Evaluating Expectations of Functionals of Brownian Motions: a Multilevel Idea

Ben Niu[1], Fred J. Hickernell[1], Thomas Müller-Gronbach[2] and Klaus Ritter[3]

[1] Department of Applied Mathematics, Illinois Institute of Technology
Chicago, IL 60616, USA
{nben,hickernell}@iit.edu
[2] Fakultät für Informatik und Mathematik , Universität Passau
94030, Passau, Germany
thomas.mueller-gronbach@uni-passau.de
[3] Fachbereich Mathematik, Technische Universität Darmstadt
Schloßgartenstr. 7, 64289, Darmstadt , Germany
ritter@mathematik.tu-darmstadt.de

Abstract. Pricing a path-dependent financial derivative, such as an Asian option, requires the computation of $E[g(B(\cdot))]$, the expectation of a payoff functional, g, that depends on a Brownian motion, $(B(t))_{t=0}^{T}$. The expectation corresponds to an infinite dimensional integral, which is approximated by the sample average of a d-dimensional approximation to the integrand. In this article, a multilevel algorithm with low discrepancy designs is used to improve the convergence rate of the worst case error with respect to a single level algorithm. The worst case error is derived as a function of each level l's sample size, n_l, and truncated dimension, d_l, for payoff functionals that arise from certain Hilbert spaces with moderate smoothness. If the error in approximating an infinite dimensional expectation by a d-dimensional integral is $\mathcal{O}(d^{-q})$, and the error for approximating a d-dimensional integral by an n-term sample average is $\mathcal{O}(n^{-p})$, independent of d, then it is shown that the error in computing the infinite dimensional expectation may be as small as $N^{-\min(p,q/s)}$ for a well-chosen multilevel algorithm, where N, the cost of the algorithm is defined as $N = n_1 d_1^s + \cdots + n_L d_L^s$ for some $s \geq 0$. This optimal convergence rate is achieved for either small or large q for rank-1 lattice rule designs, or alternatively for Niederretier net designs for large q.

Keywords. Hilbert spaces, Infinite dimension, Karhunen-Loève expansion, Low discrepancy points, Worst case error

1 Introduction

This article is an extension of [1,2], which consider the Monte Carlo simulation of stochastic integrals when the cost of function evaluation is dimension dependent. It is related to [3], which considers the error in the randomized setting, and has similarities to [4]. The problem is to approximate the expectation

$$\mu = \mu(f) = \mathbb{E}[f(X_1, X_2, \ldots)], \tag{1}$$

Dagstuhl Seminar Proceedings 09391
Algorithms and Complexity for Continuous Problems
http://drops.dagstuhl.de/opus/volltexte/2009/2298

where $(X_1, X_2, \ldots)$ is an i.i.d. sequence of random variables with common probability density function $\rho_1(\cdot)$, and f is some functional. This expectation may be thought of as an infinite dimensional integral. Note that the dependence of μ on f in (1) is sometimes written explicitly, and sometimes implied.

The aim is to efficiently estimate this expectation by an algorithm based on functional values, $f(x_{i,1}, \ldots, x_{i,d}, c, c, \ldots)$, where c is some nominal value of the X_j. For example, a simple sample average takes the form

$$\hat{\mu}_d = \hat{\mu}_d(f) = \frac{1}{n} \sum_{i=1}^{n} f\left(x_{i,1}, \ldots, x_{i,d}, c, c, \ldots\right), \tag{2}$$

for some dimension, d, and some design $\left\{\mathbf{x}_{i,1:d} = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}) \in \mathbb{R}^d\right\}_{i=1}^{n}$. The error, $\mu(f) - \hat{\mu}_d(f)$, depends on the choice of d, the design, and attributes of the functional, f. The cost of a single functional evaluation is assumed to be proportional to d^s for some $s \geq 0$. Examples where $s = 1$ is reasonable are given later in this section. The case of $s = 0$ is typically unrealistic in practical applications. Thus, the cost of algorithm $\hat{\mu}_d$ is proportional to $N = nd^s$, which corresponds to the fixed subspace sampling model of [1]. An efficient algorithm is one for which the error is $\preceq N^{-\tau}$ for τ as large as possible.

In [2], it was demonstrated that for functionals, f, lying in certain Hilbert spaces, $\mathcal{H}(K)$, defined by kernels, K, the squared worst case error of $\hat{\mu}$ can be written as the squared worst case error of approximating μ by a finite dimensional integral plus the squared worst case error of the finite dimensional cubature, i.e.,

$$\begin{aligned} &\sup_{\|f\|_{\mathcal{H}(K)} \leq 1} \left|\mu(f) - \hat{\mu}_d(f)\right|^2 \\ &= \sup_{\|f\|_{\mathcal{H}(K)} \leq 1} \left|\mu(f) - \mu_d(f)\right|^2 + \sup_{\|f\|_{\mathcal{H}(K)} \leq 1} \left|\mu_d(f) - \hat{\mu}_d(f)\right|^2. \end{aligned} \tag{3}$$

Here μ_d is defined as the conditional expectation where all coordinates numbered greater than d are set to the nominal value, c:

$$\mu_d = \mu_d(f) = \mathbb{E}[f(X_1, X_2, \ldots) | X_{d+1} = X_{d+2} = \cdots = c]. \tag{4}$$

The worst case error of approximating μ by μ_d is independent of the algorithm and might be called the bias. It is expected to vanish like $\preceq d^{-q}$ as $d \to \infty$ for some $q > 0$. The worst case of $\mu_d - \hat{\mu}_d$ corresponds to the *discrepancy* [5,6,7] and is known from strong tractability results, e.g., [8,9,10,11] to be $\preceq n^{-p}$, where the leading constant is independent of d, and p depends on the smoothness of the functional and the quality of the design. The error decomposition in (3) and the convergence rates of the two pieces on the right hand side can then be used to show that the worst case error for algorithm (2) is $\preceq N^{\frac{-pq}{ps+q}}$ [2]. Only when $s = 0$, typically not realistic, or $q \to \infty$ does one obtain an error $\preceq N^{-p}$, which is the ideal.

It is shown in Theorem 4 below that by using a multilevel algorithm one can obtain a better convergence rate of $\preceq N^{-p\min(1,\frac{q}{ps+q'})}$, where $s \geq 0$ and $0 \leq q' \leq q$, The constant q' affects the value of p, but it may be chosen, subject to the given constraints, to obtain as large a convergence order as possible. In Section 6 it is shown that for some cases one may obtain an ideal convergence rate of $\preceq N^{-\min(p,q/s)}$.

The multilevel algorithm is based on an increasing sequence of truncated dimensions, $d_1 < \cdots < d_L$. The infinite dimensional expectation is written as a telescoping sum:

$$\mu = \mu_{d_1} + \sum_{l=2}^{L} (\mu_{d_l} - \mu_{d_{l-1}}) + (\mu - \mu_{d_L}), \tag{5}$$

where the μ_{d_l} are defined in (4). Each term in this sum is approximated by an algorithm similar to $\hat{\mu}_d$ above in (2) with n_l terms. The precise form of the multilevel algorithm is given below in (14), and its cost is $N = n_1 d_1^s + \cdots + n_L d_L^s$ with $s \geq 0$. This multilevel algorithm is similar to that found in [1,3]. Multilevel algorithms can also be found in [12,13], which consider the quadrature problems of diffusion processes. There, the levels refer to the fineness of the steps in time, whereas here, the levels refer to various truncated dimensions d_l.

This article is organized as follows. This section concludes by describing an application that motivates the problem of computing the expectation of a functional of $X_1, X_2, \ldots$. Section 2 defines the Hilbert space where the functional f resides, and appropriate assumptions are made to facilitate the worst case error analysis later. Key results on the single level algorithm from [2] are reviewed in Section 3. The multilevel algorithm and its worst case error analysis are explained in Sections 4 and 5. Section 6 illustrates how to optimally choose the number of samples and truncated dimensions at each level l to achieve the minimal worst case error.

