the finite-element method;
- then compute any desired physical quantity by averaging the N values so obtained.

What would we like to achieve?

- Fast convergence (or at at least better than the MC rate $O(N^{-1/2})$);

The QMC approach to PDE with random coefficients

- We approximate the s-dimensional integral by an N-point QMC rule,
- for each QMC point $\mathbf{y}_k, k = 1, \ldots, N$ compute the field $a_s(\mathbf{x}, \mathbf{y}_k)$, and then find an approximate solution of the flow problem by the finite-element method;
- then compute any desired physical quantity by averaging the N values so obtained.

What would we like to achieve?

- Fast convergence (or at at least better than the MC rate $O(N^{-1/2})$);
- And integration errors that are independent of s.

If we are to use a lattice rule **how to choose $\mathbf{z}$?**

Recall: the lattice rule for the integral over $[0,1]^s$ is

$$Q_{N,s}(\mathbf{z};F) = \frac{1}{N}\sum_{k=0}^{N-1} F\left(\left\{k\frac{\mathbf{z}}{N}\right\}\right).$$

So we need to choose $\mathbf{z}$. But **there is no known formula for a "good" $\mathbf{z}$, beyond $s = 2$**.

If we are to use a lattice rule **how to choose $\mathbf{z}$?**

Recall: the lattice rule for the integral over $[0,1]^s$ is

$$Q_{N,s}(\mathbf{z};F) = \frac{1}{N}\sum_{k=0}^{N-1} F\left(\left\{k\frac{\mathbf{z}}{N}\right\}\right).$$

So we need to choose $\mathbf{z}$. But **there is no known formula for a "good" $\mathbf{z}$, beyond $s = 2$.**

Can we **construct** a good $\mathbf{z}$? Yes, it's possible!

The idea

The idea: Assume that $\boldsymbol{F}$ belongs to a convenient function space $\boldsymbol{H}$, one that depends on lots of parameters. Specifically, . . .

The idea

The idea: Assume that $\boldsymbol{F}$ belongs to a convenient function space $\boldsymbol{H}$, one that depends on lots of parameters. Specifically, . . .

And choose those weight parameters to minimise a certain bound on the error for the s-dimensional integral. (What bound? Later!)

The idea

The idea: Assume that F belongs to a convenient function space H, one that depends on lots of parameters. Specifically, . . .

And choose those weight parameters to minimise a certain bound on the error for the s-dimensional integral. (What bound? Later!)

And choose z to minimise the "**worst-case error**" in H.

Worst-case error

Definition: The **worst case error** in the space H of a QMC rule $Q_{N,s}(P;\cdot)$ using the point set $P = \{\mathbf{t}_0, \mathbf{t}_1, \cdots, \mathbf{t}_{N-1}\}$ is

$$\boxed{e_{N,s}(P;H) := \sup_{\|F\|_H \le 1} |I_s(F) - Q_{N,s}(P;F)|},$$

i.e. it is the largest error of $Q_{N,s}(P;F)$ for F in the unit ball of H.

Worst-case error

Definition: The **worst case error** in the space $\boldsymbol{H}$ of a QMC rule $\boldsymbol{Q_{N,s}(P;\cdot)}$ using the point set $\boldsymbol{P} = \{\mathbf{t}_0, \mathbf{t}_1, \cdots, \mathbf{t}_{N-1}\}$ is

$$\boxed{\boldsymbol{e_{N,s}(P;H) := \sup_{\|F\|_H \le 1} |I_s(F) - Q_{N,s}(P;F)|}},$$

i.e. it is the largest error of $\boldsymbol{Q_{N,s}(P;F)}$ for $\boldsymbol{F}$ in the unit ball of $\boldsymbol{H}$.

The following Hilbert space of functions has the big advantage that **the worst case error is computable**.

