ELSEVIER

Mathematics and Computers in Simulation 54 (2000) 131–143

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.nl/locate/matcom

Accuracy estimation for quasi-Monte Carlo simulations

William C. Snyder

SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, USA

Accepted 28 July 2000

Abstract

The conventional Monte Carlo approach to integration and simulation is a useful alternative to analytic or quadrature methods. It has been recognized through theory and practice that a variety of uniformly distributed sequences provide more accurate results than a purely pseudorandom sequence. The improvement in accuracy depends on the number of dimensions and the discrepancy of the sequence, which are known, and the variation of the function, which is often not known. Unlike pseudorandom methods, the accuracy of a quasirandom simulation cannot be estimated using the sample variance of the evaluations or by bootstrapping. The improvement in time-to-accuracy using quasirandom methods can be as large as several orders of magnitude, so even an empirical accuracy estimator is worth pursuing. In this paper, we discuss several methods for quasirandom empirical accuracy estimation and evaluate a modified empirical technique that appears to be useful.

Keywords: Monte Carlo method; Quasi-Monte Carlo method; Uniformly distributed sequences; Low-discrepancy sequences; Numerical integration; Accuracy assessment

1. Introduction

Quasirandom Monte Carlo methods are usually applied to numerical integration problems in physics but are gaining acceptance in other applications of computer simulation. Good reviews of the state of the art for integration are available [1–3]. Consider a model of some process of interest with a moderate number of input dimensions and that is one-to-one. In the case where traditional Monte Carlo methods are being considered, and where the function is expensive to evaluate, quasirandom Monte Carlo sampling can often achieve a given accuracy with far fewer evaluations. Implementation is often as simple as replacing the pseudorandom generator with a quasirandom one and assigning the input dimensions appropriately.

One limitation of this technique lies in estimating the accuracy of the results of a particular simulation. With pseudorandom simulations, the points are independent and we can employ one of many accuracy estimators, such as expressions based on the central limit theorem. Because the improvement from quasirandom sampling depends on properties of the function and the quasirandom point generator, it can range

E-mail address: wcsnyder@ieee.org (W.C. Snyder).

0378-4754/00/$20.00
PII: S0378-4754(00)00204-4

from none to several orders of magnitude [4–6]. In many practical cases, the nature of the function is not fully known, for instance it may be a complex computer model. Under these circumstances, an empirical accuracy estimator would enhance the range of applications of quasirandom simulation.

Convergence studies focus mainly on quasirandom integration. Because most applications of simulation are equivalent to integration, the existing studies are relevant to our wider scope [2]. Briefly, the convergence treatment is as follows. Define the absolute integration error, η, of a function $f(\cdot)$ as

$$\eta = \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \int_{(0,1)^s} f(x)\, \mathrm{d}x \right|.$$

The Koksma–Hlawka inequality is one expression of an absolute bound on the accuracy of quasirandom integration for a general class of functions,

$$\eta \leq V(f) D_n^* \tag{1}$$

The two important factors here are $V(f)$, the total variation of the function f in the sense of Hardy and Klause, and D_n^*, the star discrepancy of the point set, which depends on the number of points and the generation algorithm [7]. Given a variation, the improvement rate of the discrepancy determines the improvement rate of the accuracy. The expected discrepancy can be calculated for pseudorandom sequences and is bounded by $(\log \log n)\, n^{-1/2}$. Here is evident the $n^{-1/2}$ convergence and no dependency on dimension. Quasirandom simulations lose their advantages as the number of dimensions increases, say larger than 15 or so. Hybrid pseudorandom and quasirandom mixing techniques have been developed for larger numbers of dimensions. For instance, Ökten [8] evaluates the discrepancy of a hybrid sequence formed by concatenating a quasirandom and pseudorandom vector for each point.

For good, pure quasirandom sequences, the star discrepancy is bounded by $c_s(\log n)^s n^{-1}$; for good finite point sets the bound is slightly better, $c_s(\log n)^{s-1}/n$. The constant c_s depends only on the generation algorithm and number of dimensions. These bounds indicate that quasirandom integration will converge significantly faster than pseudorandom integration in many cases. The discrepancy, however, does not provide the absolute accuracy for a particular simulation result. That is not easily predicted, mainly because the variation of the function is unknown. In addition, even if it were known, the variation bound is a weak one. In fact, the inequality and the bounds do not fully characterize the actual convergence. The bounds are asymptotic, so in fact, the performance for finite values of n may be better or worse than would be indicated.

There are various pure quasirandom generators available. An early sequence generator was proposed by Halton and Smith [9]. Berblinger and Schlier [10] give a numerically stable integer-ratio form of this radical inverse algorithm. Sobol and Faure sequences are modifications of the Halton sequence to improve high-dimensional performance. Algorithms and code for the Sobol sequence with up to 50 dimensions are provided by Press et al. [11]. A low-discrepancy construction for finite point sets, and one that is used in the present study, is the (t, m, s)-net [12]. For base a this is a set of $n = a^m$ points in the s dimensional unit cube $(0,1)^s$ such that every elementary interval of volume a^{t-m} contains exactly a^t points. For a (t, m, s)-net the size must be a power of the base, which, for small bases, is suitable for simulations. A useful attribute of the (t, m, s)-net is that sets of size a^m contain a $(t, m-1, s)$-nets of size a^{m-1} which are available simply by dividing the full set into partitions. This fact will be employed in the accuracy assessment technique to be described later.

