Pierre L'Ecuyer
Art B. Owen *Editors*

Monte Carlo and Quasi-Monte Carlo Methods 2008

Springer

Monte Carlo and Quasi-Monte Carlo Methods 2008

Pierre L'Ecuyer • Art B. Owen
Editors

Monte Carlo and Quasi-Monte Carlo Methods 2008

Springer

Editors
Pierre L'Ecuyer
DIRO
Université de Montréal
C.P. 6128, Succ. Centre-Ville
Montreal, H3C 3J7
Canada
lecuyer@iro.umontreal.ca

Art B. Owen
Department of Statistics
Stanford University
Sequoia Hall
Stanford, CA 94305
USA
owen@stanford.edu

ISBN 978-3-642-04106-8 e-ISBN978-3-642-04107-5
DOI 10.1007/978-3-642-04107-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009940794

Mathematics Subject Classification (2000): Primary 11K45, 65-06, 65C05, 65C10; Secondary 11K38, 65D18, 65D30, 65D32, 65R20, 91B28

Cover design: VTeX, Vilnius

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume represents the refereed proceedings of the Eighth International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, which was held at the University of Montréal, from 6–11 July, 2008. It contains a limited selection of articles based on presentations made at the conference. The program was arranged with the help of an international committee consisting of:

Ronald Cools, *Katholieke Universiteit Leuven*
Luc Devroye, *McGill University*
Henri Faure, *CNRS Marseille*
Paul Glasserman, *Columbia University*
Peter W. Glynn, *Stanford University*
Stefan Heinrich, *University of Kaiserslautern*
Fred J. Hickernell, *Illinois Institute of Technology*
Aneta Karaivanova, *Bulgarian Academy of Science*
Alexander Keller, *mental images GmbH, Berlin*
Adam Kolkiewicz, *University of Waterloo*
Frances Y. Kuo, *University of New South Wales*
Christian Lécot, *Université de Savoie, Chambéry*
Pierre L'Ecuyer, *Université de Montréal* (Chair and organizer)
Jun Liu, *Harvard University*
Peter Mathé, *Weierstrass Institute Berlin*
Makoto Matsumoto, *Hiroshima University*
Thomas Müller-Gronbach, *Otto von Guericke Universität*
Harald Niederreiter, *National University of Singapore*
Art B. Owen, *Stanford University*
Gilles Pagès, *Université Pierre et Marie Curie (Paris 6)*
Klaus Ritter, *TU Darmstadt*
Karl Sabelfeld, *Weierstrass Institute Berlin*
Wolfgang Ch. Schmid, *University of Salzburg*
Ian H. Sloan, *University of New South Wales*
Jerome Spanier, *University of California, Irvine*
Bruno Tuffin, *IRISA-INRIA, Rennes*
Henryk Woźniakowski, *Columbia University*.

The local arrangements (program production, publicity, web site, registration, social events, etc.) were ably handled by Carole Dufour (GERAD), Marilyne Lavoie (GERAD), Louis Pelletier (CRM), Marie Perreault (GERAD), and Suzette Paradis (CRM). Francine Benoit (GERAD) helped with editing the proceedings.

This conference continued the tradition of biennial MCQMC conferences initiated by Harald Niederreiter. They were begun at the University of Nevada in Las Vegas, Nevada, USA, in June 1994 and followed by conferences at the University of Salzburg, Austria, in July 1996, the Claremont Colleges in Claremont, California, USA, in June 1998, Hong Kong Baptist University in Hong Kong, China, in November 2000, the National University of Singapore, Republic of Singapore, in November 2002, the Palais des Congrès in Juan-les-Pins, France, in June 2004, and Ulm University, Germany, in July 2006. The next MCQMC conference will be held in Warsaw, Poland, in August 2010.

The proceedings of these previous conferences were all published by Springer-Verlag, under the following titles:

- *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (H. Niederreiter and P.J.-S. Shiue, eds.),
- *Monte Carlo and Quasi-Monte Carlo Methods 1996* (H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof, eds.),
- *Monte Carlo and Quasi-Monte Carlo Methods 1998* (H. Niederreiter and J. Spanier, eds.),
- *Monte Carlo and Quasi-Monte Carlo Methods 2000* (K.-T. Fang, F.J. Hickernell and H. Niederreiter, eds.),
- *Monte Carlo and Quasi-Monte Carlo Methods 2002* (H. Niederreiter, ed.),
- *Monte Carlo and Quasi-Monte Carlo Methods 2004* (H. Niederreiter and D. Talay, eds.),
- *Monte Carlo and Quasi-Monte Carlo Methods 2006* (A. Keller and S. Heinrich and H. Niederreiter, eds.).

The program of the conference was rich and varied with over 135 talks being presented. Highlights were the invited plenary talks given by Josef Dick (University of New South Wales), Arnaud Doucet (University of British Columbia), Daan Frenkel (University of Cambridge), Paul Glasserman (Columbia University), Christiane Lemieux (University of Waterloo), Jun Liu (Harvard University), Klaus Ritter (TU Darmstadt), Jeffrey Rosenthal (University of Toronto), Wolfgang Schmid (University of Salzburg), and Andrew Stuart (Warwick University). The papers in this volume were carefully screened and cover both the theory and the applications of Monte Carlo and quasi-Monte Carlo methods.

We thank the anonymous reviewers for their reports and many others who contributed enormously to the excellent quality of the conference presentations and to the high standards for publication in these proceedings by careful review of the abstracts and manuscripts that were submitted.

We gratefully acknowledge generous financial support of the conference by the Centre de Recherches Mathématiques (CRM), the Groupe d'Études et de Recherche en Analyse de Décisions (GERAD), Mathematics for Information Technology

and Complex Systems (MITACS), and the American National Science Foundation (NSF).

Finally, we want to express our gratitude to Springer-Verlag for publishing this volume.

Pierre L'Ecuyer
July 2009 *Art Owen*

Contents

Part I
Tutorials

Monte Carlo and Quasi-Monte Carlo for Statistics

Art B. Owen

Abstract This article reports on the contents of a tutorial session at MCQMC 2008. The tutorial explored various places in statistics where Monte Carlo methods can be used. There was a special emphasis on areas where Quasi-Monte Carlo ideas have been or could be applied, as well as areas that look like they need more research.

1 Introduction

This survey is aimed at exposing good problems in statistics to researchers in Quasi-Monte Carlo. It has a mix of well known and not so well known topics, which both have their place in a research context. The selection of topics is tilted in the direction of problems that I have looked at. That enables me to use real examples, and examples are crucial to understanding statistics.

Monte Carlo methods are ubiquitous in statistics. Section 2 presents the bootstrap. It is a method of resampling the observed data to judge the uncertainty in a quantity. The bootstrap makes minimal assumptions about how the data were obtained. Some efforts at bringing balance to the resampling process have brought improvements, but they are not large enough to have made much impact on how the bootstrap is used. Permutation tests, considered in Section 3 have a similar flavor to the bootstrap, but there, efforts to impose balance can distort the results.

Markov chain Monte Carlo (Section 4) is used when we cannot directly sample the quantity of interest, but are at least able to find a Markov chain from whose stationary distribution the desired quantity can be sampled. Space limitations make it impossible to cover all of the topics from a three hour tutorial in depth. The work on QMC for MCMC has appeared in [31], [43] and in Tribble's dissertation [42], and so it is just sketched here.

Department of Statistics, Stanford University, Stanford CA, 94305
url: http://stat.stanford.edu/~owen

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_1,

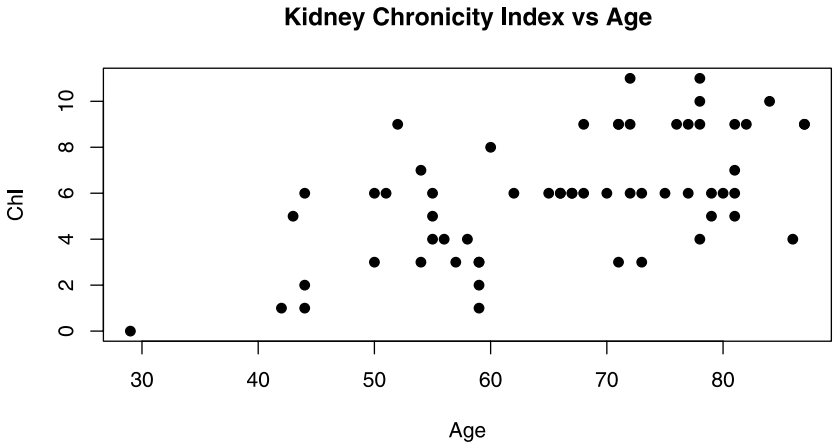


Fig. 1 A measure of kidney damage is plotted versus age for 60 subjects in [34].

Monte Carlo methods are used for search as well as for integration. Section 5 presents the method of least trimmed squares (LTS). This is the most effective method known for highly robust regression model fitting. It uses an ad hoc Monte Carlo search strategy. Quasi-Monte Carlo methods have been used in search problems [26, Chapter 6], usually to search over the unit cube [27]. The space to search over in LTS is of a combinatorial nature. Finally, Section 6 points to two more important problems from statistics where QMC may be useful.

2 The Bootstrap

Figure 1 plots a measure Y_i of kidney damage against age X_i, for 60 subjects studied in Rodwell et al. [34]. There is a clear tendency for older subjects to have greater kidney damage. This may be quantified through the correlation coefficient, which on this data takes the value 0.59. Since only 60 subjects were measured, we very much doubt that the true correlation taken over all people is exactly 0.59. Using ρ to denote a hypothetical true correlation and $\hat{\rho}$ to denote the measured one, we may just want to know the variance $V(\hat{\rho})$ of our estimate.

The variance of the sample correlation is not hard to find in closed form, so long as the data are a sample from the bivariate normal distribution. But we have no reason to expect that assumption is good enough to use. The bootstrap, introduced by Efron [5] provides a way out of that assumption. Operationally one does the following:

1. For $b = 1, \ldots, B$

2. Draw (X_i^{*b}, Y_i^{*b}), $1 \le i \le 60$ with replacement from the original data.
3. Compute $\hat{\rho}^{*b} = \hat{\rho}\big((X_1^{*b}, Y_1^{*b}), \ldots, (X_{60}^{*b}, Y_{60}^{*b})\big)$.
4. Return the variance of $\hat{\rho}^{*1}, \ldots \hat{\rho}^{*B}$.

Using $B = 9999$ the result came out to be 0.0081, so that the standard deviation of the $\hat{\rho}^*$ values is 0.090. What we actually got was a Monte Carlo estimate of the variance of the bootstrapped correlations $\hat{\rho}^*$ when resampling from the data. Even if we let $B \to \infty$ this would not be the same as the variance we want, which is that of $\hat{\rho}$ when sampling from the unknown true distribution of (X, Y) pairs. But bootstrap theory shows that the two variances become close quickly as the number n of sample values increases [7].

First impressions of the bootstrap are either that it is obviously ok, or that it is somehow too good to be true, like pulling oneself up by the bootstraps. The formal justification of the bootstrap begins with a statistical functional T defined on distributions F. In this case $T(F)$ gives the variance of the correlation measured on 60 pairs of data drawn from the distribution F on $\mathbb{R}^2$. Let F_0 be the unknown true distribution and $\widehat{F}_n$ be the distribution that puts equal probability $1/n$ on all n data points. As n increases $\widehat{F}_n$ approaches F_0. Then a continuity argument gives $T(\widehat{F}_n)$ approaching $T(F_0)$. The continuity argument holds in great generality but there are exceptions, as well as remedies in some of those cases [32].

The bootstrap can also be used to estimate and correct for biases in statistics. Let $\mathbb{E}(\hat{\rho} \mid F)$ denote the expected value of $\hat{\rho}$ when sampling n data pairs from F. Typically $\mathbb{E}(\hat{\rho}) \ne \rho$, so that the sample correlation is biased. The bootstrap estimate of the bias $B(F) \equiv \mathbb{E}(\hat{\rho} \mid F) - \rho(F)$ is $B(\widehat{F}_n) \equiv \mathbb{E}(\hat{\rho}^* \mid \widehat{F}_n) - \rho(\widehat{F}_n)$. We can estimate this bias by resampling. In the present example we find that the average value of $\hat{\rho}^* - \hat{\rho}$ in resampling is -0.0047. If we are worried that $\hat{\rho}$ is too small by 0.0047 we can add 0.0047 (i.e. subtract the estimated bias) and get 0.594 instead of 0.589. Here the bias adjustment is very small. That is typical unless the method used has a large number of parameters relative to the sample size.

Figure 2 shows a histogram of the 9,999 bootstrap samples used in this analysis. The histogram is skewed, centered near the original correlation, and is quite wide. The resampled correlations cut the real line into 10,000 intervals. A bootstrap 95% confidence interval is formed by taking the central 9500 of those intervals. If the values are sorted $\hat{\rho}^{*(1)} \le \hat{\rho}^{*(2)} \le \cdots \le \hat{\rho}^{*(9999)}$, then the 95% confidence interval goes from $\hat{\rho}^{*(250)}$ to $\hat{\rho}^{*(9750)}$. In this example we have 95% confidence that $0.391 \le \rho \le 0.755$. Similarly there is 99% confidence that $\hat{\rho}^{*(100)} \le \rho \le \hat{\rho}^{*(9900)}$, or $0.346 \le \rho \le 0.765$. Bootstrap confidence levels are not exact. They are approximate confidence intervals. Typically they have coverage probability equal to their nominal level plus $O(n^{-1})$. See [13]. The intervals presented here are known as percentile intervals. They are the simplest but not the only bootstrap confidence interval. See [7] for other choices.

The balanced bootstrap [4] is an attempt to improve on bootstrap re-sampling. Instead of sampling n observations with replacement B times, it forms a large pool of nB observations, with B copies of each original data point. Then it randomly partitions them into B subsets of equal size n. Those groups are treated as the bootstrap

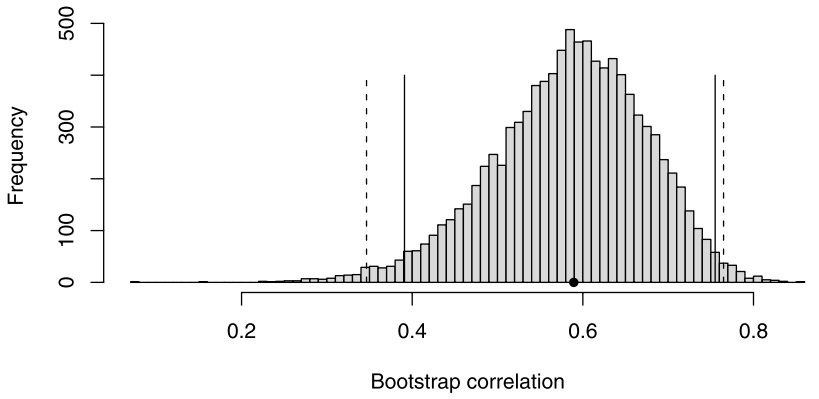


Fig. 2 This figure shows 9999 bootstrap resampled correlations for the kidney data. There is a reference point at the sample correlation. Solid vertical lines enclose the central 95% of the histogram. Dashed lines enclose the central 99%.

samples. Now each original observation appears the same number B of times among the sampled data. The balanced bootstrap is similar to QMC. Higher order balancing, based on orthogonal arrays, was proposed by [12], but that proposal requires the number n of observations to be a prime power.

To apply QMC, it helps to frame the bootstrap problem as an integral over $[0,1]^n$, as discussed in [28] and thoroughly investigated by Liu [21]. We may write X_i^* as $X_{\lceil nU_i \rceil}$ where $U_i \sim \mathbb{U}(0,1)$ are independent. Then $X^* = (X_1^*, \ldots, X_n^*)$ is a function of $U \sim \mathbb{U}(0,1)^n$. The bootstrap estimate of bias is a Monte Carlo estimate on B samples of $\int_{[0,1]^n} Q(U)\,dU$ for

$$Q(U) = T(X^*(U)) - T(X), \tag{1}$$

where $T(X)$ is a shorthand for $T(n^{-1}\sum_{i=1}^n \delta_{X_i})$ with δ_x the point mass distribution at x. The bootstrap estimate of variance has integrand

$$Q(U) = \left(T(X^*(U)) - \int_{[0,1]^n} T(X^*(U))\,dU \right)^2. \tag{2}$$

The upper end of a bootstrap 95% confidence interval is the solution $T^{0.975}$ to $\int_{[0,1]^n} Q(U)\,dU = 0.975$ where

$$Q(U) = 1_{T(X^*(U)) \le T^{0.975}}, \tag{3}$$

while the lower end uses 0.025 instead of 0.975.

