Lifting the Curse of Dimensionality

Frances Y. Kuo and Ian H. Sloan

Introduction

Richard Bellman [1] coined the phrase *the curse of dimensionality* to describe the extraordinarily rapid growth in the difficulty of problems as the number of variables (or the dimension) increases. A common experience is that the cost of an algorithm grows exponentially with dimension, making the cost prohibitive for moderate or large values of the dimension.

In this article we consider the problem of numerical integration over the d-dimensional unit cube $[0,1]^d$. If $d = 1$ and the integrand is sufficiently smooth, then the integral can be evaluated easily by, say, Simpson's rule, in which case the error of an n-point rule, with n odd, is of order $\mathcal{O}(n^{-4})$. When d is 2 or more, the most obvious strategy is to apply a rule such as Simpson's rule in each dimension, creating what is called a *product* rule. But now we meet the curse of dimensionality: the total number of points at which the integrand must be evaluated (which we may take as the cost) is $N = n^d$. And with what error? Even if the integrand is an innocuous function of only the first component, x_1, for example x_1^4, the resulting error for the product Simpson rule is clearly still of order $\mathcal{O}(n^{-4})$, since from the point of view of this integrand the integration rule is still the n-point Simpson's rule. The essential difficulty becomes apparent when that error is expressed in terms of the total number of points N: the error is now of order $\mathcal{O}(N^{-4/d})$. Put differently, perhaps we are willing to use 11 points in each coordinate direction (i.e. $n = 11$). But if the problem is 100-dimensional (i.e., $d = 100$), then the total number of function evaluations required will be 11^{100}, and the time until the end of the universe will not suffice for this calculation. Even if we take just 2 points in each direction, the computation for $d = 100$ is impossible. Of course one can improve the one-dimensional rule (for example, by using the Gauss rule), but the essential problem remains: any product rule is prohibitively costly when d is large, because the cost for a given level of accuracy increases exponentially in d.

Nevertheless, there are other ways of tackling a high-dimensional integral, as shown in dramatic fashion in 1995, when Traub and Paskov at Columbia University successfully (but without supporting theory) treated a mathematical finance problem from Wall Street as an integration problem over the 360-dimensional unit cube.

The aim of this article is to present in a nontechnical way one strand of the story of high-dimensional numerical integration (specifically, we tell the *lattice* side of the story) as it has developed over the past half century. Part of the story is concerned with the development of a theoretical setting within which the problem is *tractable*, that is to say, loosely, a setting in which a suitably defined measure of the error does not grow with the dimension d (or perhaps grows only polynomially in d). But the arguments used in the study of tractability were until

Frances Y. Kuo is Vice-chancellor's Postdoctoral Fellow in Mathematics at the University of New South Wales, Australia. Her email address is f.kuo@unsw.edu.au.

Ian H. Sloan is Scientia Professor of Mathematics at the University of New South Wales, Australia. His email address is i.sloan@unsw.edu.au.

recently not constructive. That is to say, we knew under certain conditions that good integration rules exist for large values of d, but we had no idea how to find them. Only in the last half decade has that situation turned around with the development of very fast constructions that make feasible the tailored construction of integration rules that are of known high quality and that allow d in the hundreds (or even thousands), together with any practical number of function evaluations.

Do problems with d in the hundreds really arise in practice? This introduction concludes with a qualitative description of the 360-dimensional problem referred to above, since many features of that problem are common to problems from mathematical finance.

That problem is concerned with the valuation of a parcel of mortgage-backed securities held by a bank. In brief, customers of the bank borrow money for up to thirty years. Each month every customer has the right to repay the loan, and of course repaying the loan early will reduce its value to the bank. In a simple model, the proportion of those who choose to repay will depend on the interest rate at that time: the higher the interest rate, the more likely they are to repay the loan. The interest rate is assumed to follow a (geometric) Brownian motion. The month-by-month changes in the interest rate are random variables, so the present value of the bundle of mortgages is a (suitably discounted) 360-dimensional expected value, because there are 360 possible repayment occasions. This integral over the 360-dimensional Euclidean space is then converted into an integral over the 360-dimensional unit cube by an appropriate variable transformation. Many other high-dimensional problems in finance (including options of all varieties) are also multidimensional expected values, with the dimensionality arising either from discretization in time, as here, or because there are multiple assets with different characteristics, or both.

What Strategies Are Possible?

When d is large, we can approximate an integral over the unit cube

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

by an integration rule of the form

$$Q_N f = \sum_{k=1}^{N} a_k f(\boldsymbol{x}_k); \tag{1}$$

that is, we sample f at N points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ in the unit cube and approximate If by a weighted average of these function values.

So how can we choose these sample points? We already explained that a product rule such as the one shown in Figure 1 is too costly: the 64 points in the unit square effectively collapse down to just 8 points in each coordinate direction.

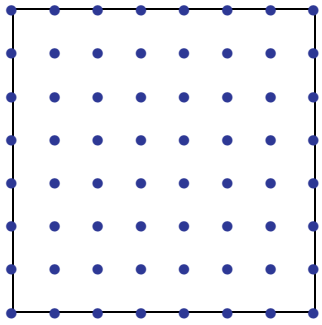


Figure 1. Product rule with 64 points.

One feasible strategy is to organize the points of a product rule in a hierarchical way and use only a few *levels* of points. This is the principle behind *sparse-grid methods*, which are generalizations of a construction first devised by Smolyak. Figure 2 shows an example of a regular sparse grid with 49 points.