So, in approximate terms, a model of the simulation accuracy is cn^{λ} and in general neither c nor λ are easy to determine for the combination of a given function and point generator. That motivates the empirical accuracy estimation method which will be presented here. Section 2 is a review of empirical accuracy methods for traditional Monte Carlo simulation. It is shown that analogs of these methods are not applicable to quasirandom results, so alternatives are explored. To begin with, a scrambling method provides several approaches to quasirandom accuracy assessment, but all require some reduction in efficiency of the simulation or a reduction in accuracy of the resulting point estimate. The partition method avoids these drawbacks, but for the typical case the result is biased in the conservative direction. A simple modification to the partition method is presented that reduces this bias. In Section 3 the effectiveness of the method is explored with several test functions.

2. Assessing simulation accuracy

2.1. Some simulation definitions

First, it is useful to digress a moment and consider the following definitions. For *simulation with a known input distribution* we generate from known distributions and apply a deterministic transform to compute output values. The output values are distributed as F. We seek to estimate an unknown parameter of F from a sample of size n:

$$\hat{\theta} = \hat{\theta}(Y_1, Y_2, \ldots, Y_n). \tag{2}$$

In the case of *simulation with an estimated input distribution* we generate inputs from an estimated distribution and apply the deterministic transform. Therefore, the output values are distributed as an estimate of the actual output distribution, $\hat{F}$, and we apply the same estimator as before. The parameter value will be approximate both because the input, and therefore the output distribution is an estimate, and because the sample has a finite size. In this case, it is useful to verify by some means that the finite sample size dominates the accuracy and to consider the input distributions to be exact.

Simulation with an estimated input distribution is similar to bootstrapping, but there are two differences. First, with simulation, the sample values are mapped through some model before evaluating a statistic. This is not usually the case for bootstrapping, which uses the sample directly. More importantly, for simulation, we do not require that the samples produce any particular sampling distribution for the statistic. For instance, we are free to generate using low-discrepancy points. This will significantly improve the sampling distribution. With bootstrapping, it is the sampling distribution that is of interest, so the resampling process must have the appropriate properties.

We also have *replicated simulations*, or "replication" for short. Here, for simulation with known or estimated input distributions, the simulation is repeated for a particular sample size and the estimator is evaluated each time to build a sample of point estimates from which we can infer properties of the sampling distribution. Replication is inefficient, but it is important in the understanding of bootstrapping and estimation accuracy, and in establishing a reference to evaluate the performance of other accuracy estimation schemes. The resamples generated by replication are what we claim to duplicate by frequentist bootstrapping.

And, finally, in its frequentist form, *bootstrapping* repeatedly generates independent samples from an output distribution estimate. The distribution estimate is based on a sample of real or simulated data. In particular, the simulation provides a point estimate from a particular sample, $\theta^* = \hat{\theta}(y_1, y_2, \ldots, y_n)$,

and we would like to know the sampling distribution of the estimate. The sample is assumed to come from some distribution, F. The bootstrap estimates this underlying distribution from the sample by the functional, $\hat{F} = \hat{F}(Y_1, Y_2, \ldots, Y_n)$, and generates new iid samples from the distribution estimate, $(Y_1', Y_2', \ldots, Y_n')_j, \quad j = 1, 2 \ldots b$. Then, the set of b estimates are computed, $(\theta_1^*, \theta_2^*, \ldots, \theta_b^*)$, from which a sampling distribution is inferred. The estimate of the distribution of Y may be either nonparametric, smoothed nonparametric, or parametric. It is often taken to be the nonparametric empirical representation and the bootstrap estimate is often evaluated numerically, as opposed to symbolically.

2.2. Analogs of conventional accuracy assessment methods

Consider the central limit theorem. For the pseudorandom case with the sample mean as the statistic, we can construct a random variable Y with $\theta = E(Y)$. Then our estimate of θ is simply the sample mean of a sample of size n. Let the variance of the random outputs be σ_Y^2. If n is sufficiently large, according to the central limit theorem, the distribution of the average $(Y_1 + Y_2 + \ldots + Y_n)/n$ is normal with mean $\mu = \theta$ and variance $\sigma^2 = \sigma_Y^2/n$. As n increases, the probability of standardized error, $(\hat{\theta} - \theta)/(\sigma_Y/\sqrt{n})$, approaches that of the standard normal variable. Therefore, regardless of the distribution of Y, we can approximate its variance from the sample and can construct confidence intervals for the accuracy of the mean. Unfortunately, this method assumes independent points. If this method is applied to a quasirandom simulation, the accuracy estimate will be nearly the same as it would be for the pseudorandom case — far too conservative.

Next, consider replication, which requires independent equivalent point sets. Generating new sets of input points is straightforward for the pseudorandom case. In spite of the fact that quasirandom points are deterministic, it turns out that we can also generate a stochastic family of quasirandom point sets with equivalent properties. One method is to apply ks quasirandom dimensions to get k independent s-dimensional point sets. With, say, 8 or 16 sets, we can get a reasonably good idea of the variability of the statistic. Even with k as low as 8, however, a typical generation algorithm with less than 50 independent dimensions will significantly limit the number of dimensions remaining for s.