To implement the ordinary bootstrap we take points $U_b = (U_{b1}, \ldots, U_{bn}) \sim \mathbb{U}(0,1)^n$ for $b = 1, \ldots, B$ and use ordinary Monte Carlo estimates of the integrals in (1), (2) and (3). To get a QMC version, we replace these points by a B point QMC rule in $[0,1]^n$.

The integrands Q are generally not smooth, because $X_{\lceil nU_i \rceil}$ is discontinuous in U_i, apart from trivial settings. Smoothness is important for QMC to be effective. Fortunately, there is a version of the bootstrap that yields smooth integrands, at least for estimating bias and variance. The weighted likelihood bootstrap (WLB), proposed by Newton and Raftery [24] uses continuous reweighting of the data points instead of discrete resampling. It is a special case of the Bayesian bootstrap of Rubin [38]. Where the ordinary bootstrap uses $T(n^{-1}\sum_{i=1}^n \delta_{X_i^*})$, the WLB uses $T(\sum_{i=1}^n w_i \delta_{X_i})$ for certain random weights w_i. To generate these weights put $v_i = -\log(U_i)$ and $w_i = v_i / \sum_{j=1}^n v_j$. Then for uniform U_i we find that v_i has the standard exponential distribution, while $w = (w_1, \ldots, w_n) = w(U)$ has a Dirichlet distribution. We substitute smooth reweighting for resampling, by using $T(\sum_{i=1}^n w_i(U)\delta_{X_i})$ in place of $T(X^*(U))$ in equations (1), (2), and (3). In a QMC version of the WLB, we take B points with low discrepancy in $[0,1]^n$ and use them as the uniform numbers that drive the reweighting.

The WLB does not give the same answers as the original bootstrap. But the original bootstrap is not exact, only asymptotically correct as $n \to \infty$. The same is true of the WLB.

In her dissertation, Ruixue Liu [21] compared various QMC methods for bootstrap problems. The underlying problem was to measure the correlation over law schools, of the average LSAT score and grade point average of newly admitted students.

In addition to the methods described above, she considered several QMC and QMC-like constructions, as follows. Latin hypercube sampling (LHS)[22] provides a sample in which each of the sample coordinates $U_{1i}, \ldots, U_{Bi}$ for $i = 1, \ldots, n$ is simultaneously stratified into intervals of width $1/B$. Randomized orthogonal array sampling [29] stratifies the bivariate or trivariate margins of the distribution of U_b. When the array has strength t, then all $\binom{n}{t}$ of the t–dimensional margins are stratified, typically into cubical regions of width $B^{-1/t}$. Orthogonal array based LHS [41] has both the LHS and the orthogonal array stratifications. Scrambled nets [30] are a randomization of a QMC method (digital nets).

Some numerical results from Liu [21] are shown in Tables 1 and 2. Bootstrap estimates of the bias, variance and 95'th percentile were repeatedly computed and their variance was found. Those variances are based on 2000 replications of each method, except for scrambled nets for which only 100 replications were used. The variances are presented as variance reduction factors relative to the variance of the plain resampling bootstrap. For example, we see that the LHS version of resampling is about 10 times as efficient as the ordinary bootstrap on the bias estimation problems.

Several trends are apparent in these results. Better variance reductions are obtained via reweighting than resampling, as one would expect because the former has smoother integrands. It is easiest to improve on bootstrap bias estimates, harder

Table 1 This table shows variance reduction factors attained from applying QMC methods to the three bootstrap problems described in the text. The values are normalized so that the ordinary bootstrap gets a value of 1.0 in each problem. Each bootstrap method used $B = 61^2 = 3721$ resamples. The quantity being bootstrapped was a sample correlation. Both reweighting and resampling bootstraps were used to estimate bias, variance and 95'th percentile. The orthogonal arrays had strength 2. The balanced bootstrap is only defined for the resampling approach.

	Resampling			Reweighting		
Method	Bias	Var	Perc	Bias	Var	Perc
Plain bootstrap	1.0	1.0	1.0	1.4	1.4	1.2
Balanced bootstrap	10.8	1.2	1.2			
Latin hypercube sampling	10.7	1.2	1.3	16.4	2.3	1.6
Randomized orthogonal array	15.7	3.1	1.7	105.2	8.3	2.9
OA-based LHS	36.0	3.2	1.6	126.1	9.0	2.7
Scrambled $(0, 61^2, 15)$–net in base 61	30.0	3.1	1.8	116.9	8.8	3.3

for variance estimates, and hardest for confidence intervals. Again we would expect this. In the von Mises expansion (e.g. [8]) the statistic T is approximated by a sum of functions of one observation at a time. When that approximation is accurate, then the bias integrands are more nearly additive in the n inputs while the variance integrands are nearly the square of an additive approximation and we might expect the resulting statistical function to be of low effective dimension in the superposition sense [2]. Unfortunately, the interest in usual applied settings is in the reverse order, percentiles, then variance then bias. Finally orthogonal array methods with high strength and few levels do poorly

3 Permutation Tests

Newborn babies have a walking reflex, in which their feet start a walking motion when placed in contact with a surface. Zelazo et al. [44] conducted an experiment to test whether regular daily encouragement of this reflex would result in babies

Table 2 This table shows the same quantities as Table 1, except that now $B = 17^3 = 4913$ and the orthogonal arrays had strength 3.

	Resampling			Reweighting		
Method	Bias	Var	Perc	Bias	Var	Perc
Plain bootstrap	1.0	1.0	1.0	1.4	1.4	1.1
Balanced bootstrap	9.9	1.2	1.1			
Latin hypercube sampling	10.2	1.2	1.3	15.7	2.5	1.5
Randomized orthogonal array	3.3	0.6	0.4	8.5	0.7	0.3
OA-based LHS	7.5	0.6	0.4	8.8	0.7	0.3
Scrambled $(0, 17^3, 15)$–net in base 17	60.7	6.5	2.2	756.0	33.6	2.9

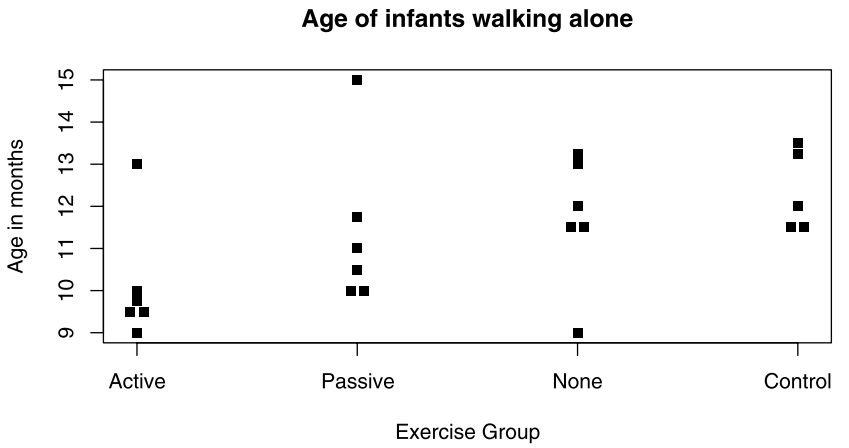


Fig. 3 Shown are ages at which babies learned to walk in each of 4 subject groups.

learning to walk at an earlier age, than without the encouragement. Figure 3 shows the results of this experiment.

There were twenty-four babies in the experiment. The age at which they could walk alone was reported for twenty-three of them. One group (active) had their reflex stimulated daily for eight weeks. Another (passive) had gross-motor stimulation of a different kind. A third group (none) got neither stimulation but was regularly tested for walking ability. The fourth group (control) was not even tested until a greater age.

For illustration, we will focus on the simple problem of testing whether the active group learned to walk sooner than the 'none' group. For differing statistical analyses of all four groups see [1, Chapter 10] and [44].

The active group learned to walk at an average age of $\bar{X} = 10.125$ months compared to $\bar{Y} = 11.7$ months for the none group. They were faster on average by 1.58 months, but perhaps this difference can be attributed to chance.

A permutation test can be used to judge the difference. The two groups combined have 12 babies. We could select 6 of them to be the active group in any of $\binom{12}{6} = 924$ ways. If there really is no difference in the distribution of walking age between the groups, then we could permute the active versus none labels without changing the distribution of the outcome. After such a permutation of the labels, we get new averages $\bar{X}^*$ and $\bar{Y}^*$. The original permutation has a 1 in 924 chance of giving the largest mean difference of them all, if the two groups are identical.

Figure 4 shows a histogram of all 924 permuted group mean differences $\bar{X}^* - \bar{Y}^*$. There is a vertical reference line at the observed value of $\bar{X} - \bar{Y} = -1.58$. There were only 49 values less than or equal to -1.58. By symmetry there were 49 values at least $+1.58$ and so there were 98 values as or more extreme than that observed.

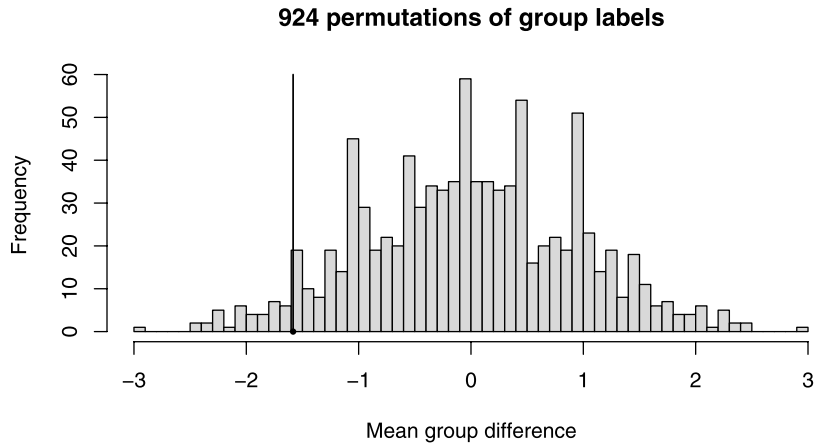


Fig. 4 The histogram shows all 924 values of $\bar{X}^* - \bar{Y}^*$ obtainable by permuting the labels (control versus none) for 12 of the babies learning to walk. The vertical reference line is at the observed value $\bar{X} - \bar{Y}$.

This permutation test allows one to claim a two-tailed p-value of $98/924 \doteq 0.106$ for the difference. A difference this large could arise by chance with probability about 10.6%. The observed difference is not at all unusual. Some statisticians (not including the author) would claim a p-value of $49/924$ here, which is closer to 5%, the conventional line at which results begin to be taken seriously.

In balanced permutations [6] we compare the two groups in a more carefully controlled way. Each time the labels are reassigned, we ensure that three members of the new treatment group come from the old treatment group and that three come from the old control group. The other six babies are similarly balanced, and they become the relabeled control group. There are $\binom{6}{3}^2 = 400$ balanced permutations for this data. It is customary to include the original sample, from an identity permutation, in an MC based permutation analysis. This avoids the possibility of $p = 0$. The identity permutation is not a balanced permutation, and so adding it in to the reference set here gives a histogram of 401 values.

The intuitive reason for balancing the permutations is as follows. If there is no difference between the groups, then the balanced permutations still give rise to the same distribution of the treatment difference as the real sample has. But when there is a difference, for example the treatment group learn to walk two months earlier on average, then things change. In relabeling we get $\bar{X}^*$ for treatment and $\bar{Y}^*$ for control. In some unbalanced permutations all or most of $\bar{X}^*$ came from the original control group while in others few or none came from there. In balanced permutations exactly half of the values contributing to $\bar{X}^*$ are at the high level and half at the low level. A mean difference between the two original groups will mostly cancel for the relabeled groups. As a result, the histogram of $\bar{X}^* - \bar{Y}^*$ for balanced permutations

can be expected to be narrower than the one for full permutations, when there is a treatment difference. Narrower histograms lead to smaller reported p-values. In the baby walking data, we find that $|\bar{X}^* - \bar{Y}^*| \geq |\bar{X} - \bar{Y}|$ holds for only 27 points in the reference set, yielding a p-value of $27/401 = 0.067$, which is smaller than for the full permutation set.

While balanced permutations have the potential to sharpen inferences, they have been applied without theoretical support. In simulations they have been found to give p–values that are too permissive. The problem with balanced permutations is that they do not form a group under composition. The group property is a key ingredient in the permutation argument [17]. Some results in [40] show that the chance of the original permutation beating all $\binom{n}{2}^2$ balanced permutations is much larger than $(1+\binom{n}{2}^2)^{-1}$ even when two groups of size n have identical distributions. The p values are too small by a factor that grows quickly with n and is already over 100 for $n = 10$.

It may be possible to repair balanced permutations, although this looks difficult at present. One approach is to try to compensate by adjusting the reported p values. Another is to search for a suitable subgroup of permutations to use.

4 MCMC

Usually in QMC problems, we write the desired answer as an integral $\mu = \int_{[0,1]^d} f(x)\,dx$. The function f takes independent uniformly distributed quantities, transforms them into the desired ones, such as dependent non-uniform values, and then computes whatever it is we want to average as a function of those values. As is well known [26], QMC methods achieve better rates of convergence than MC on such problems, making only modest smoothness assumptions on f.

In some applications however, we seek a value $\mu = \int_{\mathbb{X}} f(x)\pi(x)\,dx$ where there is no practical way to turn uniform random variables into the desired ones from π on the state space $\mathbb{X}$. We might be able to get $x \sim \pi$ by rejection sampling but with such an unfavorable acceptance rate that the method would be useless. This issue arises commonly in Bayesian computations, for statistics [10] and machine learning, as well as in the physical sciences [23].

In MCMC we generate $x_i = \phi(x_{i-1}, v_i)$ where $v_i \sim \mathbb{U}(0,1)^d$. The distribution of x_i depends on $x_{i-1}, x_{i-2}, \ldots$ only through the immediate predecessor x_{i-1}, and so it has the Markov property. With some skill and care, one can often choose ϕ so that the Markov chain has π as its stationary distribution. Sometimes it is also important to choose a good starting point x_0. Under reasonable conditions there is a law of large numbers for MCMC, so that

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \to \mu$$

and there are also central limit theorems for MCMC. The theory and practice of MCMC in statistics is presented in several books, including Liu [20] and Robert and Casella [33].

The quantity $\hat{\mu}_n$ depends on n values $v_i \sim \mathbb{U}(0,1)^d$. Its expectation is thus an nd dimensional integral which approximates, but does not equal, the desired one. Unlike crude Monte Carlo, there is a bias in MCMC. Under reasonable conditions it decays exponentially with n and so is often very small. Some other times the exponential decay is still too slow for practically useful problems, and so the constant in the exponent matters. For some studies of the bias, see Rosenthal [35].

There have been some efforts to replace n vectors v_i by quasi-Monte Carlo points. The key idea is to open up the vectors v_i into one long sequence $u_1, u_2, u_3, \ldots, u_{nd}$ where $v_i = (u_{d(i-1)+1}, u_{d(i-1)+2}, \ldots, u_{di})$. Then one replaces the independent and identically distributed (IID) points u_i for $i = 1, \ldots, nd$ by some alternative points with good equidistribution properties. Naive substitution of QMC points v_i into MCMC can fail very badly.

An early effort by Liao [19] has proven effective. Liao's approach is to take a QMC point set $a_1, \ldots, a_n \in [0,1]^d$ and randomly reorder it getting $v_i = a_{\tau(i)}$ where τ is a random permutation of $\{1, \ldots, n\}$. Then the reordered points $v_1, v_2, \ldots, v_n$ are concatenated into a single vector of nd values in $[0,1]$ with which to drive the MCMC.

The thesis of Tribble [42] has an up to date account, extending the work published in [31] and [43], and giving methods that do even better than Liao's, in numerical examples.

What is known so far is that a completely uniformly distributed (CUD) sequence $u_1, u_2, \ldots$ can be used in place of IID points. In that case the QMC answer converges to the correct result, at least for Markov chains with finite state spaces. The main theoretical technique is a coupling argument first made by Chentsov [3] for sampling Markov chains by inversion but then extended by Owen and Tribble [31] to handle Metropolis-Hastings sampling. As of 2008 the continuous state space case had not been handled. It is now covered in a technical report by Chen, Dick and Owen.

A CUD sequence is one that can be grouped into overlapping d-tuples $z_i = (u_i, \ldots, u_{i+d-1})$ such that the d dimensional star–discrepancy of $z_1, \ldots, z_{n-d+1}$ tends to 0. This must hold for all d. Extensions for random sequences and for limits of finite length sequences are given in [43]. If points u_i are independent $\mathbb{U}(0,1)$ then z_i have discrepancy that converges to 0 with probability one. But specially constructed sequences can have smaller discrepancies and hence may be more accurate.

