Automatic Monte Carlo Algorithms Where the Integrand Size Is Unknown *

Fred J. Hickernell[1], Lan Jiang[1], Yuewei Liu[2], and Art Owen[3]

[1] Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA, hickernell@iit.edu,ljiang14@hawk.iit.edu
[2] School of Mathematics and Statistics, Lanzhou University, Lanzhou City, Gansu, China 730000, ???
[3] Art's address here with email here

Summary. We attempt a probabilistic analysis of simple Monte Carlo, achieving probabilistic error bounds when the kurtosis is controlled. The algorithm uses a sample size that depends adaptively on the estimated variance of the integrand. Thus, the algorithm is nonlinear (depending essentially on the function). The advantage of what is done here over standard error analysis (complexity theory) is that the algorithm does not depend a priori on the scale of the problem (in this case the variance) to determine the number of samples. Our intention, if what is done here is correct, is to try to extend this to the more sophisticated sampling schemes and infinite dimensional problems.

1 Introduction

Monte Carlo algorithms for multidimensional integrals have been in use for half a century. The integral or mean, μ, of a d-variate function $f:\mathbb{R}^d \to \mathbb{R}$, i.e.,

$$\mu = \mu(f) = \int_{\mathbb{R}^d} f(\boldsymbol{x})\rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

is approximated by the sample mean of integrand values,

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i, \qquad Y_i = f(\boldsymbol{X}_i). \tag{1}$$

Here $\rho:\mathbb{R}^d \to [0,\infty)$ is a specified probability density function, e.g., the uniform density on a cube or the normal density. For simple Monte Carlo the points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ independent and identically distributed (i.i.d.) random variables with the marginal probability density function ρ. Other choices, such as quasi-random points, may provide greater accuracy, but this paper deals with only simple Monte Carlo.

It is natural to that practitioners would want an automatic Monte Carlo algorithm that takes as its input the integrand, f, the probability density, ρ, and an error tolerance ε, and produces

* The first and second authors were partially supported by the National Science Foundation under DMS-0923111 and DMS-1115392

a value $\hat{\mu}_n$ that is within ε of μ while requiring only a reasonable sample size, n. While there is an extensive theory on Monte Carlo algorithms, there are some key gaps that must be filled to make it an automatic algorithm. The aim of this article is to provide such an algorithm. Next this problem of approximating μ is summarized from the perspective of a statistician and then from the perspective of an information-based complexity theorist or numerical analyst.

Art will add and edit this material in this section, especially the next subsection.

1.1 The Statistician's Perspective

The statistician's sees this a problem of constructing non-parametric, fixed-width confidence intervals for μ. Setting $Y = f(\boldsymbol{X})$, the goal is to approximate its mean, $\mu = E(Y)$, from an i.i.d. sample, $Y_i = f(\boldsymbol{X}_i)$, $i = 1, \ldots, n$, to obtain an expression

$$\operatorname{Prob}\left[|\hat{\mu}_n - \mu| \le \varepsilon\right] \ge 1 - \alpha. \tag{2}$$

Here taking $\alpha = 5\%$ corresponds to a 95% confidence interval for μ with half-width ε. Although the probability distribution for Y is not readily obtainable, the Central Limit Theorem states that the sample mean, $\hat{\mu}_n$, is approximately normally distributed for large n. This fact may be used to determine the sample size n to give an approximate confidence interval,

$$\operatorname{Prob}\left[|\hat{\mu}_n - \mu| \le \frac{z_{\alpha/2}\sigma}{\sqrt{n}}\right] \approx 1 - \alpha, \tag{3}$$

where $\sigma^2 = \operatorname{var}(Y) = E[(Y-\mu)^2]$, and z_α denotes the $(1-\alpha)100\%$ percentile of the standard Gaussian distribution. This suggests taking a sample size of $n = \lceil (z_{\alpha/2}\sigma/\varepsilon)^2 \rceil$ to attain (2). Since σ is not known a priori, it must be approximated by the sample variance of the Y_i:

$$\hat{\sigma}^2 = \mathfrak{C}^2 \hat{v}_{n_\sigma}, \quad \text{where} \quad \hat{v}_{n_\sigma} = \frac{1}{n_\sigma - 1}\sum_{i=1}^{n_\sigma} (Y_i - \hat{\mu}_{n_\sigma})^2, \quad \hat{\mu}_{n_\sigma} = \frac{1}{n_\sigma}\sum_{i=1}^{n_\sigma} Y_i. \tag{4}$$

The variance inflation factor, $\mathfrak{C} > 1$, accounts for the fact the unbiased variance estimator, $\hat{v}_{n_\sigma}$, may be larger or smaller than the true variance, σ^2. Then the sample size for the sample mean may be chosen as $n = \lceil (z_{\alpha/2}\hat{\sigma}/\varepsilon)^2 \rceil$. To avoid dependence between $\hat{\sigma}^2$ and $\hat{\mu}_n$, an independent sample of Y_i, should be used to compute $\hat{\mu}_n$, i.e.,

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=n_\sigma+1}^{n_\sigma+n} Y_i, \tag{5}$$

for a total cost of $n_\sigma + n$ samples.

While this fixed-width confidence interval may be often correct, it is not guaranteed. There are two approximations, whose errors must be understood and controlled: approximating the distribution $\hat{\mu}_n$ as a Gaussian distribution, and approximating σ^2 by the sample variance. The key to controlling these two errors, as seen in Theorem 1, is in assuming a bound on the kurtosis, i.e.,

$$\kappa = \operatorname{kurt}(Y) = \frac{E[(Y-\mu)^4]}{\sigma^4} \le \kappa_{\max}. \tag{6}$$

(Note, sometimes in the literature, the kurtosis is defined as the expression above minus three.) A bounded kurtosis means that Y has or does not have ????? **help Art**. A bounded kurtosis allows one to reliably bound $\operatorname{var}(Y)$ in terms of the sample variance, and also allows one to obtain a rigorous bound on the deviation of the sample mean from the true mean. It was shown by (Bahadur and Savage, 1956, Corollary 2) that nonparametric confidence intervals are impossible for convex sets of distributions. Assuming bounded kurtosis produces a nonconvex set of possible distributions for Y.

Another approach to obtaining a fixed-width confidence interval for μ would be to assume an upper bound on σ^2 and apply Chebyshev's inequality. The disadvantage of this approach is that while it might work for $Y = f(\boldsymbol{X})$, it would not work for $cY = cf(\boldsymbol{X})$ if c were large enough, since $\mathrm{var}(cY) = c^2 \,\mathrm{var}(Y)$. However, since $\mathrm{kurt}(cY) = \mathrm{kurt}(Y)$ if the procedure for obtaining a fixed-width confidence interval that is described in this article works for Y, it also works for cY.

1.2 The Information-Based Complexity Theorist's or Numerical Analyst's Perspective

The information-based complexity theorist or numerical analyst sees this as a multivariate integration problem. There are a myriad of cubature methods, each making certain assumptions on the domain of integration and the smoothness of the integrand, f, and deriving error bounds in terms of some semi-norm of f. Simple Monte Carlo methods work for functions with low degrees of smoothness and the natural space of integrands is $\mathscr{L}_2$, where the $\mathscr{L}_p$ norm is defined as

$$\|f\|_p := \left\{ \int_{\mathbb{R}^d} |f(\boldsymbol{x})|^p \rho(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right\}^{1/p}. \tag{7}$$

The root mean square error of the simple Monte Carlo method is $\|f - \mu(f)\|_2 / \sqrt{n}$.

Again, the practical problem again in applying this error analysis to determine the sample size, n, to guarantee the desired error, one must know the size of the integrand, $\|f - \mu(f)\|_2 = \sqrt{\mathrm{var}(Y)} = \sigma$. Analogous to the statistical argument given above, the solution to this problem is not to look at balls of integrands, i.e.,

$$\mathscr{B}_{\sigma_{\max}} = \{f \in \mathscr{L}_2 : \|f - \mu(f)\|_2 \le \sigma_{\max}\}, \tag{8}$$

but to look at *cones* of integrands

$$\mathscr{C}_{\kappa_{\max}} = \{f \in \mathscr{L}_4 : \|f - \mu(f)\|_4 \le \kappa_{\max}^{1/4} \|f - \mu(f)\|_2\}. \tag{9}$$

This condition is the same as (6). Whereas f lying in the ball $\mathscr{B}_{\sigma_{\max}}$ does not guarantee cf lies in that same ball, f lying in the cone $\mathscr{C}_{\kappa_{\max}}$, does guarantee that cf lies in that same cone.