Another replication method is to shift points in an s-dimensional set by a random vector distributed uniformly in the unit cube, and apply the mod 1 function componentwise [13]. This has been shown to be effective, and is not limited by the dimensionality of the generator, but it is suboptimal at preserving the quality of the samples. Finally, with (t, m, s)-nets, we can scramble the points using Owen's method [14]. Essentially, in base 2, Owen's method pseudorandomly flips the point locations in successively smaller binary partitions. Each new permuted point set is a (t, m, s)-net with the same equidistribution properties as the original, and thus we have a stochastic family quasirandom sets. Owen's method is straightforward to implement and can be made fast by scrambling integers and then converting to floating point values. This is the method used to generate the "truth" values for the numerical experiments in Section 3. Recall that replication is not efficient for accuracy assessment.

An accuracy assessment technique that is efficient is the bootstrap because it uses only a single sample of output values. With the ability to generate independent realizations of a quasirandom point set, we can at least construct quasirandom bootstrap procedures analogous to the pseudorandom case. Bootstrapping is a more general method that uses the empirical sample in place of its unknown distribution to assess the certain properties of an estimator, such as its sampling distribution [15–18]. Bootstrapping remains effective for statistics other than the mean and variance and for normal, non-normal, and in fact, non-parametric distributions. Bootstrapping is known to be useful for characterizing the accuracy of simulation results [19,20].

The simplest nonparametric bootstrap gives equal weights to each point and resamples with replacement. In fact, at least with one-dimensional points, this is identical to generating samples of size n from the empirical distribution using the inverse cumulative distribution function (cdf) method. So, from the original data, we construct new empirical discrete distributions with weights of z_i/n at each point. Here, z is an integer that is incremented by one each time that point is chosen. We generally produce resamples of the same size as the original sample (although this is not strictly necessary) so the distribution is normalized. The empirical distribution is discrete, and if it is known that the actual distribution is continuous, smoothing will improve the accuracy of the bootstrap results. Kernel smoothing is easy to implement for iid data with the composition method [17,21]. This method takes the empirical bootstrap sample and adds a pseudorandom vector to each point. When the distribution is known to be a particular parametric one, we can estimate the parameters from the sample, and then generate from the parametric form using, for instance, the inverse cdf method.

For numerical bootstrapping of quasirandom results, we would like to generate independent, equivalent new samples from an estimate of the output distribution. The samples should duplicate the result of replication with scrambled quasirandom samples. Scrambling the output sample is an option here, but unfortunately if the pseudorandom values are simply replaced with scrambled quasirandom values in the bootstrap sampling mechanism, the results generally do not have the correct sampling distribution, at least for more than one input and output dimension. Even with a smoothed inverse cdf, resampling does not work for more than one input and output dimension it overestimates the accuracy because it does not represent the additional variability resulting from the mapping of an s-dimensional input to get an o-dimensional output with $s > o$. In fact, quasirandom bootstrapping does work for one input and output dimension by interrogating a smoothed inverse cdf with scrambled quasirandom points to generate new samples. Most applications, however, involve more than one input dimension. It appears that a successful multidimensional bootstrap technique would have to work in the input space to account for the effects of the number of input dimensions on the variability. This may be possible with a quasirandom analog of the Bayesian bootstrap.

In contrast to the frequentist bootstrap, the Bayesian bootstrap constructs new discrete distributions with continuous weights, q_i, at each point, where $\sum_{i=1}^{n} q_i = 1$ [22,23]. All points are included once. One way to construct the weights are the n gap sizes of an ordered random sample of $n-1$ points from $\mathcal{U}(0, 1)$. The statistic of interest is computed from the point set and each new set of weights. These statistics are then applied as before to estimate the sampling distribution and quantify the error in the point estimate. Because the weights are continuous, the effect is similar to that of a smoothed bootstrap, but unlike the smoothed bootstrap, we do not generate points at values that did not appear in the original sample.

For a quasirandom Bayesian bootstrap, there is in principle some point reweighting scheme in the input space that would yield reweightings of the output sample points that produce the desired sampling distribution. But this scheme appears to be difficult to formulate. The weight sets from some form of scrambling must have a particular variation and be properly interdependent in the s-dimensional space. If an appropriate procedure could be found, these weights could be assigned through the one-to-one connection to reweight the corresponding o-dimensional output points and to bootstrap the sampling distribution. This Bayesian bootstrap approach for quasirandom simulations will not be explored here but it appears to be a promising area for further study.

It is worth mentioning a novel variation of the frequentist bootstrap that combines pseudorandom and quasirandom points. It is Do and Hall's quasirandom resampling scheme [24]. This method applies quasirandom points to generate a more representative sampling distribution for a statistic of a pseudo-

random sample or real data. The method is as follows: an *n*-dimensional cube is partitioned equally into *n* parts on each axis. The partition indices into which a point in the *n*-dimensional cube falls are the indices of the points to use in forming a bootstrap sample of size *n*. In other words, any point in the cube represents a particular sample of size *n* with replacement. If the cube is interrogated with *n*-dimensional pseudorandom points, the result is identical to random sampling with replacement. We can, however, replace the pseudorandom with quasirandom interrogation, and generate a "more representative" set of resamples, and thus a more accurate bootstrap estimate. What is required is a quasirandom generator with a dimension for each point in the sample. Thus, the method is most useful with small sample sizes of pseudorandom or actual data and when the statistic is expensive to compute. Although the bootstrapping mechanism uses quasirandom points, the method cannot be used to bootstrap quasirandom data, which is our goal here.