Using a CUD sequence can be likened to using the entire period of a pseudo-random number generator. This is an old suggestion of Niederreiter [25]. Quite a different approach is to drive multiple copies of Markov chains by QMC, with reordering between steps. See for example L'Ecuyer, Lécot and Tuffin [16] and earlier work by Lécot [15].

The best numerical results for MCQMC so far used some small linear feedback shift registers, in Tribble [42]. He gets variance reduction factors well over 1,000 for the posterior means of parameters using the Gibbs sampler on the well known pump failure data set. For a higher dimensional vasorestriction data set of [9] he

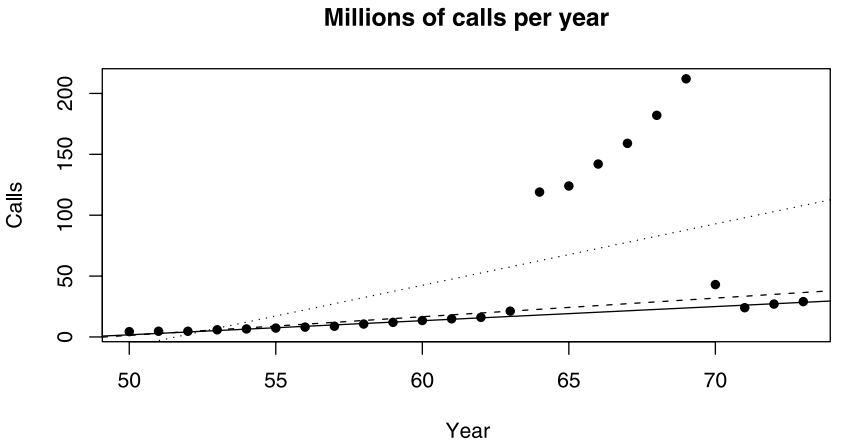


Fig. 5 This plot shows phone calls versus year. Some data values (plotted as open circles) were corrupted by counts of minutes instead of calls. From top to bottom at the right, the three fitted lines are least squares regression (dots), L_1 regression (dashes), and least trimmed squares (solid).

obtains variance reduction factors up to 100. In both cases the variance reduction factors increase with sample size. Switching from IID to CUD to randomized CUD points brings the best improvements when the function $\phi(x, v)$ is smooth. This occurs for the Gibbs sampler, which randomly updates components of x one at a time given the others. More general Metropolis-Hastings sampling methods typically have acceptance-rejection steps which make for discontinuous integrands and lessened improvements. Still, the Gibbs sampler is important enough that improvements of it are worth pursuing.

The theoretical results so far may be likened to the strong law of large numbers. They indicate that for large enough n, the answer converges. What is missing is an analogue of the central limit theorem, or of the Koksma-Hlawka inequality, to say how fast the convergence takes place. Furthermore, not enough is known about the speed with which discrepancies (for varying d) of CUD sequences can vanish. A survey of CUD constructions is given by [18].

5 Least Trimmed Squares

Figure 5 shows the Belgian telephone data of [37]. The data were supposed to portray the number of calls per year (in millions) as a function of the year (minus 1900). As it turned out minutes, and not calls, were counted for a period starting in late 1963 and ending in early 1970. The errors in the data make a big difference to the regression line, fit by least squares.

Errors or other data contamination of this nature are not as rare as we would like. In the present setting we can clearly see that something is amiss, but in problems where dozens or even thousands of explanatory variables are used to make predictions, gross errors in the data might not be easy to see.

A robust line would be better suited to this problem. Robust methods are those that are less affected by bad data. An old, but still very good reference on robust statistical methods is the book by Huber [14].

The least squares regression line shown Figure 5 was found by minimizing $\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 X_i)^2$ over β_0 and β_1, where Y_i is the i'th phone measure and X_i is the i'th year. The largest errors dominate the sum of squares, and so least squares is far from robust. A natural alternative is to minimize the L_1 error $\sum_{i=1}^{n}|Y_i - \beta_0 - \beta_1 X_i|$ instead. It can be fit by quantile regression, where it generalizes the sample median to the regression context. As such, this choice is less affected by outliers. It brings a big improvement for this data, but it is still not robust, and gets fooled badly on other data.

The current state of the art in robust fitting is to sum most of the squared errors, but not the large ones. This is the Least Trimmed Squares (LTS) method of [37]. For any $\beta = (\beta_0, \beta_1)$ let $e_i(\beta) = |Y_i - \beta_0 - \beta_1 X_i|$, and then sort these absolute errors: $e_{(1)}(\beta) \le e_{(2)}(\beta) \le \cdots \le e_{(n)}(\beta)$. We choose β to minimize

$$f(\beta) = \sum_{i=1}^{\lfloor \alpha n \rfloor} e_{(i)}(\beta)^2.$$

If we were confident that fewer than 20% of the data were bad, we could take $\alpha = 0.8$. The smallest workable value for α is $(n + p + 1)/(2n)$, which allows for just under half the data to be bad.

Figure 5 shows the least trimmed squares fit. It goes through the good points and is oblivious to the bad ones.

Figure 6 shows residuals $Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i$ for estimates $\hat{\beta}$ fit by least squares and by least trimmed squares. In least squares, the good observations have residuals of about the same size as the bad ones. For least trimmed squares, the residuals from bad points are much farther from zero than those from good ones. A data analyst could then easily decide that those points need further investigation.

When there is only one explanatory variable, then there are fast algorithms to find the least trimmed squares estimates. But when there are many such variables then the best known fitting strategy is a Monte Carlo search of Rousseeuw and van Driessen [36]. Their search is guided by some theory.

Consider a general linear model, which predicts Y_i by $\sum_{j=1}^{p}\beta_j X_{ij} = \beta' X_i$ for $p \ge 2$ and $n \ge p$. It is known that the LTS solution $\hat{\beta}$ solves $Y_i = \hat{\beta}' X_i$ for p of the points $i = 1, \ldots, n$. Thus the solution can be found by checking $\binom{n}{p}$ interpolating models. The cost of checking all those models though is often too high. They use instead a Monte Carlo strategy.

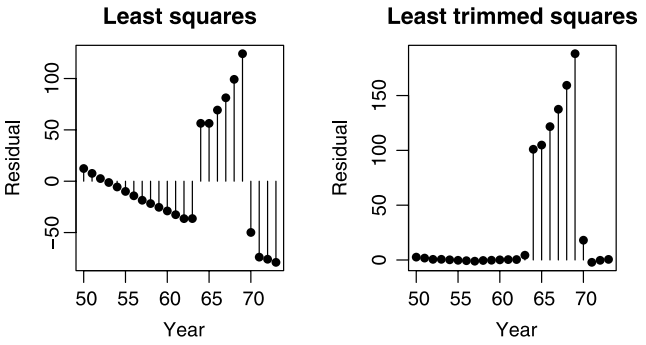


Fig. 6 The left panel shows the residuals from least squares. The good points get errors of about the same size as the bad ones. The right panel shows residuals from least trimmed squares. The bad points get very large residuals, making them numerically conspicuous.

One of their Monte Carlo search methods is presented in Figure 7. It is the one recommended when $n \leq 600$.

The search in Figure 7 has clearly been tuned empirically, for speed and effectiveness. For example the C–step brings an improvement, but they found diminishing returns to fully iterated C-steps. It is better to generate many candidates and then follow up only the best ones.

The algorithm makes numerous choices that seem arbitrary. There is clearly room for a better understanding of how to search for an optimum.

1. Sample p points: $(Y_i, X_{i1}, \ldots, X_{ip}) \quad i \in \mathbb{I} \subset \{1, \ldots, n\} \quad |\mathbb{I}| = p$
2. While linear interpolation of Y_i to X_i is not unique, add one more sample point
3. Find $\hat{\beta}$ to interpolate $Y_i = X_i'\hat{\beta}$ on sampled points.
4. Find points with the smallest $h = \lfloor n\alpha \rfloor$ absolute residuals (among all n points).
5. C-step: Fit LS to the h points and find newest h points with smallest absolute residuals
6. Do 2 more C-steps
7. Repeat steps 1 through 6, 500 times, keeping 10 best results
8. Run C-steps to convergence for these 10
9. Select best of those 10 end points

Fig. 7 This is an outline of the Monte Carlo search algorithm of [36] for solving the least trimmed squares regression problem, when $n \leq 600$.

6 Other Methods

Some other uses of MC and QMC in statistics are very important but were not described here. Notable among the gaps is the problem of fitting generalized linear mixed models. Some efforts at this problem via QMC are reported in [39]. This problem is very important in statistical applications. It features integrands which can become very spiky in even moderately high dimensions and practical problems can involve quite high dimension. Another classical quadrature problem arising in statistics is that of integrating a probability density function of dependent random variables, (e.g. Gaussian or multivariate t) over a rectangular region. For recent work in this area see [11].

Acknowledgements I thank Pierre L'Ecuyer and the other organizers of MCQMC 2008 for inviting this tutorial, and for organizing such a productive meeting. Thanks also to two anonymous reviewers and Pierre for helpful comments. This research was supported by grant DMS-0604939 of the U.S. National Science Foundation.

References

1. Brown, B.W., Hollander, M.: Statistics, A Biomedical Introduction. John Wiley & Sons, New York (1977)
2. Caflisch, R.E., Morokoff, W., Owen, A.B.: Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. Journal of Computational Finance **1**, 27–46 (1997)
3. Chentsov, N.: Pseudorandom numbers for modelling Markov chains. Computational Mathematics and Mathematical Physics **7**, 218–2332 (1967)
4. Davison, A.C., Hinkley, D.V., Schechtman, E.: Efficient bootstrap simulation. Biometrika **73**(3), 555–566 (1986)
5. Efron, B.: Bootstrap methods: Another look at the jackknife. The Annals of Statistics **7**, 1–26 (1979)
6. Efron, B., Tibshirani, R., Storey, J.D., Tusher, V.: Empirical Bayes analysis of a microarray experiment. Journal of the American Statistical Association **96**, 1151–1160 (2001)
7. Efron, B.M., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall (1993)
8. Fernholz, L.T.: von Mises calculus for statistical functionals. Springer-Verlag, New York (1983)
9. Finney, D.J.: The estimation from individual records of the relationship between dose and quantal response. Biometrika **34**, 320–334 (1947)
10. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis. Chapman & Hall, Boca Raton, FL (2003)
11. Genz, A., Bretz, F., Hochberg, Y.: Approximations to multivariate t integrals with application to multiple comparison procedures. In: Recent Developments in Multiple Comparison Procedures, vol. 47, pp. 24–32. Institute of Mathematical Statistics (2004)
12. Graham, R.L., Hinkley, D.V., John, P.W.M., Shi, S.: Balanced design of bootstrap simulations. Journal of the Royal Statistical Society, Series B **52**, 185–202 (1990)
13. Hall, P.G.: The Bootstrap and Edgeworth Expansion. Springer, New York (1992)
14. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
15. Lécot, C.: Low discrepancy sequences for solving the Boltzmann equation. Journal of Computational and Applied Mathematics **25**, 237–249 (1989)

16. L'Ecuyer, P., Lécot, C., Tuffin, B.: A randomized Quasi-Monte Carlo simulation method for Markov chains. Operations Research **56**(4), 958–975 (2008)
17. Lehmann, E.L., Romano, J.P.: Testing Statistical Hypotheses, third edn. Springer, New York (2005)
18. Levin, M.B.: Discrepancy estimates of completely uniformly distributed and pseudo-random number sequences. International Mathematics Research Notices pp. 1231–1251 (1999)
19. Liao, L.G.: Variance reduction in Gibbs sampler using quasi random numbers. Journal of Computational and Graphical Statistics **7**, 253–266 (1998)
20. Liu, J.S.: Monte Carlo strategies in scientific computing. Springer, New York (2001)
21. Liu, R.: New findings of functional ANOVA with applications to computational finance and statistics. Ph.D. thesis, Stanford University (2005)
22. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–45 (1979)
23. Newman, M.E.J., Barkema, G.T.: Monte Carlo Methods in Statistical Physics. Oxford University Press, New York (1999)
24. Newton, M.A., Raftery, A.E.: Approximate Bayesian inference with the weighted likelihood bootstrap (disc: P26-48). Journal of the Royal Statistical Society, Series B, Methodological **56**, 3–26 (1994)
25. Niederreiter, H.: Multidimensional integration using pseudo-random numbers. Mathematical Programming Study **27**, 17–38 (1986)
26. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. S.I.A.M., Philadelphia, PA (1992)
27. Niederreiter, H., Peart, P.: Quasi-Monte Carlo optimization in general domains. Caribbean Journal of Mathematics **4**(2), 67–85 (1985)
28. Owen, A.B.: Discussion of the paper by Newton and Raftery. Journal of the Royal Statistical Society, Series B **56**(1), 42–43 (1994)
29. Owen, A.B.: Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays. The Annals of Statistics **22**, 930–945 (1994)
30. Owen, A.B.: Randomly permuted (t,m,s)-nets and (t,s)-sequences. In: H. Niederreiter, P.J.S. Shiue (eds.) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, pp. 299–317. Springer-Verlag New York (1995)
31. Owen, A.B., Tribble, S.D.: A quasi-Monte Carlo Metropolis algorithm. Proceedings of the National Academy of Sciences **102**(25), 8844–8849 (2005)
32. Politis, D.N., Romano, J.P., Wolf, M.: Subsampling. Springer, New York (1999)
33. Robert, C., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer, New York (2004)
34. Rodwell, G., Sonu, R., Zahn, J.M., Lund, J., Wilhelmy, J., Wang, L., Xiao, W., Mindrinos, M., Crane, E., Segal, E., Myers, B., Davis, R., Higgins, J., Owen, A.B., Kim, S.K.: A transcriptional profile of aging in the human kidney. PLOS Biology **2**(12), 2191–2201 (2004)
35. Rosenthal, J.S.: Minorization conditions and convergence rates for Markov chain Monte Carlo. Journal of the American Statistical Association **90**, 558–566 (1995)
36. Rousseeuw, P.J., Driessen, van K.: Computing LTS regression for large data sets. Data Mining and Knowledge Discovery **12**, 29–45 (2006)
37. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley, New York (1987)
38. Rubin, D.B.: The Bayesian bootstrap. The Annals of Statistics **9**, 130–134 (1981)
39. Sloan, I.H., Kuo, F.Y., Dunsmuir, W.T., Wand, M., Womersley, R.S.: Quasi-Monte Carlo for highly structured generalised response models. Tech. rep., University of Wollongong Faculty of Informatics (2007)
40. Southworth, L.K., Kim, S.K., Owen, A.B.: Properties of balanced permutations. Journal of Computational Biology **16** (2009). In press.
41. Tang, B.: Orthogonal array-based Latin hypercubes. Journal of the American Statistical Association **88**, 1392–1397 (1993)
42. Tribble, S.D.: Markov chain Monte Carlo algorithms using completely uniformly distributed driving sequences. Ph.D. thesis, Stanford University (2007)

43. Tribble, S.D., Owen, A.B.: Construction of weakly CUD sequences for MCMC sampling. Electronic Journal of Statistics **2**, 634–660 (2008)
44. Zelazo, P.R., Zelazo, N.A., Kolb, S.: Walking in the newborn. Science **176**, 314–315 (1972)

Monte Carlo Computation in Finance

Jeremy Staum

Abstract This advanced tutorial aims at an exposition of problems in finance that are worthy of study by the Monte Carlo research community. It describes problems in valuing and hedging securities, risk management, portfolio optimization, and model calibration. It surveys some areas of active research in efficient procedures for simulation in finance and addresses the impact of the business context on the opportunities for efficiency. There is an emphasis on the many challenging problems in which it is necessary to perform several similar simulations.

1 Introduction

This tutorial describes some problems in finance that are of interest to the Monte Carlo research community and surveys some recent progress in financial applications of Monte Carlo. It assumes some familiarity with Monte Carlo and its application to finance: for an introduction, see [24, 46]. For quasi-Monte Carlo methods in finance, see [46, 72]. Section 2 provides an overview of financial simulation problems and establishes notation. Section 3 describes aspects of the business context for simulation in the financial industry and the implications for researchers. The principal theme of this tutorial is the need to solve multiple similar simulation problems and the associated opportunity to design efficient Monte Carlo procedures. The mathematical settings in which multiple similar problems arise, and the tools researchers use to deal with them, occupy Section 4. Section 5, on variance reduction, surveys database Monte Carlo and adaptive Monte Carlo. Section 6 is devoted to simulation for risk management. American options and portfolio optimization are covered in Section 7. Section 8 surveys sensitivity analysis by Monte Carlo. Some

Department of Industrial Engineering and Management Sciences, Robert R. McCormick School of Engineering and Applied Science, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208-3119, USA
url: http://users.iems.northwestern.edu/~staum/

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_2,

recent progress in simulating solutions of stochastic differential equations appears in Section 9.

