Approximating Integrals in High Dimensions — An Informal Report

Ian H Sloan

1. Introduction

For half a century or more many researchers have grappled with the problem of numerical integration in many dimensions — on the one hand to understand what is possible, on the other to construct and analyse effective algorithms. This is an informal and rather personal account of one part of that effort.

For simplicity, we consider only the problem of integration over the unit cube in s dimensions. Thus the task is to approximate the integral

$$I_s(f) = \int_{[0,1]^s} f(x)\,\mathrm{d}x, \qquad (1)$$

where s is an integer greater than 1, and f is a given continuous function (or a function from a given class), hopefully having also some additional smoothness property. The interesting case is when s is large, say in the hundreds or even thousands.

The simplest kind of s-dimensional integration rule is just a product of 1-dimensional rules, obtained by using a favourite quadrature rule (say an m-point Gauss rule) in each dimension. But when s is large no such product rule is feasible, since the total number of points for the s-dimensional rule is then $n = m^s$, an impossibly large number for large s: if s is 100 then even a product of 2-point Gauss rules would need 2^{100} function evaluations, a number unlikely ever to be feasible on a conventional computer.

2. Quasi-Monte Carlo Rules

In this short report we consider only n-point integration rules of the form

$$Q_{n,s}(f) = Q_{n,s}(t_0,\ldots,t_{n-1};f) = \frac{1}{n}\sum_{i=0}^{n-1} f(t_i), \qquad (2)$$

that use prescribed points $t_0,\ldots,t_{n-1} \in [0,1]^s$. A rule of this form is called a Quasi-Monte Carlo (or QMC) rule. In contrast, if the points $t_0,\ldots,t_{n-1} \in [0,1]^s$ are independent random samples from a uniform distribution on $[0,1]^s$ then we have the simplest Monte Carlo rule. Monte Carlo methods have many virtues, the most prominent being that smoothness of the integrand f is not needed: all that is needed is that f belong to $L_2([0,1]^s)$. But the Monte Carlo method suffers from slow convergence: the root mean square error in the sense of expectation has the celebrated convergence rate of $1/\sqrt{n}$. Our aim is to do better than that.

The central question for the QMC method is how to choose the points $t_0,\ldots,t_{n-1}$. That is a big subject, explored in a recent survey [1]. But here the view is personal and selective, so we can be brief.

There are now two main strategies for choosing QMC points, which we may call the *low discrepancy* and *lattice* approaches. Here the focus is on the second.

3. Lattice Rules

In the simplest form of lattice rule, the only kind considered here, the points are given by the simple formula

$$t_i = \left\{\frac{iz}{n}\right\}, \quad i = 0,1,\ldots,n-1, \qquad (3)$$

where $z \in \mathbb{Z}^s$, known as the *generating vector*, is an integer vector of length s having no factor in common with n, and the braces around a vector indicate that we take the fractional part of each component in the vector.

Lattice methods began life in the 1950s in the hands of number theorists, who obtained many beautiful results, especially for functions that are 1-periodic with respect to each component of x. For surveys of the classical results see [4] and [5]. Those results, based on Fourier analysis, are of limited practical use, because high-dimensional integrals arising from applications are usually not naturally periodic; and forcing periodicity through a change of variables has a tendency to turn easy high-dimensional problems into difficult ones. In the developments reported here we make no periodicity assumption.

The point set (3) depends entirely on the choice of the integer vector z. How should z be chosen? Clearly there is no sense in allowing a component

of z to lie outside the range $0, 1, \ldots, n-1$. Within these constraints, how are we to find a good choice for z? Up to the present time no closed formula for a good generating vector z (whatever "good" might mean) is known, beyond $s = 2$. And intuition seems of little use in high dimensions. To find a good z for such an s we need some mathematical structure.

4. Worst Case Error, and a Norm with Weights

We assume that our functions f all belong to some Hilbert space H, and then choose a generating vector z to minimise (or at least makes small) the *worst case error* in this function space. The worst case error in the space H of a QMC rule $Q_{n,s}(P;\cdot)$ using the point set $P = \{t_0, t_1, \cdots, t_{n-1}\}$ is the largest error for f in the unit ball of H, that is

$$e_{n,s}(P;H) := \sup_{\|f\|_H \le 1} |I_s(f) - Q_{n,s}(P;f)|.$$

So now we have again changed the problem, this time to one of choosing a good function space H. It turns out that a good choice is to take H to consist of s-variate absolutely continuous functions whose *mixed first derivatives are square-integrable*. More precisely, we take

$$\begin{aligned} H &= H_{s,\boldsymbol{\gamma}} \\ &:= \left\{ f \in L_2([0,1]^s) : \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}}{\partial x_{\mathfrak{u}}} f(x_{\mathfrak{u}}; 0) \right|^2 \mathrm{d}x_{\mathfrak{u}} < \infty \ \forall\, \mathfrak{u} \subseteq \{1:s\} \right\}, \end{aligned}$$

where $\{1:s\}$ is shorthand for $\{1, 2, \ldots, s\}$, while $x_{\mathfrak{u}}$ denotes the components x_j of x with $j \in \mathfrak{u}$, and $(x_{\mathfrak{u}}; 0)$ denotes x with all components x_j with $j \notin \mathfrak{u}$ set equal to 0, which we refer to as the *anchor*.