Thus, it turns out that, even with a scrambling technique that generates independent and equivalent sets of quasirandom points, neither the standard error method nor any of the analogs to bootstrapping are applicable to the results of a quasirandom simulation. These methods all require that the data points be independent. So we must look beyond analogs of the traditional techniques.

2.3. Other options for quasirandom accuracy assessment

One possibility that is not an analog of a conventional method would be to apply the Koksma–Hlawka inequality that we have already discussed (1). For this expression to provide an accuracy estimate in a particular case, we would need to estimate the variation of the function, $V(f)$, by some means. The variation is a complicated multidimensional integral involving partial derivatives of the function. Although we could perhaps approximate the variation from the simulation evaluations themselves, in general it is difficult to do this [2,25]. And even with a variation estimate, we have said that for finite *n* the Koksma–Hlawka bound is not particularly meaningful in practice.

So finally, consider empirical techniques that apply the behavior of statistics evaluated from subsets of the full output sample. For instance, the variability and convergence rate are related to the accuracy of the statistic as the simulation progresses using a sequence. Owen suggests two simple techniques [14]. The first method evaluates *b* replications. Each replication is an independent scramble of a set of size *n*/*b*. The *b* statistics from the samples, $\theta_j^*, \quad j = 1, \ldots, b$, can be averaged to get a point estimate for a sample of size *n*, $\hat{\theta} = \bar{\theta}$, and, because of the independence of the scrambled samples, the statistics will give an estimate of its accuracy:

$$\sigma(\hat{\theta}) = b^{-1/2} \left(\frac{1}{b-1} \sum_{j=1}^{b} (\theta_j - \bar{\theta})^2 \right)^{1/2}. \tag{3}$$

This technique uses a total of *n* evaluations, but the point estimate from averaging *b* scrambled sets of *n/b* quasirandom evaluations is not as accurate as one from a single set of *n* evaluations. Depending on the function, this may be a significant difference. In fact, as *b* tends to infinity with *n/b* fixed, such replication only achieves the pseudorandom variance rate. For example, many quasirandom simulations exhibit convergence rates near n^{-1}. For reasonably accurate estimates of the deviation, the number of partitions needs to be, say, 8 or 16. With $b = 16$, the sacrifice in accuracy in going from the statistic of a single net of size *n* to the mean of statistics from *b* nets of size *n/b* would be as large as $16/\sqrt{16} = 4$.

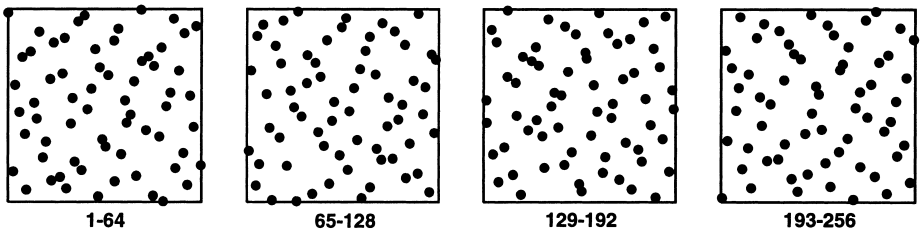

Fig. 1. Four partitions of a single (*t*, *m*, *s*)-net in two dimensions.

Thus, it is of interest to find methods that work with a single quasirandom point set of size *n*. Owen mentions a second technique, the partition method, that does this.

It appears that the estimator just described (3) gives nearly the same result with *b* partitions of a single net, each of size *n*/*b* instead of *b* scrambled nets of size *n*/*b*. This means that the point estimate is that of a complete quasirandom set of size *n*, with no sacrifice in accuracy. So this method has the dual advantages of using a single quasirandom point set and not requiring any scrambling.

In fact, if the partition size is a power of the (*t*, *m*, *s*) base, which is 2 in our case, recall that each of the partitions is also a (*t*, *m*, *s*)-net. This is important because each of these subsets will then tend to have points equally well-distributed in the hypercube. To help visualize this, Fig. 1 shows the four partitions of size 64 from a quasirandom set of size 256 in two dimensions. Because these subsets are not scrambles, but sections of a larger, single net, the statistics associated with them will be slightly negatively correlated. For a reasonable range of *n*, *b*, *s*, and the variation of *f*, it appears that the negative correlation does not affect the accuracy estimate very much and any small effect is in the conservative direction.