An important example of computing the functional of a countable number of random variables is the option pricing problem in mathematical finance. The price of an option is the expected value of the payoff, which in turn depends on the path of an asset price over an interval in time, $S(t)$, $0 \leq t \leq T$. The asset price $S(t)$ is often modeled by a stochastic differential equation (SDE) with general drift corresponding to the interest rate and a term involving a Brownian motion, $B(t)$, that describes the random fluctuations, e.g.,

$$dS(t) = r dt + \sigma(S,t) dB(t), \quad 0 < t < T, \quad S(0) \text{ given.}$$

Thus, one may express the option price as $\mu = \mathbb{E}[\text{payoff}(S(\cdot))] = \mathbb{E}[g(B(\cdot))]$, where g is the payoff functional expressed in terms of the Brownian motion driving the asset price via an SDE. For example, the discounted payoff functional of an arithmetic Asian call option is defined as:

$$g(B(\cdot)) = \text{payoff}(S(\cdot)) = e^{-rT} \max\left(\frac{1}{T}\int_0^T S(t)\,\mathrm{d}t - K, 0\right).$$

It remains to express the Brownian motion, $B(\cdot)$, in terms of a sequence of standard Gaussian random variables, $(X_1, X_2, \ldots)$, and so obtain

$$\text{payoff}(S(\cdot)) = g(B(\cdot)) = f(X_1, X_2, \ldots).$$

This then sets option pricing in the framework of computing $\mu = \mathbb{E}[f(X_1, X_2, \ldots)]$.

The *time differencing* method defines a sequence of times, t_k with $0 = t_0 < t_1 < \cdots < t_d = T$, and then sets

$$B(0) = 0, \quad B(t_k; X_1, \ldots, X_k) = B(t_{k-1}) + \sqrt{t_k - t_{k-1}} X_k, \quad k = 1, \ldots, d.$$

However, this method only defines the Brownian motion at a fixed number of different times, and does not easily allow for refinement of the time mesh to a larger number of times. Thus, the time differencing method is not suited to the problem described here.

Another approach to generate a Brownian motion is the *Brownian bridge*, which extends the time differencing approach by allowing continued refinements of the time mesh. Set the initial time to be $t_0 = 0$. For any $k \in \mathbb{N}_0$, let $t_{k+1} = (1 - \phi_2(k))\, T$, where ϕ_2 is the radical inverse function in base 2, which maps each k to a point in $[0, 1)$ by reflecting digits of the binary expansion for k about the decimal points. Specifically, if $k = k_0 + 2k_1 + \cdots + 2^{m_k - 1} k_{m_k - 1} + 2^{m_k}$ for $k_j \in \{0, 1\}$, and so $m_k = \lfloor \log_2(k) \rfloor$, then

$$\phi_2(k) = \frac{k_0}{2} + \frac{k_1}{2^2} + \cdots + \frac{k_{m_k - 1}}{2^{m_k}} + \frac{1}{2^{m_k + 1}}.$$

The Brownian Bridge definition is

$$\begin{aligned} B(t_0) &= B(0) = 0, \\ B(t_1; X_1) &= B(T) = \sqrt{T} X_1, \\ B(t_{k+1}; X_1, \ldots, X_{k+1}) &= \frac{1}{2}\left[B\left(t_{k+1} - \frac{T}{2^{m_k+1}} \right) + B\left(t_{k+1} + \frac{T}{2^{m_k+1}} \right) \right] \\ &\qquad + \sqrt{\frac{T}{2^{m_k+2}}} X_{k+1}, \quad k = 1, 2, \ldots, d, \end{aligned}$$

where $X_1, X_2, \ldots$ are standard normal random variables. Thus, one may think of the payoff functional for the option, $g(B(\cdot))$, as a functional of a countable number of standard Gaussian random variables, $f(X_1, X_2, \ldots)$. Truncating the Brownian bridge at d steps corresponds to setting the variables $X_{d+1}, X_{d+2}, \ldots$ to their nominal value $c = 0$, and then $B(t)$ becomes simply the linear interpolant between the two neighboring time mesh points t_{k_1} and t_{k_2}, where $0 \le k_1, k_2 \le d$. The cost of a d-step Brownian bridge is $\mathcal{O}(d)$ operations. Thus, the Brownian bridge generation of the Brownian motion fits the problem formulation adopted in this article.

A third method for generating a Brownian motion is the *Karhunen-Loève expansion*. The eigenvalue problem of the covariance operator for $B(t)$, i.e.,

$\text{cov}(B(t), B(s)) = \min(t, s)$, leads to an infinite series for the Brownian motion in terms of the i.i.d. standard Gaussian random variables $X_1, X_2, \ldots$ and the sinusoidal eigenfunctions:

$$B(t; X_1, X_2, \ldots) = \sqrt{2T} \sum_{j=1}^{\infty} X_j \frac{\sin\left(\left(j - \frac{1}{2}\right)\pi t/T\right)}{\left(j - \frac{1}{2}\right)\pi}.$$

Again, the payoff functional for the option, $g(B(\cdot))$ can then be thought of as a functional of a countable number of standard Gaussian random variables, $f(X_1, X_2, \ldots)$. Truncating the Karhunen-Loève expansion at d terms corresponds to $f(X_1, \ldots, X_d, 0, \ldots)$ and costs $\mathcal{O}(d)$ mathematical operations to evaluate. The truncated Karhunen-Loève expansion includes the most important low frequency terms, and can be evaluated at any time, t.

In summary option pricing where the Brownian motion driving the SDE for the asset price is generated by a d-step Brownian Bridge or a d-term Karhunen-Loève expansion is an example of finding an expectation that fits the general framework set forth in this introductory section. The functional f corresponds to the option payoff. Although it depends on a countably infinite number of random variables, one may neglect all but the first d of them as an approximation.

2 Hilbert Spaces of Functionals

In this section, the Hilbert space where the functional f resides is constructed. The functional f depends on a countably infinite number of variables, and the Hilbert space containing f is constructed as the tensor product space of a countable number of reproducing kernel Hilbert spaces. As discussed in Section 1, the main problem is to approximate $\mu = \mathbb{E}[f(X_1, X_2, \ldots)]$, where $(X_1, X_2, \ldots)$ is an i.i.d. random sequence with common probability density function $\rho_1(\cdot)$, and the support of ρ_1 is assumed to be $\overline{I}$, where I is some open, half-open, or closed interval, which may be finite, semi-finite or infinite. As discussed in [2], the domain of f can be considered to be $I^{\mathbb{N}}$, where $\mathbb{N}$ is natural number set. The vector of nominal values of the random variables is denoted $\mathbf{c} = (c, c, \ldots)$, and it is used to recursively define the effects of parts of f, which are the parts that depend on a finite number of variables.

The symmetric, positive semi-definite kernel function $K_1 : I \times I \to \mathbb{R}$ is the building block used to construct the Hilbert space containing f. This kernel may possibly be unbounded. In addition, it is assumed that $K_1(c, c) = 0$, which implies that $K_1(x, c) = 0, \forall x \in I$, since K_1 is positive semi-definite. Let $\mathcal{H}(K_1)$ denote the reproducing kernel Hilbert space of functions with reproducing kernel, K_1. The above assumption on K_1 implies that $f(c) = 0$ for all $f \in \mathcal{H}(K_1)$.

Further assumptions are made on the finiteness and integrability of K_1:

$$h_1(x) := \int_{\overline{I}} K_1(x,y)\,\rho_1(y)\,\mathrm{d}y, \quad \forall x \in I, \quad h_1 \in \mathcal{H}(K_1), \tag{6a}$$

$$m := \int_{\overline{I}^2} K_1(x,y)\,\rho_1(x)\rho_1(y)\,\mathrm{d}x\mathrm{d}y < \infty, \tag{6b}$$

$$M := \int_{\overline{I}} K_1(x,x)\rho_1(x)\,\mathrm{d}x < \infty. \tag{6c}$$

Following the notation in [2], $\mathbb{U}$ is defined as a set of subsets of $\mathbb{N}$ with finite cardinality, namely $\mathbb{U} = \{u \subset \mathbb{N} : |u| < \infty\}$. Given any set $u \in \mathbb{U}$, $\mathbf{x}_u$ denotes the vector containing the coordinates of $\mathbf{x}$ whose indices are in u. The symmetric, positive semi-definite kernel

$$K_u(\mathbf{x}_u, \mathbf{y}_u) = \prod_{j \in u} K_1(x_j, y_j)$$

defines the Hilbert space $\mathcal{H}(K_u)$ containing functionals of $|u|$ variables. The domain of those functionals in $\mathcal{H}(K_u)$ is I^u.