But how exactly should we define the norm in H? It turns out to be a fruitful idea to introduce positive numbers called *weights* into the norm: in the most general case we introduce one weight $\gamma_{\mathfrak{u}}$ for each $\mathfrak{u} \subseteq \{1:s\}$ (with $\gamma_\emptyset = 1$), taking the norm of $H_{s,\gamma}$ to be

$$\|f\|_{s,\boldsymbol{\gamma}} := \left[\sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^{-1} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}}{\partial x_{\mathfrak{u}}} f(x_{\mathfrak{u}}; 0) \right|^2 \mathrm{d}x_{\mathfrak{u}} \right]^{1/2}.$$

Note that for f to be in the unit ball of $H_{s,\boldsymbol{\gamma}}$, any term in the sum over $\mathfrak{u}$ with a small weight $\gamma_{\mathfrak{u}}$ must also have a small mixed first derivative $(\partial^{|\mathfrak{u}|}/\partial x_{\mathfrak{u}})f$. Thus the weight parameters describe the relative importance of different subsets of the variables $x_1, \ldots, x_s$: small weights correspond to unimportant subsets.

The space $H_{s,\boldsymbol{\gamma}}$ is of course a Hilbert space, with the obvious inner product, which we denote by $\langle \cdot, \cdot \rangle_{s,\boldsymbol{\gamma}}$. Less obviously, $H_{s,\boldsymbol{\gamma}}$ is a *reproducing kernel Hilbert space* with the relatively simple kernel

$$K_{s,\boldsymbol{\gamma}}(x,y) = \sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \min(x_j, y_j).$$

That is to say, $K_{s,\boldsymbol{\gamma}}(x,y)$ is a symmetric function on $[0,1]^s \times [0,1]^s$ with the property that $K_{s,\boldsymbol{\gamma}}(x,\cdot) \in H_{s,\boldsymbol{\gamma}}$, *and* the reproducing property (easily verified for $s = 1$)

$$f(x) = \langle K_{s,\boldsymbol{\gamma}}(x,\cdot), f \rangle_{s,\boldsymbol{\gamma}} \quad \forall x \in [0,1]^s, f \in H_{s,\boldsymbol{\gamma}}.$$

Why is the reproducing kernel property helpful? It is because in a reproducing kernel Hilbert space $H_s(K)$ with kernel K the worst case error is computable, via the formula

$$\begin{aligned} e^2_{n,s}(P; H_s(K)) &= \int_{[0,1]^s} \int_{[0,1]^s} K(x,y)\, \mathrm{d}x\, \mathrm{d}y \\ &\quad - \frac{2}{n} \sum_{i=0}^{n-1} \int_{[0,1]^s} K(t_i, y)\, \mathrm{d}y \\ &\quad + \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{k=0}^{n-1} K(t_i, t_k). \end{aligned}$$

This can easily be proved by simple Hilbert space arguments together with the reproducing property of the kernel.

But is the worst case error really computable for our function space $H_{s,\boldsymbol{\gamma}} = H(K_{s,\boldsymbol{\gamma}})$, given that the sum over $\mathfrak{u}$ in our expression for $K_{s,\gamma}$ contains 2^s terms? Clearly computations with general weights $\gamma_{\mathfrak{u}}$ are not feasible when s is large, but they become so for some special cases. The original weights introduced in [6] were of the *product* form

$$\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \tau_j,$$

in which case it is easily seen that $K_{s,\boldsymbol{\gamma}}$ can be expressed as the easily evaluated product

$$K_{s,\boldsymbol{\gamma}}(x,y) = \prod_{j=1}^{s} \left(1 + \tau_j \min(x_j, y_j)\right).$$

Many results take an especially simple form in the product case. In particular, the necessary and sufficient condition established in [6] for the worst case error to be bounded independently of s is just

$$\sum_{j=1}^{\infty} \tau_j < \infty.$$

This condition fails spectacularly in the classical case in which $\tau_j = 1$ for all j, but is satisfied, for example, if $\tau_j = 1/j^2$.