So in particular, a single net of size *n* is partitioned into *b* sets of *n*/*b*. The sample standard deviation of the statistic for sets of size *n*/*b* is estimated using the statistics evaluated from these partitions. Here, the sample standard deviation will be denoted $\tilde{\sigma}$ because it is not calculated from independent points. We have said that $\tilde{\sigma} > \sigma$. The value is then extended by some means from sets of size *n*/*b* to predict the deviation of sets of size *n*. If the value of *b* is too small, the deviation estimate will have a high variance. If *b* is too large, then the partitions are too small, and we cannot reliably extend the deviation from sets of size *n*/*b* to sets of size *n*. Useful values of *b* are, say, between 8 and 32. Proceeding, we have as our estimator for the deviation of the statistic for sets of size *n*/*b*,

$$\tilde{\sigma}(\hat{\theta}_{n/b}) = \left(\frac{1}{b-1} \sum_{j=1}^{b} (\theta_{n/b,j} - \bar{\theta}_{n/b})^2 \right)^{1/2}. \tag{4}$$

Next, Owen's method takes the rate of improvement of the deviation from sets of size *n*/*b* to sets of size *n* to be $n^{-1/2}$, the same as the pseudorandom case. Quasirandom simulations of smooth functions often exhibit a convergence rate closer to n^{-1}, so this is another conservative approximation. The accuracy estimator is the same as Eq. (3),

$$\tilde{\sigma}(\hat{\theta}_n) = \left(\frac{n/b}{n} \right)^{1/2} \tilde{\sigma}(\hat{\theta}_{n/b}) = b^{-1/2} \tilde{\sigma}(\hat{\theta}_{n/b}). \tag{5}$$

We have said that both the rate term and the deviation term in the product contribute to a conservative bias the accuracy estimate. In practice, the bias from the rate term dominates. The following simple modification to the partition method reduces this bias.

The modification begins by computing the deviation not at one value of b but at several different values. Then, the log of the partition sizes and the log of the deviation estimates are fit to a line which is extrapolated to the total number of points. Call this the multipartition method. In practice, the accuracies calculated from each partition size are nearly independent and the method appears to work well. A modification that decreases the variability of the estimate is to constrain the estimate of the slope (convergence rate) to be between $-1/2 < \lambda < -1$.

So, specifically, the simple modification computes a partition-based deviation estimate at, say, five points: $b = \{64, 32, 16, 8, 4\}$. Then a line is fit to the log-log points of deviation versus partition size using the constrained, weighted least squares method to estimate the slope λ and intercept, q. The points are weighted by b to account for the increasing variability for smaller values of b. The accuracy of the statistic is then computed by extrapolating to n,

$$\tilde{\sigma}(\hat{\theta}_n) = \log^{-1}(\lambda \log(n) + q). \tag{6}$$

So instead of using a convergence rate of $-1/2$ this method fits a line and approximates the actual local convergence rate. This removes the bias from the rate assumption and, in fact, also will tend to reduce the slight bias from the negative correlation of the partition statistics. The bias from the negative correlation is larger for small partitions and decreases to zero for the statistic from the single, complete point set. So the fitted line has a slope slightly steeper than the true convergence, but when it is extrapolated to n the deviation is correct.

The partition and multipartition method are depicted in Fig. 2. This is an idealized plot for a simulation that is converging with $\lambda = -1$. The single partition method is shown using $b = 4$ and a $n^{-1/2}$ rate extrapolation to the estimate. Clearly the bias is the sum of the negative correlation bias and the rate bias. The multipartition method has a rate biased slightly too steep, but extrapolates to the correct value. The amount of negative correlation bias is exaggerated for clarity. Of course, in practice there is significant variability in the fitted line and the location of the intercept for either method. In the next section, the

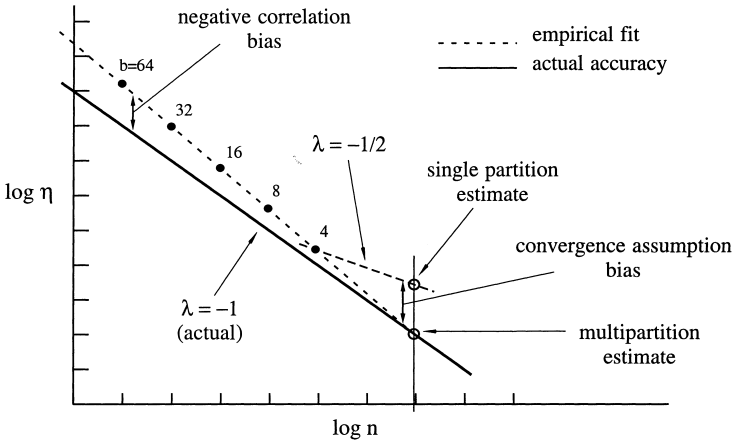

Fig. 2. Idealized diagram of the partition and multipartition estimation schemes.

pseudorandom bootstrap, the single partition method, and this multipartition method will be illustrated with several test functions.

3. Numerical examples

For engineering and applied science applications it is often the case that a function is being investigated numerically because it exists only as a computer program. In fact, the function may be quite smooth and so is easily characterized by Monte Carlo methods. For tests of the accuracy estimation techniques, however, it is better to find functions that are simple to define and have known parameter values, but present a range of difficulty. These are available from studies of traditional Monte Carlo integration. A range of functions were tested which had up to 12 dimensions. Two of these illustrate the results. The first, (7), is a typical integration test function quoted in Davis and Rabinowitz [26]. The second function, (8) is a high variation function suggested by P. Anderson (personal communication). Also shown are the number of dimensions, s, and the mean, μ, and standard deviation, σ, of the evaluations.

$$f_1(x) = \exp\left(\prod_{i=1}^{s} x_i\right) - 1.0, \quad s = 4, \mu \simeq 0.06939784, \sigma \simeq 0.1122, \tag{7}$$

$$f_2(x) = \prod_{i=1}^{s} (2x_i)^{p_i - 1} \cos(2\pi (2x_i)^{p_i}), \quad s = 5,\ p = \{3, 4, 5, 3, 4\}, \mu = 0.0, \sigma \simeq 25.68. \tag{8}$$

The test functions are evaluated on the unit cube. The quasirandom generator used for these experiments generates a (t, m, s)-net in base 2 using Niederreiter's algorithm [7]. Before use, the quasirandom point sets are scrambled once using Owen's technique. This is known to further improve the performance.