A good choice of H

A good choice for the norm squared of $\boldsymbol{F}$ in $\boldsymbol{H}$ is

$$\|\boldsymbol{F}\|^{\mathbf{2}}_{\boldsymbol{s},\boldsymbol{\gamma}} := \sum_{\mathfrak{u}\subseteq\{\mathbf{1},\ldots,\boldsymbol{s}\}} \frac{\mathbf{1}}{\gamma_{\mathfrak{u}}} \int_{[\mathbf{0},\mathbf{1}]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}\boldsymbol{F}}{\partial \mathbf{y}_{\mathfrak{u}}}(\mathbf{y}_{\mathfrak{u}}; \tfrac{\mathbf{1}}{\mathbf{2}}) \right|^{\mathbf{2}} \mathbf{d}\mathbf{y}_{\mathfrak{u}},$$

where

$$(\mathbf{y}_{\mathfrak{u}}; \tfrac{\mathbf{1}}{\mathbf{2}})_j = \begin{cases} \boldsymbol{y}_{\boldsymbol{j}} \text{ if } \boldsymbol{j} \in \mathfrak{u}, \\ \frac{1}{2} \text{ if } j \notin \mathfrak{u}. \end{cases}$$

A good choice of H

A good choice for the norm squared of $\boldsymbol{F}$ in $\boldsymbol{H}$ is

$$\|\boldsymbol{F}\|^2_{s,\boldsymbol{\gamma}} := \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}\boldsymbol{F}}{\partial \mathbf{y}_{\mathfrak{u}}}(\mathbf{y}_{\mathfrak{u}};\tfrac{\mathbf{1}}{\mathbf{2}})\right|^2 \mathbf{d}\mathbf{y}_{\mathfrak{u}},$$

where

$$(\mathbf{y}_{\mathfrak{u}};\tfrac{\mathbf{1}}{\mathbf{2}})_j = \begin{cases} y_j \ \text{if}\ j \in \mathfrak{u}, \\ \frac{1}{2}\ \text{if}\ j \notin \mathfrak{u}. \end{cases}$$

The norm squared is the SUM OVER ALL SUBSETS $\mathfrak{u}$ OF $\{1,\ldots,s\}$.

For example, for $\mathfrak{u} = \{1,3\}$ the corresponding term is

$$\frac{1}{\gamma_{\{1,3\}}} \int_0^1 \int_0^1 \left| \frac{\partial^2 F}{\partial y_1 \partial y_3}(y_1, \tfrac{1}{2}, y_3, \tfrac{1}{2}, \tfrac{1}{2}, \ldots)\right|^2 \mathbf{d}y_1\mathbf{d}y_3$$

Weights

$$\|F\|_{s,\boldsymbol{\gamma}}^2 := \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|} F}{\partial \boldsymbol{y}_{\mathfrak{u}}}(\boldsymbol{y}_{\mathfrak{u}}; \tfrac{1}{2}) \right|^2 \, d\boldsymbol{y}_{\mathfrak{u}},$$

The norm squared is the SUM OVER ALL SUBSETS $\mathfrak{u}$ OF $\{1, \ldots, s\}$.

The term for subset $\mathfrak{u}$ is divided by the **weight** $\gamma_{\mathfrak{u}}$. So there are 2^s weights!

The weight $\gamma_{\mathfrak{u}}$ is a positive number that measures the importance of the subset $\mathfrak{u}$. A small weight forces the corresponding derivative to be small.

We denote by $H = H_{s,\boldsymbol{\gamma}}$ the space with weights $\{\gamma_{\mathfrak{u}}\}$.

What's so good about this H?

It's a reproducing kernel Hilbert space with a very simple kernel:

$$K(\mathbf{y},\mathbf{y}') = \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \eta(y_j, y_j') \quad , \quad \text{with}$$

$$\eta(y,y') = \begin{cases} \min(y,y') - \frac{1}{2} & \text{if } y, y' > \frac{1}{2}, \\ \frac{1}{2} - \max(y,y') & \text{if } y, y' < \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

MATHEMATICS & STATISTICS
THE UNIVERSITY OF NEW SOUTH WALES

What’s so good about this H?

