Integration and Approximation in High Dimensions – a Tutorial

Ian Sloan

i.sloan@unsw.edu.au

University of New South Wales, Sydney, Australia
Hong Kong Polytechnic University, Hong Kong

Uncertainty Quantification, Edinburgh, May 2010

High dimensional integration

First INTEGRATION, later APPROXIMATION

Consider

\[ I_{s, \rho} f := \int_{\Omega_s} f(x_1, \ldots, x_s) \rho(x_1, \ldots, x_s) \, dx_1 \cdots dx_s = \int_{\Omega_s} f(x) \rho(x) \, dx, \]

where \( \Omega_s \subseteq \mathbb{R}^s \) and \( \rho \) is a probability density:

\[ \rho(x) \geq 0, \quad \text{and} \quad \int_{\Omega_s} \rho(x) \, dx = 1; \]

and \( s \) is LARGE.

Important special cases:
1. \( \Omega_s = \mathbb{R}^s, \quad \rho(x) = \frac{1}{(2\pi)^{s/2}} \exp(-\frac{1}{2}(x_1^2 + \ldots + x_s^2)), \)
2. \( \Omega_s = [0,1]^s, \quad \rho(x) = 1. \)

Where do we see high-dimensional integrals?

Often they are high-dimensional expected values:

**Example:** In 1995 Traub and Paskov computed values of ‘mortgage-backed obligations’. In the USA people buying a house borrow money from a bank. In return, the bank holds a ‘mortgage’. US mortgages last for 30 years, and may be repaid at the end of any month, making \( 30 \times 12 = 360 \) repayment possibilities. The monthly change in the interest rate is treated as a random variable, with some assumed (usually Gaussian) probability distribution. There are 360 random variables, so the computed quantity is a 360-dimensional expected value.

(The success of that experiment, using “quasi-Monte Carlo methods”, generated much interest in quasi-Monte Methods.)

Quadrature

or cubature, or numerical integration

\[ I_{s, \rho} f = \int_{\Omega_s} f(x) \rho(x) \, dx \approx \sum_{k=1}^{N} w_k f(t_k), \]

where \( t_1, \ldots, t_N \) are in \( \Omega_s \), and the weights \( w_k \) are real (and hopefully positive).

METHODS:

- Monte Carlo (MC)
- Quasi-Monte Carlo (QMC)
- Sparse grid
- Product rules?
**Product rule**

A product rule uses a 1-dimensional rule for each coordinate direction. (Of course this makes sense only if $\Omega_s$ is a product region.) Product rules are very bad for large $s$.

E.g. take $\Omega_s = [0, 1]^{s}$ and $\rho(x) = 1$.

If there are $N = n^s$ points, and if

$$f(x_1, \ldots, x_s) = x_1,$$

then the result is the same as for the 1-dimensional rectangle rule, so

$$\text{ERROR} = O\left(\frac{1}{n}\right) = O\left(\frac{1}{N^{1/s}}\right)!$$

This is the 'Curse of Dimensionality'.

---

**Monte Carlo (MC)**

$$I_{s,\rho}f \approx M_{N,s}f = \frac{1}{N} \sum_{k=1}^{N} f(t_k),$$

where $t_1, \ldots, t_N$ are independent random samples from $\Omega_s$ drawn from the probability density $\rho$.

Error: For $f \in L^2(\Omega_s)$, the error in the least squares sense is

$$\text{error} = \sigma(f) \frac{1}{\sqrt{N}}.$$
Pseudo-random versus random

In practice of there are no true random numbers available to us, so we must use pseudo-random numbers from some “random-number generator”.

For the science of pseudo-random numbers, see Niederreiter (1992).

Caution: the quality of available random number generators varies widely.

Quasi-Monte Carlo (QMC)

Now assume that $\Omega_s = [0, 1]^s$ and $\rho(x) = 1$. In practice this usually means that some transformation to the unit cube has already been carried out – a big subject, but one we won’t have time to address in these lectures.

$$I_s f = \int_{[0,1]^s} f(x) dx \approx Q^{\text{QMC}}_{N,s} f := \frac{1}{N} \sum_{k=1}^N f(t_k),$$

with $t_1, \ldots, t_N$ deterministic (and cleverly chosen).

How to choose $t_1, \ldots, t_N$?

- Low discrepancy points
  (Halton, Sobol’, Faure, Niederreiter, …)
- Lattice rules

Summary of quadrature rules

Algorithm:

$$I_s f \approx \sum_{k=1}^N w_k f(t_k)$$

- Product rule with 64 points
- Sparse grid with 49 points
- 64 “random” points
- First 64 points of 2D Sobol’ sequence
- A lattice rule with 64 points

Classical QMC – the Koksma-Hlawka inequality

Reference: Niederreiter (1992)

The Koksma-Hlawka inequality is

$$|I_s f - Q^{\text{QMC}}_{N,s} f| \leq D^*(t_1, \ldots, t_N) V(f).$$

Here $V(f)$ is the variation of $f$ in the sense of Hardy and Krause, and $D^*(t_1, \ldots, t_N)$ is the “star discrepancy”.

For a given function $f$, to make the bound as small as possible we want to make $D^*(t_1, \ldots, t_N)$ small.
Star Discrepancy $D^*(t_1, \ldots, t_N)$

The **star discrepancy** is the supremum, over all (half-open) rectangular regions anchored at the origin, of

\[
\frac{\text{number of QMC points in the region}}{\text{total number of points}} - \text{volume of the region}.
\]

Low discrepancy sequence

A **LOW-DISCREPANCY** sequence $t_1, t_2, \ldots$ of points in $[0, 1]^s$ is one for which the first $N$ members satisfy

\[
D^*(t_1, \ldots, t_N) \leq C_s (\log N)^s / N.
\]

For fixed $s$ the resulting error bound is “better than MC” if $N$ is large.

**But** for fixed $s$ the bound on the right grows with $N$ until $N \approx e^s$.

E.g. Sobol' (1967):

Random vs Sobol' points

A comparison of 64 (pseudo)random and 64 Sobol' points ...

Sobol' points

The Sobol' points in the picture are well distributed in this sense: If we partition the square into 64 (half-open) rectangles of size $1 \times \frac{1}{64}$ or $\frac{1}{2} \times \frac{1}{32}$ etc, then each rectangle contains exactly one point.

In the language of “nets”, we have a $(t, m, s)$ net in base 2, with $s = 2$ (the dimension), $m = 6$ (the number of points is $2^m$), and $t = 0$.

The “$t$” in a $(t, m, s)$ net is a (lack of) quality parameter: if we divide the region into $2^{m-t}$ rectangles of equal size and shape then each rectangle contains $2^t$ points. We are giving a lot away as $t$ increases!