Looking for algorithms that work well for cones of integrands, $\mathscr{C}_{\kappa_{\max}}$, leads one to *adaptive* algorithms. The sample size used to estimate the integral is determined adaptively by first computing an upper bound on $\|f - \mu(f)\|_2$. In information-based complexity theory it is known that adaptive information does not help for convex sets of integrands in the worst case and probabilistic settings (Traub et al, 1988, Chapter 4, Theorem 5.2.1; Chapter 8, Corollary 5.3.1). Here, the cone, $\mathscr{C}_{\kappa_{\max}}$ is not a convex set, so adaption can help.

Again, it should be stressed that the algorithm to be presented here is automatic. It does not require information about $\|f - \mu(f)\|_2 = \sigma$, but this quantity needs to be reliably estimated by the algorithm. Thus, the sample size needed, and consequently the time required, to estimate μ to within the prescribed error tolerance depends on how large $\|f - \mu(f)\|_2 = \sigma$ is estimated to be. The algorithm is adaptive, and its cost depends on the integrand.

1.3 Illustrative Univariate Examples of Automatic Algorithms

Several commonly used software packages have automatic algorithms for integrating functions of a single variable. These include

- `quad` in MATLAB (The MathWorks, Inc., 2012), adaptive Simpson's rule based on `adaptsim` by Gander and Gautschi (2000),
- `quadgk` in MATLAB (The MathWorks, Inc., 2012), adaptive Gauss-Kronrod quadrature based on `quadva` by Shampine (2008), and
- the `chebfun` (Hale et al, 2012) toolbox for MATLAB (The MathWorks, Inc., 2012), which approximates integrals by integrating interpolatory Chebyshev polynomial series for the integrands.

For the first three of these automatic algorithms one can easily probe where they sample the integrand, feed the algorithms zero values, and then construct fooling functions that the automatic algorithms will return a zero value for the integral. Figure 1 displays these fooling functions for the problem $\mu = \int_0^1 f(x)\,\mathrm{d}x$ for the first three algorithms. Each of these algorithms is asked to provide an answer with an absolute error no greater than 10^{14}, but in fact the the absolute error is 1 for these fooling functions. The algorithms `quad` and `chebfun` sample only about a dozen points before concluding that the function is zero, whereas the algorithm `quadgk` samples a much larger number of points (only those between 0 and 0.01 are shown in the plot). Algorithm `NIntegrate` is hard snoop, but it is examined for the next example.

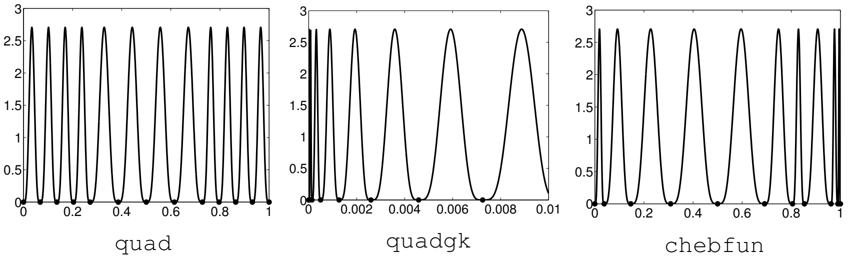


Fig. 1. Plots of fooling functions, f, with $\mu = \int_0^1 f(x)\,\mathrm{d}x = 1$, but for which the corresponding algorithms return values of $\hat{\mu} = 0$.

Accuracy and timing results have been recorded for the test function

$$f(\boldsymbol{x}) = a_0 + b_0 \prod_{j=1}^{d} \left[1 + b_j \exp\left(-\frac{(x_j - h_j)^2}{c_j^2} \right) \right]. \tag{10}$$

¡¡¡¡¡¡¡ HEAD Here $\boldsymbol{x}$ is a d dimensional vector, and the b_j, c_j, and h_j are parameters. For Figure 4 shows the results of different algorithms being used to integrate 500 different instances of f being are performed. For each instance of f, the parameters are chosen as follows: ======= Here $\boldsymbol{x}$ is a d dimensional vector, and the b_j, c_j, and h_j are parameters. For Figure 3 shows the results of different algorithms being used to integrate 500 different instances of f being are performed. For each replication, the parameters determining f are chosen as follows: ¿¿¿¿¿¿¿ change the parameter of bj [0.1 ,10]

- $b_j \in [0.1, 10]$ with $\log(b_j)$ being i.i.d. uniform, $j = 1, \ldots, d$,
- $c_j \in [10^{-6}, 1]$ with $\log(c_j)$ being i.i.d. uniform, $j = 1, \ldots, d$,
- $h_j \in [0, 1]$ with h_j being i.i.d. uniform, $j = 1, \ldots, d$,
- b_0 chosen in terms of the b_j, c_j, and h_j in order to make $\sigma^2(f) \in [10^{-2}, 10^2]$, with $\log(\sigma(f))$ being i.i.d. uniform for each instance, and
- a_0 in terms of chosen in terms of the b_j, c_j, and h_j to make $\mu(f) = 1$.

¿¿¿¿¿¿¿ More changes
Another example is illustrated in Figure 3 for the test function

$$f(\boldsymbol{x}) = b_0 \prod_{j=1}^{d} \left(a_j + b_j e^{-\frac{(x_j - h_j)^2}{c_j^2}} \right) + a_0 \tag{11}$$

where $\boldsymbol{x}$ is the d dimensional vector , a_j is the addition parameter, b_j is the function scale parameter, c_j is the function shape parameter, h_j is the center of the kernel, b_0 is the over multiplicity parameter, and a_0 is the over addition parameter. The exact integral is $\mu = \int_{[0,1]^d} f(\boldsymbol{x})\rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. where $\rho(\boldsymbol{x})$ is the probability density function.

In our numerical experiment, we did 500 replications and choose: =======

¿¿¿¿¿¿¿ change the parameter of bj [0.1 ,10]

- dimension $d = 1$
- the probability density function $\rho = 1$ which is uniformly distributed between $[0, 1]$
- $n_0 = 1024$ for the simple algorithm and $n_0 = 1024 \cdot 128$ for the heavy duty algorithm.
- For the iid sampling, kurtosis max is 9.2085; for the iid heavy duty sampling ,it is 1051.9.

======= ¡¡¡¡¡¡¡ HEAD

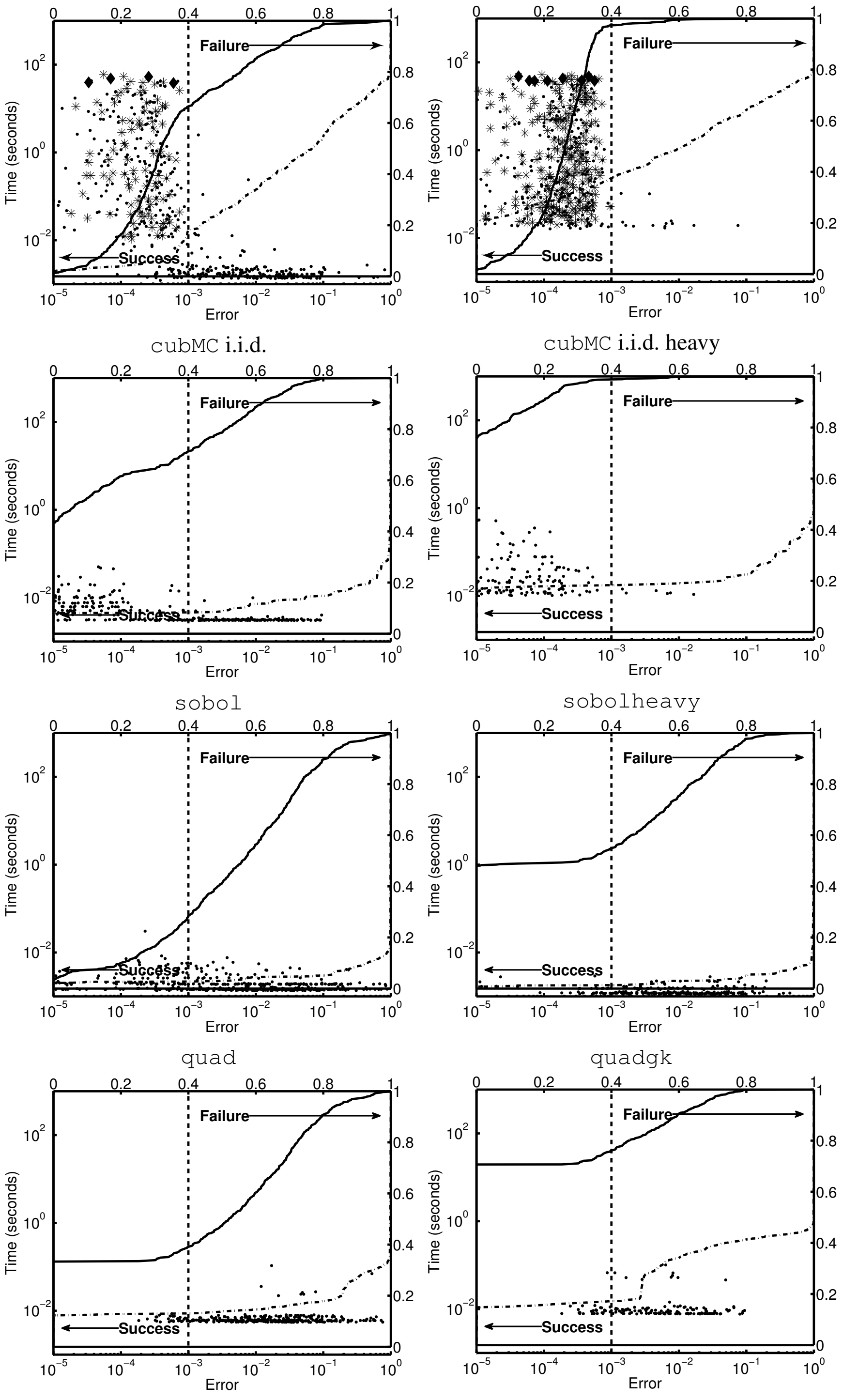
======= These 500 random constructions of f with $d = 1$ are integrated over $[0, 1]$ with $\rho = 1$ (the uniform density function), using `quad`, `quadgk`, and `chebfun`. For the first two of these algorithms, the specified absolute error tolerance is $\varepsilon = 0.001$. The algorithm `chebfun` attempts to do all calculations to near machine precision. The observed error and execution times are plotted in Figure 4. Whereas `chebfun` uses a minimum of $2^3 + 1 = 9$ function values, the figure labeled "`chebfun` heavy duty" displays the results of requiring chebfun to use at least $2^{10} + 1 = 1025$ function values. ¿¿¿¿¿¿¿ More changes