Simulations for the test functions were run with both a pseudorandom and quasirandom generator. For each type of sampling, 35 replications were generated at each of eight power-of-two sample sizes, $n = \{1024, 2048, \ldots, 131072\}$. For both the pseudo and quasirandom cases, reference accuracy values were calculated using the standard deviation of the statistic from each of the 35 replications.

For the pseudorandom simulations the accuracy of the mean was estimated by the simple bootstrap for each of the 35 replications, and at each of the eight sample sizes. The bootstrap applied 32 resamples. For the quasirandom simulations, the multipartition method was applied for each of the 35 replications using eight sample sizes, $b = \{64, 32, 16, 8, 4\}$. The plots show the $\log_{10}$ of the deviation of the mean at each of the eight sample sizes.

In Fig. 3, the 35 accuracy estimates from the frequentist bootstrap at each sample size are shown as box plots over the pseudorandom reference values, that are connected with a dashed line. The boxes contain the mean, shown as a horizontal line and contain 68% of the data. It is clear that the pseudorandom accuracy converges as $n^{-1/2}$. Also, the bootstrap accuracy estimates are within a factor of two of the true accuracy and apparently unbiased. Their variability could be further improved by increasing the number of resamples. Although only the results for the mean are plotted here, the bootstrap exhibited similar performance when assessing the accuracy of the standard deviation. The Bayesian bootstrap has approximately the same performance for the functions tested.

Also in Fig. 3, the quasirandom reference values are connected with a solid line. For f_1, over the range of n the quasirandom simulation is more accurate by between one and almost three orders of magnitude.

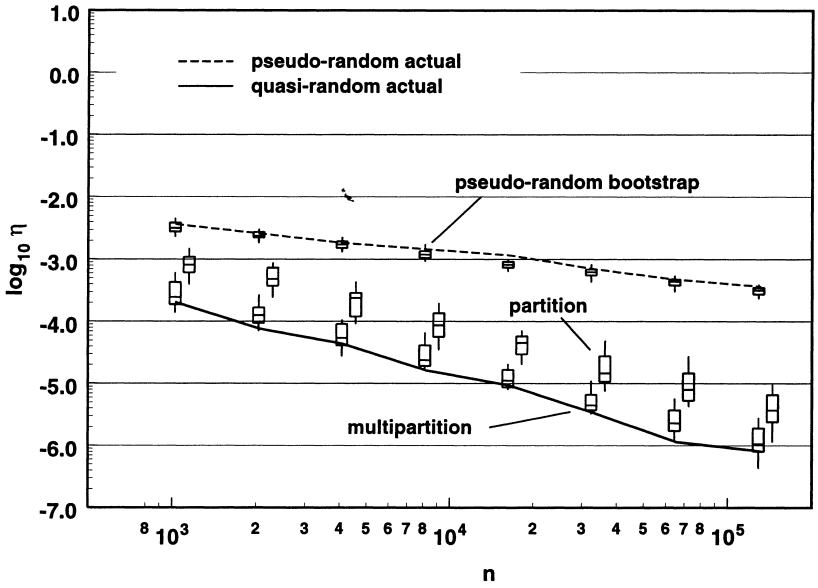

Fig. 3. Experimental results with the function f_1. The dashed line is the actual pseudorandom accuracy and the box plots show the bootstrap accuracy estimates. Each box contains 68% of the data and shows the mean as a horizontal line. The solid line is the actual quasirandom error with the partition (shifted to the right) and multipartition box plots.

The quasirandom convergence is approximately n^{-1} over the range of the plot. Also in this figure are the box plots of accuracies from the single-size partition method. For these estimates there were $b = 16$ partitions. Clearly there is a significant conservative bias. Indeed it is approximately the factor of four that results from the assumption of $n^{-1/2}$ rate when the convergence is n^{-1}. The figure also shows the box plots of the multipartition accuracy estimates for each of the 35 simulations. The estimates have approximately the same variability as the single size partition method, but are significantly less biased. In fact, because the convergence is limited to be no better than -1.0 there is still a slight conservative bias, which is desirable. This plot is typical for smooth functions of up to at least 15 dimensions.

The results for f_2 are plotted in Fig. 4. This function is an example of an extreme case for which quasirandom sampling does no better than traditional Monte Carlo. The function f_2 has a high spatial frequency content for each of its five dimensions. In addition, the evaluations of f_2 have a relatively high standard deviation. Even for the highest number of evaluations, there are only approximately $131072^{1/5} \simeq 11$ samples per dimension. There is not much difference in accuracy when the samples are more equally distributed. For sample sizes much larger than 131072, however, the bound of the quasirandom accuracy improvement, $(\log n)^4/n$, would overtake the pseudorandom rate of $n^{-1/2}$.