2 Overview of Financial Simulation Problems

Financial simulation models involve a vector stochastic process $\mathbf{S}$ of *underlying* financial variables. Let $\mathbf{S}(t)$ be the value of $\mathbf{S}$ at time t and S_j be the jth component of $\mathbf{S}$. The model is expressed as a probability measure P governing $\mathbf{S}$. A characteristic feature of finance is that valuation calls for using another measure Q, derived from P and a choice of *numéraire*, or unit of account. For example, in the Black-Scholes model, one may take the numéraire to be S_0, a money market account earning interest at a constant rate r. Its value $S_0(t) = e^{rt}$. In this model, under the *real-world measure* P, the stock price S_1 is geometric Brownian motion with drift μ and volatility σ. Using the money market account as numéraire leads to the *risk-neutral measure* Q, under which S_1 is geometric Brownian motion with drift r and volatility σ. The real-world expected price of the stock at a future time T is $\mathrm{E}_P[S_1(T)] = S_1(0)e^{\mu T}$. The stock's value now, at time 0, is $S_0(0)\mathrm{E}_Q[S_1(T)/S_0(T)] = S_1(0)$. In general, $S_0(0)\mathrm{E}_Q[H/S_0(T)]$ is a value for a security whose payoff is H at time T. Thus, we use P to simulate the real world, but we simulate under Q to value a security.

Figure 1 shows how P and Q enter into the four interrelated problems of valuing and hedging securities, risk management, portfolio optimization, and model calibration. The model specifies security values as expectations under Q. Sensitivities of these expectations, to the underlying and to the model's parameters, are used in hedging to reduce the risk of loss due to changes in those quantities. In addition to these sensitivities, real-world probabilities of losses are important in risk management. Simulating scenarios under P is one step in sampling from the distribution of profit and loss (P&L). A portfolio's P&L in each scenario involves securities' values in that scenario, and they are conditional expectations under Q. The same structure can arise in portfolio optimization, where the goal is to choose the portfolio strategy that delivers the best P&L distribution. Calibration is a way of choosing a model's parameters. It is very difficult to estimate the parameters of P statistically from the history of the underlying. Instead, one may choose the parameters of Q so that the prices of certain securities observed in the market closely match the values that the model assigns to them.

Before elaborating on these four problems, we establish some notation. We discretize a time interval $[0, T]$ into m steps, considering the times $0 = t_0, t_1, \ldots, t_m = T$, and let $\mathcal{F}_i$ represent the information available at step i after observing $\mathbf{S}(t_0)$, $\mathbf{S}(t_1)$, $\ldots$, $\mathbf{S}(t_i)$. In applying Monte Carlo, we aim to estimate an expectation or integral $\mu = \mathrm{E}[Y] = \int f(\mathbf{u})\,\mathrm{d}\mathbf{u}$. The domain of integration is often omitted; it is understood to be $[0, 1)^d$ when the variable of integration is $\mathbf{u}$. We often ignore the details of how to simulate the random variable $Y = f(\mathbf{U})$, where $\mathbf{U}$ is uniformly distributed on $[0, 1)^d$. Such details remain hidden in the background: when we generate a point set $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in order to estimate μ by $\sum_{i=1}^n f(\mathbf{u}_i)/n$, each vector $\mathbf{u}_i$ results in a sim-

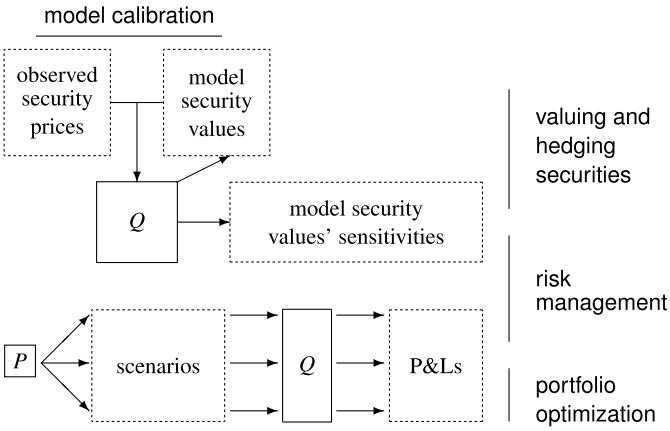


Fig. 1 Ecology of computations in finance.

ulated *path* $\mathbf{S}^{(i)}(t_1), \ldots, \mathbf{S}^{(i)}(t_m)$, where the superscript (i) indicates that this path is generated by the ith point or *replication*. The mapping ϕ from point to path is such that when $\mathbf{U}$ is uniformly distributed on $[0,1)^d$, $\phi(\mathbf{U})$ has the finite-dimensional distribution specified by P or Q, as appropriate. Sometimes we explicitly consider the intermediate step of generating a random vector $\mathbf{X}$ before computing the random variable $Y = \tilde{f}(\mathbf{X})$. We will often consider the influence of a parameter vector $\boldsymbol{\theta}$, containing initial values of the underlying, parameters of the model, characteristics of a security, or decision variables. In full generality,

$$\mu(\boldsymbol{\theta}) = \mathrm{E}[Y(\boldsymbol{\theta})] = \int_{[0,1)^d} f(\mathbf{u};\boldsymbol{\theta})\,\mathrm{d}\mathbf{u} = \int \tilde{f}(\mathbf{x};\boldsymbol{\theta}) g(\mathbf{x};\boldsymbol{\theta})\,\mathrm{d}\mathbf{x} = \mathrm{E}_{\boldsymbol{\theta}}[\tilde{f}(\mathbf{X};\boldsymbol{\theta})].$$

Derivative Securities. *Derivative securities* have payoffs that are functions of the underlying. In many models, the market is *complete*, meaning that the derivative security's payoff can be replicated by trading in the underlying securities. Then, in the absence of arbitrage, the derivative security's value should equal the cost of setting up that replicating strategy, and this is an expectation under Q [46, §1.2]. For a survey of ideas about how to price a derivative security whose payoff can not be replicated, see [92]. According to some of these ideas, price bounds are found by optimizing over hedging strategies or probability measures. Computational methods for these price bounds have received little attention; exceptions are [75, 84].

The *Greeks*, sensitivities of derivative security values to the underlying or to model parameters, are used to measure and to hedge the risk of portfolios. For example, where $\Delta_j = \mu'(S(0))$ is the sensitivity of the jth security's value to small changes in the underlying asset's price, the sensitivity of a portfolio containing w_j shares of each security j is $\Delta = \sum_j w_j \Delta_j$. Selling Δ shares of the underlying asset makes the portfolio value insensitive to small changes in the underlying asset's price. It is portfolios, rather than individual securities, that are hedged. However, it

can be helpful to know the Greeks of each security, which are its contribution to the portfolio's Greeks.

The Monte Carlo literature on finance has given a disproportionately great amount of attention to efficient methods for valuing and hedging some particular kind of exotic option in isolation. At this point, it is worth shifting attention to the other three problems or to addressing issues that arise in valuing and hedging derivative securities because of the business context. Also, research on simulating recently developed models can contribute to the solution of all four problems. For example, simulating models with jumps is an important topic of research at present. The following derivative securities are of particular interest:

- Asian options are important in commodities and foreign exchange, because they can help non-financial firms hedge risks arising from their businesses.
- Mortgage-backed securities [32] are in the news.
- So are credit derivatives, from single-name credit default swaps to portfolio credit derivatives such as collateralized debt obligations [13, 40, 41].

All of these lead to a high dimension d for integration, because they involve a large number m of time steps, and can pose challenges for Monte Carlo and quasi-Monte Carlo methods.

Risk Management. As illustrated by Figure 1, risk management is a broad subject that overlaps with the topics of hedging individual securities and of portfolio optimization. Hedging a portfolio's Greeks is one approach in risk management. Another is minimizing a risk measure of the hedged portfolio's P&L [26]. A *risk measure* is a real-valued functional of P&L or the distribution of P&L, such as variance, value at risk (VaR), or conditional value at risk (CVaR). For example, because of a regulatory requirement to report VaR, financial firms compute the 99th percentile of the loss distribution. Because limits on risk constrain activities, and because regulators impose a costly capital requirement on a financial firm proportional to its risk measure, there is also interest in decomposing the risk measure into a sum of *risk contributions* from the firm's positions or activities. Risk contributions as often computed as sensitivities of the risk measure to portfolio positions or the scale of a trading desk's activities. See [80] for an overview of risk management and [40, 48] for credit risk modeling.

Portfolio Optimization. Portfolio optimization features a decision variable that specifies a vector $\boldsymbol{\theta}$ of portfolio weights. This may be a static vector or it may be a stochastic process of portfolio weights that would be chosen in every possible scenario at each time step. The objective is often to maximize the expected utility $\mathrm{E}[u(W(T))]$ of future wealth $W(T) = \boldsymbol{\theta}(T)^\top \mathbf{S}(T)$, or to maximize the expected total discounted utility $\mathrm{E}[\sum_{i=0}^{m} e^{-\beta t_i} u(C(t_i))]$ of a consumption process C, which is another decision variable. The investor's initial wealth $W(0)$ imposes the budget constraint $\boldsymbol{\theta}^\top \mathbf{S}(0) = W(0)$. A multi-period formulation requires self-financing constraints like $\boldsymbol{\theta}(t_i)^\top \mathbf{S}(t_i) = \boldsymbol{\theta}(t_{i-1})^\top \mathbf{S}(t_i) - C(t_i)$, which may be more complicated if there are features such as transaction costs and taxes. There may also be con-

straints such as a prohibition against short-selling, $\boldsymbol{\theta} \geq 0$, or an upper bound on a risk measure of $W(T)$. For background on portfolio optimization, see [14, 28, 33].

Model Calibration. Calibrating the model to observed prices of derivative securities is an inverse problem, usually ill-posed. As shown in the upper left corner of Figure 1, the model maps a parameter vector $\boldsymbol{\theta}$ to a vector of security values $\boldsymbol{\mu}(\boldsymbol{\theta})$, and here the task is to find the $\boldsymbol{\theta}$ that yields a given vector $\mathbf{p}$ of the securities' market prices. The difficulty is that the mapping $\boldsymbol{\mu}(\cdot)$ may be non-invertible or the given $\mathbf{p}$ may not be in its range. A standard approach is to put a norm on the space of price vectors and to use $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\mu}(\boldsymbol{\theta}) - \mathbf{p}\|$. If the model has many parameters, it may be necessary to add a penalty term to the objective to prevent over-fitting. For an exposition, see [27, Ch. 13]. A recent innovation employing Monte Carlo methods in the search for good parameters is [10].

3 Financial Simulations in Context

Much research on Monte Carlo in finance focuses on computational efficiency: reducing time required to attain a target precision, or attaining better precision given a fixed computational budget. Efficiency is important: some financial simulations impose such a heavy computational burden that banks have invested in parallel computing platforms with thousands of processors. However, the value of efficiency techniques depends on the context of the business process within which computation takes place. Because computers are cheap and financial engineers are expensive, the benefit of a more efficient simulation must be weighed against the cost of analyst time required to implement it. Efficiency techniques are more valuable the easier they are to implement and the more broadly applicable they are. Efficiency is most important for computationally intensive problems, such as those in Section 4. The software engineering environment may hinder the implementation of efficient simulation procedures. Many firms use modular simulation software engines in which path generation does not depend on the security being considered. They may even generate a fixed set of paths of the underlying, which are then reused for several purposes, such as pricing many derivative securities. This is an obstacle to implementing some efficiency techniques: for example, it prevents the use of importance sampling methods tailored to each derivative security.

The Value of Speed. Faster answers are always welcome, but speed is more valuable in some applications than others. It does not matter whether it takes 0.1 or 0.01 seconds to deliver a precise estimate of one option price to a trader. However, it does matter whether it takes 60 hours or 6 hours to measure firm-wide risk over a one-day horizon: after 60 hours, the answer is useless because that day is over. Faster calibration is beneficial in delivering more frequently updated model parameters.

The Value of Precision. Faster is always better, but precision can be excessive. The reason is that *precision*, related to the reported uncertainty in an estimator, is

not the same as *accuracy*, related to how far the estimator is from the truth. In Monte Carlo, precision relates to the statistical properties of an estimator of a quantity that is specified by a model; if the estimator is consistent, it is possible to attain arbitrarily high precision by increasing the computational budget. Accuracy also involves *model error*, the difference between some quantity as specified by the model and the value it really has. Only building a better model, not more computational effort, will reduce model error. It is unhelpful to provide Monte Carlo estimates whose precision greatly exceeds the model's accuracy. Of course, this is true in any scientific computing endeavor, but model error tends to be greater in operations research and finance than in other disciplines such as physics. Therefore the useful degree of precision in financial simulations is less than in some other scientific computations.

In finance, the possibility of loss due to *model error* is known as *model risk*, and it is quite large: we can not be certain that an option's expected payoff is \$10.05 and not \$10.06, nor that value at risk is \$10 million as opposed to \$11 million. Simulation output can be too precise relative to model error. Suppose we run a long simulation and report a 99% confidence interval of $[10.33, 10.34]$ million dollars for value at risk. What this really means is that the Monte Carlo simulation left us with 99% confidence that our model says that value at risk is between \$10.33 and \$10.34 million. However, because of model error, we do not have high confidence that value at risk actually falls in this interval. Reporting excessive precision is a waste of time, and it is also dangerous in possibly misleading decision-makers into thinking that the numbers reported are very accurate, forgetting about model risk.

The utility of precision is also limited by the way in which answers are used. For example, when Monte Carlo is used in pricing derivative securities, the *bid-ask spread* provides a relevant standard: if market-makers charge ("ask") a dollar more when they sell an option than they pay ("bid") when they buy it, they do not need to price the option to the nearest hundredth of a cent.

As a rough general guideline, I suggest that 0.1% relative error for derivative security prices and 1% relative error for risk measures would not be too precise in most applications. Here *relative error* means the ratio of root mean squared error to some quantity. Usually it makes sense to take this quantity to be the price or risk measure being estimated. However, in some applications, the price is zero or nearly zero and it makes sense to take something else as the denominator of relative error. For example, in pricing swaps, one may use the swap rate or the notional principal on which the swap is written (in which case greater precision could be appropriate).

Repeating Similar Simulations. In finance, there are opportunities to improve efficiency because we often perform multiple simulations that are structurally the same and only differ slightly in the values of some parameters. Examples of three kinds of situations in which repeated similar simulations arise are:

- *Fixed set of tasks:* In electronic trading and market-making, we want to value many options which differ only in their strike prices and maturities. The strikes and maturities are known in advance.

- *Multi-step tasks:* Calibration can involve repeated simulations with different model parameters that are not known in advance, but depend on the results of previous steps.
- *Sequential tasks:* We measure a portfolio's risk every day. Tomorrow's portfolio composition and model parameters are currently unknown, but will probably differ only slightly from today's.

Section 4 describes some problems in which multiple simulations arise and methods for handling them efficiently. The variance reduction methods of Section 5 also help in this context. The aim of Database Monte Carlo is to use information generated in one simulation to reduce the variance of similar simulations. Adaptive Monte Carlo and related approaches can be applied to choose good variance reduction parameters to use in one simulation based on the output of a similar simulation.

Thinking about repeated simulations may lead to a paradigm shift in our understanding of how Monte Carlo should support computation in finance. The dominant paradigm is to treat each problem that arises as a surprise, to be dealt with by launching a new simulation and waiting until it delivers a sufficiently precise answer. Instead we might think of a business process as creating an imperative for us to invest computational resources in being able to estimate $\mu(\boldsymbol{\theta})$ for a range of $\boldsymbol{\theta}$.

4 Multiple Simulation Problems

Many of the computationally intensive problems most worthy of researchers' attention involve multiple simulations. In many cases, these are structurally similar simulations run with different parameters.

The Portfolio Context A large portfolio, containing a large number ℓ of securities, can make risk management and portfolio optimization simulations computationally expensive. The approach to portfolio valuation is often to choose the number m_i of replications in a simulation to value to the ith security large enough to value this security precisely, with the result that the total number of replications $\sum_{i=1}^{\ell} m_i$ is very large. However, [54] point out that if ℓ is large, the portfolio's value can be estimated precisely even if m is very small, as long as each security's value is estimated with independent replications: then the variance in estimating each security's value is large, but the variance in estimating the portfolio value is small.