To facilitate the definition of the Hilbert space of functionals of an infinite number of variables, a sequence of non-increasing, non-negative weights is introduced, $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots)$, which satisfies the following conditions:

$$\gamma_j \preceq j^{-1-2q}, \quad \gamma_1 \geq \gamma_2 \geq \cdots \geq 0, \tag{7}$$

where q is some positive constant. This implies that

$$\sum_{j=d}^{\infty} \gamma_j \preceq d^{-2q}. \tag{8}$$

Moreover, $\gamma_u = \prod_{j \in u} \gamma_j$ is defined as the product weight for any finite set u.

The Hilbert space of functionals with domain $I^{\mathbb{N}}$ is now defined as an infinite direct sum of reproducing kernel Hilbert spaces $\mathcal{H}(K_u)$ using the approach in [4]. The kernel K defining the Hilbert space $\mathcal{H}(K)$ is defined as

$$K(\mathbf{x}, \mathbf{y}) := \sum_{u \in \mathbb{U}} \gamma_u K_u(\mathbf{x}_u, \mathbf{y}_u) = \prod_{j=1}^{\infty} [1 + \gamma_j K_1(x_j, y_j)]. \tag{9}$$

The kernel $K(\mathbf{x}, \mathbf{y})$ defined in (9) is not necessarily finite for all $\mathbf{x}, \mathbf{y} \in I^{\mathbb{N}}$, especially for unbounded kernels K_1. Although $\mathcal{H}(K)$ may not be a reproducing kernel Hilbert space, it is a Hilbert space, and the reproducing property holds for the functional evaluation $f(\mathbf{x}_u, \mathbf{c})$ for any set $u \in \mathbb{U}$. Another useful kernel function $K^{(d)}$ is defined by

$$\begin{aligned} K^{(d)}(\mathbf{x}_{1:d}, \mathbf{y}_{1:d}) &= \sum_{u \subseteq 1:d} \gamma_u K_u(\mathbf{x}_u, \mathbf{y}_u) = \prod_{j=1}^{d} [1 + \gamma_j K_1(x_j, y_j)] \\ &= K\left((\mathbf{x}_{1:d}, \mathbf{c}), (\mathbf{y}_{1:d}, \mathbf{c})\right). \end{aligned}$$

The Hilbert space $\mathcal{H}(K^{(d)})$ contains functions depending on only the first d variables.

Following the argument in [14], it can be shown that $\mathcal{H}(K_u) \bigcap \mathcal{H}(K_v) = \{0\}$ for any $u \neq v$ because $\mathcal{H}(1) \bigcap \mathcal{H}(K_1) = \{0\}$. Consequently, the Hilbert space $\mathcal{H}(K)$ can be decomposed as

$$\mathcal{H}(K) = \bigoplus_{u \in \mathbb{U}} \mathcal{H}(\gamma_u K_u), \qquad \text{where} \quad \mathcal{H}(K_u) \perp \mathcal{H}(K_v), \quad \forall u \neq v. \tag{10}$$

The following definition recursively defines the Hilbert space $\mathcal{H}(K)$ of functionals, f, as a sum of their pieces f_u, which lie in the reproducing kernel Hilbert spaces $\mathcal{H}(K_u)$, as in [2,14].

Definition 1. *The Hilbert space $\mathcal{H}(K)$ consists of all functionals of the form*

$$f = \sum_{u \in \mathbb{U}} f_u, \qquad f_u \in \mathcal{H}(K_u),$$

such that

$$\sum_{u \in \mathbb{U}} \gamma_u^{-1} \|f_u\|^2_{\mathcal{H}(K_u)} < \infty.$$

In case of convergence, $\langle f, g \rangle_{\mathcal{H}(K)} = \sum_{u \in \mathbb{U}} \gamma_u^{-1} \langle f_u, g_u \rangle_{\mathcal{H}(K_u)}$ and $\|f\|^2_{\mathcal{H}(K)} = \sum_{u \in \mathbb{U}} \gamma_u^{-1} \|f_u\|^2_{\mathcal{H}(K_u)}$.

Since $K_u(\mathbf{x}_u, \mathbf{y}_u)$ vanishes if any one or more of the components of $\mathbf{x}_u$ equals the nominal value, c, the same holds for $f_u(\mathbf{x}_u)$. This implies that the functional $f \in \mathcal{H}(K)$ evaluated at $(\mathbf{x}_u, \mathbf{c}_{\mathbb{N} \setminus u})$ for $u \in \mathbb{U}$ is the sum of only a finite number of effects, f_v, namely, $f(\mathbf{x}_u, \mathbf{c}_{\mathbb{N} \setminus u}) = \sum_{v \subseteq u} f_v$, since $f_v(\mathbf{x}_u, \mathbf{c}_{\mathbb{N} \setminus u})$ vanishes for $v \not\subseteq u$. Therefore, the effects, f_u, can be obtained recursively by the formula $f_u = f(\mathbf{x}_u, \mathbf{c}_{\mathbb{N} \setminus u}) - \sum_{v \subset u} f_v$.

3 Single Level Algorithm

The main result of the worst case error analysis for the single level algorithm defined in (2) is that the worst case error can be decomposed as the bias due to approximating an infinite dimensional integral by a finite dimensional approximation to the functional, and the sampling error for approximating a d-dimensional integral by a sample average, i.e., the discrepancy. The following theorem summarizes the main result from [2].

Theorem 1. *Suppose that K_1 is a symmetric, real-valued, positive semi-definite kernel function defined on I^2 that satisfies the assumptions (6). Consider a Hilbert space $\mathcal{H}(K)$ of functionals $f : I^{\mathbb{N}} \to \mathbb{R}$, which is defined above in terms of K_1 and the weights γ_j satisfying assumption (8). Then the worst case error*

for approximating the expectation of the functional f by a Monte Carlo type algorithm of the form (2) is

$$\begin{aligned} \textit{worst-err}(\{\mathbf{x}_{i,1:d}\}_{i=1}^n; K) &= \sup_{||f||_{\mathcal{H}(K)} \le 1} |\mu(f) - \hat{\mu}(f)| \\ &= \sqrt{\textit{worst-bias}^2(d; K) + \mathcal{D}^2\left(\{\mathbf{x}_{i,1:d}\}_{i=1}^n; K^{(d)}\right)}, \end{aligned}$$

where

$$\begin{aligned} \textit{worst-bias}^2(d; K) &= \prod_{j=1}^{d}[1+\gamma_j m]\left[\prod_{j=d+1}^{\infty}[1+\gamma_j m]-1\right] \preceq d^{-2q}, \\ \mathcal{D}^2(\{\mathbf{x}_{i,1:d}\}_{i=1}^n; K^{(d)}) &= \prod_{j=1}^{d}[1+\gamma_j m] - \frac{2}{n}\sum_{i=1}^{n}\prod_{j=1}^{d}[1+\gamma_j h_1(x_{i,j})] \\ &\quad + \frac{1}{n^2}\sum_{i,k=1}^{n}\prod_{j=1}^{d}[1+\gamma_j K_1(x_{i,j}, x_{k,j})]. \end{aligned} \tag{11}$$

Results on strong tractability, e.g., [8,9,10,11] imply that the discrepancy $\preceq n^{-p}$, and so worst-err$^2(\{\mathbf{x}_{i,1:d}\}_{i=1}^n; K) \preceq d^{-2q} + n^{-2p}$, where the implied constants are independent of d and n. As mentioned in the introduction the cost of a single function evaluation is proportional to d^s for some $s \ge 0$, and so the computational cost of the single level algorithm is proportional to $N = nd^s$. Minimizing the error upper bound with respect to d and n given the computational cost N yields $d \asymp n^{p/q}$, and

$$\min_{\substack{n,d \\ nd=N}} \text{worst-err}(\mathbf{x}_{i,1:d}; K) \preceq N^{-pq/(ps+q)}. \tag{12}$$

4 Multilevel Algorithms

As noted in the introduction, this single level algorithm convergence is worse than the rate of $\preceq N^{-\min(p,q/s)}$ that one might dream of. The multilevel algorithm allows the dream convergence rate to be realized by focusing more sampling effort on the lower dimensions.