Because of the higher variance and the higher spatial frequency content, both the pseudo- and quasirandom accuracy estimates for f_2 have a high variability, especially at the smaller sample sizes. For this function, the single size partition method would give approximately the same results as the multipartition method because the local convergence rate is $n^{-1/2}$. Of course we would like to have lower variability in the accuracy estimates, but this performance is quite useful. It turns out that the variabilities from the scrambling methods are not much better, and the results can be strongly biased toward the conservative direction, as can those from the single size partition method.

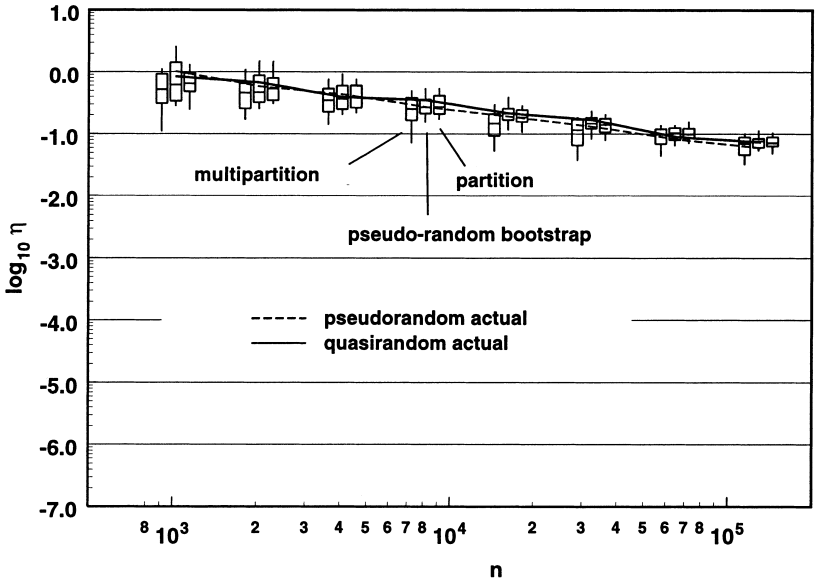

Fig. 4. Experimental results with the function f_2. These are the same plots shown in Fig. 1. Here they all overlap because the convergence of the quasirandom simulation in this extreme case is the same as pseudorandom.

To further illustrate the capability of the multipartition method, the accuracies with six integration test functions from Genz [27] were evaluated:

$$f_{\text{osc}}(x) = \cos\left(2\pi u_1 + \sum_{j=1}^{s} a_j x_j\right) \quad \text{(Oscillatory)},$$

$$f_{\text{ppk}}(x) = \prod_{j=1}^{s} (a_j^{-2} + (x_j + u_j)^2)^{-1} \quad \text{(Product peak)},$$

$$f_{\text{cpk}}(x) = \left(1 + \sum_{j=1}^{s} a_j x_j\right)^{-(s+1)} \quad \text{(Corner peak)},$$

$$f_{\text{gaus}}(x) = \exp\left(-\sum_{j=1}^{s} a_j^2 (x_j - u_j)^2\right) \quad \text{(Gaussian)},$$

$$f_{C_0}(x) = \exp\left(-\sum_{j=1}^{s} a_j |x_j - u_j|\right) \quad (C_0),$$

$$f_{\text{disc}}(x) = \exp\left(\sum_{j=1}^{s} a_j x_j\right) 0_{x_1 > u_1} 0_{x_2 > u_2} \quad \text{(Discontinuous)}.$$

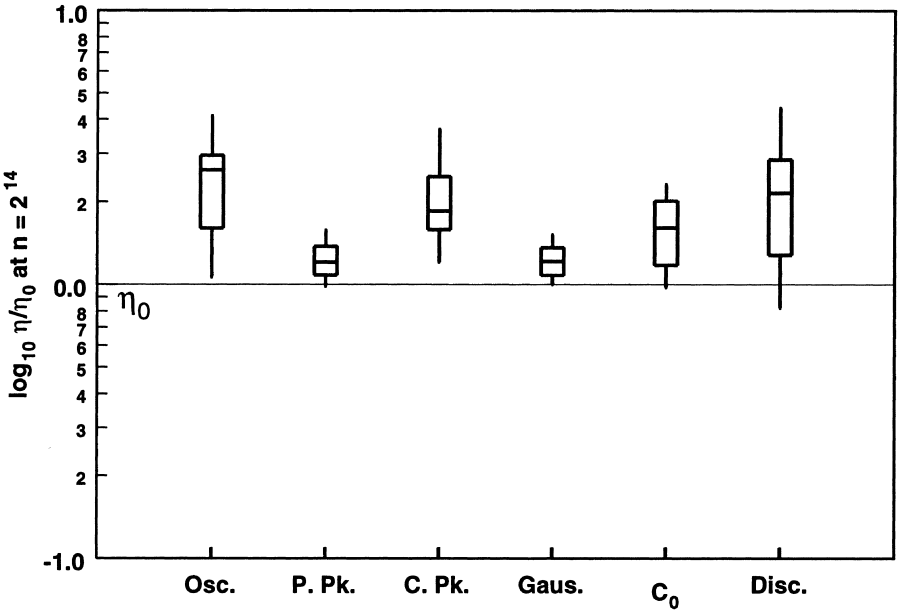

Fig. 5. These are box plots of the variability of the multipartition method for the six test functions (see text). There are 35 replications for each function consisting of generating a quasirandom sample using scrambling, and evaluating its accuracy. The data are log of the accuracy estimate divided by the actual accuracy. The boxes contain 68% of the data. The means are shown as a horizontal line in each box.