Figure 4 shows that `quad` and `quadgk` are quite fast, nearly always providing an answer in less that 0.01 seconds. Unfortunately, they successfully meet the error tolerance only about 30% of the time for `quad` and 50–60% of the time for `quadgk`. The difficult cases are those where c_1 is quite small, and these algorithms miss the sharp peak. The performance of `chebfun` is similar to that of `quad` and `quadgk`. The heavy duty version of `chebfun` fares somewhat better.

¡¡¡¡¡¡¡ HEAD =======

¿¿¿¿¿¿¿ add example of high dimension ¿¿¿¿¿¿¿ More changes

For the iid sampling, the successful rate is only 70% whereas the heavy duty iid sampling, the successful rate is above 95%. in the graph, the diamond means it reaches the sample limit;



`cubMC` i.i.d.

`cubMC` i.i.d. heavy

`sobol`

`sobolheavy`

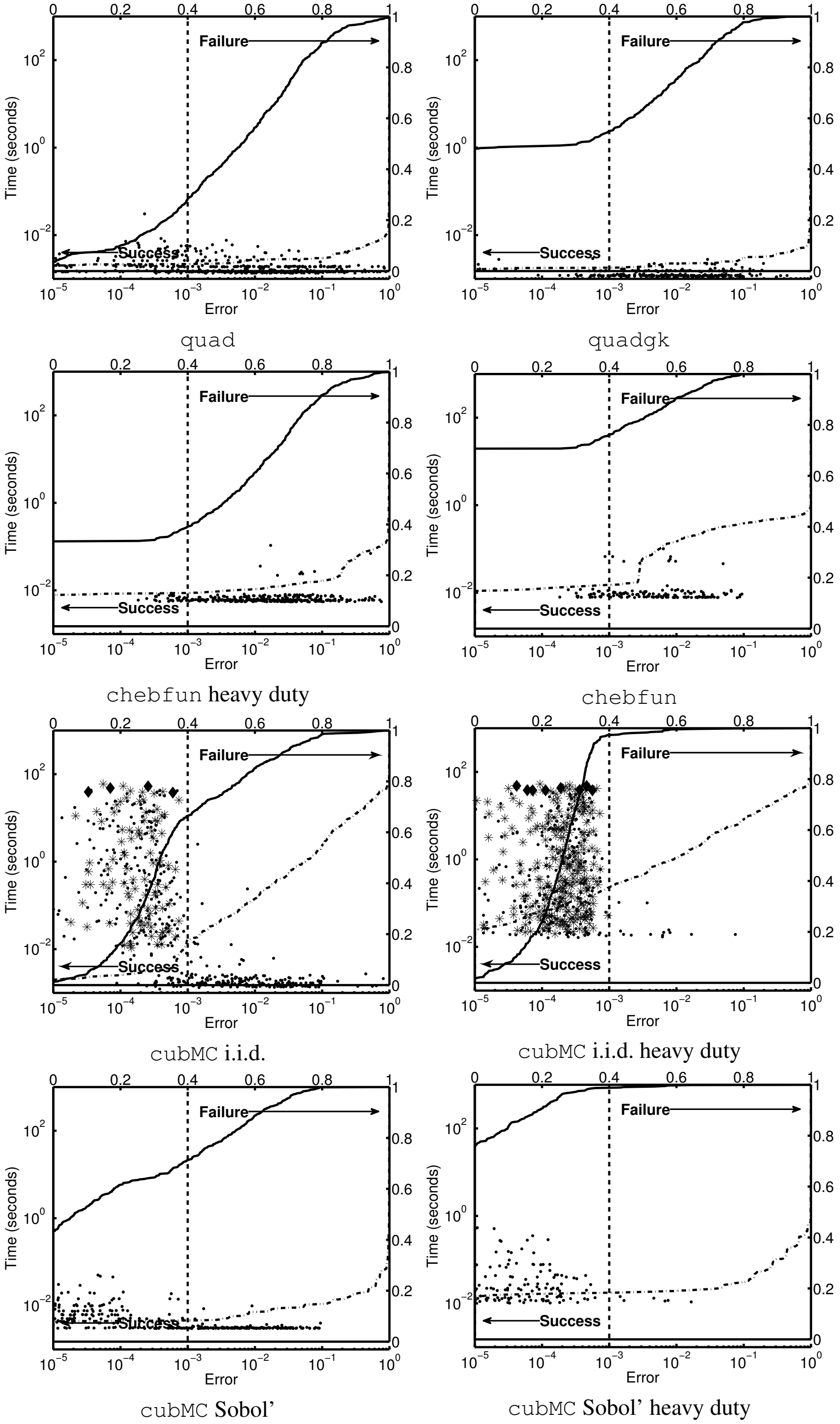
`quad`

`quadgk`

`chebfun`

`chebfunheavy`

Fig. 2. Execution times and errors for test function (1.3) for $d = 1$ and $\varepsilon = 10^{-3}$, and a variety of parameters giving a range of $\sigma(f)$ and $\kappa(f)$. The solid line shows that cumulative distribution of actual errors, and the dot-dashed line shows the cumulative distribution of execution times. For the `cubMC` i.i.d. and i.i.d. heavy duty the points labeled * are those for which the Theorem 1 guarantees the error tolerance.



`quad` `quadgk`

`chebfun` heavy duty `chebfun`

`cubMC` i.i.d. `cubMC` i.i.d. heavy duty

`cubMC` Sobol' `cubMC` Sobol' heavy duty

Fig. 3. Execution times and errors for test function (1.3) for $d = 1$ and $\varepsilon = 10^{-3}$, and a variety of parameters giving a range of $\sigma(f)$ and $\kappa(f)$. The solid line shows that cumulative distribution of actual errors, and the dot-dashed line shows the cumulative distribution of execution times. For the `cubMC` i.i.d. and i.i.d. heavy duty the points labeled * are those for which the Theorem 1 guarantees the error tolerance.

quad-eps-converted-to.pdf

quad

quadgk-eps-converted-to.pdf

quadgk

chebfun-eps-converted-to.pdf

chebfun

chebfunheavy-eps-converted-to.pdf

chebfun heavy duty

iid-eps-converted-to.pdf

cubMC i.i.d.

iidheavy-eps-converted-to.pdf

cubMC i.i.d. heavy duty

sobol-eps-converted-to.pdf

cubMC Sobol'

sobolheavy-eps-converted-to.pdf

cubMC Sobol' heavy duty

the star means the kurtosis is less than the kurtosis maximum. the dot is the general case. For the Sobol sampling the successful rate is around 70% , the heavy duty case, the successful rate is almost 100%. Clearly, Sobol takes less time than iid sampling.For the quad,quadgk, chebfun and chebfun heavy duty, they take less time but could not reach a higher tolerance as we expected.