Small $t$ is best! **But** as the dimension increases $t$ must increase.
Example of a bad 2-dimensional projection

From Joe and Kuo (2008), showing what can go wrong. This is the (19, 28) two-dimensional projection of $4096 = 2^{12}$ Sobol’ points. The value of $t$ is 6, so this is a $(6, 12, 2)$ net.

Some practical comments on Sobol’ points

Remember that the Sobol’ points have $s$ dimensions. The points are constructed one component at a time. For component $j$, the designer has to choose a primitive polynomial in the field $\mathbb{Z}_2$ (a polynomial that has not been used for earlier components) and of degree say $d_j$. This polynomial is used in a recurrence relation, so you also need $d_j$ starting numbers, called “direction numbers”.

Bratley and Fox (1998) give an efficient implementation using graycode. Primitive polynomials and direction numbers are given on the web site of Frances Kuo, http://www.maths.unsw.edu.au/~fkuo/sobol, for dimensions up to 21201. They are designed to optimize the 2-dimensional projections, following Joe and Kuo (2008).

More about Sobol’ points

In practice Sobol’ points are always presented as an infinite sequence. This is done by ingenious filling in between the powers of 2. It brings us to the idea of a $(t, s)$ sequence: an infinite sequence of points in $[0, 1]^s$ is a $(t, s)$ sequence in base 2 if every block of $2^m$ points is a $(t, m, s)$ net. The definition does not restrict the way other points are filled in.

As the dimension $s$ increases, the supply of primitive polynomials in $\mathbb{Z}_2$ of a fixed degree becomes exhausted, so the degree $d_s$ of the primitive polynomial for the latest component $s$ is eventually forced to increase. This is bad for the $t$ value since $t$ for the $s$-dimensional sequence is given by

$$t = \sum_{j=1}^{s} (d_j - 1).$$

An important QMC principle

Thus we come to an important QMC principle:

When using Sobol’ points, and indeed any QMC rule, you should always order the variables in terms of their importance.

Contrast this with Monte Carlo: it does not care about the order of the variables!
Other low discrepancy sequences

Other low discrepancy sequences are due to Halton (1960), Faure (1982), Niederreiter (1980s on), Xing ...

The Halton sequence uses successive primes 2, 3, 5 ... for the successive coordinate directions; it is not usually recommended. All the rest are \((t, m, s)\) nets, but with different bases.

Refs: Niederreiter (1992), Dick and Pillichshammer (2010). (The latter describes potentially important recent developments of analogous sequences with higher order convergence.)

Lattice rules

Lattice rules for periodic functions

Korobov, 1959 and Hlawka, 1961
L K Hua and Y Wang
Frolov, 1977
Sloan & Kachoyan, 1984
Lyness
Sloan & Joe (book), 1994

Lattice rules for non-periodic functions

L’Ecuyer, .... , in \(L_2\) context
Sloan & Woźniakowski, Hickernell, Joe, Kuo, Dick, Larcher, Wang, Waterhouse, ...., from 2001

Lattice rule definition

Lattice rule (of rank 1)

\[
Q_{\text{lattice}}^{N,s,z} f = \frac{1}{N} \sum_{k=1}^{N} f \left( \left\{ \frac{k z}{N} \right\} \right),
\]

where \(z \in \{1, \ldots, N - 1\}^s\), and the braces around a vector indicate that each component is to be replaced by its fractional part.


Shifted lattice rule

\[
Q_{\text{lattice}}^{N,s,z,\Delta} f = \frac{1}{N} \sum_{k=1}^{N} f \left( \left\{ \frac{k z}{N} + \Delta \right\} \right),
\]

Here \(\Delta\) (the "shift") \(\in [0, 1]^s\).

Example of lattice & shifted lattice rules

\(N = 34, z = (1, 21)\)
\(N = 34, z = (1, 21), \Delta = (0.8, 0.1)\)
Randomly shifted QMC rules

Given the QMC rule

\[ Q_{N,s}^{\text{QMC}} f = \frac{1}{N} \sum_{k=1}^{N} f(t_k), \]

with \( t_1, \ldots, t_N \) deterministic, the corresponding randomly shifted QMC rule is

\[ Q_{N,s,\Delta}^{\text{QMC}} f := \frac{1}{N} \sum_{j=1}^{N} f(\{ t_k + \Delta \}), \]

where \( \Delta \) is chosen from a uniform distribution on \([0, 1)^s\).

This family is an unbiased estimator of \( I_s f \) (exercise).

Randomized QMC in practice

In practice we recommend making \( q \) random selections of \( \Delta \), for say \( q = 10 \) or \( 30 \), and then taking as our estimate the mean

\[ \bar{Q}_{N,s} f := \frac{1}{q} \sum_{i=1}^{q} Q_{N,s,\Delta_i}^{\text{QMC}} f \]

where \( \Delta_1, \ldots, \Delta_q \) are chosen randomly from a uniform distribution on \([0, 1)^s\). Now the approximation has Monte Carlo features - the error can be estimated: the error estimate is

\[ \frac{1}{\sqrt{q}} \left( \frac{1}{q-1} \sum_{i=1}^{q} (Q_{N,s,\Delta}^{\text{QMC}} f - \bar{Q}_{N,s} f)^2 \right)^{1/2}. \]

Remark: For the Sobol’ points random shifting destroys the net property. We really should here do a digital shift – see Dick & Pillichshammer (2010).

A key insight into QMC rules

Both Sobol’ points (with \( 2^m \) points) and lattice rules will do very well for the function

\[ f(x) = \cos(x_1^2) + 3 \sqrt{x_2^2 + 1}, \]

and more generally for any integrand of the form

\[ f(x_1, \ldots, x_s) = f_1(x_1) + \ldots + f_s(x_s). \]

This is because the projection of the QMC points onto any axis is a set of \( N \) equally spaced points. Thus for a function of a single component \( x_j \) the effect is of a 1-dimensional \( N \)-point rectangle rule – and remember that \( N \) is typically large. By linearity the same is true for a sum of functions each of which depends only on one component of \( x \).

Of course this is a special case, but often a representation like this is true in an approximate sense.

ANOVA decomposition

Every \( f \) can be written (in many ways) as a sum of the form

\[
\begin{align*}
  f(x) &= f_0 + f_1(x_1) + f_2(x_2) + \ldots + f_s(x_s) \\
        &+ f_{1,2}(x_1, x_2) + f_{1,3}(x_1, x_3) + \ldots + f_{s-1,s}(x_{s-1}, x_s) \\
        &+ \ldots + f_{1,\ldots,s}(x_1, \ldots, x_s) \\
        &= \sum_{u \subseteq \{1, \ldots, s\}} f_u(x_u).
\end{align*}
\]

Here \( x_u \), for \( u \) a given subset of \( \{1, \ldots, s\} \), is the vector made up of the components \( x_j \) of \( x \) for which \( j \) lies in \( u \).