It’s a reproducing kernel Hilbert space with a very simple kernel:

$$K(\mathbf{y},\mathbf{y}') = \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \eta(y_j, y_j') \quad , \quad \text{with}$$

$$\eta(y,y') = \begin{cases} \min(y,y') - \frac{1}{2} & \text{if } y, y' > \frac{1}{2}, \\ \frac{1}{2} - \max(y,y') & \text{if } y, y' < \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

What does this mean? It means

$K(\mathbf{y},\cdot) \in H$, and

$$\langle f(\cdot), K(\mathbf{y},\cdot)\rangle_H = f(\mathbf{y}) \quad \forall \mathbf{y} \in [0,1]^s \ \ \forall f \in H.$$

What's so good about this H?

It's a reproducing kernel Hilbert space with a very simple kernel:

$$K(\mathbf{y},\mathbf{y}') = \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \eta(y_j, y_j') \quad , \quad \text{with}$$

$$\eta(y,y') = \begin{cases} \min(y,y') - \frac{1}{2} & \text{if } y, y' > \frac{1}{2}, \\ \frac{1}{2} - \max(y,y') & \text{if } y, y' < \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

What does this mean? It means

$K(\mathbf{y},\cdot) \in H$, and

$$\langle f(\cdot), K(\mathbf{y},\cdot)\rangle_H = f(\mathbf{y}) \quad \forall \mathbf{y} \in [0,1]^s \;\; \forall f \in H.$$

The great thing about a RKHS with kernel is that there is a simple formula for the worst-case error:

The worst case error in a RKHS

For the QMC rule with points $P = \{\mathbf{t}_0, \ldots, \mathbf{t}_{N-1}\}$ the squared WCE is:

$$\begin{aligned} e^2_{N,s}(P;H) &= \int_{[0,1]^s} \int_{[0,1]^s} K(\mathbf{y}, \mathbf{y}')\, d\mathbf{y}\, d\mathbf{y}' \\ &\quad - \frac{2}{N} \sum_{i=0}^{N-1} \int_{[0,1]^s} K(\mathbf{t}_i, \mathbf{y})\, d\mathbf{y} + \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} K(\mathbf{t}_i, \mathbf{t}_k), \end{aligned}$$

Thus in a RKHS with a known kernel the WCE can be **computed**

The worst case error in a RKHS

For the QMC rule with points $P = \{\mathbf{t}_0, \ldots, \mathbf{t}_{N-1}\}$ the squared WCE is:

$$\begin{aligned} e^2_{N,s}(P;H) &= \int_{[0,1]^s} \int_{[0,1]^s} K(\mathbf{y}, \mathbf{y}')\, \mathrm{d}\mathbf{y}\, \mathrm{d}\mathbf{y}' \\ &\quad - \frac{2}{N} \sum_{i=0}^{N-1} \int_{[0,1]^s} K(\mathbf{t}_i, \mathbf{y})\, \mathrm{d}\mathbf{y} + \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} K(\mathbf{t}_i, \mathbf{t}_k), \end{aligned}$$

Thus in a RKHS with a known kernel the WCE can be **computed**

except for the little fact that 2^s terms are needed for each $K(\mathbf{y}, \mathbf{y}')$! .

The case of product weights

Originally we considered only **product weights** (IHS and H Wozniakowski, 98),

$$\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \alpha_j .$$

In this case the WCE is easily computed:

$$\left(\frac{1}{N} \sum_{k=0}^{N-1} \prod_{j=1}^{s} \left(1 + \alpha_j \left[B_2\left(\left\{\frac{kz_j}{N}\right\}\right) + \frac{1}{12} \right]\right) - \prod_{j=1}^{s} \left(1 + \frac{\alpha_j}{12}\right) \right)^{1/2},$$

where $B_2(x) := x^2 - x + 1/6$.