Consider a sequence of increasing dimensions $0 = d_0 < d_1 < \cdots < d_{L+1} = \infty$. For any $d \in \mathbb{N}$, define the projection Φ_d from $\mathcal{H}(K)$ to $\mathcal{H}(K^{(d)})$ as

$$(\Phi_d f)(x_1, \ldots, x_d) = f(x_1, \ldots, x_d, c, c, \ldots), \quad \forall f \in \mathcal{H}(K).$$

By convention define $\Phi_0 f = 0$ and $\Phi_\infty f = f$. Because $\Phi_d f \perp f - \Phi_d f$, Φ_d is an orthogonal projection onto $\mathcal{H}(K^{(d)})$. Using this notation, the telescopic sum for the infinite dimensional expectation in (5) can be rewritten as

$$\mu(f) = \sum_{l=1}^{L+1} \mu_{d_l}\left(\Phi_{d_l} f - \Phi_{d_{l-1}} f\right), \tag{13}$$

where the d-dimensional expectation μ_d is defined in (4).

The multilevel algorithm for approximating $\mu(f)$ approximates each term in (13), except for the last one, by a Monte Carlo or quasi-Monte Carlo algorithm of the form of (2). The last term in (13) is approximated by 0 since it is assumed to be small. Specifically, the multilevel algorithm takes the form

$$\hat{\mu}(f) = \sum_{l=1}^{L+1} \hat{\mu}_{d_l}(\Phi_{d_l} f - \Phi_{d_{l-1}} f), \tag{14a}$$

$$\hat{\mu}_{d_l}(\Phi_{d_l} f - \Phi_{d_{l-1}} f) = \frac{1}{n_l} \sum_{i=1}^{n_l} \left[f(\mathbf{x}_{i,1:d_l}^{(l)}, \mathbf{c}) - f(\mathbf{x}_{i,1:d_{l-1}}^{(l)}, \mathbf{c}) \right], \quad l = 1, \ldots, L, \tag{14b}$$

$$\hat{\mu}_{d_{L+1}}(\Phi_{d_{L+1}} f - \Phi_{d_L} f) = 0. \tag{14c}$$

An n_l-point design of dimension d_l is used to compute the level l approximation, $\hat{\mu}_{d_l}$. All dimensions are used to evaluate $f(\mathbf{x}_{i,1:d_l}^{(l)}, \mathbf{c})$ and the first d_{l-1} dimensions are used to evaluate $f(\mathbf{x}_{i,1:d_{l-1}}^{(l)}, \mathbf{c})$.

Since the cost of evaluating $f(\mathbf{x}_{i,1:d_l}^{(l)}, \mathbf{c}) - f(\mathbf{x}_{i,1:d_{l-1}}^{(l)}, \mathbf{c})$ for a single i is proportional to d_l^s, the total computational cost of $\hat{\mu}(f)$ proportional to

$$N = \sum_{l=1}^{L} n_l d_l^s. \tag{15}$$

In many cases $s = 1$ matches the practical situation, but the analysis here allows s to be any non-negative number. As shall be seen, the advantage of the multilevel algorithm is that one can choose n_l to decrease as d_l increases to obtain greater efficiency than the single level algorithm.

5 Multilevel Algorithm Worst-case Error Analysis

The worst case error of the multilevel algorithm $\hat{\mu}$ defined in the previous section is now analyzed for functionals f lying in the Hilbert space $\mathcal{H}(K)$ defined in Definition 1. It is found in Theorem 2 that this worst case error is the square root of the square bias plus the sum of differences of discrepancies with different dimensions at each level. Although Theorem 2 provides a tight upper bound for the error, it is technically more difficult to get the explicit convergence rate, and a more convenient loose upper bound is provided in Corollary 1.

To facilitate the error analysis and simplify the notation, the following subsets of $\mathbb{U}$ are defined:

$$\Omega_l = 2^{1:d_l} \setminus 2^{1:d_{l-1}} = \{u \in \mathbb{U} : u \in 1 : d_l \text{ and } u \notin 1 : d_{l-1}\}, \quad l = 1, \ldots, L+1.$$

Given the convention that $d_0 = 0$ and $d_{L+1} = \infty$, this corresponds to $\Omega_1 = 2^{1:d_1}$ and $\Omega_{L+1} = \mathbb{U} \setminus 2^{1:d_L}$. Define the following projections for any $f \in \mathcal{H}(K)$:

$$\Psi_l f = \Phi_{d_l} f - \Phi_{d_{l-1}} f = \sum_{u \in \Omega_l} f_u \in \mathcal{H}\left(\sum_{u \in \Omega_l} \gamma_u K_u \right), \qquad l = 1, 2, \ldots, L+1.$$

Whereas $\Phi_{d_l} f$ yields the part of f depending only on the first d_l coordinates, $\Psi_l f$ yields the part of f depending only on the first d_l coordinates, but not on the first d_{l-1} coordinates. It follows from (10), that the Ψ_l are mutually orthogonal projections and

$$f = \sum_{l=1}^{L+1} \Psi_l f, \quad \|f\|^2_{\mathcal{H}(K)} = \sum_{l=1}^{L+1} \|\Psi_l f_l\|^2_{\mathcal{H}(K)}, \qquad \forall f \in \mathcal{H}(K).$$

The main task is to get a tight upper bound for the worst case error

$$\text{worst-err} = \sup_{\|f\|_{\mathcal{H}(K)} \le 1} |\mu(f) - \hat{\mu}(f)| = \sup_{\|f\|_{\mathcal{H}(K)} \le 1} \left| \sum_{l=1}^{L+1} (\mu_{d_l} - \hat{\mu}_{d_l})(\Psi_l f) \right|. \quad (16)$$

Each term in the sum above is a bounded linear functional of $\Psi_l(f)$. Since f lies in a Hilbert space, these bounded linear functionals may be written as the inner products of $\Psi_l(f)$ with the corresponding representers.

As in [2], the infinite dimensional expectation is defined as a countable sum of the finite dimensional expectations of pieces, i.e.,

$$\mu(f) = \mathbb{E}[f(X_1, X_2, \ldots)] := \sum_{u \in \mathbb{U}} \mathbb{E}[f_u(\mathbf{X}_u)] = \sum_{u \in \mathbb{U}} \langle h_u, f \rangle_{\mathcal{H}(K)} = \langle h, f \rangle_{\mathcal{H}(K)},$$

where the representers h and h_u are given by

$$h_u(\mathbf{x}_u) := \prod_{j \in u} \gamma_j h_1(x_j) \in \mathcal{H}(K_u),$$

$$h(\mathbf{x}) := \sum_{u \in \mathbb{U}} h_u(\mathbf{x}_u) = \prod_{j=1}^{\infty} [1 + \gamma_j h_1(x_j)].$$

For any $d \in \mathbb{N}$ the bounded linear functionals μ_d and $\hat{\mu}_d$ are expressed as inner products by

$$\mu_d(f) = \langle \Phi_d h, f \rangle_{\mathcal{H}(K)} = \langle h(\cdot, \mathbf{c}_{d+1:\infty}), f \rangle_{\mathcal{H}(K)},$$
$$\hat{\mu}_d(f) = \left\langle \frac{1}{n} \sum_{i=1}^{n} K(\cdot, (\mathbf{x}_{i,1:d}, \mathbf{c})), f \right\rangle_{\mathcal{H}(K)} = \left\langle \frac{1}{n} \sum_{i=1}^{n} K^{(d)}(\cdot, \mathbf{x}_{i,1:d}), f \right\rangle_{\mathcal{H}(K)}.$$

This means that $(\mu_{d_l} - \hat{\mu}_{d_l})(\Psi_l f) = \langle \xi_l, \Psi_l f \rangle_{\mathcal{H}(K)}$, where

$$\xi_l = h(\cdot, \mathbf{c}_{d_l+1:\infty}) - \frac{1}{n_l} \sum_{i=1}^{n_l} K^{(d_l)}(\cdot, \mathbf{x}^{(l)}_{i,1:d_l}), \quad l = 1, \ldots L,$$
$$\xi_{L+1} = h.$$

Note that since the Ψ_l are orthogonal projections, one may also write $(\mu_{d_l} - \hat{\mu}_{d_l})(\Psi_l f) = \langle \Psi_l \xi_l, \Psi_l f \rangle_{\mathcal{H}(K)}$.