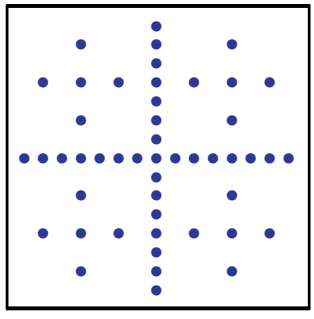


Figure 2. Sparse grid with 49 points.

Modern sparse-grid methods are dimension-adaptive: they find the important dimensions automatically and use more integration points in those dimensions. For details on sparse-grid methods, we refer readers to the recent survey article by Bungartz and Griebel [2].

Another possible strategy is the *Monte Carlo method*, the real workhorse of present-day high-dimensional integration. In its simplest form the Monte Carlo method is an equal weight rule (i.e. $a_k = 1/N$), with the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ generated *randomly* from a uniform distribution on $[0,1]^d$. Figure 3 shows 64 (pseudo) random points.

The well-known probabilistic error estimate for the Monte Carlo method is

$$\frac{\sigma(f)}{\sqrt{N}},$$

where $\sigma^2(f) = If^2 - (If)^2$ is the variance of f. Perhaps the most remarkable aspect of the Monte

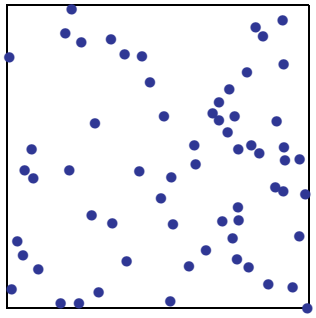


Figure 3. 64 random points.

Carlo method is that it does not suffer from the curse of dimensionality: in particular the $\mathcal{O}(N^{-1/2})$ convergence rate, while slow and erratic, does not depend on the dimension d so long as f is square integrable. Furthermore, it is cheap and easy to provide an effective error estimate, since the first term If^2 of the variance can be estimated by making use of the same function values as already used for approximating If.

Quasi-Monte Carlo methods are equal weight rules, just like the Monte Carlo method, except that the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are now designed in a clever way to be more uniformly distributed than random points so that a convergence rate close to $\mathcal{O}(N^{-1})$ is possible. (Note, however, that the implied constant can depend exponentially on d.) Figure 4 shows the first 64 points of a 2-dimensional *Sobol′ sequence*, the first example of the now widely renowned concept of *(t,m,s)-nets and (t,s)-sequences* established by Niederreiter.

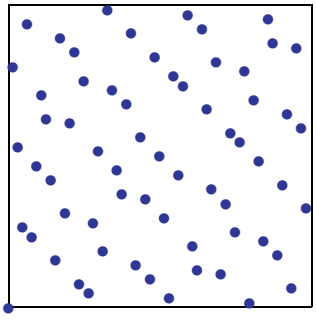


Figure 4. First 64 points of 2D Sobol′ sequence.

Informally, the basic idea is to have the right number of points in various subcubes. For example, if in Figure 4 we divide the unit square into strips of size 1 by $1/64$, then there is exactly one point in each of the 64 strips, with any point on the boundary counting toward the next strip. Similarly, if we divide the unit square into squares of size $1/8$ by $1/8$, we get exactly one point in each square. In fact, as long as we partition the unit square into 64 rectangles of the same shape and size, each rectangle will include exactly one point. Details on both theory and construction of nets and sequences can be found in the book of Niederreiter [3].

Lattice rules are a different kind of quasi-Monte Carlo method. The points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ of a lattice rule are so regular that they form a group under the operation of addition modulo the integers. Figure 5 shows a lattice rule with 55 points.

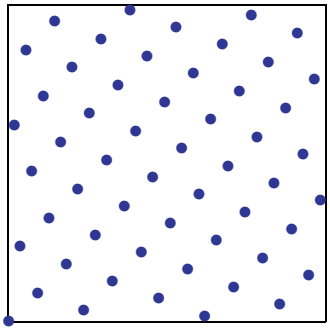


Figure 5. "Fibonacci" lattice rule with 55 points.

In this article we focus on lattice rules. Before continuing we should acknowledge a disadvantage of all deterministic methods, such as sparse-grid or quasi-Monte Carlo, when compared to the Monte Carlo method—namely, that they come without any practical information about the error. (The *a priori* estimates involving, for example, higher derivatives of f are essentially never useful for practical error estimation.) This has led to a growing interest in hybrid methods that are essentially deterministic but that also have some element of randomness, thereby seeking to capture the benefits of both approaches. We shall see an example of a hybrid method later in the article.

More on Lattice Rules

There are many kinds of lattice rules (indeed, even the product of left-rectangle rules is a lattice rule), but for our purposes it is enough to consider just the oldest and simplest kind, technically known now as a *rank*-1 *lattice rule*, which takes the form

$$Q_N f = \frac{1}{N} \sum_{k=1}^{N} f\left(\left\{k\frac{\boldsymbol{z}}{N}\right\}\right). \tag{2}$$

Here $\boldsymbol{z} \in \mathbb{Z}^d$ is the *generating vector*, and the braces indicate that each component is to be replaced by its fractional part in $[0,1)$. In this case the additive group formed by the points is the cyclic group generated by $\{\boldsymbol{z}/N\}$. Without loss of generality we may restrict the components of $\boldsymbol{z}$ to the set $\{0,1,\ldots,N-1\}$. Furthermore, we will drop 0 from this set, since if any component of $\boldsymbol{z}$ is zero, then so is the corresponding component of each integration point, which is clearly not interesting.