The case of product weights

Originally we considered only **product weights** (IHS and H Wozniakowski, 98),

$$\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \alpha_j .$$

In this case the WCE is easily computed:

$$\left(\frac{1}{N}\sum_{k=0}^{N-1}\prod_{j=1}^{s}\left(1+\alpha_j\left[B_2\left(\left\{\frac{kz_j}{N}\right\}\right)+\frac{1}{12}\right]\right)-\prod_{j=1}^{s}\left(1+\frac{\alpha_j}{12}\right)\right)^{1/2},$$

where $B_2(x) := x^2 - x + 1/6$.

Actually, this is the worst case error in an appropriate root mean square sense for a **randomly shifted** lattice rule.

How to minimise the worst case error?

- Since $\mathbf{z}$ is an integer vector of length s, with values between 1 and $N-1$, in principle we could evaluate the WCE for all possible values of $\mathbf{z}$.

How to minimise the worst case error?

- Since $\mathbf{z}$ is an integer vector of length s, with values between 1 and $N-1$, in principle we could evaluate the WCE for all possible values of $\mathbf{z}$.

- But that would be exponentially costly.

How to minimise the worst case error?

- Since $\mathbf{z}$ is an integer vector of length s, with values between 1 and $N-1$, in principle we could evaluate the WCE for all possible values of $\mathbf{z}$.

- But that would be exponentially costly.

- Fortunately, we can provably get close to the best WCE by the **component-by-component (CBC) algorithm**:

CBC

The CBC algorithm: Korobov, IHS, Reztsov, Kuo, Joe

With CBC, a good generator $\mathbf{z} = (z_1, \ldots, z_s)$ is constructed one component at a time:

- choose $z_1 = 1$
- choose z_2 to minimise WCE for $s = 2$, then
- choose z_3 to minimise WCE for $s = 3$, then
- $\ldots$

so that at each step there are only (at most) $N - 1$ choices.

A naive implementation costs $O(s^2 N^2)$ operations.

Fast CBC

In 2006 Cools and Nuyens developed **Fast CBC** – it uses FFT ideas, and requires a time of order only $\boldsymbol{O(s\ N\log N)}$.

Fast CBC

In 2006 Cools and Nuyens developed **Fast CBC** – it uses FFT ideas, and requires a time of order only $O(s\ N\log N)$.

The Nuyens and Cools implementation allows the CBC algorithm for product weights to be run with s in the thousands, N in the millions.

Now to fix the weights. This is what's new!

Recall the worst case error for integration over $[0,1]^s$:

$$e_{N,s,\boldsymbol{\gamma}}(\mathbf{t}_1,\ldots,\mathbf{t}_N) := \sup_{\|F\|_{s,\boldsymbol{\gamma}}\le 1} |I_s(F) - Q_{s,N}(F)| \,.$$

Now to fix the weights. This is what's new!

Recall the worst case error for integration over $[0,1]^s$:

$$e_{N,s,\boldsymbol{\gamma}}(\mathbf{t}_1,\ldots,\mathbf{t}_N) := \sup_{\|F\|_{s,\boldsymbol{\gamma}}\le 1} |I_s(F) - Q_{s,N}(F)| .$$

Given F, we can use the resulting error bound:

$$\boxed{|I_s(F) - Q_{s,N}(F)| \le e_{N,s,\boldsymbol{\gamma}}(\mathbf{t}_1,\ldots,\mathbf{t}_N) \ \times\ \|F\|_{s,\boldsymbol{\gamma}}}$$

and choose weights that minimize the right-hand side

or some upper bound on the right-hand side.

Skipping details, for the PDE problem

$$\text{Error} \;\leq \frac{\boldsymbol{C}}{\boldsymbol{N}^{(1/2\boldsymbol{\lambda})}} \left(\sum_{0<|\mathfrak{u}|<\infty} \boldsymbol{\gamma}_{\mathfrak{u}}{}^{\boldsymbol{\lambda}} \boldsymbol{A}_{\mathfrak{u}} \right)^{1/2\boldsymbol{\lambda}} \times \left(\sum_{|\mathfrak{u}|<\infty} \frac{\boldsymbol{B}_{\mathfrak{u}}}{\boldsymbol{\gamma}_{\mathfrak{u}}} \right)^{1/2},$$

for all $\boldsymbol{\lambda} \in (\frac{1}{2}, 1]$, where $\boldsymbol{A}_{\mathfrak{u}} = \ldots$ and $\boldsymbol{B}_{\mathfrak{u}} = \ldots$.