Nested Simulations Nested simulation arises when, during a simulation, we would like to know a conditional expectation. If it is not known in closed form, we may resort to an *inner-level* simulation to estimate it. That is, within an *outer-level* simulation in which we want to estimate $\int f(\mathbf{u})\,\mathrm{d}\mathbf{u}$ by $\sum_{i=1}^{n} f(\mathbf{u}_i)/n$ but can not evaluate the function f, we may nest an inner level of simulation, in which we estimate $f(\mathbf{u}_1), \ldots, f(\mathbf{u}_n)$. See [69] for a general framework for two-level simulation in which we wish to estimate a functional of the distribution of $f(\mathbf{U})$ and estimate f by Monte Carlo. For examples of nested simulation, see Section 6 on risk management, where inner-level simulation estimates a portfolio's value in each scenario simulated

at the outer level, and Section 7 on American option pricing, where inner-level simulation estimates the option's continuation value at every potential exercise date on each path simulated at the outer level. For the sake of computational efficiency, it is desirable to avoid a full-blown nested simulation, which tends to require a very large total number of replications: mn if each of n outer-level scenarios or paths receives m inner-level replications. One way of avoiding nested simulation is metamodeling.

Metamodeling *Metamodeling* of a simulation model is the practice of building an approximation $\hat{\mu}$ to a function μ, using a simulation model that enables estimation of $\mu(\boldsymbol{\theta})$ for any θ in a domain Θ. One purpose of metamodeling is to be able to compute an approximation $\hat{\mu}(\boldsymbol{\theta})$ to $\mu(\boldsymbol{\theta})$ quickly. Many simulation models are slow to evaluate $\mu(\boldsymbol{\theta})$, but metamodels are constructed so that they are fast to evaluate. This makes them useful in dealing with repeated similar simulations (§3). It can be faster to build a metamodel and evaluate it repeatedly than to run many separate simulations, so metamodeling can reduce the computational burden of a fixed set of tasks or a multi-step task. In dealing with sequential tasks, metamodeling enables an investment of computational effort ahead of time to provide a rapid answer once the next task is revealed. Another benefit of metamodeling is that it supports visualization of the function's behavior over the whole domain Θ, which is more informative than merely estimating local sensitivity.

Metamodeling is better developed for deterministic simulations than for stochastic simulations, but it is becoming more widespread in stochastic simulation: for references, see [3]. In deterministic simulation, the metamodel is built by running the simulation model at some *design points* $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k$ and using the observed outputs $\mu(\boldsymbol{\theta}_1), \ldots, \mu(\boldsymbol{\theta}_k)$ to construct $\hat{\mu}$ by regression, interpolation, or both. In stochastic simulation, this is not exactly possible, because $\mu(\boldsymbol{\theta}_i)$ can only be estimated; we explain below how to deal with this conceptual difficulty. The two main approaches to metamodeling are regression and kriging. Regression methods impose on the metamodel $\hat{\mu}$ a particular form, such as quadratic, composed of splines, etc. Then the unknown coefficients are chosen to minimize the distance between the vectors $(\mu(\boldsymbol{\theta}_1), \ldots, \mu(\boldsymbol{\theta}_k))$ and $(\hat{\mu}(\boldsymbol{\theta}_1), \ldots, \hat{\mu}(\boldsymbol{\theta}_k))$. Finance is one of the applications in which it may be hard to find a form for $\hat{\mu}$ that enables it to approximate μ well over a large domain Θ. However, in some applications such as sensitivity estimation and optimization, it may only be necessary to approximate μ well locally. Unlike regression, *kriging* is an interpolation method that forces the metamodel to agree with the simulation outputs observed at all design points. However, it can be combined with estimation of a trend in $\mu(\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta}$, as in regression. There are two principal difficulties for metamodeling of financial simulations.

One is that metamodeling is hard when $\boldsymbol{\theta}$ is high-dimensional and when μ is discontinuous or non-differentiable. One remedy for the latter problem is to construct separate metamodels in different regions, such that μ is differentiable on each region. In some cases, the troublesome points are known a priori. For example, in a typical option pricing example, the option price μ is non-differentiable where the stock price equals the strike price and time to maturity is zero. In other cases, it is not known in advance whether or where μ may be badly behaved. It may help

to apply methods that automatically detect the boundaries between regions of Θ in which there should be separate metamodels [55].

The second obstacle is common to all stochastic simulations. It involves the conceptual difficulty that we can not observe the true value $\mu(\boldsymbol{\theta})$ at any input $\boldsymbol{\theta}$, and a related practical shortcoming. We might deal with the conceptual difficulty in one of two ways. One way is to use quasi-Monte Carlo or fix the seed of a pseudo-random number generator and regard the output ν of the stochastic simulation as deterministic, given that these *common random numbers* (CRN) are used to simulate at any input $\boldsymbol{\theta}$. We can build a metamodel $\hat{\nu}$ of ν, but its output $\hat{\nu}(\boldsymbol{\theta})$ is an approximation to $\nu(\boldsymbol{\theta})$, the output that the simulation would produce if it were run at $\boldsymbol{\theta}$ with CRN, not necessarily a good approximation to the expectation $\mu(\boldsymbol{\theta})$ that we want to know. Then the practical shortcoming is that $\nu(\boldsymbol{\theta})$ needs to be a precise estimate of the expectation $\mu(\boldsymbol{\theta})$, so the number of replications used at each design point must be large. The second way to deal with the conceptual difficulty is to use different pseudo-random numbers at each design point $\boldsymbol{\theta}_i$, but build a metamodel by plugging in a precise simulation estimate for $\mu(\boldsymbol{\theta}_i)$ anyway. This entails the same practical shortcoming, and the Monte Carlo sampling variability makes it harder to fit a good metamodel. It is a practical shortcoming to need many replications at each design point because, given a fixed computational budget, it might be more efficient to have more design points with fewer replications at each. *Stochastic kriging* [3] is one solution to these problems. It shows how uncertainty about the expectation $\mu(\boldsymbol{\theta})$ arises from the combination of interpolation and the Monte Carlo sampling variability that affects the stochastic simulation output as an estimate of $\mu(\boldsymbol{\theta}_i)$ for each design point. Stochastic kriging makes it possible to get a good approximation to $\mu(\boldsymbol{\theta})$ even when the number of replications at each design point is small and provides a framework for analyzing the trade-off between having many design points and having more replications at each of them.

Metamodeling is closely related in its aims to database Monte Carlo (§5).

Optimization Another reason that one might need to obtain Monte Carlo estimates of $\mu(\boldsymbol{\theta})$ for multiple values of θ is when optimizing over $\boldsymbol{\theta}$, if simulation is needed in evaluating the objective or constraints of the optimization problem. This is *optimization via simulation* (OvS): for an overview, see [34, 60]. In the following, we concentrate on the problem $\min_{\boldsymbol{\theta}\in\Theta} \mu(\boldsymbol{\theta})$ of minimizing an expectation that must be estimated by Monte Carlo over a continuous decision space Θ defined by constraints that can be evaluated without Monte Carlo. The typical pattern is that an optimization procedure visits candidate solutions $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ sequentially, at each step j using information generated by Monte Carlo to choose $\boldsymbol{\theta}_j$. It is quite useful in choosing $\boldsymbol{\theta}_j$ to be able to estimate the gradient $\nabla\mu(\boldsymbol{\theta}_{j-1})$: see Section 8.

- **Sample Average Approximation.** The simplest approach is to approximate the objective value $\mu(\boldsymbol{\theta})$ by the sample average $\hat{\mu}(\boldsymbol{\theta}) = \sum_{i=1}^{n} f(\mathbf{u}_i; \theta)/n$. That is, the common random numbers $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are used to estimate the objective at any candidate solution θ_j. To minimize $\hat{\mu}$, one can use a gradient-free optimization procedure or use the gradient $\nabla\hat{\mu}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} f(\mathbf{u}_i; \boldsymbol{\theta})/n$ if available.

- **Metamodeling and Stochastic Approximation.** Another approach involves running more simulations at each step, using increasing total simulation effort as the optimization procedure converges to the optimal $\boldsymbol{\theta}$. *Sequential metamodeling* [7] considers a neighborhood Θ_j of $\boldsymbol{\theta}_{j-1}$ at step j, and builds a metamodel $\hat{\mu}_j$ that approximates μ locally, on Θ_j. The gradient $\nabla\hat{\mu}_j$ helps in choosing $\boldsymbol{\theta}_j$. (Because of the difficulty of building metamodels that fit well globally, it has not been common practice in OvS simply to build one metamodel and minimize over it.) *Stochastic approximation* depends on ways of computing an estimate $\widehat{\nabla}\mu(\boldsymbol{\theta})$ of the gradient that are described in Section 8 and [35]. At step j, the next candidate solution is $\boldsymbol{\theta}_j = \boldsymbol{\theta}_{j-1} - \gamma_j\widehat{\nabla}\mu(\boldsymbol{\theta}_{j-1})$. It can be troublesome to find a sequence of step sizes $\{\gamma_j\}_{j\in\mathbb{N}}$ that works well for one's particular optimization problem [34]. For recent progress, see [18, 82]. Other questions include whether it is best to estimate the optimal $\boldsymbol{\theta}$ by $\boldsymbol{\theta}_n$ or a weighted average of $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$, or to constrain $\boldsymbol{\theta}_j$ from moving too far from $\boldsymbol{\theta}_{j-1}$; see [60].
- **Metaheuristics.** Various *metaheuristic* methods, such as simulated annealing and genetic algorithms, use Monte Carlo to solve optimization problems heuristically, even if the objective μ can be evaluated without Monte Carlo: they randomly select the next candidate solution $\boldsymbol{\theta}_j$. See [83] for an overview in the simulation context, where it is typical to employ the metaheuristic simply by using a simulation estimate $\hat{\mu}(\boldsymbol{\theta}_j)$ in place of each $\mu(\boldsymbol{\theta}_j)$. Metaheuristics can solve difficult optimization problems, such as model calibration problems that are non-convex, with multiple local minima and regions in which the objective is very flat. However, they are called *meta*heuristics because they require tailoring to the specific problem to produce an algorithm that works well. Randomness over candidate solutions can have more benefits than escaping from local minima: for example, [10] uses a metaheuristic optimization procedure to account for the parameter uncertainty that remains after model calibration.
- **Approximate Dynamic Programming.** This discussion of optimization has not yet explicitly taken into account optimization over policies that include decisions at multiple times, which is important for American options and dynamic portfolio optimization. This is the subject of dynamic programming, in which the optimal decision at each time maximizes a value function, such as the expected utility of terminal wealth as a function of underlying prices and the composition of the portfolio. Approximate dynamic programming (ADP) is a solution method for dynamic programs that are too large to solve exactly. Instead of computing the exact value of each state, ADP constructs an approximate value function. Monte Carlo can help in approximating the value function: then ADP is closely related to simulation metamodeling. For more on ADP, see [11, 88, 89].

5 Variance Reduction

Here we discuss only two active areas of research in variance reduction that have important applications in finance.

Database Monte Carlo. The idea of database Monte Carlo (DBMC) is to invest computational effort in constructing a database which enables efficient estimation for a range of similar problems [16]. In finance, it is often important to solve a whole set of similar problems (§3). The problems are indexed by a parameter $\boldsymbol{\theta}$, and we want to estimate $\mu(\boldsymbol{\theta}) = \int f(\mathbf{u};\boldsymbol{\theta})\,d\mathbf{u}$ for several values of $\boldsymbol{\theta}$, for example, to price several options of different strike prices. DBMC involves choosing a base value $\boldsymbol{\theta}_0$ of the parameter and evaluating $f(\cdot;\boldsymbol{\theta}_0)$ at many points $\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_N$ in $[0,1)^d$. The set $\{(\boldsymbol{\omega}_i, f(\boldsymbol{\omega}_i;\boldsymbol{\theta}_0))\}_{i=1,\dots,N}$ constitutes the database. DBMC provides a generic strategy for employing a variance reduction technique effectively: the purpose of investing computational effort in the database is that it enables powerful variance reduction in estimating $\mu(\boldsymbol{\theta})$ for values of $\boldsymbol{\theta}$ such that $f(\cdot;\boldsymbol{\theta})$ is similar to $f(\cdot;\boldsymbol{\theta}_0)$. It may be possible to estimate $\mu(\boldsymbol{\theta})$ well with $f(\cdot,\boldsymbol{\theta})$ evaluated at only a small number n of points. DBMC has been implemented with stratification and control variates [16, 96, 97, 98]. All but one of the methods in these papers are structured database Monte Carlo (SDMC) methods, in which further effort is expended in structuring the database: the database is sorted so that $f(\boldsymbol{\omega}_i;\boldsymbol{\theta}_0)$ is monotone in i [98].

SDMC with stratification partitions $\{1,\dots,N\}$ into $n \ll N$ strata $I_1 = \{1,\dots,i_1\}$, $I_2 = \{i_1+1,\dots,i_2\}$, …, $I_n = \{i_{n-1}+1,\dots,N\}$. (How best to partition is a subject of active research.) It then performs stratified resampling of $\mathbf{u}_1,\dots,\mathbf{u}_n$ from $\{\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_N\}$. That is, $\mathbf{u}_1$ is drawn uniformly from the set $\{\boldsymbol{\omega}_i : i \in I_1\}$, $\mathbf{u}_2$ uniformly from $\{\boldsymbol{\omega}_i : i \in I_2\}$, etc. SDMC then estimates $\mu(\boldsymbol{\theta})$ by $\sum_{j=1}^n p_j f(\mathbf{u}_j;\boldsymbol{\theta})$ where $p_j = |I_j|/N = (i_j - i_{j-1})/N$. If this stratification provides good variance reduction, then $\sum_{j=1}^n p_j f(\mathbf{u}_j;\boldsymbol{\theta})$ is a good estimator of $\sum_{i=1}^N f(\boldsymbol{\omega}_i;\boldsymbol{\theta})/N$. In turn, $\sum_{i=1}^N f(\boldsymbol{\omega}_i;\boldsymbol{\theta})/N$ is a good estimator of $\mu(\boldsymbol{\theta})$ because N is large. Then, even though n is small, $\sum_{j=1}^n p_j f(\mathbf{u}_j;\boldsymbol{\theta})$ is a good estimator of $\mu(\boldsymbol{\theta})$.

The advantage of SDMC can be understood by viewing it as a scheme for automatically creating good strata. Ordinary stratification requires partitioning $[0,1)^d$ into strata, and it is time-consuming and difficult to find a good partition, especially because the partition must be such that we know the probability of each stratum and how to sample uniformly within each stratum. Although SDMC actually stratifies the database, it is similar to partitioning $[0,1)^d$ into strata $\mathcal{X}_1,\dots,\mathcal{X}_n$ such that $\{\boldsymbol{\omega}_i : i \in I_j\} \subseteq \mathcal{X}_j$ for all $j = 1,\dots,n$. Typically, this partition is better than one that an analyst could easily create, because SDMC takes advantage of knowledge about $f(\cdot;\boldsymbol{\theta}_0)$ that is encoded in the database. If $f(\boldsymbol{\omega}_i,\boldsymbol{\theta})$ is close to monotone in the database index i, then SDMC with stratification provides excellent variance reduction [97]. SDMC avoids issues that make it hard for analysts to find good partitions. We need not know the stratum probabilities, because they are estimated by sample proportions from the database. Nor do we need to know how to sample from the conditional distribution of $f(\mathbf{U};\boldsymbol{\theta})$ given that it falls in a certain stratum, because stratified sampling is performed using the database indices.

DBMC applied to control variates [16] leads to the idea of a quasi-control variate [31], i.e., a random variable used as a control variate even though its mean is unknown and merely estimated by Monte Carlo [85]. In DBMC, one can use $f(\mathbf{u};\boldsymbol{\theta}_0)$ as a quasi-control variate, with estimated mean $\sum_{i=1}^N f(\boldsymbol{\omega}_i;\boldsymbol{\theta}_0)/N$. One may resample $\mathbf{u}_1,\dots,\mathbf{u}_n$ from $\{\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_N\}$ or instead use fresh points $\mathbf{u}_1,\dots,\mathbf{u}_n$, and then

estimate $\mu(\boldsymbol{\theta})$ by $\sum_{j=1}^{n} f(\mathbf{u}_j;\boldsymbol{\theta})/n - \beta(\sum_{j=1}^{n} f(\mathbf{u}_j;\boldsymbol{\theta}_0)/n - \sum_{i=1}^{N} f(\boldsymbol{\omega}_i;\boldsymbol{\theta}_0)/N)$. There are also SDMC methods which involve sorting the database and using the database index as a control variate [96].

DBMC is a powerful and exciting new strategy for variance reduction when handling multiple similar problems. DBMC methods are generic and provide automated variance reduction, requiring relatively little analyst effort. Open questions remain, especially in experiment design for DBMC. What is the optimal database size N when one must estimate $\mu(\boldsymbol{\theta}_1), \ldots, \mu(\boldsymbol{\theta}_k)$ given a fixed budget $C = N + kn$ of function evaluations? We may be interested in some values of $\boldsymbol{\theta}$ that are near the base value $\boldsymbol{\theta}_0$ and others that are far: when is it worthwhile to restructure the database or create a new database at another base value?