The sum is unique (exercise), and is the ANOVA decomposition, if

\[ \int_{0}^{1} f_u(x_u) dx_j = 0 \quad \text{for} \quad j \in u \quad \text{(which implies} \quad f_0 = I_s f). \]
ANOVA — Analysis of Variance

Using the ANOVA decomposition of $f$, define

$$\sigma_u(f)^2 := \int_{[0,1]^s} f_u(x_u)^2 \, dx, \quad u \neq \emptyset.$$  

Because the ANOVA terms are $L_2$ orthogonal, that is (exercise)

$$\int_{[0,1]^s} f_u(x_u)f_v(x_v) \, dx = 0 \quad \text{for} \quad u \neq v,$$

we have:

$$\sigma(f)^2 = \sum_{u \neq \emptyset} \sigma_u(f)^2.$$  

Thus the variance of $f$ is the sum of variances of the ANOVA terms.

Effective dimension

By now it is well accepted that QMC methods, when they work well, do so because the effective dimension of the problem is in some sense much less than the nominal dimension. Caflisch, Morokoff and Owen (1997) made this idea quantitative:

Definition: The truncation dimension of $f$ is $d_t$ if the ANOVA terms $f_u$ with $u \subseteq \{1, \ldots, d_t\}$ capture 99% or more of the total variance $\sigma(f)^2$.

Definition: The superposition dimension of $f$ is $d_s$ if the ANOVA terms $f_u$ with $|u| \leq d_s$ capture 99% or more of the total variance $\sigma(f)^2$.

Effective dimension and QMC

Effective dimension is a key to understanding QMC. Consider the 100-dimensional integral

$$\int_{[0,1]^{100}} \prod_{j=1}^{100} (x_j - \frac{1}{2}) \, dx.$$  

For this $f$ there is just one ANOVA term, namely $f_{\{1,2,\ldots,100\}}$. The effective dimension in both senses is equal to the nominal dimension 100. And the problem is truly hard, unless you use special knowledge.

Success with QMC generally requires that the effective dimension is relatively small.

Function space settings

We can’t do more until we choose a function space setting. Classically, lattice rules were designed for periodic functions with fast decaying Fourier coefficients

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

A classical setting is the Banach space $B_{\text{Korobov}}$, with the norm

$$\|f\|_{\text{Korobov}} := \sup\{ |\hat{f}(h)| \prod_{j=1}^{s} \max(|h_j|, 1)^2 : h \in \mathbb{Z}^s \}.$$  

$$\Rightarrow |\hat{f}(h)| \leq \frac{\|f\|_{\text{Korobov}}}{\prod_{j=1}^{s} \max(|h_j|, 1)^2} \quad \text{for} \quad f \in B_{\text{Korobov}}.$$  

(In one dimension the Fourier coefficients $\hat{f}(h)$ decay like $h^{-2}$.)
A classical lattice-rule theorem

**Definition:** The “worst-case error” for $Q_{N,s}$ in a Banach space $B$ is

$$e(B, Q_{N,s}) := \sup \{||I_s f - Q_{N,s} f|| : f \in B, \|f\| = 1\},$$

which implies $|I_s f - Q_{N,s} f| \leq e(B, Q_{N,s}) \|f\|$ for $f \in B$.

**A theorem of Korobov:** For $N$ prime, there exists a rank-1 lattice rule $Q_{N,s,z}$ such that the worst-case error in $B_{\text{Korobov}}$ satisfies

$$e(B_{\text{Korobov}}, Q_{N,s,z}) \leq c_s N^2.$$

But in practice most functions are not periodic. And although there exist useful periodizing transformations in low dimensions, in higher dimensions they tend to make even easy problems very hard by increasing the effective dimension.

Modern versus classical settings

The classical function space settings above assumed either that $f$ has bounded variation $V(f)$ in the sense of Hardy and Krause, or that $f$ is periodic with suitably decaying Fourier series.

While the classical results are very interesting with respect to what happens as $N$ increases, they are much less helpful with respect to increasing dimension $s$.

Modern function space settings are generally **Reproducing Kernel Hilbert Spaces**, simply because they are easy to work with.

The curse of dimensionality in $B_{\text{Korobov}}$

The dependence of $c_s$ on $s$ in

$$e(B_{\text{Korobov}}, Q_{N,s}^{\text{lattice}}) \leq \frac{c_s}{N^2}$$

is bad. How bad is bad? Very bad!

Theorem: (Sloan & Woźniakowski, 1997)

For $N < 2^s$, and for every quadrature rule $Q_{N,s}$,

$$e(B_{\text{Korobov}}, Q_{N,s}) \geq 1.$$

For $N < 2^s$ the result is as bad as it could be, since we get $e(B, Q_{N,s}) = 1$ even for the quadrature rule $Q_{N,s} f \equiv 0$!

RKHS – definition

Suppose $H$ is a Hilbert space of functions on a domain $\Omega$, with inner product $(f, g)_H$. It is a Reproducing Kernel Hilbert Space (RKHS) if there exists a kernel $K(x, y)$ with these three properties:

1. $K(x, y) = K(y, x)$,
2. For $y \in \Omega$ we need $K(\cdot, y) \in H$,
3. For $f \in H$,

$$(f, K(\cdot, y))_H = f(y), \ \forall \ y \in \Omega.$$

Mathematically, $H$ is a RKHS iff point evaluation in $H$ is a bounded linear functional.

Notation: we often write a RKHS with kernel $K$ as $H_K$. 

Modern versus classical settings

Integ. & approxn in high dims p. 33/101

Modern function space settings are generally **Reproducing Kernel Hilbert Spaces**, simply because they are easy to work with.
An important 1-dimensional RKHS

The space of $L_2$ functions on $[0, 1]$ with square integrable first derivatives, and with norm

$$(f, g)_H = f(0)g(0) + \int_0^1 \frac{df}{dx}(x)\frac{dg}{dx}(x)\,dx,$$

is a RKHS with reproducing kernel

$$K(x, y) = 1 + \min(x, y).$$

This RKHS has been used in many publications. An advantage is that it is easy to prove the reproducing property:

Proof of the reproducing property

$$K(x, y) = \begin{cases} 1 + x & \text{if } x < y \\ 1 + y & \text{if } x > y \end{cases}$$

hence

$$\frac{\partial K}{\partial x} = \begin{cases} 1 & \text{if } x < y \\ 0 & \text{if } x > y \end{cases}.$$ 

From the definition $(f, g)_H = f(0)g(0) + \int_0^1 f'(x)g'(x)\,dx$ of the inner product we have

$$(f, K(\cdot, y))_H = f(0)1 + \int_0^y \frac{df}{dx}(x)\,dx$$

$$= f(0) + f(y) - f(0) = f(y).$$

The worst-case error in a RKHS

Why are RKHS useful? Because in a RKHS we can compute the worst-case error:

Recall the worst-case error of the rule $Q_{N,s}$ in a Hilbert space $H$:

$$e(H, Q_{N,s}) := \sup \left\{ \left| \int_s f - \sum_{k=1}^N w_k f(t_k) \right| : f \in H, \|f\|_H \leq 1 \right\}.$$ 