Nevertheless, it has to be admitted that product weights, in spite of their popularity, were introduced for mathematical convenience, rather than being guided by application. In this report general weights have been retained, because certain non-product weights of so-called *product and order dependent* or POD form, have arisen recently in applications, see [2]. The POD weights have the form

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{j\in\mathfrak{u}} \tau_j, \quad \mathfrak{u} \subseteq \{1:s\},$$

where $\Gamma_0 = 1, \Gamma_1, \cdots, \Gamma_s$ are given positive numbers. As with product weights, computations with POD weights turn out to be feasible (see below).

5. A Little Randomisation

It turns out to be useful, for reasons of both theory and practice, to modify slightly the QMC algorithm by introducing some randomisation: specifically, we replace the lattice rule given by (2) and (3) by the *randomly shifted lattice rule*

$$Q^{\text{ran}}_{n,s}(z;f) := \frac{1}{n}\sum_{i=0}^{n-1} f\left(\left\{\frac{iz}{n} + \boldsymbol{\Delta}\right\}\right),$$

where $\boldsymbol{\Delta}$ is a random vector of length s drawn from a uniform distribution on $[0,1]^s$. This has some flavour of the Monte Carlo method: indeed it reduces to the Monte Carlo method if $n = 1$. In practice the randomly shifted lattice rule is implemented, once z is chosen, by picking some fixed number (say 10 or 30) of random (or rather pseudo-random) shifts $\boldsymbol{\Delta}_k$, and averaging the cubature results for the separate shifts to obtain an estimate of the integral, while using the spread of the results to estimate the error, just as with the Monte Carlo method.

The worst case error is now replaced by the root mean square expected value of the error, known as the *shift-averaged worst case error*,

$$e^{\text{sh}}_{n,s}(z;H) := \left(\mathbb{E}\left[\sup_{\|f\|_H\le 1} \left|I_s(f) - Q^{\text{ran}}_{n,s}(z;f)\right|^2\right]\right)^{1/2}.$$

For the case of our particular Hilbert space $H_{s,\boldsymbol{\gamma}}$ it can be computed by the formula (see [1, Lemma 5.7])

$$e^{\text{sh}}_{n,s}(z;H_{s,\boldsymbol{\gamma}})$$

$$= \left(\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j\in\mathfrak{u}}\left[B_2\left(\left\{\frac{kz_j}{n}\right\}\right)+\frac{1}{3}\right] - \left(\frac{1}{3}\right)^{|\mathfrak{u}|}\right)\right)^{1/2},$$

where $B_2(x) = x^2 - x + 1/6$ is the Bernoulli polynomial of degree 2. For the case of product weights the expression reduces to the easily computable

$$e^{\text{sh}}_{n,s}(z;H_{s,\boldsymbol{\gamma}})$$

$$= \left(-\prod_{j=1}^{s}\left(1+\frac{\tau_j}{3}\right) + \frac{1}{n}\sum_{k=0}^{n-1}\prod_{j=1}^{s}\left(1+\tau_j\left[B_2\left(\left\{\frac{kz_j}{n}\right\}\right)+\frac{1}{3}\right]\right)\right)^{1/2}.$$

6. Good Lattice Rules Exist!

As explained below, we now know the following: For every n there exists $z \in \{0,1,\ldots,n-1\}^s$ such that, for all $\lambda \in (1/2, 1]$,

$$e^{\text{sh}}_{n,s}(z;H_{s,\boldsymbol{\gamma}})$$

$$\le \left(\frac{1}{\varphi(n)}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda}\left(\frac{2\zeta(2\lambda)}{(2\pi^2)^{\lambda}} + \left(\frac{1}{3}\right)^{\lambda}\right)^{|\mathfrak{u}|}\right)^{1/(2\lambda)}. \quad (4)$$

Here $\zeta(\cdot)$ is the Riemann zeta function, and $\varphi(n)$ is the Euler totient function, that is, it is the number of integers between 1 and $n-1$ that are relatively prime to n. If n is prime then $\phi(n) = n-1$, and in that case the order of convergence is $n^{-1/(2\lambda)}$. We would set $\lambda = 1/2$ if that were possible, because then the order of convergence would be n^{-1}, but that is not possible because $\zeta(2\lambda)$ diverges as 2λ approaches 1 from above. But the result shows (since λ can be chosen arbitrarily close to 1/2) that for every $f \in H_{s,\boldsymbol{\gamma}}$ the rate of convergence will ultimately be arbitrarily close to n^{-1}. Can good rates of convergence be obtained independently of dimension? The answer depends on the weights $\gamma_{\mathfrak{u}}$: the answer is yes if the weights $\gamma_{\mathfrak{u}}$ decay sufficiently quickly.

7. Good Lattice Rules are (sometimes) Constructible!

Lattice rules that achieve the bound (4) are in principle constructible — indeed, the proof of (4) (see [1, Theorem 5.9]) is by an inductive argument based on an explicit construction. That construction is the so-called *component by component* (or CBC) construction.