The lattice rule in Figure 5 is a *Fibonacci* lattice rule with $N = 55$ and $\boldsymbol{z} = (1, 34)$. (The Fibonacci lattice rule with $N = F_k$ points, where F_k is the kth number of the Fibonacci sequence $1, 1, 2, 3, 5, \ldots$, has generating vector $\boldsymbol{z} = (1, F_{k-1})$.) While the Fibonacci rules are in some sense optimal for $d = 2$, higher-dimensional generalizations, where they exist, are less persuasive.

Lattice rules were developed by number theorists, especially Hlawka and Korobov, during the late 1950s in the context of numerical integration of periodic functions. The periodicity of smooth integrands allows a convergence rate faster than $\mathcal{O}(N^{-1})$ for equal weight rules. For historical results on lattice rules applied to periodic integrands, readers are referred to the books of Niederreiter [3] and Sloan and Joe [5].

While the usefulness of lattice rules for numerical integration of periodic functions has been recognized for fifty years, their value for nonperiodic functions in high dimensions has been known only for the past half decade. That is a story to be told later in the article. The best possible convergence rate for any equal weight rule is then at best $\mathcal{O}(N^{-1})$, but that may seem like a very good rate indeed if the problem is 100-dimensional and if the implied constant is independent of d.

What makes a "good" lattice rule? This depends very much on the integrand at hand. A good lattice rule is one that is tailored to the integrand, taking into account its special features. Thus first we must consider the nature of the integrands we may encounter.

How to Model Reality—What Is Really the Dimension?

Consider a simple function

$$f(\boldsymbol{x}) = f(x_1, x_2, x_3, x_4) = x_1 \cos x_2 + x_3.$$

What is really the dimension? The nominal dimension is 4, yet clearly only the first three variables are present. Moreover, f is a sum of two terms, with each term depending on at most 2 variables. Thus we could say that the *effective dimension* of f is only 3 or even, from the point of view of each individual term, only 2.

More generally, it is possible to write any d-dimensional function f as a sum of 2^d terms, with each term depending on only a subset of variables:

$$f(\boldsymbol{x}) = \sum_{\mathfrak{u} \subseteq \{1,\ldots,d\}} f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}).$$

Here $\boldsymbol{x}_{\mathfrak{u}}$ denotes the $|\mathfrak{u}|$-dimensional vector containing those components of $\boldsymbol{x}$ whose indices belong to the set $\mathfrak{u}$. Each term $f_{\mathfrak{u}}$ depends only on variables in $\boldsymbol{x}_{\mathfrak{u}}$, and we may say that $f_{\mathfrak{u}}$ describes the *interaction* between these variables. The terms $f_{\mathfrak{u}}$ with $|\mathfrak{u}| = \ell$ are referred to collectively as the order-ℓ terms. Note that such a decomposition of f is by no means unique. The nicest way to ensure uniqueness is to insist that for each nonempty subset $\mathfrak{u}$ we have $\int_0^1 f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}})\, dx_j = 0$ for all $j \in \mathfrak{u}$. In this case the decomposition corresponds exactly to the *ANOVA* (for analysis of variance) *decomposition* used in the statistics literature. A special feature associated with the ANOVA decomposition is that the variance of the function f is simply the sum of the variances of the individual terms $f_{\mathfrak{u}}$.

For some functions it may happen that all the terms involving variables other than, say, $x_1, \ldots, x_{10}$ are negligible, or it may be that all variables are present and equally important, but the higher-order terms are negligible compared with the lower-order ones. In these situations we can ignore the effect of variables other than $x_1, \ldots, x_{10}$ or drop unimportant higher-order terms. In both cases such functions are said to have *low effective dimension*.

The concept of effective dimension was formally introduced by Caflisch, Morokoff, and Owen in 1997. By considering the ANOVA decomposition of f and the way that the variance is allocated among the ANOVA terms, they defined two concepts of effective dimension: f has *truncation dimension* d_t if the combined variance captured by the ANOVA terms $f_{\mathfrak{u}}$ with $\mathfrak{u} \subseteq \{1, \ldots, d_t\}$ exceeds 99 percent of the total variance $\sigma^2(f)$; on the other hand, f has *superposition dimension* d_s if the combined variance captured by the ANOVA terms $f_{\mathfrak{u}}$ with order $|\mathfrak{u}| \le d_s$ exceeds 99 percent of the total variance. For the example at the beginning of this section, the truncation dimension is 3, whereas the superposition dimension is 2.

Designing the Function Spaces and Introducing Weights

Many high-dimensional problems that occur in practical applications are of low effective dimension. To model these situations, Sloan and Woźniakowski introduced in 1998 the concept of *weighted* function spaces (see [7]). They assumed the variables are ordered so that successive variables become less and less important. To be more precise, they considered a function space that is formed by taking a *tensor-product* of one-dimensional Hilbert spaces. (Loosely speaking, functions in the tensor-product space are products, sums of products, and limits of sums of products of functions from the one-dimensional spaces.) The one-dimensional building blocks for the tensor-product space are Sobolev spaces consisting of absolutely continuous functions on $[0, 1]$ with square-integrable first derivatives. The norm in the jth one-dimensional space is parameterized by a *weight* γ_j which controls the variability of f with respect to the jth variable x_j. A small value of γ_j means that f depends only weakly on x_j.