Such questions emphasize differences between DBMC, in its present state of development, and metamodeling. DBMC and metamodeling are two ways of using an investment of computational effort to get fast estimates of $\mu(\boldsymbol{\theta})$ for many values of $\boldsymbol{\theta}$. However, they work quite differently. Metamodeling provides an estimate of $\mu(\boldsymbol{\theta})$ without any further simulation, but the estimate is biased, in general; when metamodeling works badly, large errors can result. Metamodeling works by exploiting properties of the function μ, whereas DBMC works by exploiting properties of f. DBMC estimates $\mu(\boldsymbol{\theta})$ with a small simulation of n replications, and the resulting estimate is unbiased (ignoring bias due to estimating coefficients of control variates). The parallel to metamodeling suggests extending DBMC to incorporate information from multiple simulations, not just one at $\boldsymbol{\theta}_0$.

Adaptive Monte Carlo. The fundamental idea of adaptive Monte Carlo is to improve the deployment of a variance reduction technique during the simulation, using information generated during the simulation. That is, the variance reduction technique is parameterized by $\boldsymbol{\vartheta}$, where the special notation $\boldsymbol{\vartheta}$ indicates that parameter does not affect the mean $\mu = \mathrm{E}[f(\mathbf{U};\boldsymbol{\vartheta})]$. However, it does affect the variance $\mathrm{Var}[f(\mathbf{U};\boldsymbol{\vartheta})]$. Adaptive Monte Carlo uses simulation output to choose $\boldsymbol{\vartheta}$ to reduce the variance $\mathrm{Var}[f(\mathbf{U};\boldsymbol{\vartheta})]$. A number of Monte Carlo methods can be viewed as adaptive to some extent, even the long-standing practice of using regression to choose the coefficients of control variates based on simulation output.

This standard way of implementing control variates illustrates a recurrent question in adaptive Monte Carlo: should one include the replications used to choose $\boldsymbol{\vartheta}$ in the estimator of μ, or should one throw them out and include only fresh replications in the estimator? If separate batches of replications are used to choose the coefficients and to estimate the expectation, the estimator with control variates is unbiased. However, it is preferable to use the same replications for both tasks, despite the resulting bias, which goes to zero as the sample size goes to infinity [46, §4.1.3]. Many adaptive Monte Carlo methods include all the replications in the estimator, which is nonetheless asymptotically unbiased under suitable conditions. In some portfolio risk measurement and American option pricing problems, the bias may be large at the desired sample size. There are methods for these problems, discussed in Sections 6 and 7, that use a fresh batch of replications to reduce bias or to deliver probabilistic bounds for bias.

There are two main approaches to adaptive Monte Carlo. In one approach, the analyst chooses a parameterized variance reduction scheme, and adaptive Monte Carlo tries to choose $\boldsymbol{\vartheta}$ to attain variance near $\inf_{\boldsymbol{\vartheta}} \mathrm{Var}[f(\mathbf{U};\boldsymbol{\vartheta})]$. The other approach is oriented towards learning a value function which, if known, would enable zero-variance simulation. This kind of adaptive Monte Carlo achieves variance reduction by making use of an approximation to the value function. In finance, both approaches employ optimization via simulation (§4), either to minimize variance or to find the approximate value function that best fits the simulation output. Stochastic approximation (SA) and sample average approximation (SAA) have been employed as optimization methods. Importance sampling and control variates are the most common variance reduction methods in this literature.

In the variance-minimization approach, [21] uses SAA while [4, 93] use SA. The procedures using SA can have multiple stages: at stage n, variance reduction is performed using the parameter $\boldsymbol{\vartheta}_{n-1}$, and then the parameter is updated to $\boldsymbol{\vartheta}_n$ based on the new simulation output. The estimator is computed by [93] as an average of fresh replications in the last stage, which were never used to choose the variance reduction parameter; it is an average of all replications in [4] and papers that follow it. Under suitable conditions, the variance reduction parameter $\boldsymbol{\vartheta}_{\mathbf{n}}$ converges to an optimal choice, and the average over all replications is a consistent, asymptotically normal estimator. Still, it would also be well to confirm the bias is negligible at the relevant sample sizes. Another issue is how many replications should be in each stage, between updates of $\boldsymbol{\vartheta}$. Although classic SA procedures may update ϑ after each replication, that will usually entail too much computational effort when the goal is variance reduction, or rather, a reduction in work-normalized variance.

The approach that approximates a value function V is surveyed by [73] in a Markov-chain setting. In finance, $V(t,\mathbf{S}(t))$ may be an option's price when the underlying is $\mathbf{S}(t)$ at time t, for example. An approximate value function $\hat{V}$ is built by metamodeling (§4). Adaptive control variates work by using $\hat{V}$ to construct a martingale whose ith increment is $\hat{V}(t_i,\mathbf{S}(t_i)) - \mathrm{E}[\hat{V}(t_i,\mathbf{S}(t_i))|\mathcal{F}_{i-1}]$, and using it as a control variate. Adaptive importance sampling works by setting the likelihood ratio for step i to $\hat{V}(t_i,\mathbf{S}(t_i))/\mathrm{E}[\hat{V}(t_i,\mathbf{S}(t_i))|\mathcal{F}_{i-1}]$. For the sake of computational efficiency, $\hat{V}$ should be such that $\mathrm{E}[\hat{V}(t_i,\mathbf{S}(t_i))|\mathcal{F}_{i-1}]$ can be computed in closed form. If the true value function V could be substituted for the approximation $\hat{V}$, then the control variate or importance sampling would be perfect, resulting in zero variance [63]. A bridge between the two approaches is [66], using SA and SAA methods to construct $\hat{V}$ by minimizing the variance that remains after it is used to provide a control variate. In finance, this approach to adaptive Monte Carlo has been used above all for American options: [30, 63] use SAA and regression metamodeling for this purpose. Because metamodeling is commonly used anyway in American option pricing, to identify a good exercise policy, the marginal computational cost of using the metamodel to find a good control variate or importance sampling distribution can be small, making this adaptive Monte Carlo approach very attractive.

6 Risk Management

Monte Carlo methods in risk management are an active area of research. A straightforward Monte Carlo approach is to sample scenarios $\mathbf{S}^{(1)}(T), \ldots, \mathbf{S}^{(n)}(T)$ and in each scenario to compute P&L $V(T, \mathbf{S}(T)) - V(0, \mathbf{S}(0))$, the change in the portfolio's value by time T. It is natural to have a high dimensional for $\mathbf{S}$ because a portfolio's value can depend on many factors. There are two main computational challenges in risk measurement.

One challenge is that risk measures such as VaR and CVaR focus on the left tail of the distribution of P&L, containing large losses. It is a moderately rare event for loss to exceed VaR, so straightforward Monte Carlo estimation of a large portfolio's risk can be slow. This makes it worthwhile to pursue variance reduction: see [46, Ch. 9] for general techniques and [8, 25, 48, 49, 51] for techniques specific to credit risk.

The second challenge arises when the portfolio value function $V(T, \cdot)$ is unknown, so P&L in each scenario must be estimated by Monte Carlo. This leads to a computationally expensive nested simulation (§4): simulation of scenarios under P (as in the lower left corner of Figure 1) and a nested simulation under Q conditional on each scenario, to estimate the portfolio value $V(T, \mathbf{S}(T))$ in that scenario. In particular, nested simulation is generally biased, which causes a poor rate of convergence for the Monte Carlo estimate as the computational budget grows. This makes it worthwhile to explore ways to make the simulation more efficient:

- Jackknifing can reduce the bias [54, 74].
- Although variance is not always a good portfolio risk measure, it can be useful in evaluating hedging strategies. Unbiased estimation of the variance of P&L by nested simulation is possible. Indeed, a nested simulation with small computational effort devoted to each scenario, and thus inaccurate estimation of P&L in each scenario, can provide an accurate estimator of the variance of P&L [94].
- It helps to optimize the number n of scenarios to minimize MSE or confidence interval width given a fixed computational budget [54, 68].
- When the risk measure emphasizes the left tail of the distribution, is desirable to allocate more computational effort to simulating the scenarios that seem likely to be near VaR (when estimating VaR) or to belong to the left tail (for CVaR). This suggests adaptive simulation procedures, in which the allocation of replications at one stage depends on information gathered at previous stages. One approach is to eliminate scenarios once they seem unlikely to belong to the left tail [70, 78]. Another is to make the number of replications somehow inversely proportional to the estimated distance from a scenario to the left tail or its boundary [54].
- Metamodeling (§4) and database Monte Carlo (§5) can be useful in portfolio risk measurement because it involves many similar simulation problems: estimating P&L in many scenarios. Metamodeling can be successful because P&L is often a well-behaved function of the scenario. It has been applied in [9] and in an adaptive procedure for estimating CVaR by [79], where more computational effort is allocated to design points near scenarios with large losses.

Estimating sensitivities of risk measures is studied in [47, 61, 62, 76]. They can provide risk components or be useful in optimization.

7 Financial Optimization Problems

The finance problem most clearly linked to optimization is portfolio optimization. Before discussing Monte Carlo methods for portfolio optimization, we turn to American option pricing. It involves a simple optimization problem, and Monte Carlo methods for American option pricing have been more thoroughly studied. See Section 4 for background on optimization via simulation.

American Options. Monte Carlo is best suited for *European options*, which can be exercised only at maturity. *American options* can be exercised at any time until maturity. The owner of an American option faces an optimal stopping problem. Let τ represent the exercise policy: the random variable $\tau = \tau(\mathbf{U})$ is the stopping time at which exercise occurs. The resulting payoff is $f(\mathbf{U}; \tau)$. Pricing methods for American options involve computing the optimal exercise policy τ^* that maximizes the value $\mathrm{E}[f(\mathbf{U}; \tau)]$ of the option, while computing the price $\mathrm{E}[f(\mathbf{U}; \tau^*)]$. It is optimal to exercise at time t if the payoff $f(\mathbf{U}; t)$ of doing so exceeds the *continuation value*, the conditional expectation of the payoff earned by exercising at the optimal time after t. Because a continuous-time optimal stopping problem is troublesome for simulation, much research on the topic of American options actually deals with *Bermudan options*, which can be exercised at any one of the times $\{t_1, \ldots, t_m\}$. A Bermudan option with a sufficiently large set of possible exercise times is treated as an approximation of an American option. Even Bermudan options are not straightforward to price by Monte Carlo methods: at every step on every path, one needs to know the continuation value to make the optimal decision about whether to exercise. A naive approach, which is impractical due to excessive computational requirements, is nested simulation (§4): at every step on every path, estimate the continuation value by an inner-level simulation. For overviews of Monte Carlo methods in American option pricing, see [15, 24, 39, 46]. Here we merely emphasize connections to themes of financial simulation.

- The most popular approach to American option pricing, regression-based Monte Carlo, is a form of approximate dynamic programming (ADP). The optimal stopping problem is relatively easy for ADP because there are only two actions, continue or exercise, and they do not affect the dynamics of the underlying.
- After choosing a sub-optimal exercise policy τ and sampling $\mathbf{U}$ independently, $f(\mathbf{U}; \tau)$ is an estimator of the American option price with negative bias. Duality yields an estimator with positive bias: see [56] and references therein, particularly [2]. This enables a conservative confidence interval that is asymptotically valid for large simulation sample sizes. A bias reduction method is developed in [65].
- Adaptive Monte Carlo (§5) is very useful in American option pricing. It is connected to duality: according to [63], "the perfect control variate solves the ad-

ditive duality problem and the perfect importance sampling estimator solves the multiplicative duality problem."

American option pricing remains an active research area because there are many rival methods that are amenable to improvement. There is potential to gain efficiency by adaptive simulation that allocates extra simulation effort to design points near the boundary where estimated exercise and continuation values are equal. High-dimensional problems remain challenging. It would also be good to better understand and to reduce the error in approximating an American by a Bermudan option.

Portfolio Optimization. An introduction to this topic, stressing the connection between American option pricing and portfolio optimization, while emphasizing the value of dual methods, is [56]. The purpose of the dual methods is to provide an upper bound on the optimal expected utility: one can use simulation to estimate both the expected utility a candidate portfolio strategy provides and the upper bound on the optimal expected utility, and compare these estimates to see if the candidate is nearly optimal [57]. Other ADP methods in portfolio optimization include [17, 81, 95]. ADP is not the only Monte Carlo approach to portfolio optimization. For an overview, see [14]. Another method uses Monte Carlo to estimate conditional expectations involving Malliavin derivatives, which are proved to be the optimal portfolio weights for a portfolio optimization in a complete market [29].

8 Sensitivity Analysis

Many problems in finance call for estimation of the sensitivity $\mu'(\theta)$ of a mean $\mu(\theta)$ to a parameter θ: the Greeks are of direct interest in hedging, and sensitivities are needed in gradient-based optimization. Approaches to estimating sensitivities via simulation include:

- *Finite differences* (FD). Run the simulation at two values θ_1 and θ_2 in the neighborhood of θ, using common random numbers. The FD estimator is $(f(\mathbf{U};\theta_1) - f(\mathbf{U};\theta_2))/(\theta_1 - \theta_2)$. This approach is biased and computationally inefficient.
- Metamodeling (M, §4) can be viewed as a variant of FD that is helpful when estimating sensitivities with respect to many parameters: where FD would require running many simulations, metamodeling can provide an answer based on simulations at only a few design points. To estimate first-order sensitivities, fit a linear metamodel locally, in a neighborhood of θ. To get second-order sensitivities too, fit a quadratic metamodel locally.
- The *pathwise* (PW) method, known outside finance as *infinitesimal perturbation analysis* (IPA). Under some conditions, $\mu'(\theta) = \mathrm{E}[Y'(\theta)]$, so an unbiased estimator is $Y'(\theta) = (\partial f/\partial\theta)(\mathbf{U};\theta)$. It may be easy to compute this if θ is a parameter, such as a strike price, that has a simple, direct effect on the payoff, but it might be hard if θ is a parameter that governs the distributions of random variables in the simulation. This method can only be applied if Y is suitably differentiable; there are a number of cases in finance in which it does not apply.

- *Smoothed perturbation analysis* (SPA) is an extension of IPA. It works by reformulating the simulation model: if there is a conditional expectation $\tilde{Y}(\theta) = \mathrm{E}[Y(\theta)|\mathcal{F}]$ that can be computed and $\tilde{Y}$ is a smoother function of θ than Y is, then the estimator $\tilde{Y}'(\theta)$ can be used when IPA does not apply. This approach requires the analyst to identify a good set of information $\mathcal{F}$ on which to condition, and to compute the conditional expectation.
- IPA can have problems in first or second derivative estimation because of discontinuity or non-differentiability of the integrand in the commonplace case where $Y(\theta) = f(\mathbf{U};\theta)$ has the form $f_1(\mathbf{U};\theta)1\{f_2(\mathbf{U};\theta) \geq 0\}$. Kernel smoothing leads to the estimator

$$\frac{\partial f_1}{\partial \theta}(\mathbf{U};\theta)1\{f_2(\mathbf{U};\theta) \geq 0\} + \frac{1}{\delta} f_1(\mathbf{U};\theta)\frac{\partial f_2}{\partial \theta}(\mathbf{U};\theta)\phi\left(\frac{f_2(\mathbf{U};\theta)}{\delta}\right),$$

  where ϕ is the kernel and δ is the bandwidth [77]. In contrast to SPA, kernel smoothing requires no analyst ingenuity: a Gaussian kernel and automated bandwidth selection perform well. This estimator is biased, although it is consistent under some conditions which may be hard to verify.
- The *likelihood ratio* (LR) method, also known outside finance as the *score function* method, involves differentiating a density $g(\cdot;\theta)$ instead of differentiating a payoff. Here we require a representation $\mu(\theta) = \int f(\mathbf{u};\theta)\,\mathrm{d}\mathbf{u} = \int \tilde{f}(\mathbf{x})g(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x}$, framing the simulation as sampling the random vector $\mathbf{X}(\mathbf{U};\theta)$ which has density $g(\cdot;\theta)$. In the new representation, $Y(\theta) = f(\mathbf{U};\theta) = \tilde{f}(\mathbf{X}(\mathbf{U};\theta))$, so $\tilde{f}$ has no explicit dependence on θ: applying the method requires θ to be a parameter only of the density. Under some conditions,

$$\mu'(\theta) = \int \tilde{f}(\mathbf{x})\frac{\partial g(\mathbf{x};\theta)/\partial\theta}{g(\mathbf{x};\theta)}g(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x} = \mathrm{E}\left[Y(\theta)\frac{\partial g(\mathbf{X};\theta)/\partial\theta}{g(\mathbf{X};\theta)}\right],$$

  so an unbiased estimator is $Y(\theta)(\partial g(\mathbf{X};\theta)/\partial\theta)/g(\mathbf{X};\theta)$. If the density is not known in closed form, one may apply the LR method instead to a discretized version of the underlying stochastic process.
- Malliavin calculus can provide estimators of sensitivities. Implementing these estimators generally requires that time be discretized. The resulting estimators are asymptotically equivalent, as the number of time steps $m \to \infty$, to combinations of PW and LR estimators for the discretized process [23]. Combinations of PW and LR methods are also used to overcome the limitations of PW and of LR in isolation. For a unified view of the PW and LR methods, see [71].
- The method of *weak derivatives* (WD) can be explained based on LR [37]: suppose $\partial g(\mathbf{x};\theta)/\partial\theta$ can be written in the form $c(\theta)(g_1(\mathbf{x};\theta) - g_2(\mathbf{x};\theta))$, where $g_1(\cdot;\theta)$ and $g_2(\cdot;\theta)$ are densities. If the LR approach is valid, then

$$\begin{aligned}\mu'(\theta) &= c(\theta)\left(\int \tilde{f}(\mathbf{x})g_1(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x} - \int \tilde{f}(\mathbf{x})g_2(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x}\right)\\ &= c(\theta)\mathrm{E}\left[\tilde{f}(\mathbf{X}_1) - \tilde{f}(\mathbf{X}_2)\right],\end{aligned}$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are sampled according to the densities $g_1(\cdot;\theta)$ and $g_2(\cdot;\theta)$ respectively: an unbiased estimator is $c(\theta)(\tilde{f}(\mathbf{X_1}) - \tilde{f}(\mathbf{X_2}))$. (However, the WD approach does not actually require differentiating the density.) Here we did not specify how the original pseudo-random numbers would be used to simulate $\mathbf{X}_1$ and $\mathbf{X}_2$. The whole structure of the simulation is changed, and the dependence or *coupling* of $\mathbf{X}_1$ and $\mathbf{X}_2$ has a major effect on the estimator's variance.