2 Simple Monte Carlo with Guaranteed Error Estimation

=======

3 Adaptive Monte Carlo with Guaranteed Error Estimation

This section describes an adaptive Monte Carlo algorithm based on i.i.d. sampling that has probabilistic guarantees on satisfying an absolute error tolerance. A software implementation of this algorithm, `cubMC`, is being developed by the authors and their collaborators. Numerical experiments using `cubMC` are shown in Figure 4 above and in Section 5 below. ¿¿¿¿¿¿¿ More changes

3.1 Reliably Bounding the Variance

To estimate or bound the variance of $Y = f(\boldsymbol{X})$ one must have a finite fourth moment. The definitions of the variance and kurtosis of Y in Section 1.1 may be extended naturally to $f \in \mathscr{L}_4$ in terms of $\mathscr{L}_p$ norms defined in (7) as follows:

$$\sigma^2 = \text{var}(Y) = \text{var}(f) = \|f - \mu(f)\|_2^2, \qquad \kappa = \text{kurt}(Y) = \text{kurt}(f) = \frac{\|f - \mu(f)\|_4^4}{\|f - \mu(f)\|_2^4}.$$

Thus, one may speak about Y or f interchangeably. If $1 \le q \le p$, then by Hölder's inequality,

$$\begin{aligned} \|f\|_q &= \left\{ \int_{\mathbb{R}^d} |f(\boldsymbol{x})|^q \rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \right\}^{1/q} \\ &\le \left\{ \int_{\mathbb{R}^d} |f(\boldsymbol{x})|^p \rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \right\}^{1/p} \left\{ \int_{\mathbb{R}^d} 1^{p/(p-q)} \rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \right\}^{(p-q)/(pq)} = \|f\|_p . \end{aligned}$$

Thus, $\kappa = \text{kurt}(f) \ge 1$, provided $\sigma^2 = \text{var}(f) > 0$. For $\sigma^2 = \text{var}(f) = 0$, one defines $\kappa = \text{kurt}(f) = 1$.

As mentioned in (4), a practical upper bound on σ^2 may be obtained in terms of the sample variance. The justification for this statement is contained in the lemma below. Two well-known probability inequalities that are needed here and later are quoted in the following lemma.

Lemma 1. *Suppose that Z is any random variable with finite second moment. For any $a > 0$,*

$$\text{Prob}\left[|Z - E(Z)| \ge a\right] \le \frac{\text{var}(Z)}{a^2} \qquad \textit{Chebyshev's inequality (Lin and Bai, 2010, 6.1c)},$$

$$\text{Prob}[Z - E[Z] \ge a] \le \frac{\text{var}(Z)}{a^2 + \text{var}(Z)} \qquad \textit{Cantelli's Inequality (Lin and Bai, 2010, 6.1e)}.$$

Lemma 2. *Suppose that Y is a random variable with finite fourth moment, and with mean μ, variance σ^2, and kurtosis κ. Let $\hat{v}_n$ denote the unbiased sample variance based on n i.i.d. samples, $Y_1,\ldots,Y_n$ as defined in* (4). *Then for any $\alpha \in (0,1]$ it follows that*

$$\operatorname{Prob}\left[\frac{\hat{v}_n}{1-\sqrt{\left(\kappa-\frac{n-3}{n-1}\right)\left(\frac{1-\alpha}{\alpha n}\right)}} > \sigma^2\right] \geq 1-\alpha, \tag{12a}$$

$$\operatorname{Prob}\left[\frac{\hat{v}_n}{1+\sqrt{\left(\kappa-\frac{n-3}{n-1}\right)\left(\frac{1-\alpha}{\alpha n}\right)}} < \sigma^2\right] \geq 1-\alpha. \tag{12b}$$

Proof. It is known that the sample variance is an unbiased estimator of the variance, i.e., $E(\hat{v}_n) = \sigma^2$. It is also known from (Miller, 1986, (7.16)) that the variance of the sample variance can be expressed in terms of σ^2 and κ as

$$\operatorname{var}(\hat{v}_n) = \frac{\sigma^4}{n}\left(\kappa - \frac{n-3}{n-1}\right).$$

Choosing

$$a = \sqrt{\operatorname{var}(\hat{v}_n)\frac{1-\alpha}{\alpha}} = \sigma^2\sqrt{\left(\kappa-\frac{n-3}{n-1}\right)\left(\frac{1-\alpha}{\alpha n}\right)} > 0,$$

it follows from Cantelli's inequality (Lemma 1) that

$$\begin{aligned}\operatorname{Prob}\left[\hat{v}_n - \sigma^2 \geq \sigma^2\sqrt{\left(\kappa-\frac{n-3}{n-1}\right)\left(\frac{1-\alpha}{\alpha n}\right)}\right] &= \operatorname{Prob}\left[\hat{v}_n - \sigma^2 \geq a\right] \\ &\leq \frac{\operatorname{var}(\hat{v}_n)}{a^2+\operatorname{var}(\hat{v}_n)} = \frac{\operatorname{var}(\hat{v}_n)}{\operatorname{var}(\hat{v}_n)\frac{1-\alpha}{\alpha}+\operatorname{var}(\hat{v}_n)} = \frac{1}{\left(\frac{1-\alpha}{\alpha}\right)+1} = \alpha.\end{aligned}$$

Then (12a) follows directly. By a similar argument, applying Cantelli's inequality to the expression $\operatorname{Prob}\left[-\hat{v}_n + \sigma^2 \geq a\right]$ implies (12b). □

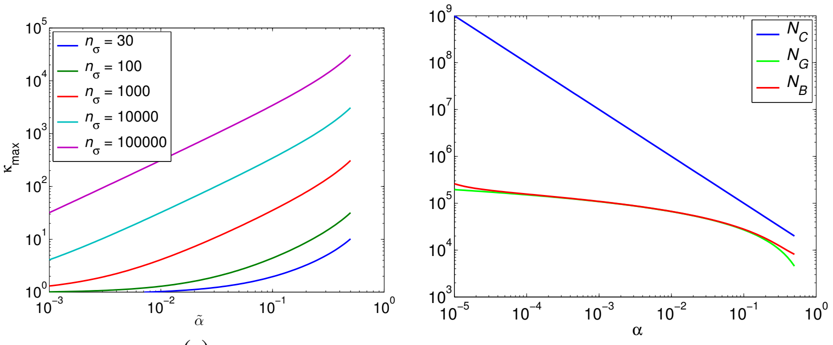
Lemma 2 provides probabilistic justification to use $\hat{\sigma}^2 = \mathfrak{C}^2\hat{v}_{n_\sigma}$ as a reliable upper bound for $\sigma^2 = \operatorname{var}(Y) = \operatorname{var}(f)$. One can claim that $\operatorname{Prob}(\hat{\sigma}^2 \geq \sigma^2) \geq 1-\alpha$ assuming that

$$\begin{aligned}&\frac{1}{1-\sqrt{\left(\kappa-\frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\alpha}{\alpha n_\sigma}\right)}} \leq \mathfrak{C}^2 \\ &\iff \kappa \leq \frac{n_\sigma-3}{n_\sigma-1} + \left(\frac{\alpha n_\sigma}{1-\alpha}\right)\left(1-\frac{1}{\mathfrak{C}^2}\right)^2 =: \kappa_{\max}(\alpha, n_\sigma, \mathfrak{C}).\end{aligned} \tag{13}$$

Figure 5a shows how large a kurtosis can be accommodated for a given n_σ, α, and variance inflation factor $\mathfrak{C} = 1.5$. Note that for $n = 30$, a common rule of thumb for applying the central limit theorem, even $\alpha = 0.1$ gives $\kappa_{\max}$ of only about 2, which is rather restrictive.

3.2 Determining the Sample Size

The other issue that needs to be addressed is a tight probabilistic error bound to replace the asymptotic or approximate error bound given by the Central Limit Theorem, (3). Chebyshev's



(a)

(b)

Fig. 5. (a) The maximum kurtosis, $\kappa_{\max}(\alpha, n_\sigma, 1.5)$, as defined in (13); (b) comparison of sample sizes $N_G(0.01,\alpha)$, $N_C(0.01,\alpha)$, and $N_B(0.01,\alpha,\kappa_{\max}^{3/4}(\alpha,1000,1.5))$.

inequality implies that the number of function evaluations needed to ensure that $\hat{\mu}_n$ satisfies the error tolerance with high probability is

$$\text{Prob}\left[|\hat{\mu}_n - \mu| \le \varepsilon\right] \ge 1-\alpha \quad \text{for } n \ge N_C(\varepsilon/\sigma,\alpha),\ f \in \mathscr{L}_2, \tag{14a}$$

where

$$N_C(\varepsilon,\alpha) := \left\lceil \frac{1}{\alpha\varepsilon^2} \right\rceil. \tag{14b}$$

However, this sample size is much larger than that given by the Central Limit Theorem, as shown in Figure 5b.

Since higher order moments of the integrand are required to guarantee an upper bound on the true variance in terms of the sample variance, it is sensible to use these higher order moments to obtain smaller sample sizes. A smaller sample size than (14b) with a rigorous probabilistic bound can be found by invoking the following inequality.