By now there are many variants of these weighted spaces. A popular variant (and the only one we consider here) is the *unanchored* Sobolev space $\mathcal{F}$ which corresponds directly to the ANOVA decomposition. If f and g have the ANOVA decompositions $f = \sum_{\mathfrak{u}} f_{\mathfrak{u}}$ and $g = \sum_{\mathfrak{u}} g_{\mathfrak{u}}$ respectively, then their inner product in $\mathcal{F}$ is given by

$$\langle f, g \rangle_{\mathcal{F}} = \sum_{\mathfrak{u} \subseteq \{1,\ldots,d\}} \langle f_{\mathfrak{u}}, g_{\mathfrak{u}} \rangle_{\mathcal{F}_{\mathfrak{u}}},$$

where

$$\langle f_{\mathfrak{u}}, g_{\mathfrak{u}} \rangle_{\mathcal{F}_{\mathfrak{u}}} = \frac{1}{\prod_{j \in \mathfrak{u}} \gamma_j} \int_{[0,1]^{|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|} f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}})}{\partial \boldsymbol{x}_{\mathfrak{u}}} \frac{\partial^{|\mathfrak{u}|} g_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}})}{\partial \boldsymbol{x}_{\mathfrak{u}}} \, \mathrm{d}\boldsymbol{x}_{\mathfrak{u}}. \tag{3}$$

(Technically speaking, the space $\mathcal{F}$ is a direct sum of function spaces $\mathcal{F}_{\mathfrak{u}}$, with each $\mathcal{F}_{\mathfrak{u}}$ depending only on variables in $\boldsymbol{x}_{\mathfrak{u}}$ and every $f \in \mathcal{F}$ having the ANOVA decomposition $f = \sum_{\mathfrak{u}} f_{\mathfrak{u}}$, with $f_{\mathfrak{u}} \in \mathcal{F}_{\mathfrak{u}}$.)

We have described here the original idea for the weights, in which each variable x_j has associated with it a weight γ_j, and a subset of variables $\boldsymbol{x}_{\mathfrak{u}}$ is automatically assigned the weight $\prod_{j \in \mathfrak{u}} \gamma_j$. This is not always satisfactory, because the interactions between variables may not be modeled correctly. For example, for functions with low superposition dimension, it may well be that all variables are equally important but that only their lower-order interactions matter. To give more flexibility, the concept of weights has recently been generalized so that a weight $\gamma_{\mathfrak{u}}$ is associated with each subset of variables $\boldsymbol{x}_{\mathfrak{u}}$. Thus for a d-dimensional space, we have altogether 2^d weights instead of just d weights as before. The inner product for the unanchored Sobolev space with generalized weights can be obtained by replacing $\prod_{j \in \mathfrak{u}} \gamma_j$ in (3) by $\gamma_{\mathfrak{u}}$. Note that these generalized spaces are no longer tensor-product spaces. Following the generalization of weights, the traditional choice of weights, which leads to tensor-product spaces, is now referred to as the *product weights*. In some cases the interaction between variables in $\boldsymbol{x}_{\mathfrak{u}}$ depends only on the cardinality of $\mathfrak{u}$; this leads to *order-dependent weights*.

An important development is the modeling of functions with low superposition dimension by *finite-order weights* (i.e., there exists a fixed integer q such that $\gamma_{\mathfrak{u}} = 0$ for all $|\mathfrak{u}| > q$). Many practical problems do appear to be approximately of small order, for example, of order 2 or order 3.

To keep the story simple, throughout this article we shall consider only product weights.

The Magic of Reproducing Kernels

It makes sense to ask of an integration rule that it perform well not just for a single function f but rather for some family of functions. Our choice, if $\mathcal{F}$ is a Banach space, is to study the *worst-case error*: for a quasi-Monte Carlo rule Q_N and a particular space $\mathcal{F}$ with norm $\|\cdot\|_{\mathcal{F}}$, the worst-case error is defined to be the greatest error for any function f in the unit ball of $\mathcal{F}$,

$$e_N = e(Q_N, \mathcal{F}) = \sup_{\|f\|_{\mathcal{F}} \le 1} |\, Q_N f - I f \,|.$$

It is an annoying truth that the worst-case error is generally impossible to compute accurately or even to estimate other than loosely. This means that the worst-case error cannot generally be used to decide which is the better of two quasi-Monte Carlo rules. In one beautiful scenario, however, the situation is quite different, namely, when $\mathcal{F}$ is a *reproducing kernel Hilbert space* (RKHS). (In brief, the Hilbert space $\mathcal{F}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is an RKHS with kernel $K(\boldsymbol{x}, \boldsymbol{y})$ if $K(\cdot, \boldsymbol{y}) \in \mathcal{F}$ for all $\boldsymbol{y}$, $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}$ and all $\boldsymbol{y}$, and

$$\langle K(\cdot, \boldsymbol{y}), f \rangle_{\mathcal{F}} = f(\boldsymbol{y})$$

for all $f \in \mathcal{F}$ and all $\boldsymbol{y}$.) By the Riesz representation theorem, a Hilbert space is an RKHS if and only if point evaluations are bounded linear functionals in $\mathcal{F}$. The usefulness of reproducing kernel spaces in computational analysis and statistics was pointed out by Wahba fifteen years ago.