For introductions to these methods, see [24, 37, 43, 46]. Important early references include [19, 38]. The different methods have different realms of applicability and, when two of them apply, they can yield estimators with very different variances.

A recent advance has been in speeding up PW computations of multiple Greeks of the same derivative security price using adjoint methods [43, 45]. Another active area of research is estimation of sensitivities when the underlying stochastic process has jumps: see e.g. [52]. A further topic for future work is the application of WD to estimating sensitivities in financial simulation: although weak derivatives were applied to simulating the sensitivities of option prices in [58], the WD method has not received enough attention in finance. For results on WD when underlying distributions are normal, as happens in many financial models, see [59].

9 Discretization of Stochastic Differential Equations

Many financial simulations involve *stochastic differential equations* (SDEs). The solution $\mathbf{S}$ to an SDE is a continuous-time stochastic process, but it is standard to discretize time and simulate $\mathbf{S}(t_1), \ldots, \mathbf{S}(t_m)$. In some models, it is possible to simulate *exactly*, that is, from the correct distribution for $(\mathbf{S}(t_1), \ldots, \mathbf{S}(t_m))$. However, in many models, it is not known how to do so. *Discretization error* is the difference between the distribution of $(\mathbf{S}(t_1), \ldots, \mathbf{S}(t_m))$ as simulated and the distribution it should have according to the SDE. Discretization error causes *discretization bias* in the Monte Carlo estimator. To reduce the discretization bias, one increases the number m of steps, which increases the computational cost of simulating $\mathbf{S}(t_1), \ldots, \mathbf{S}(t_m)$. On quantifying and reducing this discretization bias, see [46, 67], or [24, 53] for introductions. Some research on SDE discretization is specific to one model, that is, to one SDE, while some is generic.

Model-specific research may consist of showing how to simulate a certain model exactly or how to reduce discretization error. For example, recently there have been major improvements in simulating the Heston model [1, 20, 50]. On simulation of Lévy processes, see [5] and [27, Ch. 6]. Lévy processes used in finance include VG and CGMY: on simulating these, see [6, 36, 64, 87].

The generic research includes the study of different discretization schemes and the rate at which discretization bias decreases as the number m of steps increases. This rate may be unaffected by replacing the normal random variables typically used in SDE discretization by simpler random variables which are faster to simulate, e.g. having discrete distributions with only three values [46, pp. 355-6]. It would be interesting to explore the application of quasi-Monte Carlo to a simulation scheme

using these discrete random variables. One active research topic, based on [86], involves new discretization schemes, the quadratic Milstein scheme and a two-stage Runge-Kutta scheme, along with a new criterion, microscopic total variation, for assessing a scheme's quality.

We next consider two important recent developments in simulating SDEs.

Multi-Grid Extrapolation. One method for reducing discretization error is *extrapolation* [46, §6.2.4]. Let $\hat{\mu}(m)$ be a simulation estimator based on discretizing an SDE with m time steps, and $\hat{\mu}(2m)$ be the estimator when $2m$ time steps are used. Because of bias cancelation, the estimator $2\hat{\mu}(2m) - \hat{\mu}(m)$ can have lower bias and a better rate of convergence. This idea is extended by [44] to multiple grids of different fineness, instead of just two. The estimator given L grids, with N_ℓ paths simulated on the ℓth grid which has m_ℓ steps, is $\sum_{\ell=1}^{L} \sum_{i=1}^{N_\ell} (\hat{\mu}^{(i)}(m_\ell) - \hat{\mu}^{(i)}(m_{\ell-1}))/N_\ell$, where $\hat{\mu}^{(i)}(m_\ell)$ involves simulating the same Wiener process sample path $\{W^{(i)}(t)\}_{0\le t\le T}$ for all grids. It is efficient to simulate fewer paths using the fine grids than with the coarse grids. For one thing, even if N_ℓ is small for a fine grid, including this ℓth grid contributes a correction term $\mathrm{E}[\hat{\mu}^{(i)}(m_\ell) - \hat{\mu}^{(i)}(m_{\ell-1})]$ that reduces bias. Furthermore, simulating paths on a fine grid is computationally expensive, while the variance of $\hat{\mu}^{(i)}(m_\ell) - \hat{\mu}^{(i)}(m_{\ell-1})$ tends to be small for the fine grids. Consequently, computational resources are better spent on coarser grids where it is cheap to attack large components of the variance. The result is reduced bias and better rates of convergence. QMC should be useful particularly when applied to the coarser grids. A related approach involving multiple grids [91] is based on the idea that coarse grids provide biased control variates [90].

Exact Simulation of SDEs. Surprisingly, it is sometimes possible to simulate a scalar diffusion S exactly even when it is not possible to integrate the SDE in closed form to learn the distribution of $(S(t_1), \ldots, S(t_m))$ [12, 22]. The basic idea is to sample according to the law of S by acceptance-rejection sampling of paths of a Wiener process W. If a path $\{W(t)\}_{0\le t\le T}$ is accepted with probability proportional to the Radon-Nikodym derivative between the law of the S and the law of W, the path is sampled from the law of S. The log of the Radon-Nikodym derivative has the form $A(W(T)) - \int_0^T \phi(t, W(t))\,\mathrm{d}t$ where A and ϕ depend on the coefficients of the SDE. The problem lies in simulating $\int \phi(t, W(t))\,\mathrm{d}t$, which is an awkward functional of the entire continuous-time path $\{W(t)\}_{0\le t\le T}$. The key insight is that $\exp(-\int_0^T \phi(t, W(t))\,\mathrm{d}t)$ is the conditional probability, given the path of the Wiener process, that no arrivals occur by time T in a doubly stochastic Poisson process whose arrival rate at time t is $\phi(t, W(t))$. This may be simulated by straightforward or sophisticated stochastic thinning procedures, depending on the characteristics of the function ϕ [12, 22, 42]. This approach is a significant development: it is of theoretical interest and, when applicable, it eliminates the need for the analyst to quantify and reduce discretization bias. More work is needed to render this approach widely applicable in finance and to study the efficiency gains it produces. Acceptance-rejection sampling can be very slow, when the acceptance probability is low, so this way of simulating SDEs exactly could be slower to attain a target MSE than exist-

ing methods of SDE discretization. The speed of acceptance-rejection sampling can be improved by drawing the original samples from another law. When the Radon-Nikodym derivative between the law of S and the original sampling law is smaller, acceptance occurs faster. In this case, one might think of drawing the original samples from the law of some other integrable Itō process, not a Wiener process. For example, one might sample from the law of geometric Brownian motion or of an Ornstein-Uhlenbeck process, because in many financial models, S is closer to these than to a Wiener process. An interesting question is how best to choose the original sampling law given the SDE one wishes to simulate.

Acknowledgements The author acknowledges the support of the National Science Foundation under Grant No. DMI-0555485. He is very grateful to Mark Broadie, Michael Fu, Kay Giesecke, Paul Glasserman, Michael Gordy, Bernd Heidergott, Shane Henderson, Jeff Hong, Pierre L'Ecuyer, Elaine Spiller, Pirooz Vakili, and an anonymous referee for providing comments, corrections, and references which led to major improvements to this article.

References

1. Leif Andersen. Efficient simulation of the Heston stochastic volatility model. Working paper, Banc of America Securities, January 2007.
2. Leif Andersen and Mark N. Broadie. A primal-dual simulation algorithm for pricing multi-dimensional American options. *Management Science*, 50(9):1222–1234, 2004.
3. Bruce Ankenman, Barry L. Nelson, and Jeremy Staum. Stochastic kriging for simulation metamodeling. *Operations Research*. Forthcoming.
4. Bouhari Arouna. Adaptative Monte Carlo method, a variance reduction technique. *Monte Carlo Methods and Applications*, 10(1):1–24, 2004.
5. Søren Asmussen and Jan Rosiński. Approximations of small jumps of Lévy processes with a view towards simulation. *Journal of Applied Probability*, 38(2):482–493, 2001.
6. Athanassios N. Avramidis and Pierre L'Ecuyer. Efficient Monte Carlo and quasi-Monte Carlo option pricing under the variance-gamma model. *Management Science*, 52(12):1930–1944, 2006.
7. Russell R. Barton and Martin Meckesheimer. Metamodel-based simulation optimization. In S.G. Henderson and B.L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science pages 535–574. Elsevier, Amsterdam, 2006.
8. Achal Bassamboo, Sandeep Juneja, and Assaf Zeevi. Portfolio credit risk with extremal dependence. *Operations Research*, 56(3):593–606, 2008.
9. R. Evren Baysal, Barry L. Nelson, and Jeremy Staum. Response surface methodology for hedging and trading strategies. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 629–637, Piscataway, N.J., 2008. IEEE Press.
10. Sana Ben Hamida and Rama Cont. Recovering volatility from option prices by evolutionary optimization. *Journal of Computational Finance*, 8(4):43–76, 2005.
11. Dimitri P. Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Nashua, N.H., 1996.
12. Alexandros Beskos and Gareth O. Roberts. Exact simulation of diffusions. *Annals of Applied Probability*, 15(4):2422–2444, 2005.
13. Tomasz R. Bielecki, Stéphane Crépey, Monique Jeanblanc, and Marek Rutkowski. Valuation of basket credit derivatives in the credit migrations environment. In J.R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science pages 471–507. Elsevier, Amsterdam, 2008.

14. John R. Birge. Optimization methods in dynamic portfolio management. In J.R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science pages 845–865. Elsevier, Amsterdam, 2008.
15. Nomesh Bolia and Sandeep Juneja. Monte Carlo methods for pricing financial options. *Sādhanā*, 30(2-3):347–385, 2005.
16. Tarik Borogovac and Pirooz Vakili. Control variate technique: a constructive approach. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 320–327, Piscataway, N.J., 2008. IEEE Press.
17. Michael W. Brandt, Amit Goyal, Pedro Santa-Clara, and Jonathan R. Stroud. A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies*, 18(3):831–873, 2005.
18. Mark N. Broadie, Deniz M. Cicek, and Assaf Zeevi. General bounds and finite-time improvement for stochastic approximation algorithms. Working paper, Columbia University, February 2009. Available via `http://www2.gsb.columbia.edu/faculty/azeevi`.
19. Mark N. Broadie and Paul Glasserman. Estimating security price derivatives using simulation. *Management Science*, 42(2):269–285, 1996.
20. Mark N. Broadie and Özgur Kaya. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, 54(2):217–231, 2006.
21. Luca Capriotti. Least squares importance sampling for Monte Carlo security pricing. *Quantitative Finance*, 8(5):485–497, 2008.
22. Nan Chen. Localization and exact simulation of Brownian motion driven stochastic differential equations. Working paper, Chinese University of Hong Kong, May 2009.
23. Nan Chen and Paul Glasserman. Malliavin Greeks without Malliavin calculus. *Stochastic Processes and their Applications*, 117:1689–1723, 2007.
24. Nan Chen and L Jeff. Hong. Monte Carlo simulation in financial engineering. In S.G. Henderson, B. Biller, H. Hsieh, J. Shortle, J.D. Tew, and R.R. Barton, editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 919–931, Piscataway, N.J., 2007. IEEE Press.
25. Zhiyong Chen and Paul Glasserman. Fast pricing of basket default swaps. *Operations Research*, 56(2):286–303, 2008.
26. Thomas F. Coleman, Yuying Li, and Maria-Cristina Patron. Total risk minimization using Monte Carlo simulations. In J.R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science pages 593–635. Elsevier, Amsterdam, 2008.
27. Rama Cont and Peter Tankov. *Financial Modelling with Jump Processes*. Chapman & Hall/CRC, Boca Raton, 2004.
28. Gerard Cornuejols and Reha Tütüncü. *Optimization Methods in Finance*. Cambridge University Press, New York, 2007.
29. Jérôme Detemple, René Garcia, and Marcel Rindisbacher. Intertemporal asset allocation: a comparison of methods. *Journal of Banking and Finance*, 29:2821–2848, 2005.
30. Samuel M.T. Ehrlichman and Shane G. Henderson. Adaptive control variates for pricing multidimensional American options. *Journal of Computational Finance*, 11(1), 2007.
31. Markus Emsermann and Burton Simon. Improving simulation efficiency with quasi control variates. *Stochastic Models*, 18(3):425–448, 2002.
32. Frank J. Fabozzi, editor. *The Handbook of Mortgage-Backed Securities*. McGraw-Hill, New York, 5th edition, 2001.
33. Frank J. Fabozzi, Petter N. Kolm, Dessislava Pachamanova, and Sergio M. Focardi. *Robust Portfolio Optimization and Management*. John Wiley & Sons, Hoboken, N.J., 2007.
34. Michael C. Fu. Optimization for simulation: theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.
35. Michael C. Fu. Gradient estimation. In S.G. Henderson and B.L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science pages 575–616. Elsevier, Amsterdam, 2006.