The idea of the CBC construction is that the components of the generating vector $z = (z_1, z_2, \ldots, z_s)$ are determined one after the other, starting with $z_1 = 1$, with the successive components z_d for $d = 2,\ldots,s$ being minimisers of a quantity related to the shift-averaged worst case error for the d-dimensional problem. For our particular choice of norm, the quantity to be minimised in the case of general weights $\boldsymbol{\gamma}_{\mathfrak{u}}$ is not exactly the shift-averaged worst case error, see [1, Theorem 5.9], but if we change slightly the

definition of the norm, to

$$\|f\|_{s,\boldsymbol{\gamma}} := \left[\sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{-1} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial x_{\mathfrak{u}}} f(x_{\mathfrak{u}}; x_{-\mathfrak{u}})\, \mathrm{d}x_{-\mathfrak{u}} \right|^2 \mathrm{d}x_{\mathfrak{u}} \right]^{1/2},$$

then the CBC algorithm precisely minimises the shift-averaged worstcase error. Here $x_{-\mathfrak{u}}$ denotes the components x_j of x with j in $\{1:s\}$ but *not* in $\mathfrak{u}$, so that now the components of x that do not appear in the mixed derivative are integrated out instead of being anchored at 0. The formula for the shift-averaged worst case error is the same as before, except that the $\frac{1}{3}$ terms are now replaced by zero. The bound achieved by the CBC construction (proved inductively in [1, Theorem 5.8]) is exactly (4), except that again the $\frac{1}{3}$ term is replaced by zero. Finally, fast CBC construction is possible for both product weights and POD weights, see respectively [1, Section 5.5] and [1, Section 5.6].

8. Conclusion

We have seen that lattice rules for high dimensional integration are at an interesting stage of development: for given dimensionality s and given weights $(\gamma_{\mathfrak{u}})$ we can in principle construct a generating vector z for which the order of convergence for f in the unit ball of $H_{s,\boldsymbol{\gamma}}$ is arbitrarily close to $O(n^{-1})$, with an implied constant that is known explicitly for a given convergence rate, and that is independent of the dimensionality s if the weights are small enough. The construction is feasible for both product weights and POD weights.

This informal report is incomplete in many ways. Not reported on are higher order QMC methods, which aim to achieve a rate of convergence faster than $O(n^{-1})$. Also not reported is infinite-dimensional integration, a topic now showing high levels of activity. We have not done justice to the important work on fast implementation of CBC, initiated by Dirk Nuyens and Ronald Cools. We have not discussed how the weights should be chosen for a given practical problem. We have not considered lattice rules that are extensible in number of points or dimension. We have not considered integration over unbounded regions. All these matters are considered in [1].

Finally, many people have contributed to the recent developments outlined here, too many to mention them all, but especially I want to acknowledge Henryk Woźniakowski, Stephen Joe, Frances Kuo and Josef Dick. I am grateful to these and all others for their contributions, and to the Australian Research Council for its sustained support.

References

[1] J. Dick, F. Y. Kuo and I. H. Sloan, Numerical integration in high dimensions — the quasi-Monte Carlo way, *Acta Numerica*, 13 (2013) 133–288.
[2] F. Y. Kuo, C. Schwab and I. H. Sloan, Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients, *SIAM J., Numerical Analysis*, 50 (2012) 3351–3374.
[3] F. Y. Kuo and I. H. Sloan, Lifting the curse of dimensionality, *Notices of the AMS* 52 (2005) 1320–1328.
[4] H. Niederreiter, Quasi-Monte Carlo methods and pseudo random numbers, *SIAM* (1992).
[5] I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration* (Oxford University Press, 1994).
[6] I. H. Sloan and H. Woźniakowski, When are quasi-Monte Carlo algorithms, efficient for high-dimensional integrals? *J. Complexity* 14 (1998) 1–33.

Ian Sloan
University of New South Wales, Australia
I.Sloan@unsw.edu.au

Ian Sloan completed physics and mathematics degrees at Melbourne University, a Master's degree in mathematical physics at Adelaide, and a PhD in theoretical atomic physics (under the supervision of HSW Massey) at the University of London, finishing in 1964. After a decade of research on few-body collision problems in nuclear physics, his main research interests shifted to computational mathematics.

He was Head of the School of Mathematics in University of New South Wales and member of the ARC's Research Grants Committee, and is a former President of the Australian Mathematical Society. His awards/recognitions include Fellow of the Australian Academy of Science, ANZIAM Medal, etc. In 2008 he was appointed an Officer of the Order of Australia (AO). He is currently Deputy Director of MASCOS, the ARC Centre of Excellence for Mathematics and Statistics of Complex Systems.