These were all tested with 10 dimensions. Fig. 5 shows the variability of the multipartition estimates. The six box plots are the range of 35 replications, each with a scrambled quasirandom sample of $n = 16,384$ points. This plot shows the log of the accuracy divided by the actual value. Practically all of the estimates are biased conservatively because of the constraint on the slope. The accuracy estimates are primarily within a factor of three of the actual accuracy value.

So, Owen's partition method is an empirical approach that gives an empirical accuracy estimate from the evaluations of a single net or sequence. With a simple modification the partition method appears to work well on many test functions which were designed to challenge Monte Carlo integration methods. Based on the test functions, the multipartition method predicts empirically the accuracy of a simulation to well within an order of magnitude using the quasirandom outputs only.

As the simulation progresses, the method estimates the current accuracy, and in addition, gives the improvement rate that can be used to predict the number of additional evaluations needed for and accuracy goal. Many simulations would not be practical without the gains in efficiency. This is true particularly for models that take a significant time to evaluate or for exploratory work on a model which involves many simulations, such as for optimization or sensitivity analysis.

The multipartition method should work well for a variety of generators including pseudorandom and hybrid ones. It's only assumptions are that locally, the accuracy is cn^{λ} and that partitioning the sequence produces a reasonable estimate of the accuracy for each partition size. In conclusion, the multipartition method appears to be a robust and useful accuracy estimator.

References

[1] W.J. Morokoff, R.E. Caflisch, Quasi-Monte Carlo integration, J. Comput. Phys. 122 (1995) 218–230.
[2] F. James, J. Hoogland, R. Kleiss, Multidimensional sampling for simulation and integration: measures, discrepancies, and quasirandom numbers, Comput. Phys. Commun. 99 (1997) 180–220.
[3] J.M. Sobol, On quasi-Monte Carlo integrations, Math. Comput. Simul. 47 (1998) 103–112.
[4] W.H. Press, S.A. Teukolsky, Quasi- (that is, sub-) random numbers, Comput. Phys. 3 (1989) 76–79.
[5] W.J. Morokoff, R.E. Caflisch, Quasirandom sequences and their discrepancies, SIAM J. Sci. Comput. 15 (1994) 1251–1279.
[6] J. Hoogland, R. Kleiss, Discrepancy-based error estimates for Quasi-Monte Carlo. I: General formalism, Comput. Phys. Commun. 98 (1–2) (1996) 111–127.
[7] H. Niederreiter, Low-discrepancy and low-dispersion number sequences, J. Number Theory 30 (1988) 51–70.
[8] G. Ökten, A probabilistic result on the discrepancy of a hybrid-Monte Carlo sequence and applications, Monte Carlo Meth. Appl. 2 (1996) 255–346.
[9] J.H. Halton, G.B. Smith, Algorithm 247, radical-inverse quasirandom point sequence, Comm. ACM 7 (1964) 701–702.
[10] M. Berblinger, C. Schlier, Monte-Carlo integration with quasirandom numbers: some experience, Comput. Phys. Commun. 66 (1991) 157–166.
[11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, 2nd Edition, Cambridge University Press, Cambridge, 1992.
[12] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, 1992.
[13] R. Cranley, T.N.L. Patterson, Randomization of number theoretic methods for multiple integration, SIAM J. Numer. Anal. 13 (1976) 904–914.
[14] A.B. Owen, Monte Carlo variance of scrambled equidistribution quadrature, SIAM J. Num. Anal. 34 (1997) 1884–1910.
[15] B. Efron, Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods, Biometrika 68 (1981) 589–599.
[16] C. Leger, D.N. Politis, J.P. Romano, Bootstrap technology and applications, Technometrics 34 (1992) 378–398.
[17] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, London, 1993.
[18] G.A. Young, Bootstrap: More than a stab in the dark? Stat. Sci. 9 (1994) 382–415.
[19] L.W. Friedman, H.H. Friedman, Analyzing simulation output using the bootstrap method, Simulation 64 (1995) 95–100.
[20] R.C.H. Cheng, Bootstrap methods in computer simulation experiments, in: C. Alexopoulos, K. Kang, W.R. Lilegdon, D. Goldsman (Eds.), Proceedings of the 1995 Winter Simulation Conference, IEEE, 1995 pp. 171–177.
[21] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman and Hall, London, 1986.
[22] D.B. Rubin, The bayesian bootstrap, Ann. Statist. 9 (1981) 130–134.
[23] A.Y. Lo, A large sample study of the Bayesian bootstrap, Ann. Statist. 15 (1987) 360–375.
[24] K.-A. Do, P. Hall, Quasirandom resampling for the bootstrap, Stat. Comput. 1 (1991) 13–22.
[25] R. Kleiss, Quasi-Monte Carlo, quasirandom numbers and quasierror estimates, Int. J. Modern Phys. C 4 (1993) 323–330.
[26] P.J. Davis, P. Rabinowitz, Methods of Numerical Integration, 2nd Edition, Academic Press, Orlando, FL, 1984.
[27] A. Genz, Testing multidimensional integration routines, in: B. Ford, J.C. Rault, F. Thomasset (Eds.), Tools, Methods, and Languages for Scientific and Engineering Computation, North-Holland, Amsterdam, 1984, pp. 81–94.