If $\mathcal{F}$ is an RKHS, then it is easy to show that the worst-case error for a quasi-Monte Carlo rule can be written explicitly in terms of the reproducing kernel. This is especially useful if the kernel is available as a simple analytic expression.

In particular, our unanchored Sobolev space is an RKHS, since point evaluations are bounded. Less obvious, perhaps, is that the kernel has the simple expression

$$K(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{d} \Big(1 + \gamma_j \Big[\tfrac{1}{2} B_2(|x_j - y_j|) + (x_j - \tfrac{1}{2})(y_j - \tfrac{1}{2}) \Big] \Big),$$

where $B_2(x) = x^2 - x + 1/6$ is the Bernoulli polynomial of degree 2. For this kernel, the worst-case error of a quasi-Monte Carlo rule with points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ is given by

$$e_N^2 = -1 + \frac{1}{N^2} \sum_{k=1}^{N} \sum_{\ell=1}^{N} K(\boldsymbol{x}_k, \boldsymbol{x}_\ell), \tag{4}$$

which can be computed in $\mathcal{O}(N^2 d)$ operations.

There Is Always One Choice as Good as Average

How is it possible to prove, even nonconstructively, that there exists a "good" choice of quasi-Monte Carlo points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ (whatever "good" may mean)? Here we use a beautiful argument borrowed from the number theorists: there is always one choice for which the worst-case error is as

good as the average over all possible choices. As a result, it is enough to show that some average of the worst-case error has the desired property.

The simplest average to compute is

$$\overline{e}_N = \left(\int_{[0,1]^d} \cdots \int_{[0,1]^d} e_N^2 \, \mathrm{d}\boldsymbol{x}_1 \cdots \mathrm{d}\boldsymbol{x}_N \right)^{1/2},$$

the root mean square average of the worst-case error e_N over all possible choices for each point $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ in a quasi-Monte Carlo rule. An explicit formula for $\overline{e}_N$ is easily obtained if $\mathcal{F}$ is a tensor-product RKHS. In our unanchored Sobolev space the result is

$$\begin{aligned} \overline{e}_N &= \frac{1}{\sqrt{N}} \left(\prod_{j=1}^{d} \left(1 + \frac{\gamma_j}{6} \right) - 1 \right)^{1/2} \\ &\le \frac{1}{\sqrt{N}} \exp\left(\frac{1}{12} \sum_{j=1}^{d} \gamma_j \right). \end{aligned} \tag{5}$$

So if we fix d, then for each increasing value of N there exists at least one choice of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ for which the Monte Carlo rate of convergence $\mathcal{O}(N^{-1/2})$ is achieved.

Actually, we shall see that faster convergence than $\mathcal{O}(N^{-1/2})$ is possible. To set a limit to our ambition, note that we could not hope to do better than $\mathcal{O}(N^{-1})$, since that is known to be the best possible order even for $d = 1$. Thus a reasonable aim would be to obtain (for fixed d) a convergence order close to $\mathcal{O}(N^{-1})$. We shall see that this aim can be achieved, but first we need to consider the behavior of the error with increasing d.

The Tractability of Integration Depends on the Weights

Earlier we introduced *weights* into our definitions of function spaces. We did this to help model certain problems that arise, for example, in mathematical finance, but we did this without explaining very convincingly why it might be necessary to bother about this level of detail. Now we come to the key point: that *without the weights the integration problem in our function space $\mathcal{F}$ is intractable*. That is to say, if $\gamma_j = 1$ for all j, then it is impossible to choose sequences Q_N of d-dimensional N-point quasi-Monte Carlo rules in such a way that

$$e_N \le c \frac{d^b}{N^a} \tag{6}$$

for positive constants a, b, c. The claim is that such a bound is impossible no matter how large we may choose c and b nor how close to zero we may choose a.

The intractability of the unweighted problem follows from an explicit lower bound that holds for all choices of Q_N. In particular, for our unanchored space it can be shown from (4) that

$$e_N^2 \ge \frac{1}{N} \left(\frac{13}{12} \right)^d - 1. \tag{7}$$

Thus if N and d are both allowed to go to infinity, then the error is bounded away from zero unless N grows *exponentially* with respect to d. For most cases it is known, too, that allowing general integration weights a_k in rule (1) does not help to break the curse of dimensionality. For a recent survey of tractability results for integration, see the review article by Novak and Woźniakowski [4]. For the foundations of tractability and intractability, we refer readers to the book *Information-Based Complexity* by Traub, Wasilkowski, and Woźniakowski [8].

So now we come to the key role of the weights γ_j: it is a remarkable fact that the integration problem is *strongly tractable* (i.e., (6) holds with $b = 0$) if and only if the weights satisfy

$$\sum_{j=1}^{\infty} \gamma_j < \infty. \tag{8}$$

If the condition is not satisfied, then one has a lower bound similar to (7) but with $(13/12)^d$ replaced by another sequence that goes to infinity as $d \to \infty$. On the other hand, if the condition (8) is satisfied, then from (5) we know already (but not constructively) that there is some choice of quasi-Monte Carlo rule Q_N for which

$$e_N \le \frac{1}{\sqrt{N}} \exp\left(\frac{1}{12} \sum_{j=1}^{\infty} \gamma_j \right),$$

giving an error bound that is independent of d and that converges to zero with the Monte Carlo rate $\mathcal{O}(N^{-1/2})$.