Theorem. In a RKHS $H_K$ the worst-case error satisfies

$$e(H_K, Q_{N,s})^2 = \iint_{[0,1]^2} K(x, y)\,dx\,dy$$

$$-2 \sum_{k=1}^N w_k \int_{[0,1]} K(x, t_k)\,dx + \sum_{k=1}^N w_k \sum_{k'=1}^N w_{k'} K(t_k, t_k').$$

Proof that $e(H_K, Q_{N,s})^2 = \ldots$ in a RKHS

We can write

$$f(x) = (f, K(\cdot, x))_{H_K}$$

$$\Rightarrow \sum_{k=1}^N w_k f(t_k) = (f, \sum_{k=1}^N w_k K(\cdot, t_k))_{H_K}$$

and similarly

$$I_s f = \left( f, \int_{[0,1]} K(\cdot, x)\,dx \right)_{H_K}$$

$$\Rightarrow I_s f - \sum_{k=1}^N w_k f(t_k) = \left( f, \int_{[0,1]} K(\cdot, x)\,dx - \sum_{k=1}^N w_k K(\cdot, t_k) \right)_{H_K}.$$ 

Now use: for $g$ in a Hilbert space $H$,

$$\sup\{|(f, g)_H : f \in H, \|f\|_H \leq 1\} = \|g\|_H.$$ 

(Hint: Use Cauchy-Schwarz, and note that the inequality is sharp if $f = g/\|g\|_H$.)
Proof that $e(H, Q_{N,s})^2 = \ldots$ (continued)

Thus

$$e(H, Q_{N,s}) := \sup\{ |I_s f - \sum_{k=1}^{N} w_k f(t_k)| : f \in H, \|f\|_H \leq 1 \}$$

$$= \sup\{ \left( f, \int_{[0,1]^s} K(\cdot, x)dx - \sum_{k=1}^{N} w_k K(\cdot, t_k) \right)_{H_K} : f \in H, \|f\|_H \leq 1 \}$$

$$= \left\| \int_{[0,1]^s} K(\cdot, x)dx - \sum_{k=1}^{N} w_k K(\cdot, t_k) \right\|_{H_K}.$$

Now the result follows (exercise) if we square both sides, use

$$\|g\|_H^2 = (g, g)_H,$$

then expand the inner product into four terms, and finally use the reproducing property:

A second RKHS (the “unanchored” space)

I prefer another 1-dimensional RKHS, with an equivalent norm:

$$(f, g)_H = \left( \int_0^1 f(x)dx \right) \left( \int_0^1 g(x)dx \right) + \int_0^1 \frac{df}{dx}(x) \frac{dg}{dx}(x)dx.$$

It is a RKHS with reproducing kernel (it’s a take-home exercise to verify this)

$$K(x, y) = 1 + \frac{1}{2} B_2(|x - y|) + (x - \frac{1}{2})(y - \frac{1}{2}),$$

where $B_2(x)$ is the Bernoulli polynomial of degree 2,

$$B_2(x) := x^2 - x + \frac{1}{6}.$$

We will mostly use this RKHS from now on.

A corresponding $s$-dimensional space

The easiest way to define an $s$-dimensional space is as a tensor product space. The tensor product norm in our preferred space is:

$$(f, g)_H = \left( \int_0^1 f(x_1, x_2)dx_1 dx_2 \right) \left( \int_0^1 g(x_1, x_2)dx_1 dx_2 \right)$$

$$+ \int_0^1 \left| \frac{\partial f}{\partial x_1}(x_1, x_2) \right| dx_1 + \int_0^1 \left| \frac{\partial f}{\partial x_2}(x_1, x_2) \right| dx_1 dx_2$$

Note that if $f(x_1, x_2) = g(x_1)h(x_2)$ then $$\|f\|_2 = \|g\|_1 \|h\|_1.$$
A concise notation for general $s$

$$
\|f\|_s^2 = \sum_{u \subseteq \{1, \ldots, s\}} \int_{[0,1]^u} \int_{[0,1]^{s-u}} \left| \frac{\partial f}{\partial x_u}(x_u, x_{-u}) \right| dx_u dx_{-u}^2
$$

where for each subset $u \subseteq \{1, \ldots, s\}$,

- $x_u$ stands for the components $x_j$ of $x$ with $j \in u$,
- $x_{-u}$ stands for the remaining components,
- $[0,1]^u$ is the unit cube for the variables $x_u$,
- $[0,1]^{s-u}$ is the unit cube for the variables $x_{-u}$,
- $|u|$ is the cardinality of $u$.

The kernel of the $s$-dimensional space

This reproducing kernel of this space is a simple product:

$$
K_s(x, y) = \prod_{j=1}^{s} K_1(x_j, y_j).
$$

where

$$
K_1(x, y) := 1 + \frac{1}{2} B_2(|x - y|) + (x - \frac{1}{2})(y - \frac{1}{2}).
$$

Weighted function spaces

Sloan & Woźniakowski (1998) introduced “weighted” spaces, i.e.
spaces in which the coordinate directions $x_1, x_2, \ldots$ are weighted by parameters $\gamma_1, \gamma_2, \ldots$, with

$$
\gamma_1 \geq \gamma_2 \geq \cdots > 0,
$$

BECAUSE IN SOME PROBLEMS THE VARIABILITY OF $f$ IS:

- greatest in $x_1$ direction
- less in $x_2$ direction
- still less in $x_3$ direction

...
Or in general, using the concise notation,
\[ \|f\|_{2,s,\gamma}^2 = \sum_{u \subseteq \{1, ..., s\}} \frac{1}{\gamma_u} \int_{[0,1]^u} \left| \int_{[0,1]^{1-u}} \frac{\partial f}{\partial x_u}(x_u, x-u) \, dx_u \right|^2 \, dx_u, \]
where
\[ \gamma_\emptyset = 1, \quad \gamma_u := \prod_{j \in u} \gamma_j \text{ for } u \neq \emptyset. \]

**The kernel of the weighted RKHS**

This space is a RKHS with kernel:
\[ K_{s,\gamma}(x, y) = \prod_{j=1}^s K_{1,\gamma_j}(x_j, y_j) = \prod_{j=1}^s (1 + \gamma_j \eta(x_j, y_j)), \]
where
\[ \eta(x, y) = \frac{1}{2} B_2(|x - y|) + (x - \frac{1}{2})(y - \frac{1}{2}). \]

Expanding out the product, we get
\[ K_{s,\gamma}(x, y) = \sum_{u \subseteq \{1, ..., s\}} \gamma_u \eta_u(x_u, y_u), \]
where
\[ \eta_u(x_u, y_u) = \prod_{j \in u} \eta(x_j, y_j). \]