Continuing from (16) one can now write

$$\text{worst-err} = \sup_{\|f\|_{\mathcal{H}(K)}\le 1} \left| \sum_{l=1}^{L+1} (\mu_{d_l} - \hat{\mu}_{d_l})(\Psi_l f) \right| = \sup_{\|f\|_{\mathcal{H}(K)}\le 1} \left| \sum_{l=1}^{L+1} \langle \Psi_l \xi_l, \Psi_l f \rangle_{\mathcal{H}(K)} \right|$$
$$= \sup_{\|f\|_{\mathcal{H}(K)}\le 1} |\langle \xi, f \rangle_{\mathcal{H}(K)}| = \|\xi\|_{\mathcal{H}(K)} = \sqrt{\sum_{l=1}^{L+1} \|\Psi_l \xi_l\|^2_{\mathcal{H}(K)}},$$

where $\xi := \sum_{l=1}^{L+1} \Psi_l \xi_l$. This leads to the following theorem:

Theorem 2. *Under the assumptions for the Hilbert space of functionals, f, in Section 2, the multilevel algorithm $\hat{\mu}$ defined in (14) has the following worst case error:*

$$\begin{aligned} \textit{worst-err}^2 &= \sup_{\|f\|_{\mathcal{H}(K)}\le 1} |\mu(f) - \hat{\mu}(f)|^2 \\ &= \sum_{l=1}^{L} \left[\mathcal{D}^2(P_l; K^{(d_l)}) - \mathcal{D}^2(P_l; K^{(d_{l-1})}) \right] \\ &\qquad + \prod_{j=1}^{d_L} [1 + \gamma_j m] \left[\prod_{j=d_L+1}^{\infty} [1 + \gamma_j m] - 1 \right], \end{aligned}$$

where $\mathcal{D}^2(P_l; K^{(d_l)})$ is the discrepancy defined in (11) dependent on the design $P_l = \{\mathbf{x}_{i,1:d_l}\}_{i=1}^{n_l}$, and the reproducing kernel for functions of d_l variables, $K^{(d_l)}$.

Proof. Most of the proof is contained in the preceding arguments. It is shown in [2] that the last term in the expression for the squared worst case error corresponds to $\|\Psi_{L+1}\xi_{L+1}\|^2_{\mathcal{H}(K)}$. What remains to be shown is that $\|\Psi_l \xi_l\|^2_{\mathcal{H}(K)} = \mathcal{D}^2(P_l; K^{(d_l)}) - \mathcal{D}^2(P_l; K^{(d_{l-1})})$ for $l = 1, \ldots, L$.

Note that $\mathcal{D}(P_l; K^{(d_l)}) = \|\xi_l\|_{\mathcal{H}(K)}$. Furthermore, the fact that $\Phi_{d_{l-1}} = \Psi_1 + \cdots + \Psi_{l-1}$ implies that Ψ_l and $\Phi_{d_{l-1}}$ are mutually orthogonal projections. Moreover, by definition, $\Phi_{d_l}\xi_l = \xi_l$. Thus, for $l = 1, \ldots, L$,

$$\begin{aligned} \mathcal{D}^2(P_l; K^{(d_l)}) &= \|\xi_l\|^2_{\mathcal{H}(K)} = \|\Phi_{d_l}\xi_l\|^2_{\mathcal{H}(K)} = \left\|(\Phi_{d_{l-1}} + \Psi_l)\xi_l\right\|^2_{\mathcal{H}(K)} \\ &= \left\|\Phi_{d_{l-1}}\xi_l\right\|^2_{\mathcal{H}(K)} + \|\Psi_l\xi_l\|^2_{\mathcal{H}(K)} = \mathcal{D}^2(P_l; K^{(d_{l-1})}) + \|\Psi_l\xi_l\|^2_{\mathcal{H}(K)}, \end{aligned}$$

which completes the proof.

Theorem 2 makes it possible to write the worst case error as a combination of the difference between square discrepancies of the same design, but with different dimensional kernel functions. Note that this difference of square discrepancies is always non-negative. The goal is to minimize the worst case error given the computational cost constraint. A key point here is how to make a good balance between the errors at different levels, l, by choosing d_l and n_l appropriately. To facilitate this a looser bound is derived.

Consider another non-increasing, non-negative sequence of coordinate weights, $\boldsymbol{\gamma}' = (\gamma_1', \gamma_2', \ldots)$ satisfying $\gamma_j \le \gamma_j'$, and use these new coordinate weights to define a kernel function K', where

$$K^{'}(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^{\infty} [1 + \gamma_j^{'} K_1(x_j, y_j)].$$

The kernel function K and $K^{'}$ differ only in terms of the different coordinate weights used to define them. The projections Φ_d, $d \in \mathbb{N}$, and Ψ_l, $l = 1, \ldots, L$ act on $\mathcal{H}(K')$ in the same way that they act on $\mathcal{H}(K)$. It follows that

$$\mathcal{H}(K) \subseteq \mathcal{H}(K^{'}), \quad \text{with} \quad \|f\|_{\mathcal{H}(K')} \le \|f\|_{\mathcal{H}(K)}, \quad f \in \mathcal{H}(K). \tag{17}$$

Lemma 1. *For any $g = \sum_{u \in \mathbb{U}} g_u \in \mathcal{H}(K)$ where $g_u \in \mathcal{H}(K_u)$, let $g' = \sum_{u \in \mathbb{U}} g_u'$, where $g_u' = (\gamma_u'/\gamma_u) g_u$. It follows that $\langle g, f \rangle_{\mathcal{H}(K)} = \langle g', f \rangle_{\mathcal{H}(K')}$ for all $f \in \mathcal{H}(K)$. Moreover, $\|\Psi_l g\|_{\mathcal{H}(K)} \le \sqrt{k_l} \, \|\Psi_l g'\|_{\mathcal{H}(K')}$ and $\mathcal{D}^2(P_l; K^{(d_l)}) - \mathcal{D}^2(P_l; K^{(d_{l-1})}) \le k_l \mathcal{D}^2(P_l; K'^{(d_l)})$ for $l = 1, \ldots, L$, where $k_l = \gamma_{d_{l-1}+1} / \gamma_{d_{l-1}+1}^{'}$.*