An even better result is known: by exploiting a connection between nonperiodic and periodic spaces, Hickernell and Woźniakowski showed in 2000 that if the weights satisfy the stronger condition

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

then there exists some quasi-Monte Carlo rule Q_N such that

$$e_N \le \frac{C_\delta}{N^{1-\delta}}, \quad \delta > 0,$$

where C_δ is independent of N and d. This rate of convergence $\mathcal{O}(N^{-1+\delta})$ is optimal in the sense of being arbitrarily close to $\mathcal{O}(N^{-1})$.

The result established by Hickernell and Woźniakowski is impressive but does not help us toward a construction. We shall say no more about it, because in fact a better result is now known—better not in the sense of order of convergence, but in the sense that the same $\mathcal{O}(N^{-1+\delta})$ can be achieved, and under the same condition, within the much more

limited class of *shifted lattice rules*. While the proofs are still nonconstructive, that topic, in the next section, will take us much closer to a construction, because the allowed set of rules is so much smaller.

Lattice Rules Can Beat the Average

Earlier we introduced the notion of a rank-1 lattice rule (2). Because we are dealing with functions that are not generally periodic, it turns out to be useful to introduce a small generalization: the *shifted rank-1 lattice rule* is defined by

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

where $\boldsymbol{z}$ is the generating vector as before and $\boldsymbol{\Delta} \in [0,1)^d$ is the *shift*.

In 2001 Sloan and Woźniakowski proved a result equivalent to the following: if $\sum_{j=1}^{\infty} \gamma_j < \infty$, then for the unanchored space and for each prime N there exist $\boldsymbol{z}$ and $\boldsymbol{\Delta}$ such that

$$e_N(\boldsymbol{z},\boldsymbol{\Delta}) \le \frac{1}{\sqrt{N}} \exp\left(\frac{1}{12}\sum_{j=1}^{\infty}\gamma_j\right).$$

That is, at least for prime N one can do as well as the average quasi-Monte Carlo method even within the small class of shifted lattice rules. Moreover, they showed that the improved result of Hickernell and Woźniakowski also holds: that is, if N is prime and $\sum_{j=1}^{\infty} \gamma_j^{1/2} < \infty$, then there exist $\boldsymbol{z}$ and $\boldsymbol{\Delta}$ such that

$$e_N(\boldsymbol{z},\boldsymbol{\Delta}) \le \frac{C_\delta}{N^{1-\delta}}, \quad \delta > 0, \tag{9}$$

with C_δ again independent of N and d.

Both results were obtained by variants of the previous averaging argument. To be precise, the averaging argument proceeds in two steps. The first step is to determine the root mean square average of $e_N(\boldsymbol{z},\boldsymbol{\Delta})$ over all $\boldsymbol{\Delta} \in [0,1)^d$, which we denote by $\overline{e}_N(\boldsymbol{z})$. For the $\mathcal{O}(N^{-1/2})$ result above, the next step is to determine the root mean square of $\overline{e}_N(\boldsymbol{z})$ over all $\boldsymbol{z} \in \{1,\ldots,N-1\}^d$. It turns out that both these steps can be done in a closed form for our unanchored Sobolev space. (The primality of N simplifies the argument in the second of these steps.) To obtain the improved rate of convergence, the second averaging step uses a different form of average over $\boldsymbol{z} \in \{1,\ldots,N-1\}^d$, namely,

$$\left(\frac{1}{(N-1)^d}\sum_{\boldsymbol{z}\in\{1,\ldots,N-1\}^d} \overline{e}_N(\boldsymbol{z})^{1/(1-\delta)}\right)^{1-\delta},$$

which after a considerable struggle can be bounded in the way indicated in (9).

To summarize the story to this point, we now know that in the worst-case setting and for suitably weighted tensor-product Hilbert spaces there exist quasi-Monte Carlo rules (and indeed even shifted lattice rules) that achieve an error bound that is independent of d and that (under suitable conditions) goes to zero with rate $\mathcal{O}(N^{-1+\delta})$ for arbitrary $\delta > 0$. But we do not yet know how to find any rules that achieve this result. That is the story of the next section.

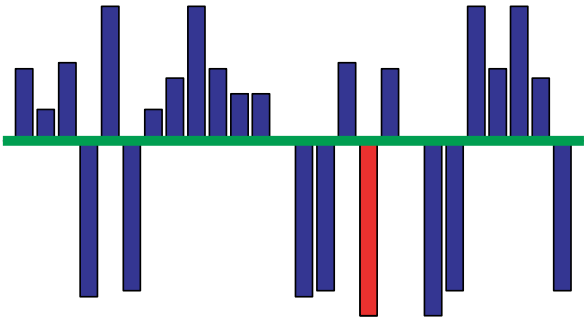
Finding a Good Lattice Rule One Component at a Time

Now the tale takes a surprising twist: what began as a theoretical study of tractability turned out, against expectation, to yield a method of construction.

Actually, all existence results of the kind reported here (asserting the existence of integration rules for which the worst-case error is smaller than some explicit upper bound) already contain within them the germ of a method of construction: one need only search among all the possible rules until one finds a rule for which the worst-case error achieves the desired bound (assuming, of course, that the worst-case error is explicitly computable). And the search can be carried out in complete confidence that it will be successful. The problem is that a full search of this kind is either impossible in principle (one cannot search over all values of a continuous variable) or else exponentially expensive (a search over all values of $\boldsymbol{z}$ in $\{1,\ldots,N-1\}^d$ requires $(N-1)^d$ evaluations of the worst-case error). Somehow the cost of the search must be reduced without sacrificing the quality of the result.