**Existence of a good QMC rule**

Recall: the “worst-case-error” for a QMC rule in \( H_{s,\gamma} \) (with an obvious change of notation) is
\[ e_{N,s,\gamma}(t_1, \ldots, t_N) := \sup \left\{ \left| I_s f - \frac{1}{N} \sum_{k=1}^N f(t_k) \right| : \|f\|_{H_{s,\gamma}} \leq 1 \right\} \]

**Theorem** Sloan & Woźniakowski (1998)

If (and only if) \[ \sum_{j=1}^\infty \gamma_j < \infty, \] then there exist points \( t_1, \ldots, t_N \in [0,1]^s \) such that
\[ e_{N,s,\gamma}(t_1, \ldots, t_N) \leq \frac{D_\gamma}{\sqrt{N}}, \]
with \( D_\gamma \) independent of \( s \).
Remarks

1. The result holds e.g. for \( \gamma_j = 1/j^2 \). It does not hold for the classical weights \( \gamma_j = 1 \).

2. \( D_\gamma \) is known explicitly: for example, we can take

\[
D_\gamma = \exp \left( \frac{1}{12} \sum_{j=1}^{\infty} \gamma_j \right).
\]

3. The condition \( \sum_{j=1}^{\infty} \gamma_j < \infty \) is necessary as well as sufficient. There is a lower bound on the worst-case error that grows unboundedly with \( s \), for every choice of points, if the condition fails.

4. The existence proof is not constructive!

The 1998 paper used essentially our first RKHS, not the unanchored one used here.

Proof of existence theorem

Let \( M_{N,s,\gamma} \) be the RMS average of \( e_{N,s,\gamma} \) over all possible choices of \( t_1, \ldots, t_N \), that is

\[
M_{N,s,\gamma}^2 := \int_{[0,1]^s} \cdots \int_{[0,1]^s} e_{N,s,\gamma}(t_1, \ldots, t_N)^2 dt_1 \cdots dt_N
\]

\[
= \int_{[0,1]^s} \cdots \int_{[0,1]^s} \left[ \int \int K(x,y)dx\,dy \right]
\]

\[ -2\frac{1}{N} \sum_{k=1}^{N} \int K(x,t_k)dx + \frac{1}{N^2} \sum_{k=1}^{N} \sum_{k'=1}^{N} K(t_k,t_{k'})dt_1 \cdots dt_N \]

(Now split the sum into off-diagonal and diagonal parts)

Proof of existence theorem (continued)

\[
= - \int_{[0,1]^{2s}} K(x,y)dx\,dy + \frac{1}{N^2} N(N-1) \int_{[0,1]^{2s}} K(t,t')dt\,dt'
\]

\[ + \frac{1}{N^2} \int_{[0,1]^{s}} K(t,t)dt \]

(Now combine the first two terms)

\[
= \left( -1 + \frac{N(N-1)}{N^2} \right) \int_{[0,1]^{2s}} K(x,y)dx\,dy + \frac{N}{N^2} \int_{[0,1]^{s}} K(x,x)dx
\]

\[ = \frac{1}{N} \left( -\int_{[0,1]^{2s}} K(x,y)dx\,dy + \int_{[0,1]^{s}} K(x,x)dx \right)
\]

\[ = \frac{1}{N} \left( -1 + \prod_{j=1}^{s} \left( 1 + \gamma_j \int_{0}^{1} \eta(x,x)dx \right) \right) = \frac{1}{N} \left( -1 + \prod_{j=1}^{s} \left( 1 + \frac{\gamma_j}{6} \right) \right). \]
Proof of existence theorem (continued)

Thus

\[ M^2_{N,s,\gamma} = \frac{1}{N} \left( -1 + \prod_{j=1}^{s} \left( 1 + \frac{\gamma_j}{6} \right) \right) \]

\[ \leq \frac{1}{N} \exp \left( \frac{1}{6} \sum_{j=1}^{s} \gamma_j \right) \leq \frac{1}{N} \exp \left( \frac{1}{6} \sum_{j=1}^{\infty} \gamma_j \right) = \frac{D^2}{N}. \]

(Get the first inequality by dropping the \(-1\), then use

\[ 1 + x = \exp(\log(1 + x)) \leq \exp(x) \text{ for } x \geq 0. \]

The final step: observe that there is at least one choice of

\[ t_1, \ldots, t_N \] that is as good as average!

\[ \square \]

Sobol' sequences in a weighted space

**THEOREM** X. Wang (2002):

If

\[ \sum_{j=1}^{\infty} \gamma_j^{1/2} j \log j \log \log j < \infty, \quad \text{e.g. } \gamma_j = \frac{1}{j^5} \]

then the first \( N \) terms of the Sobol' sequence satisfy

\[ e_{s,\gamma}(Q_{N,s}^{\text{Sobol}}) \leq \frac{C_{\gamma,\delta}}{N^{1-\delta}} \quad \forall \ \delta > 0. \]

Wang in that paper also found a similar result for the Niederreiter sequence (without \( \log \log j \), but still with the \( j \log j \) factor).

A sparse grid construction

**THEOREM** Wasilkowski & Woźniakowski (1999)

If

\[ \sum_{j=1}^{\infty} \gamma_j^{1/3} < \infty \quad \text{e.g. } \gamma_j = \frac{1}{j^{1/4}} \]

then a certain sparse grid quadrature rule \( Q_{N,s}^{\text{sparse}} \) yields

\[ e_{s,\gamma}(Q_{N,s}^{\text{sparse}}) \leq \frac{C_{\gamma,\delta}}{N^{1-\delta}} \quad \forall \ \delta > 0. \]

N.B.: The sparse grid rule has unequal weights, so is not a QMC rule.
Existence of a good lattice rule

**Theorem** Sloan & Woźniakowski (2001):

If

$$\sum_{j=1}^{\infty} \gamma_j^{1/2} < \infty,$$

and if $N$ is prime, then \exists a shifted lattice rule $Q_{N,s,z,\Delta}$ such that

$$e_{s,\gamma}(Q_{N,s,z,\Delta}) \leq \frac{C_{\gamma,\delta}}{N^{1-\delta}} \quad \forall \delta > 0.$$  

Recall: a shifted lattice rule is a QMC rule of special form

$$Q_{N,s,z,\Delta}f = \frac{1}{N} \sum_{k=1}^{N} f \left( \left\{ \frac{kz}{N} + \Delta \right\} \right),$$

$z \in \{1, 2, \ldots, N-1\}^s$, $\Delta \in [0,1)^s$

Proof is by averaging in a different way over $\Delta$ and $z$, then using Jensen’s inequality. I skip the details.

---

Construction of good lattice rules

The original idea of component-by-component (or CBC) construction was to construct $z$ and $\Delta$ one component at a time:

Writing $z = (z_1, \ldots, z_s)$, $\Delta = (\Delta_1, \ldots, \Delta_s)$,

we determine successively $z_1, \Delta_1, z_2, \Delta_2, z_3, \Delta_3, \ldots$,

at each step minimizing some average of the worst-case error.