Proof. From the definition of the inner product in Definition 1 it follows that

$$\begin{aligned} \langle g, f \rangle_{\mathcal{H}(K)} &= \sum_{u \in \mathbb{U}} \frac{1}{\gamma_u} \langle g_u, f_u \rangle_{\mathcal{H}(K_u)} = \sum_{u \in \mathbb{U}} \frac{1}{\gamma_u'} \left\langle \frac{\gamma_u'}{\gamma_u} g_u, f_u \right\rangle_{\mathcal{H}(K_u)} \\ &= \sum_{u \in \mathbb{U}} \frac{1}{\gamma_u'} \langle g_u', f_u \rangle_{\mathcal{H}(K_u)} = \langle g', f \rangle_{\mathcal{H}(K')}, \end{aligned}$$

which proves the first part of the lemma. Moreover,

$$\begin{aligned} \|\Psi_l g\|_{\mathcal{H}(K)}^2 &= \sum_{u \in \Omega_l} \frac{1}{\gamma_u} \langle g_u, g_u \rangle_{\mathcal{H}(K_u)} = \sum_{u \in \Omega_l} \frac{\gamma_u}{\gamma_u'} \frac{1}{\gamma_u'} \left\langle \frac{\gamma_u'}{\gamma_u} g_u, \frac{\gamma_u'}{\gamma_u} g_u \right\rangle_{\mathcal{H}(K_u)} \\ &= \sum_{u \in \Omega_l} \frac{\gamma_u}{\gamma_u'} \frac{1}{\gamma_u'} \langle g_u', g_u' \rangle_{\mathcal{H}(K_u)} \le \sup_{u \in \mathbb{U}} \left(\frac{\gamma_u}{\gamma_u'} \right) \|\Psi_l g'\|_{\mathcal{H}(K')}^2 = k_l \, \|\Psi_l g'\|_{\mathcal{H}(K')}^2, \end{aligned}$$

which completes the second part of the proof. Finally, combining this fact from what is known from the proof of Theorem 2 it follows that

$$\begin{aligned} \mathcal{D}^2(P_l; K^{(d_l)}) - \mathcal{D}^2(P_l; K^{(d_{l-1})}) &= \|\Psi_l \xi_l\|_{\mathcal{H}(K)}^2 \le k_l \, \|\Psi_l \xi_l'\|_{\mathcal{H}(K')}^2 \\ &= k_l \left[\mathcal{D}^2(P_l; K'^{(d_l)}) - \mathcal{D}^2(P_l; K'^{(d_{l-1})}) \right] \le k_l \mathcal{D}^2(P_l; K'^{(d_l)}). \end{aligned}$$

for $l = 1, \ldots, L$.

This lemma allows the worst case error of the multilevel algorithm to be bounded in terms of discrepancies, not differences of discrepancies in the corollary below. The advantage is a simpler form that makes it possible to determine the optimal choice of the samples sizes and dimensions for each level. The disadvantage is that the error bound is no longer tight.

Corollary 1. *Under the assumptions of Theorem 2 and for the kernel K' defined in terms of the coordinate weights $\boldsymbol{\gamma}'$, it follows that the worst case error of the multilevel algorithm $\hat{\mu}$ is bounded loosely above by*

$$\textit{worst-err}^2 \le \sum_{l=1}^{L} k_l \mathcal{D}^2(P_l; K'^{(d_l)}) + \prod_{j=1}^{d_L}[1+\gamma_j m] \left[\prod_{j=d_L+1}^{\infty} [1+\gamma_j m] - 1 \right], \quad (18)$$

where $k_l = \gamma_{d_{l-1}+1}/\gamma'_{d_{l-1}+1}$.

The discrepancy, $\mathcal{D}(P_l; K'^{(d_l)})$, is that defined in (11), but for the Hilbert space $\mathcal{H}(K'^{(d_l)})$ defined in terms of the weights γ'_j. The advantage of introducing the new set of weights is that Corollary 1 has an upper bound with a k_l that can be made to become small as l increases. The next section explores the choice of γ'_j, n_l, and d_l to make the error bound as small as possible given a budget of mathematical operations.

6 Optimal choice of n_l and d_l

Studies of strong tractability in [10,11,15], and related articles show that it is possible to obtain

$$\mathcal{D}\left(P; K'^{(d)}\right) \preceq n^{-p}, \quad (19)$$

for some positive p, where the right hand side is independent of d. The order p is dependent on the following: i) the quality of the design, P, ii) the smoothness of the kernel K', which determines the smoothness of the associated Hilbert space of functionals, and iii) the rate of decay of the weights, γ'_j, defining K'.

The non-increasing, non-negative sequence of weights $\boldsymbol{\gamma}' = (\gamma'_1, \gamma'_2, \ldots)$ is assumed to satisfy

$$\gamma'_j \asymp j^{2(q-q')}\gamma_j, \quad (20)$$

from some positive q' no greater than q. This means that the γ'_j decays to zero more slowly than the γ_j, so $\mathcal{H}(K'^{(d)})$ is a larger space of functions than $\mathcal{H}(K^{(d)})$. This condition on the γ'_j together with (7) imply that

$$\gamma'_j \asymp j^{2(q-q')}\gamma_j \asymp j^{2(q-q')} j^{-1-2q} \preceq j^{-1-2q'}.$$

Table 1, reproduced from [2], provides some values for p, the order of convergence of discrepancy in (19) depending on q' for rank-1 lattices and digital nets. Note that $\mathcal{H}(K)$ contains functions with mixed partial first derivative that are squared integrable.

Condition (20) implies that

$$k_1 = 1, \qquad k_l = \frac{\gamma_{d_{l-1}+1}}{\gamma'_{d_{l-1}+1}} \preceq d_{l-1}^{-2(q-q')}, \quad l = 2, 3, \ldots. \quad (21)$$

Table 1. The choice of p and q based on different designs

Design	p
Simple random sequence	$1/2$
Rank-1 lattices [8]	$\min\left(1, q' + \frac{1}{2}\right) - \varepsilon$
Niederreiter (T, s)-net $(q' > 1/2)$ [11]	$\min\left(1, \frac{q'}{2} + \frac{1}{4}\right) - \varepsilon$

This condition and (19), imply that $k_l \mathcal{D}^2\left(P_l; K'^{(d_l)}\right) \preceq d_{l-1}^{-2(q-q')} n_l^{-2p}$ for $l = 2, 3, \ldots$. As in Theorem 1, the last term in (18) is $\preceq d_L^{-2q}$. Thus, the task is to optimally choose n_l and d_l such that the worst case error bound

$$\text{worst-err}^2 \preceq \text{w-up}^2 := n_1^{-2p} + n_2^{-2p} d_1^{-2(q-q')} + \cdots + n_L^{-2p} d_{L-1}^{-2(q-q')} + d_L^{-2q} \quad (22)$$

is minimized under the computational cost constraint (15). The following theorem describes the best that can be hoped for for this constrained optimization problem. Theorem 4 demonstrates that this can be achieved.

Theorem 3. *Suppose $n_1 \geq \cdots \geq n_L$ and $d_1 < \cdots < d_L$, and let the computational cost of the multilevel algorithm, N, be defined as in (15). For any values of $p, q, q' > 0$ with $q \geq q'$ it follows that w-up, has the lower bound*

$$\text{w-up} \succeq N^{-p \min\left(1, \frac{q}{ps+q'}\right)}. \quad (23)$$

Proof. Since $N \succeq n_1 d_1^s \succeq n_1$ and $N \succeq n_L d_L^s \succeq n_L d_{L-1}^s$, so w-up^2 is bounded below by its first term and last two terms:

$$\begin{aligned}
\text{w-up}^2 &\succeq n_1^{-2p} + n_L^{-2p} d_{L-1}^{-2(q-q')} + d_L^{-2q} \\
&\succeq N^{-2p} + n_L^{-2p} \left(\frac{N}{n_L}\right)^{-2(q-q')/s} + \left(\frac{N}{n_L}\right)^{-2q/s} \\
&\succeq N^{-2p} + N^{-\frac{2pq}{ps+q'}} \left[\left(\frac{n_L}{N^{\frac{q'}{ps+q'}}}\right)^{-2(ps+q'-q)/s} + \left(\frac{n_L}{N^{\frac{q'}{ps+q'}}}\right)^{2q/s} \right].
\end{aligned}$$

If $ps + q' \leq q$, then both powers of $n_L N^{-\frac{q'}{ps+q'}}$ are positive and are minimized by taking $n_L \asymp 1$, which implies that

$$\text{w-up}^2 \succeq N^{-2p} + N^{-2(q-q')/s} + N^{-2q/s} \asymp N^{-2p}.$$

If $ps + q' > q$, then the two powers of $n_L N^{-\frac{q'}{ps+q'}}$ have opposite signs, and their sum is minimized by taking $n_L N^{-\frac{q'}{ps+q'}} \asymp 1$. This means that

$$\text{w-up}^2 \succeq N^{-2p} + N^{-\frac{2pq}{ps+q'}} \asymp N^{-\frac{2pq}{ps+q'}}.$$