We have seen already that the search can be restricted (at least if N is prime) to shifted lattice rules. There are two ways of handling the problem caused by the continuous nature of the shift $\boldsymbol{\Delta} \in [0,1)^d$. One way is to limit the components of $\boldsymbol{\Delta}$ to odd multiples of $1/(2N)$, since it can be shown that this restriction can only reduce the root mean square average of the error. The approach we prefer is to leave $\boldsymbol{\Delta}$ continuous but to treat its components as continuous random variables uniformly and independently distributed on $[0,1)$. The first advantage of using a *random* shift $\boldsymbol{\Delta}$ is that we no longer need to find the shift! The second is that, just as in the Monte Carlo method, we can obtain a probabilistic error estimate if we repeat the calculation with several independently chosen random shifts (see below). The third advantage is that we can work with the root mean square worst-case error $\overline{e}_N(\boldsymbol{z})$ defined previously, which is much easier to compute than the worst-case error $e_N(\boldsymbol{z},\boldsymbol{\Delta})$, since it involves only a single sum: explicitly, it can be shown that

$$\overline{e}_N^2(\boldsymbol{z}) = -1 + \frac{1}{N}\sum_{k=1}^{N}\prod_{j=1}^{d}\left(1+\gamma_j B_2\left(\left\{k\frac{z_j}{N}\right\}\right)\right).$$



The worst-case error for one component of the generating vector. The green line represents the average worst-case error over all choices of this component. Clearly "there is always one choice as good as average" and indeed there are many such choices. The component-by-component algorithm takes the best choice, which is typically very much better than average.

Now we need to find a good choice of $\boldsymbol{z} \in \{1, \ldots, N-1\}^d$. The question we face is how to shorten the search over the values of $\boldsymbol{z}$ yet still find a $\boldsymbol{z}$ for which the theoretical error estimate

$$\overline{e}_N(\boldsymbol{z}) \le \frac{C_\delta}{N^{1-\delta}}, \quad \delta > 0, \tag{10}$$

is achieved, assuming $\sum_{j=1}^{\infty} \gamma_j^{1/2} < \infty$.

It turns out that the following component-by-component algorithm, introduced by Sloan, Kuo, and Joe in 2002 (see [6]), achieves the desired result. In this algorithm the components of $\boldsymbol{z} = (z_1, \ldots, z_d)$ are determined successively as follows:

1. Set $z_1 = 1$.
2. For j from 2 to d, with $z_1, \ldots, z_{j-1}$ fixed, choose z_j from $\{1, \ldots, n-1\}$ to minimize $\overline{e}_N(z_1, \ldots, z_j)$.

That such a simple "greedy" algorithm can succeed is in a certain sense surprising, since in the classical literature of lattice rules it is well accepted that a good lattice rule in d dimensions does not extend to a good lattice rule in $d+1$ dimensions.

The proof that the simple algorithm achieves the desired bound (10) is, naturally enough, by induction. And as the reader will by now expect, the inductive step itself relies on an averaging argument: with the components $z_1, \ldots, z_{j-1}$ held fixed and assumed to yield the desired bound with d replaced by $j-1$, one shows that an appropriate average over all possible choices of the next component z_j satisfies the analogous bound with d replaced by j. In the original paper of Sloan, Kuo, and Joe, only a bound of order $\mathcal{O}(N^{-1/2})$ was established in this way. That the same algorithm yields the essentially optimal order of convergence in (10) was established by Kuo in 2003 by a further refinement of the averaging argument.

In the practical implementation of the rule, once we have a generating vector $\boldsymbol{z}$, we generate a number of independent random shifts $\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_m$ and form the approximations $Q_1, \ldots, Q_m$, where Q_i is the shifted lattice rule approximation to the integral If based on the generating vector $\boldsymbol{z}$ and the shift $\boldsymbol{\Delta}_i$. Then the mean

$$\overline{Q} = \frac{1}{m}(Q_1 + \cdots + Q_m)$$

is our final approximation to If. An unbiased estimate for the error of our approximation is given by

$$\sqrt{\frac{1}{m} \cdot \frac{1}{m-1} \sum_{i=1}^{m} (Q_i - \overline{Q})^2}.$$

Fast Construction of a Good Lattice Rule

The obvious implementation of the component-by-component algorithm of the last section requires $\mathcal{O}(N^2 d)$ operations to compute all components of $\boldsymbol{z} = (z_1, \ldots, z_d)$. Since this is only polynomial in d, it certainly overcomes the cost aspect of the curse of dimensionality, but it is still too expensive when N is large. In 2005 Nuyens and Cools developed a revolutionary implementation of the algorithm that reduces the computational cost to $\mathcal{O}(N(\log N)\, d)$ operations. This allows fast construction of good lattice rules with N up in the millions and brings our lattice story to a satisfying conclusion.