---

CBC for randomly shifted lattice rules

But it is better to work with randomly shifted lattice rules.

The first advantage: we don’t need to compute the components of $\Delta$!

The algorithm chooses $z_1, z_2, \ldots$ from $\{1, 2, \ldots, N-1\}$, one component at a time, for use in randomly shifted lattice rules.

At each stage we minimise the worst-case error averaged over shifts,

$$e_{\text{rms}}(H_K, Q_{N,s}^{\text{QMC}}) := \left( \int_{[0,1]^s} e(H_K, Q_{N,s}^{\text{QMC}})^2 d\Delta \right)^{1/2}.$$

For lattice rules this is fast, because (as we shall see) the average over shifts: can be computed with a cost of only $O(Ns)$.

---

The root mean square worst-case error

Hickernell and Woźniakowski (2000) showed that for a RKHS $H_K$ with kernel $K$, and a QMC rule

$$Q_{N,s}^{\text{QMC}} = \frac{1}{N} \sum_{k=1}^{N} f(t_k),$$

the squared root mean square error for the randomly shifted rule is

$$e_{\text{rms}}(H_K, Q_{N,s}^{\text{QMC}})^2 = - \int_{[0,1]^{2s}} K(x,y) dxdy + \frac{1}{N^2} \sum_{k=1}^{N} \sum_{k'=1}^{N} K^*(t_k, t_{k'}),$$

where $K^*$ is the "shift-invariant" kernel corresponding to $K$,

$$K^*(x, y) = \int_{[0,1]^{s}} K^*\left(\{x + \Delta\}, \{y + \Delta\}\right) d\Delta.$$

Proof: Exercise. (Use expression for $e(H_K, Q_{N,s}^{\text{QMC}})^2$.)
**Shift invariant kernel**

*K* is “shift invariant” because (this is easy)

\[ K^*(x, y) = K^*(\{x - y\}, 0) =: k_{shift-inv}(x - y). \]

**Example** For our second RKHS the shift-invariant kernel is

\[ k_{shift-inv}(x) = \prod_{j=1}^{s} \left(1 + \gamma_j B_2(x_j)\right), \]

thus the RMS worst-case error for a lattice rule \[ Q_{lattice}^{N,s,\gamma,z} \] is

\[ e_{rim}^{N,s,\gamma}(z)^2 = -1 + \frac{1}{N} \sum_{k=1}^{N} \prod_{j=1}^{s} \left(1 + \gamma_j B_2(\{kz_j/N\})\right), \]

\[ = -1 + \frac{1}{N} \sum_{k=1}^{N} \prod_{j=1}^{s} \left(1 + \gamma_j B_2(\{kz_j/N\})\right). \]

**CBC for random shifts (N prime)**

- \( z_1 = 1 \)
- For \( s = 2, 3, \ldots, s_{\text{max}} \), choose \( z_s \) from \( \{1, 2, \ldots, N - 1\} \) so as to minimise

\[ e_{rim}^{N,s,\gamma}(z_1, z_2, \ldots, z_s)^2 = -1 + \frac{1}{N} \sum_{k=1}^{N} \prod_{j=1}^{s} \left(1 + \gamma_j B_2(\{kz_j/N\})\right). \]

Here \( B_2(x) = x^2 - x + \frac{1}{6} \).

The total cost of a naive implementation is \( O(N^2 s_{\text{max}}^2) \).

**Error of CBC with random shifts**

**THEOREM** Sloan, Kuo, Joe (2002) Let \( N \) be prime, and let \( z_1, \ldots, z_s \) be chosen by the new algorithm. Assume

\[ \sum_{j=1}^{\infty} \gamma_j < \infty. \]

Then

\[ e_{rim}^{N,s,\gamma}(z) \leq \frac{D_{\gamma}}{N^{1/2}} \]

As before,

\[ D_{\gamma} = \exp\left(\frac{1}{12} \sum_{j=1}^{\infty} \gamma_j\right) \]

Proof: By induction, using an averaging argument at each step.

**Numerical example**

(Actually for the first RKHS.)

**EXAMPLE:** \( \gamma_j = \frac{1}{j^2}, j = 1, 2, \ldots \)

<table>
<thead>
<tr>
<th>( s )</th>
<th>( z_1 )</th>
<th>( e_{\text{rms}}^{N,s,\gamma}(z) )</th>
<th>( z_1 )</th>
<th>( e_{\text{rms}}^{N,s,\gamma}(z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = 2,003 )</td>
<td>( n = 32,003 )</td>
<td>( n = 2,003 )</td>
<td>( n = 32,003 )</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2.04(-4)</td>
<td>1</td>
<td>1.28(-5)</td>
</tr>
<tr>
<td>2</td>
<td>765</td>
<td>3.02(-4)</td>
<td>9376</td>
<td>2.03(-5)</td>
</tr>
<tr>
<td>3</td>
<td>605</td>
<td>3.72(-4)</td>
<td>11835</td>
<td>2.58(-5)</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>31</td>
<td>450</td>
<td>6.51(-4)</td>
<td>13604</td>
<td>5.51(-5)</td>
</tr>
<tr>
<td>32</td>
<td>241</td>
<td>6.53(-4)</td>
<td>3393</td>
<td>5.53(-5)</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>99</td>
<td>973</td>
<td>7.01(-4)</td>
<td>15017</td>
<td>6.07(-5)</td>
</tr>
<tr>
<td>100</td>
<td>304</td>
<td>7.02(-4)</td>
<td>10489</td>
<td>6.08(-5)</td>
</tr>
</tbody>
</table>

The apparent rate of convergence is better than \( O\left(\frac{1}{\sqrt{N}}\right) \)!
The CBC algorithm has optimal convergence!

**THEOREM** Frances Kuo (2003) Let $N$ be prime, and let $z_1, z_2, \ldots, z_s$ be chosen by the CBC algorithm for randomly shifted lattice rules.

Assume

$$\sum_{j=1}^{\infty} \frac{\gamma_j^{1/2}}{j} < \infty.$$ 

Then $\forall \delta > 0$

$$e_{\text{rms}}^{N,s,\gamma}(z) \leq \frac{C_{\gamma,\delta}^u}{N^{1-\delta}}.$$ 

Thus the optimal rate is achieved by the new algorithm!

The proof (not given here) is by a non-trivial averaging argument.

Faster construction

Nuyens & Cools (2006) have shown that the cost of the CBC construction for random shifts and $N$ prime can be reduced to $O(N \log N \, s_{\max})$.