We cannot take $\boldsymbol{\lambda} = \frac{1}{2}$ because $\boldsymbol{A}_{\mathfrak{u}} \to \infty$ as $\boldsymbol{\lambda} \to \frac{1}{2}+$.

Skipping details, for the PDE problem

$$\text{Error} \;\le\; \frac{C}{N^{(1/2\lambda)}} \left(\sum_{0<|\mathfrak{u}|<\infty} {\gamma_{\mathfrak{u}}}^{\lambda} A_{\mathfrak{u}} \right)^{1/2\lambda} \times \left(\sum_{|\mathfrak{u}|<\infty} \frac{B_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}} \right)^{1/2},$$

for all $\lambda \in (\frac{1}{2}, 1]$, where $A_{\mathfrak{u}} = \ldots$ and $B_{\mathfrak{u}} = \ldots$.

We cannot take $\lambda = \frac{1}{2}$ because $A_{\mathfrak{u}} \to \infty$ as $\lambda \to \frac{1}{2}+$.

Minimising the product yields:

$$\gamma_{\mathfrak{u}} = \left(\frac{B_{\mathfrak{u}}}{A_{\mathfrak{u}}} \right)^{1/(1+\lambda)} = (|\mathfrak{u}|!)^{\frac{2}{1+\lambda}} \prod_{j\in\mathfrak{u}} \alpha_j, \quad \alpha_j = \ldots$$

The $\gamma_{\mathfrak{u}}$ are “POD” (for product and order dependent) weights.

For the PDE application assuming exact PDE solution

Convergence theorem. (Kuo/Schwab/IHS 2012) If $\mathbf{z}$ is chosen by CBC, using the minimising weights, then with $F(\mathrm{y}) = G(u(\mathrm{y}))$,

$$\text{Cubature error for approximating } F \le \frac{C_\delta}{N^{1-\delta}}. \quad \text{for all } \delta > 0.$$

And generalizations ....

For the PDE application assuming exact PDE solution

Convergence theorem. (Kuo/Schwab/IHS 2012) If $\mathbf{z}$ is chosen by CBC, using the minimising weights, then with $F(\mathbf{y}) = G(u(\mathbf{y}))$,

$$\text{Cubature error for approximating } F \le \frac{C_\delta}{N^{1-\delta}}. \quad \text{for all } \delta > 0.$$

And generalizations ....

The error is independent of s provided $\sum_{j=1}^{\infty} \|\psi_j\|_{\infty}^{2/3} < \infty$

Thus the convergence is **provably** faster than the Monte Carlo rate $N^{-1/2}$ – **and the CURSE OF DIMENSIONALITY has disappeared!**

What else is new?

- Similar analysis of the **lognormal** case.
- Analysis of a **multilevel** method for both uniform and lognormal cases:

[Kuo, Schwab, Sloan (to appear)]

$$I(G(u)) \approx Q_*^L(G(u)) = \sum_{\ell=0}^{L} Q_{s_\ell,n_\ell}\left(G(u_{h_\ell}^{s_\ell} - u_{h_{\ell-1}}^{s_{\ell-1}})\right)$$

$$\mathrm{cost} = \mathcal{O}\left(\sum_{\ell=0}^{L} s_\ell n_\ell {h_\ell}^{-d}\right)$$

What's newer still?

We know how to construct **higher order** QMC rules, with e.g. O(N^{-2}), and even how to apply them the above PDE problem.

(Dick, Kuo, Le Gia, Nuyens and Schwab 2015 for the uniform case)

[The best convergence rate achievable with a lattice rule is (close to) O(N^{-1}).]