Theorem 3 suggests that the convergence rate for the multilevel method might be as fast as $N^{-p\min\left(1,\frac{q}{ps+q'}\right)}$. The next step is to choose the n_l and d_l optimally to achieve this rate of convergence. This is done by choosing n_l to decrease exponentially and d_l to increase exponentially, as follows:

$$n_l \asymp 2^{L-bl}, \quad d_l \asymp 2^{cl}, \quad l = 1, \ldots, L, \tag{24}$$

where b and c are positive constants to be determined, and $b \leq 1$ to ensure that $n_L \succeq 1$. Correspondingly the computational cost of the algorithm, as defined in (15), is

$$N \asymp \sum_{l=1}^{L} 2^{L-bl} 2^{cls} = 2^L \sum_{l=1}^{L} 2^{(cs-b)l} \asymp \begin{cases} 2^L, & \text{if } b > cs, \\ 2^{L(1+cs-b)}, & \text{if } b < cs. \end{cases} \tag{25}$$

Moreover, the upper bound on the square worst case error becomes

$$\begin{aligned} \text{worst-err}^2 &\preceq \sum_{l=1}^{L} 2^{-2c(l-1)(q-q')} 2^{-2p(L-bl)} + 2^{-2cLq} \\ &\asymp 2^{-2pL} \left[\sum_{l=1}^{L} 2^{2(bp-c(q-q'))(l-1)} + 2^{2L(p-cq)} \right]. \end{aligned} \tag{26}$$

The following theorem describes the optimal choice of b and c such that the above convergence order can be achieved.

Theorem 4. *Consider any $s \geq 0$ and $p, q, q' > 0$ with $q \geq q'$ and $ps + q' \neq q$. Let the computational cost of the multilevel algorithm is defined as $N = \sum_{l=1}^{L} n_l d_l^s$. Then the worst case error of this algorithm is bounded above by*

$$\textit{worst-err} \preceq N^{-p\min\left(1,\frac{q}{ps+q'}\right)}.$$

This convergence rate is achieved by choosing b and c to satisfy $0 < b \leq 1$, $c > 0$, and

i. $p^2 s/q \leq cps < bp < c(q - q')$ in the case $ps + q' < q$, and
ii. $1 + cq'/p = b < 1 - q'/q$ in the case $ps + q' > q$.

Proof. The proof is proceeds by considering two cases, as in the proof of the previous theorem. First, consider the case of $ps + q' < q$ and the choices of b and c described above. Since $b > cs$, it follows from (25) that $N \asymp 2^L$. Since $bp - c(q - q') < 0$ and $p - cq \leq 0$, it follows from (26) that worst-err$^2 \preceq 2^{-2pL} \asymp N^{-2p}$.

Next, consider the case of $ps + q' > q$. The proof of Theorem 3 suggests choosing the last two terms on the right hand side of (26) to be of the same order, which implies that $b = 1 + cq'/p$ and $c = (1-b)p/q'$. Thus, $pb - c(q-q') = pq(b - 1 + q'/q)/q' < 0$ under the inequality constraint on b, and the last term

in the sum in (26) is the dominant one. Moreover, the inequality constraint on b also implies that

$$cs - b = \frac{ps - b(ps+q')}{q'} > \frac{ps - (q-q')(ps+q')/q}{q'} = \frac{ps+q'-q}{q} > 0,$$

so $N \asymp 2^{L(1+cs-b)}$ by (25). Thus, from (26) it follows that

$$\text{worst-err} \preceq 2^{-Lcq} \asymp N^{-cq/(1+cs-b)} = N^{-pq/(ps+q')}.$$

Theorems 3 and 4 still leave open the question of the optimal choice of q'. Thus, one has worst-err $\preceq N^{-\tau}$, where

$$\tau = \max_{q'} \; p \min\left(1, \frac{q}{ps+q'}\right) \le \min\left(p, q/s\right). \tag{27}$$

Since p typically depends on q', this is a nontrivial, but often solvable problem. In some cases, the upper bound above is attainable.

Consider the low discrepancy designs given in Table 1. For rank-1 lattice points, since $p = \min(1, q' + \frac{1}{2})$, ignoring the arbitrarily small positive number ε, the order of convergence for the best q' is

$$\begin{aligned}\tau &= \min\left(\frac{q}{s}, \frac{q+1/2}{s+1}, 1\right)\\ &= \begin{cases} \frac{q}{s}, & 0 < q < \frac{s}{2}, \text{ by choosing } q' = 0,\\ \frac{q+1/2}{s+1}, & \frac{s}{2} \le q < s + \frac{1}{2}, \text{ by choosing } q' = \frac{q-s/2}{s+1},\\ 1, & s + \frac{1}{2} \le q, \text{ by choosing } q' = \frac{1}{2}. \end{cases}\end{aligned} \tag{28}$$

Correspondingly, the best convergence exponent that can be obtained for a Niederreiter (T, d)-net is

$$\begin{aligned}\tau &= \min\left(\frac{q}{s+1}, \frac{q+1/2}{s+2}, 1\right)\\ &= \begin{cases} \frac{q}{s+1}, & \frac{1}{2} \le q < \frac{s+1}{2}, \text{ by choosing } q' = \frac{1}{2},\\ \frac{q+1/2}{s+2}, & \frac{s+1}{2} \le q < s + \frac{3}{2}, \text{ by choosing } q' = \frac{2q-s/2}{s+2},\\ 1, & s + \frac{3}{2} \le q, \text{ by choosing } q' = \frac{3}{2}. \end{cases}\end{aligned} \tag{29}$$

Figure 1 compares the order of convergence in the case $s = 1$ for the multilevel algorithm, the single level algorithm described in Section 3, and results from [4]. One plot is for rank-1 lattice designs and the other for Niederreiter net designs. The convergence order for the multilevel algorithm is given by (28) and (29). The convergence order for the single level algorithm is based on (12) and Table 1 with q' replaced by q. The upper bound on the order of convergence is $\min(1, q)$, which comes from (27). It is independent of the design but dependent on the smoothness of the space $\mathcal{H}(K)$. Unfortunately, the upper bound is only attained for a certain range of q. In the plot of rank-1 lattice rules, the convergence order