The Nuyens and Cools implementation allows the CBC algorithm to be run with $s$ in thousands, $N$ in millions. How does it work? Recall

$$e_{\text{rms}}^{N,s,\gamma}(z_1, z_2, \ldots, z_s)^2 = -1 + \frac{1}{N} \sum_{k=1}^{N} \prod_{j=1}^{s} \left( 1 + \gamma_j B_2(\{kz_j/N\}) \right) \left( 1 + \gamma_s B_2(\{kz_s/N\}) \right)$$

The main cost is that of multiplying the matrix $(B)_{z,k} = B_2(\{kz/N\})$ by a vector. With a suitable permutation of rows and columns the matrix becomes a circulant matrix, permitting the use of FFT.

Choosing the weights?

For a particular application, how should the weights $\gamma_1, \gamma_2, \ldots$ be chosen? Actually we need not stick to “product weights”. So far we have taken

$$\gamma_u = \prod_{j \in u} \gamma_j$$

but there is also much interest in more general weights. In principle, for each $s$ we could choose a new weight for each of the $2^s$ subsets of $\{1, 2, \ldots, s\}$. But then we would have even more weights to choose! Often discussed are the “finite order” weights, for which $\gamma_u = 0$ for $|u|$ greater than some specified order.

The question of how best to choose the weights in particular applications is very much a topic of current research.
In practice

In practice it is more useful, following Cools, Kuo, Nuyens (2006), if the number of lattice points $N$ is not prime, but instead a power of 2, with the $2^m$ points for each $m$ containing as a subset the $2^{m-1}$ points from the previous level; and moreover if the new points to reach the $m$th level are added one at a time. In effect, the points (like the Sobol' points) are available as a very long sequence – the cited paper has available all values of $N$ from $2^{10}$ to $2^{20}$. The dimensionality is essentially unlimited, by a variant of the CBC algorithm.

For example, in a recent application (related to the valuation of a certain financial instrument) Kuo and Waterhouse used rules for dimension 9,125 and up to approximately 1 million points.

A general not-too-technical reference is Kuo and Sloan (2005).

Sparse grids

Introduction: Sparse grids were perhaps first introduced by Smolyak (1963), in the context of integration. Since then they have been rediscovered many times.

Developed for approximation and pde etc by many people, including Babenko, Zenger, Griebel, Temlyakov, ... Here we develop sparse grids for approximation rather than integration, because for approximation sparse grids are presently unrivalled.

Acknowledgement: Some images and content are taken, with permission, from http://www.math.tu-berlin.de/~garcke/, Jocken Garcke's web-based “Sparse grid tutorial".

Bungartz and Griebel (2004) is a good reference on sparse grid approximation.

The 1-dimensional case

Start with piecewise-linear interpolation on $[0, 1]$.

For $\ell \geq 0$, partition the interval into $2^\ell$ sub-intervals of equal length

$$h_\ell := \frac{1}{2^\ell}.$$

We shall refer to $\ell$ as the level.

For $i = 0, \ldots, \ell$, define

$$\phi_{\ell,i}(x) = \begin{cases} 1 - \left| \frac{x}{h_\ell} - i \right| : & x \in [(i - 1)h_\ell, (i + 1)h_\ell] \cap [0, 1], \\ 0 & \text{otherwise}, \end{cases}$$

= piecewise-linear "hat" function centred at $x_{\ell,i} := ih_\ell$ and of half-width $h_\ell$.

The space of 1-dim. piecewise-linear functions

For $\ell \geq 0$, define the piecewise-linear space at level $\ell$:

$$V_\ell := \text{the linear space of piecewise-linear functions on } [0, 1] \text{ with breakpoints on the grid } 0, h_\ell, 2h_\ell, \ldots, 1, \text{ with } h_\ell := \frac{1}{2^\ell}.$$

There are two different ways of defining a basis for $V_\ell$: the "nodal" basis and the "hierarchical" basis – see the next images for $\ell = 3$. 
Nodal basis in 1-dimension for level 3

Here \( h_3 = \frac{1}{2^3} = \frac{1}{8} \), and there are \( 2^3 + 1 = 9 \) nodal basis functions.

Hierarchical basis in 1-dim. up to level 3

The linear span of this set is the same as that of the nodal basis.

The nodal and hierarchical bases for \( V_3 \)

For the space \( V_3 \) (= piecewise linears with step-length \( \frac{1}{2^3} = \frac{1}{8} \)):

The nodal basis is \( \{ \phi_{3,0}, \phi_{3,1}, \ldots, \phi_{3,8} \} \).

The hierarchical basis (Faber 1909, Yserantant and others) is

\( \{ \phi_{0,0}, \phi_{0,1} \} \) (which spans \( W_0 \))

\( \cup \)

\( \{ \phi_{1,1} \} \) (which spans \( W_1 \))

\( \cup \)

\( \{ \phi_{2,1}, \phi_{2,3} \} \) (which spans \( W_2 \))

\( \cup \)

\( \{ \phi_{3,1}, \phi_{3,3}, \phi_{3,5}, \phi_{3,7} \} \) (which spans \( W_3 \)).

Convergence with respect to hierarchical basis

With the hierarchical basis, the size of the contributions becomes smaller as the level increases (for a smooth function \( f \)).
Decomposition of $V_L$ into difference spaces

$V_L = W_0 \bigoplus W_1 \bigoplus \ldots \bigoplus W_L,$

where $W_\ell$ is the “difference” space at level $\ell$:

$W_\ell = \begin{cases} \text{span}\{\phi_{\ell,0}, \phi_{\ell,1}\} & \text{for } \ell = 0 \\ \text{span}\{\phi_{\ell,1}, \phi_{\ell,3}, \ldots, \phi_{\ell,2^\ell-1}\} & \text{for } \ell \geq 1. \end{cases}$

$= \text{span}\{\phi_{\ell,i} : i \in b_\ell\},$

where $b_\ell := \begin{cases} \{0, 1\} & \text{for } \ell = 0, \\ \{1, 3, \ldots, 2^\ell - 1\} & \text{for } \ell \geq 1. \end{cases}$

Note that $|b_0| = 2$ for $\ell = 0$, $|b_\ell| = 2^\ell - 1$ for $\ell \geq 1$.

Thus $|b_\ell| = 2, 1, 2, 4, 8, \ldots$ for $\ell = 0, 1, 2, 3, 4, \ldots$.

Tensor product piecewise-linear spaces

For $\ell = (\ell_1, \ldots, \ell_s)$ with all $\ell_j \geq 0$, define the tensor product “difference” space

$W_\ell = W_{\ell_1} \otimes W_{\ell_2} \otimes \ldots \otimes W_{\ell_s}$

$= \text{span}\{\phi_{\ell,i}(x), \ i \in B_\ell\}$

where

$\phi_{\ell,i}(x) = \prod_{j=1}^s \phi_{\ell_j,i_j}(x_j),$

and

$B_\ell = \{i : i_j \in b_{\ell_j}, \ j = 1 \ldots s\}.$

The full tensor product piecewise-linear space

The full tensor product space for piecewise linear functions up to level $L$ is

$V_L = \bigoplus_{\ell_1=0}^L \bigoplus_{\ell_2=0}^L \ldots \bigoplus_{\ell_s=0}^L W_{(\ell_1,\ldots,\ell_s)}$

$= \bigoplus_{\|\ell\|_\infty \leq L} W_\ell.$

The following image has the basis for $V_3$, that is for all piecewise linear functions for $s = 2$ up to level 3.