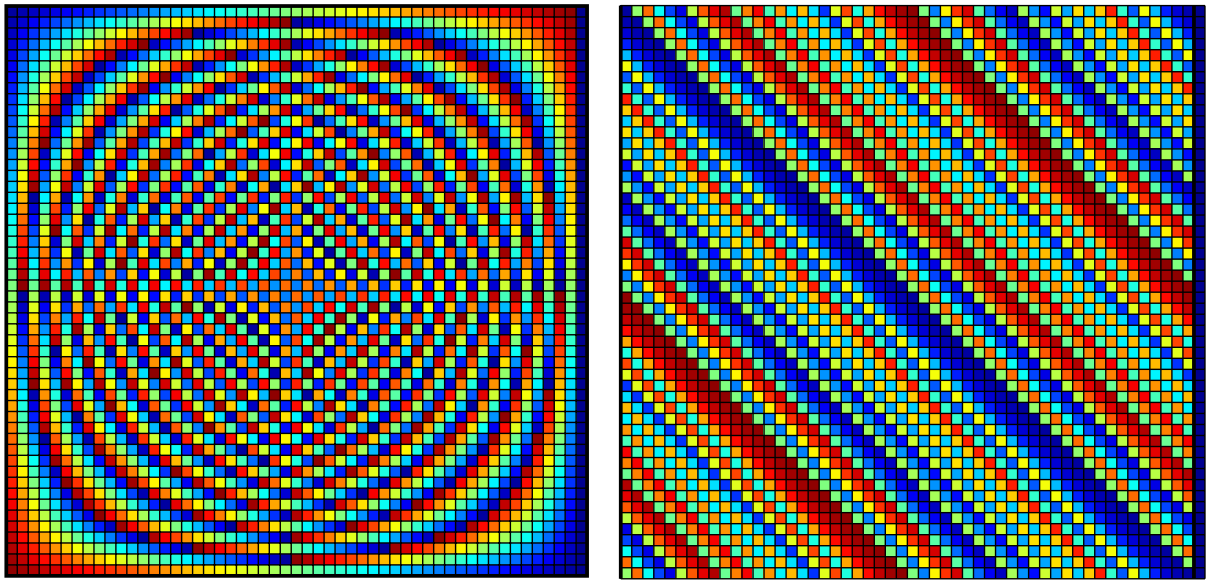
The main idea is as follows. In the jth step of the component-by-component algorithm, we need to evaluate the worst-case error with z_j taking the values $1, 2, \ldots, N-1$. This corresponds to some matrix-vector product, with the core of the matrix being (with N prime)

$$[kz \bmod N]_{\substack{1 \le z \le N-1 \\ 1 \le k \le N}}.$$

Leaving out the $k = N$ column, we can permute the rows and the columns of this matrix in such a way that we get a *circulant* matrix of order $N-1$. While a matrix-vector product in general requires $\mathcal{O}(N^2)$ operations, for a circulant matrix it can be achieved in $\mathcal{O}(N(\log N))$ operations by making use of the Fast Fourier Transform. Therein lies the secret behind the fast construction.

An Unfinished Story

The story, as told above, finishes with the development of a fast algorithm, one that yields integration rules of guaranteed quality in a specific setting (i.e., worst-case error in a specified weighted space). But many challenges remain before the resulting rules can be used sensibly for practical problems. The principal difficulty is in knowing how to choose appropriate weights for a particular



The structured matrix arising from the component-by-component algorithm and its permuted circulant form when N is prime. In its natural ordering the matrix exhibits a flower-like pattern or a wave interference effect as shown on the left. Using number theoretic permutations, this matrix can be transformed into a circulant form as shown on the right. The circulant structure means that the matrix-vector product can be computed in a fast way using the Fast Fourier Transform. When N is not prime, the same pattern shows up in multitude; see the cover and its corresponding description for N a power of 2.

problem or class of problems. That is the subject of much current research. Related problems are how best to transform a given problem to the unit cube and whether or not to use the variance reduction techniques that are familiar to the Monte Carlo community. (Practical experience suggests that variance reduction procedures are as useful as they are in the Monte Carlo method.) Much interest now lies in finding integration rules that are of acceptable quality for a range of weights. Perfection in the choice of weights is of course hard to achieve and fortunately not needed in practice, especially if we remember that the use of random shifts allows estimation of the error. (However, it is now known that the classical weights, as used implicitly for example in the book of Sloan and Joe, are much too large to yield good results.)

Many extensions have been considered by other researchers. The results have been extended to non-Hilbert space setting, to general weights, to integration over unbounded regions, and to the problem of approximation. The assumption that N is prime has been removed at the cost of complicating the analysis and slightly weakening the results.

Perhaps the most interesting extension has been the recent adaptation by Dick and Pillichshammer of the entire framework above, from the definition of weighted spaces through averaging arguments to component-by-component construction, to yield not a shifted lattice rule but rather a *digitally shifted digital net*, which is a particular kind of (t, m, s)-net. The story continues!

Acknowledgment

The support of the Australian Research Council under its Centres of Excellence Program is gratefully acknowledged.

References

[1] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[2] H.-J. Bungartz and M. Griebel, Sparse grids, *Acta Numerica* **13** (2004), 147–269.

[3] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, PA, 1992.

[4] E. Novak and H. Woźniakowski, When are integration and discrepancy tractable?, *Foundations of Computational Mathematics* (R. A. DeVore, A. Iserles, and E. Süli, eds), Cambridge University Press, Cambridge, 2001, pp. 211–266.

[5] I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration*, Oxford University Press, Oxford, 1994.

[6] I. H. Sloan, F. Y. Kuo, and S. Joe, Constructing randomly shifted lattice rules in weighted Sobolev spaces, *SIAM J. Numer. Anal.* **40** (2002), 1650–1665.

[7] I. H. Sloan and H. Woźniakowski, When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?, *J. Complexity* **14** (1998), 1–33.

[8] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, *Information-Based Complexity*, Academic Press, New York, NY, 1988.