Lemma 3 (Non-uniform Berry-Esseen Inequality). *(Petrov, 1995, Theorem 5.16, p. 168) Let $Y_1,\ldots,Y_n$ be i.i.d. random variables. Suppose that $\mu = E(Y_i)$, $\operatorname{var}(Y_i) = \sigma^2 > 0$, and $M = E\left|Y_i - \mu\right|^3/\sigma^3 < \infty$. Then*

$$\left|\text{Prob}\left[\frac{1}{\sigma\sqrt{n}}\sum_{i=1}^{n}(Y_i-\mu) < x\right] - \Phi(x)\right| \le \frac{AM}{\sqrt{n}(1+|x|)^3}.$$

for all x, where Φ is the cumulative distribution function of a standard Gaussian (normal) random variable, and A is some number satisfying $0.4097 \le A \le 0.5600$.

The right hand side of this inequality bounds how far the probability distribution of sample mean deviates from the Central Limit Theorem approximation. This right hand side vanishes as either the sample size or $|x|$ tends to infinity. This inequality requires that $Y = f(\boldsymbol{X})$, have a finite third moment, i.e., $f \in \mathscr{L}_3$, which is a weaker assumption than is needed to bound the variance in terms of the sample variance in Lemma 2. Recalling that $Y_i = f(\boldsymbol{X}_i)$, $\mu = E(Y_i)$, and $\hat{\mu}_n = (Y_1 + \cdots + Y_n)/n$, this Berry-Esseen inequality then implies that for positive x,

$$\begin{aligned}\text{Prob}\left[|\hat{\mu}_n - \mu| < \frac{\sigma x}{\sqrt{n}}\right] &= \text{Prob}\left[\hat{\mu}_n - \mu < \frac{\sigma x}{\sqrt{n}}\right] - \text{Prob}\left[\hat{\mu}_n - \mu < -\frac{\sigma x}{\sqrt{n}}\right] \\ &\ge \left[\Phi(x) - \frac{0.56M}{\sqrt{n}(1+|x|)^3}\right] - \left[\Phi(-x) + \frac{0.56M}{\sqrt{n}(1+|x|)^3}\right] \\ &= 1 - 2\left(\Phi(-x) - \frac{0.56M}{\sqrt{n}(1+|x|)^3}\right). \qquad (15)\end{aligned}$$

Letting $x = \varepsilon\sqrt{n}/\sigma$, the probability of making an error less than ε is bounded below by $1-\alpha$, i.e.,

$$\text{Prob}[|\hat{\mu}_n - \mu| < \varepsilon] \ge 1-\alpha, \quad \text{provided } n \ge N_B(\varepsilon/\sigma, \alpha, M),\ f \in \mathscr{L}_3, \qquad (16a)$$

where

$$N_B(b,\alpha,M) := \min\left\{ n \in \mathbb{N} : \Phi\left(-b\sqrt{n}\right) + \frac{0.56M}{\sqrt{n}\left(1+b\sqrt{n}\right)^3} \le \frac{\alpha}{2} \right\}. \qquad (16b)$$

As shown in Figure 5b, for a range of α, the sample size guaranteeing coverage of the confidence interval, N_B, is quite close to the sample size for the approximate Central Limit Theorem confidence interval, N_G, however, N_B may be somewhat larger for very small or rather large α. In general N_B is significantly smaller than N_C, but not always. A disadvantage of (16) is that class of integrands, $\mathscr{L}_3$, is smaller than that in (14), but this typically a small price to pay given the much smaller cost of computation.

The theorem below combines the results on estimating the variance with the sample sizes arising from Chebyshev's inequality and the Berry-Esseen inequality. These lead to an adaptive Monte Carlo algorithm with a probabilistic error guarantee.

Theorem 1. *Specify the following parameters defining the algorithm:*

- *sample size for variance estimation, $n_\sigma \in \mathbb{N}$,*
- *inflation factor for variance estimation, $\mathfrak{C} \in (1,\infty)$,*
- *uncertainty tolerance, $\alpha \in (0,1)$, and $\tilde{\alpha} = 1-\sqrt{1-\alpha}$, and*
- *absolute error tolerance, $\varepsilon \in (0,\infty)$.*

For $\kappa_{\max} = \kappa_{\max}(n_\sigma, \tilde{\alpha}, \mathfrak{C})$, as defined in (13)*, define the cone of integrands functions with bounded kurtosis, $\mathscr{C}_{\kappa_{\max}}$, according to* (9)*. For any $f \in \mathscr{C}_{\kappa_{\max}}$, compute the sample variance, $\hat{v}_{n_\sigma}$ using a simple random sample of size n_σ. Use this to approximate the variance of f by $\hat{\sigma}^2 = \mathfrak{C}^2\hat{v}_{n_\sigma}$ as in* (4)*. Next define a sample size $n = N_{CB}(\varepsilon/\hat{\sigma}, \tilde{\alpha}, \kappa_{\max}^{3/4})$, where*

$$N_{CB}(b,\alpha,M) := \min\left(N_C(b,\alpha), N_B(b,\alpha,M)\right), \qquad (17)$$

N_C is defined in (14b) *and N_B is defined in* (16b)*. Compute $\hat{\mu}_n$, the simple Monte Carlo estimator of μ based on n samples, as in* (5)*. A probabilistic absolute error bound is given by*

$$\text{Prob}\left[|\hat{\mu}_n - \mu| \le \varepsilon\right] \ge 1-\alpha.$$

Proof. Note that (1) implies that the third moment can be bounded in terms of the fourth moment, namely M in Lemma 3 is no greater than $\kappa^{3/4}$. There are three primary random variables in this algorithm: the estimated upper bound on the standard deviation, $\hat{\sigma}$, the sample size to estimate the mean, n, which is an explicit function of $\hat{\sigma}$, and the estimated mean, $\hat{\mu}_n$. By (14) and (16) it then follows that $\text{Prob}\left[|\hat{\mu}_n - \mu| \le \varepsilon\right] \ge 1-\tilde{\alpha}$, provided that $\hat{\sigma} \ge \sigma$. Thus,

$$\begin{aligned}\operatorname{Prob}\left[|\hat{\mu}_n-\mu|\le\varepsilon\right] &= E_{\hat{\sigma}}\left\{\operatorname{Prob}\left[|\hat{\mu}_n-\mu|\le\varepsilon \mid \hat{\sigma}\right]\right\}\\ &\ge E_{\hat{\sigma}}\left\{(1-\tilde{\alpha})1_{[\sigma,\infty)}(\hat{\sigma})\right\}\\ &\ge (1-\tilde{\alpha})(1-\tilde{\alpha}) = 1-\alpha,\end{aligned}$$

since $\hat{\sigma}\ge\sigma$ with probability $1-\tilde{\alpha}$ by (13). □

Remark 1. If one is willing to invest n_σ samples to estimate σ, it makes practical sense to to choose the sample size for the sample mean at least that large, i.e,

$$n = \max(n_\sigma, N_{CB}(\varepsilon/\hat{\sigma}, \tilde{\alpha}, \kappa_{\max}^{3/4})).$$

By the error bound following from Chebyshev's inequality, (14), this means that the probabilistic absolute error bound in Theorem 1 also holds for integrands, f, lying in the ball $\mathscr{B}_{\sigma_{\max}}$, defined in (8), where $\sigma_{\max} = \varepsilon\sqrt{\alpha n_\sigma}$.

3.3 Cost of the Algorithm

The sample size of the adaptive algorithm defined in Theorem 1 is a random variable, and so the cost of this algorithm might best be defined defined probabilistically. Moreover, the cost depends strongly on σ as well as the ε, and its definition should reflect this dependence.

Let A be any random algorithm defined for a set of integrands $\mathscr{F}$ that takes as its input an error tolerance, ε, an uncertainty level, α, and a procedure for computing values of $f\in\mathscr{F}$. The algorithm then computes an approximation to the integral, $A(f,\varepsilon,\alpha)$. This approximation is based solely on $f(\boldsymbol{x}_1),\ldots,f(\boldsymbol{x}_n)$, where the choice of each $\boldsymbol{x}_i$ may depend iteratively on $(\boldsymbol{x}_1,f(\boldsymbol{x}_1)),\ldots,(\boldsymbol{x}_{i-1},f(\boldsymbol{x}_{i-1}))$, and the decision to stop with n data depends on all function data collected, up to and including the n^{th}. The cost of this algorithm for this set of inputs is denoted $\operatorname{cost}(A,\varepsilon,\alpha,f)$, is the number of function values, n, which as noted is a random variable. The probabilistic cost of the algorithm, with uncertainty β, for integrands of variance no greater than $\sigma_{\max}^2$ is defined as

$$\operatorname{cost}(A,\varepsilon,\alpha,\beta,\mathscr{F},\sigma_{\max}) := \sup_{\substack{f\in\mathscr{F}\\ \operatorname{var}(f)\le\sigma^2}} \min\left\{N : \operatorname{Prob}[\operatorname{cost}(A,\varepsilon,\alpha,f)\le N]\ge 1-\beta\right\}. \quad (18)$$