The basis for $V_3$

The basis function centred at a point is represented by a dot at that point.
Now for sparse grids

Instead of including all \( \ell = (\ell_1, \ldots, \ell_s) \) with \( \ell_1 \leq L, \ldots, \ell_s \leq L \), now we include only a subset, those with \( \ell_1 + \ell_2 + \ldots + \ell_s \leq L \).

That is, our sparse grid subspace of \( V_L \) is

\[
V_L^{\text{sparse}} := \bigoplus_{\|\ell\|_1 \leq L} W_\ell,
\]

with \( \|\ell\|_1 = \ell_1 + \ldots + \ell_s \).

We shall see that the sparse grid is truly sparse.

The level-4 sparse grid in 2 dimensions

The level-4 sparse grid in 3 dimensions
The cost of a sparse grid calculation

The cost is (at least) that of some multiple of the number of function evaluations, that is, some multiple of the number of points in the sparse grid.

The curse of dimensionality is already apparent: merely to evaluate \( f \) at each vertex of the cube we need \( 2^s \) function evaluations, an exponentially large number for large \( s \).

In many applications (e.g. elliptic PDE with homogeneous Dirichlet boundary conditions) we should omit the basis functions \( \phi_{\ell,i} \) centred for all points on the boundary. This can be achieved by omitting all basis functions \( \phi_{\ell,i} \) with \( \ell_j = 0 \) for some \( j \).

The number of (interior) points

The number of interior points can be computed: at level \( L \) it is

\[
\sum_{\ell_1 + \cdots + \ell_s \leq L} 2^{\ell_j - 1} = \sum_{|\ell| \leq L} 2^{m - s} \sum_{|\ell| = m} 1
\]

(with the sums over \( \ell \) restricted to \( \ell_j \geq 1 \forall j \))

\[
= \ldots \quad \text{(see Bungartz and Griebel (2004) for the combinatorial argument)}
\]

\[
= O(2^L L^{s-1}) = O(n_L (\log(n_L))^{s-1}) = O(h_L^{-1} (\log(h_L^{-1}))^{s-1}),
\]

where \( n_L = 2^L, \ h_L = \frac{1}{n_L} = 2^{-L} \).

Compare with the full tensor product, which uses \( n_L^s = h_L^{-s} \) points!

The accuracy of sparse grid approximation

In the 1-dimensional case we know that piecewise-linear interpolation on a grid of size \( \frac{1}{n} \) of a function with two integrable derivatives will yield an error of order \( O\left(\frac{1}{n^2}\right) \).

In \( s \) dimensions the full tensor product grid will give the same order of convergence (but at enormous cost).

In a sparse grid space we can get almost the same result, but we have to work in mixed derivative spaces.

Mixed derivative spaces

Some notation: let

\[
D^\alpha f = \frac{\partial f[\alpha]}{\partial x_1^{\alpha_1} \cdots \partial x_s^{\alpha_s}},
\]

where \( |\alpha| = \alpha_1 + \cdots + \alpha_s \). The mixed second-derivative space is

\[
H^{2,\text{mix}} := \{ f \in L_2 : D^\alpha \in L_2 \forall \alpha \text{ with } \alpha_j \leq 2, j = 1, \ldots, s \},
\]

with the norm and seminorm

\[
\|f\|_{H^{2,\text{mix}}} = \left( \sum_{\alpha_1, \alpha_j \leq 2 \forall j} \|D^\alpha f\|_{L_2}^2 \right)^{1/2},
\]

\[
|f|_{H^{2,\text{mix}}} = \|D^{(2,2,\ldots,2)} f\|_{L_2}.
\]
Error for sparse grid approximation

Bungartz and Griebel (2004) show: if $\Lambda_L f$ is the sparse grid interpolant of $f$ at level $L$, then

$$\|f - \Lambda_L f\|_L^2 \leq C_s n_{L}^{-2} (\log(n_L))^{s-1} |f|_{H^2,\text{mix}}.$$  

$$= C_s h_L^2 (\log(h_L^{-1}))^{s-1} |f|_{H^2,\text{mix}}.$$  

Other aspects of sparse grid approximation

See Griebel and Bungartz (2004) for many other aspects of sparse grids, including:

- Measuring the error in the energy norm (there is no $(\log(h_L^{-1}))^{s-1}$ factor)
- Adaptive sparse grid
- Higher order piecewise-polynomial spaces
- Applications to pde

Sparse grid quadrature

For $\ell = 0, 1, \ldots, L$, let $T_\ell f$ be the trapezoidal rule on $[0, 1]$ with step length $h_\ell = 2^\ell$, and therefore with $2^\ell + 1$ equally spaced points.

Let $\Delta_\ell := T_\ell - T_{\ell-1}$, with $T_{-1} = 0$.

Then the corresponding sparse grid (Smolyak) quadrature approximation on $[0, 1]^s$ is:

$$\int_{[0,1]^s} f(x) \, dx \approx \sum_{\ell_1 + \ell_2 + \cdots + \ell_s \leq L} \Delta_{\ell_1} \otimes \Delta_{\ell_2} \otimes \cdots \otimes \Delta_{\ell_s} f(x_{\ell,1}).$$

where the grid points are exactly as before:
Comments about sparse grid quadrature

The result is equivalent to integrating exactly the sparse grid piecewise-linear interpolant (exercise), so the error is bounded by the error of piecewise linear interpolation, $C_s n^{-2(\log n)^s-1} |f|_{H^{2,\text{mix}}}$, where $n = 2^L$.

The convergence order (for fixed dimension $s$) can be improved easily, by increasing the order of the 1-dimensional rule.

But sparse grid quadrature has problems for high dimensions. (Remember the $(\log n)^{s-1}$ factor.) And it does not adapt in an easy way to the weighted space idea, that some directions are more important than others.

And finally, the sparse grid quadrature weights are not only not equal: they can even be negative!

A final word

Both QMC and sparse grid require $f$ to be in some mixed-derivative space. This is much more restrictive than might at first sight appear: consider

$$f(x) = f(x_1, x_2, \ldots, x_s) = |x_1 + x_2 + \ldots + x_s|, \; x \in [-1,1]^s.$$  

This absolute value function is in $H^{1,\text{mix}}$ for $s = 1$, not for any $s \geq 2$.

If we consider instead the characteristic function of $x_1 + \ldots + x_s$, then $f$ is not in $H^{1,\text{mix}}$ for any $s \geq 1$!

Motto: There’s no such thing as a free lunch.

Bibliography