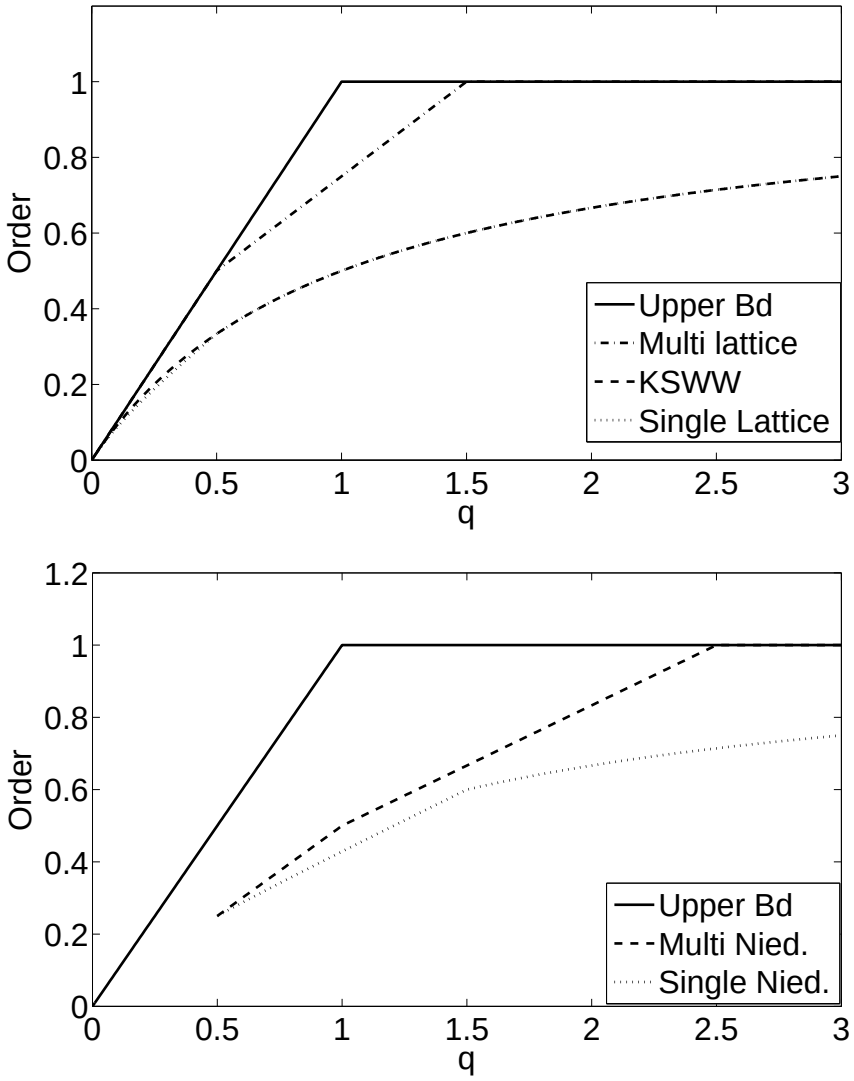

Fig. 1. The order of convergence for a multilevel algorithm based on rank-1 lattice rules (top) and a Niederreiter (T, d)-net (bottom) compared to the single level algorithm, the upper bound in (27) and the lower bound of [4].

for the single level quite similar to the results from [4], except that [4] is slightly superior value for $q \leq \frac{1}{2}$. The multilevel algorithm based on lattice rules gives the best orders of convergence among all algorithms compared.

The lattice and net designs referred to in Table 1 have the advantage of being extensible in both dimension and sample size. This allows one to use parts of one large design for each level of the multilevel algorithm. Specifically, one can remove the superscript (l) labeling the design points in (14) and re-arrange the terms to arrive at an equivalent formula that uses fewer computational operations. Recall that $d_0 = 0$ and $f(\mathbf{x}_{i,1:d_0}, \mathbf{c}) = 0$, and furthermore assume that $n_1 > n_2 > \cdots > n_L > n_{L+1} = 0$. Then one obtains the following equivalent expression for the multilevel algorithm:

$$\begin{aligned}
\hat{\mu}(f) &= \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left[f(\mathbf{x}_{i,1:d_l}, \mathbf{c}) - f(\mathbf{x}_{i,1:d_{l-1}}, \mathbf{c}) \right] \\
&= \sum_{l=1}^{L} \sum_{k=l}^{L} \sum_{i=n_{k+1}+1}^{n_k} \frac{1}{n_l} \left[f(\mathbf{x}_{i,1:d_l}, \mathbf{c}) - f(\mathbf{x}_{i,1:d_{l-1}}, \mathbf{c}) \right] \\
&= \sum_{k=1}^{L} \sum_{i=n_{k+1}+1}^{n_k} \sum_{l=1}^{k} \frac{1}{n_l} \left[f(\mathbf{x}_{i,1:d_l}, \mathbf{c}) - f(\mathbf{x}_{i,1:d_{l-1}}, \mathbf{c}) \right] \\
&= \sum_{k=1}^{L} \sum_{i=n_{k+1}+1}^{n_k} \left\{ \frac{1}{n_k} f(\mathbf{x}_{i,1:d_k}, \mathbf{c}) + \sum_{l=1}^{k-1} \left[\frac{1}{n_l} - \frac{1}{n_{l+1}} \right] f(\mathbf{x}_{i,1:d_l}, \mathbf{c}) \right\} \\
&= \sum_{k=1}^{L} \left\{ \frac{1}{n_k} \sum_{i=n_{k+1}+1}^{n_k} f(\mathbf{x}_{i,1:d_k}, \mathbf{c}) + \sum_{l=1}^{k-1} \left[\frac{1}{n_l} - \frac{1}{n_{l+1}} \right] \sum_{i=n_{k+1}+1}^{n_k} f(\mathbf{x}_{i,1:d_l}, \mathbf{c}) \right\}.
\end{aligned}$$

Here, each point of the design, $\mathbf{x}_i$ has theoretically unbounded dimension. One uses the first $d_1, \ldots, d_l$ dimensions of this design point where l is defined by the relation $n_{l+1} < i \leq n_l$. The savings here does not affect order of operations required for the multilevel algorithm but does affect the leading constant.

7 Conclusion

This paper investigates the worst case error of the multilevel algorithm for computing the expectation of functions of infinitely many variables. The quality of the sampling points or design are essential in the determination of the convergence order of the worst case error, since the worst case error is dependent on the discrepancy at each level. Moreover, the strong tractability of the algorithm, i.e., the independence of the discrepancy on the dimension, is also necessary. Our main result is to achieve a superior convergence order to the single level algorithm. For a range of values of q, a measure of how quickly the importance of the higher numbered variables decays, one can obtain the ideal convergence rate of $\preceq N^{-\min(p,q/s)}$. In addition, it has been shown how the sample size and the truncated dimension should be chosen in terms of q to achieve this rate.

An outstanding question is how to compute the truncated dimensions and sample sizes on the fly without a prior knowledge of q or even p. This question, for which even an empirical answer might be useful, is a topic for future research.

References

1. J. Creutzig, S. Dereich, T. Müller-Gronbach, K. Ritter, Infinite-dimensional quadrature and approximation of distributions, Found. Comput. Math.
2. B. Niu, F. J. Hickernell, Monte Carlo simulation of stochastic integrals when the cost of function evaluation is dimension dependent, in: P. L'Ecuyer, A. Owen (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2008, Springer-Verlag, Berlin, 2010, to appear.
3. F. J. Hickernell, T. Müller-Gronbach, B. Niu, K. Ritter, Multi-level monte carlo algorithms for infinite-dimensional integration on $\mathbb{R}^{\mathbb{N}}$, submitted for publication (2009).
4. F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski, H. Woźniakowski, Liberating the dimension, submitted for publication (2009).
5. F. J. Hickernell, A generalized discrepancy and quadrature error bound, Math. Comp. 67 (1998) 299–322.
6. F. J. Hickernell, Goodness-of-fit statistics, discrepancies and robust designs, Statist. Probab. Lett. 44 (1999) 73–78.
7. H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, 1992.
8. F. J. Hickernell, I. H. Sloan, G. W. Wasilkowski, The strong tractability of multivariate integration using lattice rules, in: H. Niederreiter (Ed.), Monte Carlo and Quasi-Monte Carlo Methods 2002, Springer-Verlag, Berlin, 2004, pp. 259–273.
9. E. Novak, H. Woźniakowski, When are integration and discrepancy tractable?, in: Foundations of Computational Mathematics, Vol. 284 of London Math. Soc. Lecture Note Ser., Cambridge University Press, Cambridge, 2001, pp. 211–266.
10. H. Woźniakowski, Efficiency of quasi-Monte Carlo algorithms for high dimensions, in: H. Niederreiter, J. Spanier (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 1998, Springer-Verlag, Berlin, 2000, pp. 114–136.
11. R. X. Yue, F. J. Hickernell, Strong tractability of multivariate integration over Banach spaces, SIAM J. Numer. Anal. 44 (2006) 2559–2583.
12. M. B. Giles, Multilevel Monte Carlo path simulation, Oper. Res. 56 (2008) 607—617.
13. M. B. Giles, B. J. Waterhouse, Multilevel quasi-Monte Carlo path simulation, Radon Series Comp. Appl. Math 8 (2009) 1–18.
14. F. J. Hickernell, X. Wang, The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimension, Math. Comp. 71 (2002) 1641–1661.
15. F. J. Hickernell, H. Niederreiter, The existence of good extensible rank-1 lattices, J. Complexity 19 (2003) 286–300.