The cost of the particular adaptive Monte Carlo algorithm defined in Theorem 1, denoted `cubMC`, for the cone of integrands $\mathscr{C}_{\kappa_{\max}}$ is

$$\operatorname{cost}(\texttt{cubMC},\varepsilon,\alpha,\beta,\mathscr{C}_{\kappa_{\max}},\sigma_{\max}) := \sup_{\substack{f\in\mathscr{C}_{\kappa_{\max}}\\ \operatorname{var}(f)\le\sigma_{\max}^2}} \min\left\{N : \operatorname{Prob}(n_\sigma+n\le N)\ge 1-\beta\right\}. \quad (19)$$

Since n_σ is fixed, bounding this cost depends on bounding n, which depends on $\hat{\sigma}$ as given by Theorem 1. Moreover, $\hat{\sigma}$ can be bounded above using (12b) in Lemma 2. For $f\in\mathscr{C}_{\kappa_{\max}}$,

$$\begin{aligned}1-\beta &\le \operatorname{Prob}\left[\hat{v}_{n_\sigma} < \sigma^2\left\{1+\sqrt{\left(\kappa-\frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\beta}{\beta n_\sigma}\right)}\right\}\right]\\ &\le \operatorname{Prob}\left[\hat{\sigma}^2 = \mathfrak{C}^2\hat{v}_{n_\sigma} < \mathfrak{C}^2\sigma^2\left\{1+\sqrt{\left(\kappa_{\max}(n_\sigma,\tilde{\alpha},\mathfrak{C})-\frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\beta}{\beta n_\sigma}\right)}\right\}\right]\\ &= \operatorname{Prob}\left[\hat{\sigma}^2 < \sigma^2\gamma^2(\tilde{\alpha},\beta,\mathfrak{C})\right],\end{aligned}$$

where

$$\gamma^2(\tilde{\alpha},\beta,\mathfrak{C}) := \mathfrak{C}^2 \left\{ 1 + \sqrt{\left(\frac{\tilde{\alpha}}{1-\tilde{\alpha}}\right)\left(\frac{1-\beta}{\beta}\right)\left(1-\frac{1}{\mathfrak{C}^2}\right)^2} \right\} > 1.$$

Noting that $N_{CB}(\cdot,\alpha,M)$ is a non-increasing function allows one to derive the following upper bound on the cost of the adaptive Monte Carlo algorithm.

Theorem 2. *The adaptive Monte Carlo algorithm based on i.i.d. sampling described in Theorem 1, denoted* `cubMC`*, has a probabilistic cost bounded above by*

$$\mathrm{cost}(\texttt{cubMC},\varepsilon,\alpha,\beta,\mathscr{C}_{\kappa_{\max}},\sigma_{\max}) \le n_\sigma + N_{CB}(\varepsilon/(\sigma_{\max}\gamma(\tilde{\alpha},\beta,\mathfrak{C})),\tilde{\alpha},\kappa_{\max}^{3/4}).$$

The cost of the adaptive Monte Carlo algorithm `cubMC` is roughly proportional to $\sigma^2_{\max}\varepsilon^{-2}$. The set $\mathscr{C}_{\kappa_{\max}}$ contains integrands with arbitrarily large variance, $\sigma^2 = \mathrm{var}(f)$, and thus with potentially arbitrarily large algorithmic cost. On the other hand, since the algorithm is adaptive, the cost may be small if σ^2 is small. The upper bound in Theorem 2 certainly scales with the $\sigma^2_{\max}$ as one might hope if $\sigma^2_{\max}$ were known. The variable cost of the algorithm for integrands in $\mathscr{C}_{\kappa_{\max}}$ is actually an advantage, rather than a drawback, of this analysis. One need not make any a priori assumptions about the size of the integrand, σ, only about its kurtosis, κ, which is unchanged when the integrand is multiplied by an arbitrary nonzero constant.

¡¡¡¡¡¡¡ HEAD

4 More Numerical Examples

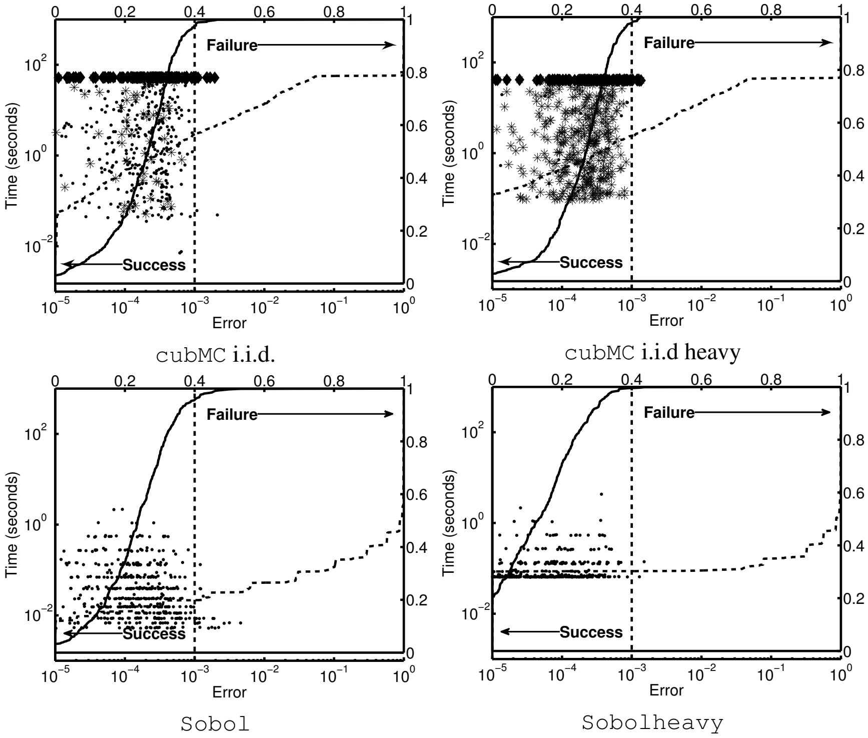
We are using the function 1.3 and the parameters on page 5 to test the higher dimension case, set the dimension d be random between 2 to 8, then we have the graphs below: =======

5 Numerical Examples with `cubMC`

Figure 4 shows timing and observed errors for the adaptive algorithm, `cubMC`, with i.i.d. sampling, as described in the previous section. The parameters chosen are $\varepsilon = 0.001$, $\alpha = 5\%$, and $\mathfrak{C} = 1.5$. For the plot on the left, $n_\sigma = 2^{10} = 1024$, which corresponds to $\kappa_{\max} = 9.2$. For the heavy duty plot on the right, $n_\sigma = 2^{17} = 131\,072$, which corresponds to $\kappa_{\max} = 1052$. In both of these plots the points labeled with a * are those for which $\kappa(f) \le \kappa_{\max}$ and so Theorem 1 guarantees that the answer is correct $1-\alpha = 95\%$ of the time. For these plots all of the points labeled * fall within the prescribed error tolerance. For `cubMC` i.i.d. heavy duty plot $\kappa_{\max}$ is larger, so there are more points for which the guarantee holds. Those points labeled with a dot, are those for which $\kappa(f) > \kappa_{\max}$, and so no guarantee holds. The points labeled with a square are those for which `cubMC` attempts to exceed the cost budget, i.e., it wants to choose n such that $(n_\sigma + n)d > N_{\max} = 10^9$.

The `cubMC` algorithm performs somewhat more robustly than `quad`, `quadgk`, and `chebfun`, because `cubMC` does requires a very low degree of smoothness and makes a fairly large minimum sample. The more important point is that `cubMC` has a guarantee, where to our knowledge, the other routines do not.

Figure 4 also exhibits the results of using `cubMC` with scrambled Sobol' sampling (Owen, 1995, 1997a,b; Matoušek, 1998; Hong and Hickernell, 2003; Dick and Pillichshammer, 2010),



cubMC i.i.d.

cubMC i.i.d heavy

Sobol

Sobolheavy

Fig. 6. Execution times and errors for test function (1.3) for $d = 7$ and $\varepsilon = 10^{-3}$, and a variety of parameters giving a range of $\sigma(f)$ and $\kappa(f)$. The solid line shows that cumulative distribution of actual errors, and the dot-dashed line shows the cumulative distribution of execution times. For the cubMC i.i.d. and i.i.d. heavy duty the points labeled * are those for which the Theorem 1 guarantees the error tolerance.

i.e., the sample mean, $\hat{\mu}_n$ is based on sampling the integrand on a Sobol' net with n points. Since the points of the Sobol' net are purposefully correlated, the error of $\hat{\mu}_n$ does not depend on $\mathrm{var}(f)$, but on some measure of variation of f (see (Owen, 1995, 1997a,b; Dick and Pillichshammer, 2010)) that is difficult to estimate in practice. Tony Warnock and John Halton proposed the following error bound estimate, called the quasi-standard error (**??**):

$$\mathrm{qse}(f;m) = \sqrt{\frac{1}{m(m-1)} \sum_{j=1}^{m} [\hat{\mu}_n^{(j)} - \hat{\mu}_n]^2}. \tag{20}$$

where $\hat{\mu}^{(j)}$ corresponds to the sample mean of the function values for the j^{th} partition out of the Sobol' net. In cubMC the error of $\hat{\mu}_n$ is assumed to be no greater than $\mathfrak{C}\mathrm{qse}(f;m)$ with $m = 8$ and $\mathfrak{C} = 1.5$. The number of samples, n, is doubled until $\mathfrak{C}\mathrm{qse}(f;m) \le \varepsilon$. Unfortunately, there is no theory yet that intuitively describes the cone of integrands for which this stopping criterion guarantees that the error tolerance is met. This is an area of ongoing research.