Lattice rules are now replaced by **interlaced polynomial lattice rules**, and POD weights by **SPOD** weights (standing for "smoothness-driven product and order-dependent weights")

Now randomisation is not needed, and the rate of convergence is better: the theoretical convergence rate is $O(N^{-1/p})$, instead of $O(N^{-1/p+1/2})$ for lattice rules with $\frac{2}{3} < p \leq 1.$

But higher order QMC is still very new, and there are few calculations.

Summarising the PDE+QMC story

Summarising the PDE+QMC story

- we truncate the ∞-dimensional problem at a high enough s;

Summarising the PDE+QMC story

- we truncate the ∞-dimensional problem at a high enough s;
- we use a theoretical bound on the norm of the quantity of interest to deduce good (POD) weights $\boldsymbol{\gamma}_{\mathfrak{u}}$.

Summarising the PDE+QMC story

- we truncate the ∞-dimensional problem at a high enough s;
- we use a theoretical bound on the norm of the quantity of interest to deduce good (POD) weights $\boldsymbol{\gamma}_{\mathfrak{u}}$.
- using those good weights, we compute good a lattice parameter $\boldsymbol{z}$ by the fast CBC algorithm;

Summarising the PDE+QMC story

- we truncate the ∞-dimensional problem at a high enough s;
- we use a theoretical bound on the norm of the quantity of interest to deduce good (POD) weights $\boldsymbol{\gamma}_{\mathfrak{u}}$.
- using those good weights, we compute good a lattice parameter $\mathbf{z}$ by the fast CBC algorithm;
- using the corresponding lattice points $\mathbf{y}_k, k = 1, \ldots, N$ we compute the field $a(\mathbf{x}, \mathbf{y}_k)$, then find the approximate solution of the PDE and compute the quantity of interest (QOI), and then average the QOI over all N realisations;

Summarising the PDE+QMC story

- we truncate the ∞-dimensional problem at a high enough s;
- we use a theoretical bound on the norm of the quantity of interest to deduce good (POD) weights $\boldsymbol{\gamma}_{\mathfrak{u}}$.
- using those good weights, we compute good a lattice parameter $\mathbf{z}$ by the fast CBC algorithm;
- using the corresponding lattice points $\mathbf{y}_k, k = 1, \ldots, N$ we compute the field $a(\mathbf{x}, \mathbf{y}_k)$, then find the approximate solution of the PDE and compute the quantity of interest (QOI), and then average the QOI over all N realisations;
- and so obtain rigorously and constructively **a convergence rate better than MC, and an error bound independent of s.**

But some cautions

- The constants may be very large.
- The numerical evidence is so far not completely convincing. We always do better than MC, but often no better than with an off-the-shelf QMC rule.
- For other high-dimensional applications the present theory cannot be applied at all (option pricing), or is way in the future (weather and climate).

Nevertheless, high-dimensional problems will not go away, and we are perhaps making some progress.

In the world of high dimensions, we live in interesting times!

Some reading

- I H Sloan, *What's new in high dimensional integration? – designing quasi-Monte Carlo for applications*, Proceedings of ICIAM 2015, Beijing, China, Higher Education Press, 2015. pp. 365–386.
- J. Dick, F.Y. Kuo, & I.H. Sloan, *Numerical integration in high dimensions – the Quasi Monte Carlo way*, Acta Numerica 2013.
- F.Y. Kuo and D Nuyens, *Application of quasi-Monte Carlo methods to PDEs with random coefficients – survey of analysis and implementation*, in preparation.

Collaborators

Josef Dick, UNSW
Ivan Graham, Bath
Michael Griebel, Bonn
Fred Hickernell, IIT
Stephen Joe, Waikato
Frances Kuo, UNSW
Q Thong Le Gia, UNSW
James Nichols, UNSW
Erich Novak, Jena
Dirk Nuyens, Leuven
Leszek Plaskota, Warsaw
Andrei Reztsov
Rob Scheichl, Bath
Christoph Schwab, ETH
Elisabeth Ullmann, Hamburg
Grzegorz Wasilkowski, Kentucky
Henryk Woźniakowski, Warsaw and Columbia