For the iid sampling, the successful rate is only 70% whereas the heavy duty iid sampling, the successful rate is above 95%. in the graph, the square means it reaches the sample limit;

the star mean the kurtosis is less than the kurtosis maximum. the dot is the general case. For the Sobol sampling the successful rate is around 70% , the heavy duty case, the successful rate is almost 100%. Clearly, Sobol takes less time than iid sampling.For the quad,quadgk, chebfun and chebfun heavy duty, they take less time but could not reach a higher tolerance as we expected.

¿¿¿¿¿¿¿ More changes

6 A General Error Criterion

In many practical situations, one needs to approximate the integral with a certain relative accuracy. For example, one wants an answer that is correct to three significant digits. In this case, given a tolerance, ε, and a significance level α, with $\varepsilon, \alpha \in (0,1)$, one seeks a random $\tilde{\mu}$ such that

$$\operatorname{Prob}\left[\left|\frac{\tilde{\mu}-\mu}{\mu}\right| \le \varepsilon\right] \ge 1-\alpha.$$

A more general form of this criterion would be

$$\operatorname{Prob}\left[\frac{|\tilde{\mu}-\mu|}{1-\theta+\theta\,|\mu|} \le \varepsilon\right] \ge 1-\alpha. \tag{21}$$

for some fixed $\theta \in [0,1]$, where $\theta = 0$ corresponds to absolute error, and $\theta = 1$ corresponds to relative error. Clearly, one must have $(1-\theta)+|\mu| \neq 0$ for such a statement to be possible.

If $\varepsilon_A \ge 0$ is an absolute error tolerance, and $\varepsilon_R \ge 0$ is a relative error tolerance, then letting

$$\varepsilon = \frac{\varepsilon_A \varepsilon_R}{\theta\varepsilon_A + (1-\theta)\varepsilon_R},$$

it follows that for all $\theta \in [0,1]$,

$$[1-\theta+\theta\,|\mu|]\varepsilon = (1-\gamma)\varepsilon_A + \gamma\varepsilon_R\,|\mu| \le \max(\varepsilon_A, \varepsilon_R\,|\mu|),$$

where

$$\gamma = \frac{\theta\varepsilon_A}{\theta\varepsilon_A + (1-\theta)\varepsilon_R} \in [0,1].$$

Thus, error criterion (21) implies that one has satisfied either an absolute or a relative error criterion,

$$\operatorname{Prob}\left[\frac{|\tilde{\mu}-\mu|}{1-\theta+\theta\,|\mu|} \le \varepsilon\right] \ge 1-\alpha \implies \operatorname{Prob}\left[|\tilde{\mu}-\mu| \le \varepsilon_A \text{ or } |\tilde{\mu}-\mu| \le |\mu|\,\varepsilon_R\right] \ge 1-\alpha.$$

A value of γ close to zero implies a preference to fulfill the absolute error criterion while a value of γ close to one implies a preference to fulfill the relative error criterion.

Obtaining a confidence interval of the form (21), proceeds in three stages: i) obtaining an upper bound on σ^2, ii) obtaining a lower bound on $1-\theta+\theta\,|\mu|$, and iii) then using these to obtain (21). What differs from the absolute error case is step ii). For this step it is noted that

$$\begin{aligned}&\operatorname{Prob}[|\hat{\mu}-\mu| \le \hat{\varepsilon}] \ge 1-\alpha \implies \operatorname{Prob}[\max(|\hat{\mu}|-\hat{\varepsilon},0) \le |\mu| \le |\hat{\mu}|+\hat{\varepsilon}] \ge 1-\alpha \\ &\implies \operatorname{Prob}[1-\theta+\theta\max(|\hat{\mu}|-\hat{\varepsilon},0) \le 1-\theta+\theta\,|\mu| \le 1-\theta+\theta(|\hat{\mu}|+\hat{\varepsilon})] \ge 1-\alpha. \end{aligned} \tag{22}$$

Although one might be happy the left side of this inequality being positive, if it is too much smaller than the right side, then one might be eventually expending too much extra work in step iii). Thus, it makes sense to require

$$\begin{gathered}1-\theta+\theta\max(|\hat{\mu}|-\hat{\varepsilon},0) \geq \hat{\delta}[1-\theta+\theta(|\hat{\mu}|+\hat{\varepsilon})] \\ \iff \hat{\varepsilon}=\begin{cases}\dfrac{(1-\hat{\delta})(1-\theta)}{\hat{\delta}\theta}-|\hat{\mu}|, & 0 \leq |\hat{\mu}| < \dfrac{(1-\hat{\delta})(1-\theta)}{2\hat{\delta}\theta}, \\ \dfrac{1-\hat{\delta}}{1+\hat{\delta}}\left[\dfrac{1-\theta}{\theta}+|\hat{\mu}|\right], & \dfrac{(1-\hat{\delta})(1-\theta)}{2\hat{\delta}\theta} \leq |\hat{\mu}| < \infty.\end{cases}\end{gathered}$$

This is done iteratively in the algorithm described in Theorem 3 below. One needs to prevent $\hat{\varepsilon}$ from becoming too small. This means that $\hat{\delta}$ should kept away from 1, which means that the lower bound on $1-\theta+\theta|\mu|$ is allowed to be somewhat smaller than the upper bound. Preventing $\hat{\varepsilon}$ from becoming too small also means that $1-\theta+|\hat{\mu}|$ cannot be too small. This may be unavoidable if one is interested in relative error $\theta=1$, and the true answer, μ, is small.

Some notation is needed for this theorem. For any fixed $\alpha \in (0,1)$, and $M>0$, define the inverse of the functions $N_C(\cdot,\alpha)$, $N_B(\cdot,\alpha,M)$, and $N_{CB}(\cdot,\alpha,M)$,

$$\begin{gathered}N_C^{-1}(n,\alpha) := \frac{1}{\sqrt{n\alpha}}, \\ N_B^{-1}(n,\alpha,M) := \min\left\{b>0 : \Phi\left(-b\sqrt{n}\right)+\frac{0.56M}{\sqrt{n}\left(1+b\sqrt{n}\right)^3} \leq \frac{\alpha}{2}\right\}, \\ N_{CB}^{-1}(n,\alpha,M) := \min(N_C^{-1}(n,\alpha), N_B^{-1}(n,\alpha,M)).\end{gathered}$$

It then follows then by Chebyshev's inequality and the Berry-Esseen Inequality (see (15)) that

$$\operatorname{Prob}[|\hat{\mu}_n-\mu|<\hat{\varepsilon}] \geq 1-\alpha, \quad \text{provided } f \in \mathscr{C}_{\kappa_{\max}}, \text{ where } \hat{\varepsilon}=\sigma(f)N_{CB}^{-1}(n,\alpha,\kappa_{\max}^{3/4}),$$

and $\sigma(f)=\sqrt{\operatorname{var}(f)}$ is the standard deviation of the integrand. Given a significance level, $\alpha \in (0,1)$, let $\alpha_\sigma, \alpha_\mu, \alpha_1, \alpha_2, \ldots$ be an infinite sequence of positive numbers all less than one, such that

$$(1-\alpha_\sigma)(1-\alpha_\mu)(1-\alpha_1)(1-\alpha_2)\cdots = 1-\alpha. \tag{23}$$

For example, one might choose $\alpha_\sigma, \alpha_\mu$, and $\hat{\alpha}$ such that $(1-\alpha_\sigma)(1-\alpha_\mu)(1-\hat{\alpha})=1-\alpha$, and then

$$\alpha_i = 1-(1-\hat{\alpha})^{(a-1)a^{-i}}, \; i \in \mathbb{N}, \quad \text{where } a \in (1,\infty). \tag{24}$$

Theorem 3. *Specify the following parameters defining the algorithm:*

- *sample size for variance estimation, $n_\sigma \in \mathbb{N}$,*
- *initial sample size for mean estimation, $n_1 \in \mathbb{N}$,*
- *variance inflation factor for variance estimation, $\mathfrak{C} \in (1,\infty)$,*
- *factors for Step 2, $\hat{\delta}, \delta, \tilde{\delta} \in (0,1)$, with $\delta < \tilde{\delta}$.*
- *uncertainty tolerance, $\alpha \in (0,1)$, and a sequence $\alpha_\sigma, \alpha_\mu, \alpha_1, \alpha_2, \ldots$ satisfying* (23)*,*
- *the parameter $\theta \in [0,1]$, used to define the general error criterion* (21)*, and*
- *the error tolerance, $\varepsilon > 0$.*

Let $\kappa_{\max}=\kappa_{\max}(n_\sigma,\alpha_\sigma,\mathfrak{C})$ as defined in (13)*. For any f lying in the cone of functions with bounded kurtosis, $\mathscr{C}_{\kappa_{\max}}$, do the following:*

1. **Bounding the variance of the integrand from above.** *Compute the sample variance, $\hat{v}_{n_\sigma}$ using a simple random sample of size n_σ. Use this to approximate the variance of f by $\hat{\sigma}^2 = \mathfrak{C}^2 \hat{v}_{n_\sigma}$, as in* (4)*. Compute the width of initial the confidence interval for the mean, $\hat{\varepsilon}_1 = \hat{\sigma} N_{CB}^{-1}(n_1, \alpha_1, \kappa_{\max}^{3/4})$.*

2. **Bounding the denominator in the error criterion from below.** *For $i = 1, 2, \ldots$, do the following:*

a) Compute the sample average $\hat{\mu}_{n_i}$ using a simple random sample that is independent of those used to compute $\hat{v}_{n_\sigma}$ and $\hat{\mu}_{n_1}, \ldots, \hat{\mu}_{n_{i-1}}$.

b) Compute $\mathfrak{c} = 1 - \theta + \theta \max(|\hat{\mu}_{n_i}| - \hat{\varepsilon}_i, 0)$, a confident lower bound on $1 - \theta + \theta\,|\mu|$, according to (22)*. If $\mathfrak{c} \geq \delta[1 - \theta + \theta(|\hat{\mu}_{n_i}| + \hat{\varepsilon})]$, then $\mathfrak{c}$ is large enough. Set $\tau = i$ and go to Step 3.*

c) Else, compute the next tolerance for the sample mean

$$\hat{\varepsilon}_0 = \begin{cases} \dfrac{(1-\hat{\delta})(1-\theta)}{\hat{\delta}\theta} - |\hat{\mu}_{n_i}|, & 0 \leq |\hat{\mu}_{n_i}| < \dfrac{(1-\hat{\delta})(1-\theta)}{2\hat{\delta}\theta}, \\ \dfrac{1-\hat{\delta}}{1+\hat{\delta}} \left[\dfrac{1-\theta}{\theta} + |\hat{\mu}_{n_i}| \right], & \dfrac{(1-\hat{\delta})(1-\theta)}{2\hat{\delta}\theta} \leq |\hat{\mu}_{n_i}| < \infty, \end{cases}$$

$$\hat{\varepsilon}_{i+1} = \max(\min(\hat{\varepsilon}_0, \tilde{\delta}\hat{\varepsilon}_i), \delta\hat{\varepsilon}_i).$$

d) Define the next sample size, $n_{i+1} = N_{CB}(\hat{\varepsilon}_{i+1}/\hat{\sigma}, \alpha_{i+1}, \kappa_{\max}^{3/4})$, increase i by one, and go to step a).

3. **Computing the sample mean to sufficient accuracy.** *Compute the sample size $n = N_{CB}(\mathfrak{c}\varepsilon/\hat{\sigma}, \alpha_\mu, \kappa_{\max}^{3/4})$. Compute $\tilde{\mu} = \hat{\mu}_n$ using a simple random sample that is independent of those used to compute $\hat{v}_{n_\sigma}$ and $\hat{\mu}_{n_1}, \ldots, \hat{\mu}_{n_\tau}$. Terminate the algorithm.*

If this algorithm terminates, then the general error criterion, (21)*, is satisfied.*

Proof. In this algorithm there are a number of important random variables: the estimated upper bound on the standard deviation, $\hat{\sigma}$, the sample sizes $n_1, \ldots, n_\tau, n$, the number of iterations, τ, required to get a good lower bound $\mathfrak{c}$, and the final estimate of the mean $\tilde{\mu} = \hat{\mu}_n$. These sample means are conditionally independent given the sequence of sample sizes. The probability that the final confidence interval is correct, is then no less than the probability that all of the confidence intervals are correct, conditioned on the sample sizes. Specifically,

$$\begin{aligned} \operatorname{Prob}\left[\frac{|\tilde{\mu} - \mu|}{1 - \theta + \theta\,|\mu|} \leq \varepsilon\right] &\geq \operatorname{Prob}\left[|\hat{\mu}_n - \mu| \leq \mathfrak{c}\varepsilon \;\&\; \mathfrak{c} \leq 1 - \theta + \theta\,|\mu|\right] \\ &= E\left\{\operatorname{Prob}\left[|\hat{\mu}_n - \mu| \leq \mathfrak{c}\varepsilon \;\&\; |\hat{\mu}_{n_\tau} - \mu| \leq \hat{\varepsilon}_\tau \mid \hat{\sigma}, \tau, n_1, \ldots, n_\tau, n\right]\right\} \\ &\geq E\left\{\operatorname{Prob}\left[|\hat{\mu}_n - \mu| \leq \mathfrak{c}\varepsilon \;\&\; |\hat{\mu}_{n_i} - \mu| \leq \hat{\varepsilon}_i \; \forall i \mid \hat{\sigma}, \tau, n_1, \ldots, n_\tau\right]\right\} \\ &\geq E_{\hat{\sigma}}\left\{[(1-\alpha_\mu)(1-\alpha_1))(1-\alpha_2)\cdots]1_{[\sigma,\infty)}(\hat{\sigma})\right\} \\ &\geq (1-\alpha_\sigma)(1-\alpha_\mu)(1-\alpha_1)(1-\alpha_2)\cdots = 1 - \alpha. \qquad \square \end{aligned}$$

Remark 2. Step 2 in this algorithm is not needed for the case of pure absolute error $\theta = 0$, because $\mathfrak{c} = 1$ automatically, which is large enough. As suggested earlier, a difficulty may arise if $\mu \approx 0$ and $\theta \approx 1$, in which the algorithm may fail to converge in a reasonable number of steps and overall sample size. Step 2c has safeguards against making $\hat{\varepsilon}_{i+1}$ too small compared to $\hat{\varepsilon}_i$, but this may also increase the number of iterations, τ, necessary for completion of Step 2. Because it is difficult to knowing how large τ is for a given integrand, there is no rigorous bound on the cost of this algorithm yet.

7 Discussion

Put something here.

Acknowledgements

The authors gratefully acknowledge discussions with Erich Novak and Henryk Woźniakowski. The plots of the univariate fooling functions were prepared with the help of Nicholas Clancy and Caleb Hamilton.

References

Bahadur RR, Savage LJ (1956) The nonexistence of certain statistical procedures in nonparametric problems. Ann Math Stat 27:1115–1122

Dick J, Pillichshammer F (2010) Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration. Cambridge University Press, Cambridge

Gander W, Gautschi W (2000) Adaptive quadrature — revisited. BIT 40:84–101

Hale N, Trefethen LN, Driscoll TA (2012) Chebfun Version 4

Hong HS, Hickernell FJ (2003) Implementing scrambled digital nets. ACM Trans Math Software 29:95–109

Lin Z, Bai Z (2010) Probability Inequalities. Science Press and Springer-Verlag, Beijing and Berlin

Matoušek J (1998) On the L_2-discrepancy for anchored boxes. J Complexity 14:527–556

Miller R (1986) Beyond ANOVA, Basics of Applied Statistics. John Wiley & Sons, Inc., New York

Owen AB (1995) Randomly permuted (t,m,s)-nets and (t,s)-sequences. In: Niederreiter H, Shiue PJS (eds) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Springer-Verlag, New York, Lecture Notes in Statistics, vol 106, pp 299–317

Owen AB (1997a) Monte Carlo variance of scrambled net quadrature. SIAM J Numer Anal 34:1884–1910

Owen AB (1997b) Scrambled net variance for integrals of smooth functions. Ann Stat 25:1541–1562

Petrov VV (1995) Limit Theorems of Probability Theory:Sequences of Independent Random Variables. Clarendon Press, Oxford

Shampine LF (2008) Vectorized adaptive quadrature in matlab. J Comput Appl Math 211:131–140

The MathWorks, Inc (2012) MATLAB 7.12. The MathWorks, Inc., Natick, MA

Traub JF, Wasilkowski GW, Woźniakowski H (1988) Information-Based Complexity. Academic Press, Boston