36. Michael C. Fu. Variance gamma and Monte Carlo. In M.C. Fu, R.A. Jarrow, J.-Y.J. Yen, and R.J. Elliott, editors, *Advances in Mathematical Finance*, pages 21–34. Springer-Verlag, New York, 2008.
37. Michael C. Fu. What you should know about simulation and derivatives. *Naval Research Logistics*, 55(8):723–736, 2008.
38. Michael C. Fu and Jian-Qiang Hu. Sensitivity analysis for Monte Carlo simulation of option pricing. *Probability in the Engineering and Informational Sciences*, 9(3):417–446, 1995.
39. Michael C. Fu, Scott B. Laprise, Dilip B. Madan, Yi Su, and Rongwen Wu. Pricing American options: a comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3):39–88, 2001.
40. Kay Giesecke. Portfolio credit risk: top down vs. bottom up approaches. In R. Cont, editor, *Frontiers in Quantitative Finance: Credit Risk and Volatility Modeling*, pages 251–268. John Wiley & Sons, Hoboken, N.J., 2008.
41. Kay Giesecke. An overview of credit derivatives. Working paper, Stanford University, March 2009. Available via http://www.stanford.edu/dept/MSandE/people/faculty/giesecke/publications.html
42. Kay Giesecke, Hossein Kakavand, and Mohammad Mousavi. Simulating point processes by intensity projection. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 560–568, Piscataway, N.J., 2008. IEEE Press.
43. Michael B. Giles. Monte Carlo evaluation of sensitivities in computational finance. In E.A. Lipitakis, editor, *HERCMA 2007 Conference Proceedings*, 2007. Available via http://www.aueb.gr/pympe/hercma/proceedings2007/H07-FULL-PAPERS-1/GILES-INVITED-1.pdf.
44. Michael B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
45. Michael B. Giles and Paul Glasserman. Smoking adjoints: fast Monte Carlo Greeks. *Risk*, 19:88–92, 2006.
46. Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2004.
47. Paul Glasserman. Measuring marginal risk contributions in credit portfolios. *Journal of Computational Finance*, 9(1):1–41, 2005.
48. Paul Glasserman. Calculating portfolio credit risk. In J.R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science pages 437–470. Elsevier, Amsterdam, 2008.
49. Paul Glasserman, Wanmo Kang, and Perwez Shahabuddin. Fast simulation of multifactor portfolio credit risk. *Operations Research*, 56(5):1200–1217, 2008.
50. Paul Glasserman and Kyoung-Kuk Kim. Gamma expansion of the Heston stochastic volatility model. *Finance and Stochastics*. Forthcoming.
51. Paul Glasserman and Jingyi Li. Importance sampling for portfolio credit risk. *Management Science*, 51(11):1643–1656, 2005.
52. Paul Glasserman and Zongjian Liu. Estimating Greeks in simulating Lévy-driven models. Working paper, Columbia University, October 2008. Available via http://www.paulglasserman.net.
53. Peter W. Glynn. Monte Carlo simulation of diffusions. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 556–559, Piscataway, N.J., 2008. IEEE Press.
54. Michael B. Gordy and Sandeep Juneja. Nested simulation in portfolio risk measurement. Finance and Economics Discussion Series 2008-21, Federal Reserve Board, April 2008. Available via http://www.federalreserve.gov/Pubs/feds/2008/200821.
55. Robert B. Gramacy and Herbert K.H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

56. Martin B. Haugh and Leonid Kogan. Duality theory and approximate dynamic programming for pricing American options and portfolio optimization. In J.R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science pages 925–948. Elsevier, Amsterdam, 2008.
57. Martin B. Haugh, Leonid Kogan, and Jiang Wang. Evaluating portfolio policies: a dual approach. *Operations Research*, 54(3):405–418, 2006.
58. Bernd Heidergott. Option pricing via Monte Carlo simulation: a weak derivative approach. *Probability in the Engineering and Informational Sciences*, 15:335–349, 2001.
59. Bernd Heidergott, Felisa J. Vázquez-Abad, and Warren Volk-Makarewicz. Sensitivity estimation for Gaussian systems. *European Journal of Operational Research*, 187:193–207, 2008.
60. Shane G. Henderson and Sujin Kim. The mathematics of continuous-variable simulation optimization. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 122–132, Piscataway, N.J., 2008. IEEE Press.
61. L. Jeff Hong. Estimating quantile sensitivities. *Operations Research*, 57(1):118–130, 2009.
62. L. Jeff Hong and Guangwu Liu. Simulating sensitivities of conditional value at risk. *Management Science*, 55(2):281–293, 2009.
63. Sandeep Juneja and Himanshu Kalra. Variance reduction techniques for pricing American options. *Journal of Computational Finance*, 12(3):79–102, 2009.
64. Vladimir K. Kaishev and Dimitrina S. Dimitrova. Dirichlet bridge sampling for the variance gamma process: pricing path-dependent options. *Management Science*, 55(3):483–496, 2009.
65. K H. Felix Kan, R. Mark Reesor, Tyson. Whitehead, and Matt. Davison. Correcting the bias in Monte Carlo estimators of American-style option values. Submitted to *Monte Carlo and Quasi-Monte Carlo Methods 2008*.
66. Sujin Kim and Shane G. Henderson. Adaptive control variates for finite-horizon simulation. *Mathematics of Operations Research*, 32(3):508–527, 2007.
67. Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, New York, 1992.
68. Hai Lan. Tuning the parameters of a two-level simulation procedure with screening. Working paper, Northwestern University, available via `http://users.iems.northwestern.edu/~staum`, March 2009.
69. Hai Lan, Barry L. Nelson, and Jeremy Staum. Two-level simulations for risk management. In S. Chick, C.-H. Chen, S.G. Henderson, and E. Yücesan, editors, *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*, pages 102–107, Fontainebleau, France, 2007. INSEAD. Available via `http://www.informs-cs.org/2007informs-csworkshop/23.pdf`.
70. Hai Lan, Barry L. Nelson, and Jeremy Staum. Confidence interval procedures for expected shortfall risk measurement via two-level simulation. Working paper 08-02, Department of IEMS, Northwestern University, November 2008. Available via `http://users.iems.northwestern.edu/~staum`.
71. Pierre L'Ecuyer. A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science*, 36(11):1364–1383, 1990.
72. Pierre L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 13(3):307–349, 2009
73. Pierre L'Ecuyer and Bruno Tuffin. Approximate zero-variance simulation. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 170–181, Piscataway, N.J., 2008. IEEE Press.
74. Shing-Hoi Lee. *Monte Carlo computation of conditional expectation quantiles*. PhD thesis, Stanford University, 1998.
75. Vadim Lesnevski, Barry L. Nelson, and Jeremy Staum. Simulation of coherent risk measures based on generalized scenarios. *Management Science*, 53(11):1756–1769.
76. Guangwu Liu and L. Jeff Hong. Kernel estimation of quantile sensitivities. *Naval Research Logistics*. Forthcoming.

77. Guangwu Liu and L. Jeff Hong. Pathwise estimation of the Greeks of financial options. Working paper, Hong Kong University of Science and Technology, August 2008. Available via http://ihome.ust.hk/~liugw.
78. Ming Liu, Barry L. Nelson, and Jeremy Staum. An adaptive procedure for point estimation of expected shortfall. Working paper 08-03, Department of IEMS, Northwestern University, October 2008. Available via http://users.iems.northwestern.edu/~staum.
79. Ming Liu and Jeremy Staum. Estimating expected shortfall with stochastic kriging. Working paper, Northwestern University, March 2009. Available via http://users.iems.northwestern.edu/~staum.
80. Alexander J. McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative Risk Management*. Princeton University Press, Princeton, N.J., 2005.
81. Kumar Muthuraman and Haining Zha. Simulation-based portfolio optimization for large portfolios with transactions costs. *Mathematical Finance*, 18(1):115–134, 2008.
82. Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.
83. Sigurdur Ólafsson. Metaheuristics. In S.G. Henderson and B.L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science pages 633–654. Elsevier, Amsterdam, 2006.
84. Soumik Pal. Computing strategies for achieving acceptability: a Monte Carlo approach. *Stochastic Processes and their Applications*, 117(11):1587–1605, 2007.
85. Raghu Pasupathy, Bruce W. Schmeiser, Michael R. Taafe, and Jin Wang. Control variate estimation using estimated control means. *IIE Transactions*. Forthcoming.
86. Jose Antonio Perez. *Convergence of numerical schemes in the total variation sense*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, 2004.
87. Jérémy Poirot and Peter Tankov. Monte Carlo option pricing for tempered stable (CGMY) processes. *Asia-Pacific Financial Markets*, 13:327–344, 2006.
88. Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, Hoboken, N.J., 2007.
89. Warren B. Powell. What you should know about approximate dynamic programming. *Naval Research Logistics*, 56(3):239–249, 2009.
90. Bruce W. Schmeiser, Michael R. Taafe, and Jin Wang. Biased control-variate estimation. *IIE Transactions*, 33(3):219–228, 2001.
91. Adam Speight. A multilevel approach to control variates. *Journal of Computational Finance*. Forthcoming.
92. Jeremy Staum. Incomplete markets. In J.R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science pages 511–563. Elsevier, Amsterdam, 2008.
93. Yi Su and Michael C. Fu. Optimal importance sampling in securities pricing. *Journal of Computational Finance*, 5(4):27–50, 2002.
94. Yunpeng Sun, Daniel W. Apley, and Jeremy Staum. $1\frac{1}{2}$-level simulation for estimating the variance of a conditional expectation. Working paper, Northwestern University, 2009.
95. Jules H. van Binsbergen and Michael W. Brandt. Solving dynamic portfolio choice problems by recursing on optimized portfolio weights or on the value function? *Computational Economics*, 29:355–367, 2007.
96. Gang Zhao, Tarik Borogovac, and Pirooz Vakili. Efficient estimation of option price and price sensitivities via structured database Monte Carlo (SDMC). In S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, and R.R. Barton, editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 984–990, Piscataway, N.J., 2007. IEEE Press.
97. Gang Zhao and Pirooz Vakili. Monotonicity and stratification. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 313–319, Piscataway, N.J., 2008. IEEE Press.
98. Gang Zhao, Yakun Zhou, and Pirooz Vakili. A new efficient simulation strategy for pricing path-dependent options. In L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, and R.M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference*, pages 703–710, Piscataway, N.J., 2006. IEEE Press.

Part II
Invited Articles

Particle Markov Chain Monte Carlo for Efficient Numerical Simulation

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein

Abstract Markov Chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC) methods are the two most popular classes of algorithms used to sample from general high-dimensional probability distributions. The theoretical convergence of MCMC algorithms is ensured under weak assumptions, but their practical performance is notoriously unsatisfactory when the proposal distributions used to explore the space are poorly chosen and/or if highly correlated variables are updated independently. We show here how it is possible to systematically design potentially very efficient high-dimensional proposal distributions for MCMC by using SMC techniques. We demonstrate how this novel approach allows us to design effective MCMC algorithms in complex scenarios. This is illustrated by a problem of Bayesian inference for a stochastic kinetic model.

1 Introduction

Assume that we are interested in sampling from a probability distribution $\pi(\mathbf{x})$ where $\mathbf{x} = (x_1, \ldots, x_T)$ for some $T > 1$. For ease of presentation, we assume that each $x_i \in \mathcal{X}$ for some space $\mathcal{X}$. For complex problems, it is impossible to sample directly from $\pi(\mathbf{x})$.

Christophe Andrieu
Department of Mathematics, Bristol University, UK
url: http://www.stats.bris.ac.uk/~maxca

Arnaud Doucet
The Institute of Statistical Mathematics, Japan
url: http://www.cs.ubc.ca/~arnaud

Roman Holenstein
Department of Computer Science, University of British Columbia, Canada
url: http://www.cs.ubc.ca/~romanh

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_3,

The standard MCMC approach consists of sampling long realisations of ergodic Markov chains with invariant distribution $\pi(\mathbf{x})$. The Metropolis-Hastings (MH) algorithm is the main known generic mechanism to define such updates. It requires the choice of proposal distributions that sample possible states for the Markov chain which are either accepted or rejected. A popular application of this principle consists, for example, of repeatedly updating in turn the lower-dimensional components x_i of $\mathbf{x}$ conditional upon the remaining components $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_T)$. The size reduction often allows for a better choice of local proposal distributions. Although this strategy can result in an improvement over the full updating of $\mathbf{x}$ in one block, it can still be ineffective when highly dependent components are not updated simultaneously.

SMC methods are an alternative to MCMC methods where a swarm of samples, named particles, evolves towards the distribution of interest according to a combination of importance sampling (IS) and resampling; see [6] for a collection of articles on the subject and [11, chapters 3 and 4]. Where traditional IS would try to directly produce weighted samples to approximate $\pi(\mathbf{x})$, and most likely fail for the same reason that an independent MH (IMH) algorithm would fail, SMC methods decompose the problem of sampling from $\pi(\mathbf{x})$ into a series of "simpler" sub-problems. We introduce a sequence of intermediate "bridging" probability distributions of increasing dimension $\{\pi_n(\mathbf{x}_n), n = 1, \ldots, T-1\}$ with $\mathbf{x}_n = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$, then we sequentially sample approximately from $\pi_1(\mathbf{x}_1), \pi_2(\mathbf{x}_2), \ldots, \pi_{T-1}(\mathbf{x}_{T-1})$ and $\pi_T(\mathbf{x}) = \pi(\mathbf{x})$. As is the case for MCMC algorithms this dimension reduction usually allows for the design of better proposal distributions. In this paper we present a recent addition to the Monte Carlo toolbox named Particle MCMC (PMCMC) which aims to take advantage of the differing strengths of MCMC and SMC methods.

The rest of this paper is organised as follows. In Section 2, we briefly review SMC methods and discuss some of their properties. In Section 3 we present the *particle IMH sampler*, a recently developed IMH update targeting $\pi(\mathbf{x})$ which has the capability of using SMC approximations of $\pi(\mathbf{x})$ as a proposal mechanism [1]. In Section 4, we review extensions of this basic update to the case where we are interested in sampling from $\pi(\theta, \mathbf{x})$ on $\Theta \times \mathcal{X}^T$: the *particle marginal MH sampler* and the *particle Gibbs sampler*. As shown in [1], such updates are of particular interest in the context of inference in state-space models, but their relevance is not limited to such models. Connections to previous work are discussed in Section 5. Finally in Section 6, we demonstrate the performance of the methodology in the context of inference in a stochastic kinetic model. Space constraints prevent us from detailing all the results and proofs; we refer the reader to [1] for details.

2 Sequential Monte Carlo Methods

We briefly review here the principle of SMC methods to sample from a given target $\pi(\mathbf{x})$. We first introduce an artificial sequence of bridging distributions

$\{\pi_n(\mathbf{x}_n); n = 1, \ldots, T-1\}$ of increasing dimension and define $\pi_T(\mathbf{x}_T) = \pi(\mathbf{x})$. Each distribution is assumed known up to a normalising constant, that is

$$\pi_n(\mathbf{x}_n) = \frac{\gamma_n(\mathbf{x}_n)}{Z_n},$$

where $\gamma_n : \mathcal{X}^n \to \mathbb{R}^+$ can be evaluated pointwise, but Z_n is unknown. We will use the notation Z for Z_T. An SMC algorithm also requires us to specify an importance distribution $q_1(x_1)$ on $\mathcal{X}$ in order to initialise the recursion at time 1 and a family of proposal distributions $\{q_n(x_n|\mathbf{x}_{n-1}); n = 2, \ldots, T\}$ in order to extend $\mathbf{x}_{n-1} \in \mathcal{X}^{n-1}$ by sampling $x_n \in \mathcal{X}$ conditional upon $\mathbf{x}_{n-1}$ at time instants $n = 2, \ldots, T$. Guidelines on how to best select $q_n(x_n|\mathbf{x}_{n-1})$ are well known, and the main recommendation is to use the conditional distribution $\pi_n(x_n|\mathbf{x}_{n-1})$ or an approximation [6], [11]. An SMC algorithm also involves a resampling procedure of the N particles, which relies on a family of probability distributions $\{r(\cdot|\mathbf{w}), \mathbf{w} \in [0,1]^N\}$ on $\{1, \ldots, N\}^N$. The resampling step is usually necessary as in most applications the variance of the importance weights would otherwise typically increase exponentially with n.

The algorithm proceeds as follows to produce a sequence of samples $\{\mathbf{X}_n^i, i = 1, \ldots, N\}$ for $n = 1, \ldots, T$. Note that we adopt below the convention that whenever the index i is used we mean "for all $i \in \{1, \ldots, N\}$." Further on, we also use the standard convention whereby capital letters are used for random variables while lower case letters are used for their values. We also use the notation $\mathbf{W}_n = \left(W_n^1, \ldots, W_n^N\right)$ and $\mathbf{A}_n = \left(A_n^1, \ldots, A_n^N\right)$.

Sequential Monte Carlo Algorithm

$\underline{n = 1}$

- Sample $\mathbf{X}_1^i \sim q_1(\cdot)$.
- Update and normalise the weights

$$w_1\left(\mathbf{X}_1^i\right) = \frac{\gamma_1(\mathbf{X}_1^i)}{q_1(\mathbf{X}_1^i)}, \quad W_1^i = \frac{w_1\left(\mathbf{X}_1^i\right)}{\sum_{k=1}^N w_1\left(\mathbf{X}_1^k\right)}. \tag{1}$$

For $n = 2, \ldots, T$

- Sample $\mathbf{A}_{n-1} \sim r(\cdot|\mathbf{W}_{n-1})$.
- Sample $X_n^i \sim q_n(\cdot|\mathbf{X}_{n-1}^{A_{n-1}^i})$ and set $\mathbf{X}_n^i = (\mathbf{X}_{n-1}^{A_{n-1}^i}, X_n^i)$.
- Update and normalise the weights

$$w_n\left(\mathbf{X}_n^i\right) = \frac{\gamma_n\left(\mathbf{X}_n^i\right)}{\gamma_{n-1}\left(\mathbf{X}_{n-1}^{A_{n-1}^i}\right) q_n\left(X_n^i \middle| \mathbf{X}_{n-1}^{A_{n-1}^i}\right)}, \quad W_n^i = \frac{w_n\left(\mathbf{X}_n^i\right)}{\sum_{k=1}^N w_n\left(\mathbf{X}_n^k\right)}. \tag{2}$$

The variable A_{n-1}^i plays an important role in our formulation of SMC methods, and represents the index of the "parent" at time $n-1$ of particle $\mathbf{X}_n^i$ for $n = 2, \ldots, T$. The vector $\mathbf{A}_n$ is thus a random mapping defined on $\{1, \ldots, N\} \to \{1, \ldots, N\}^N$, and

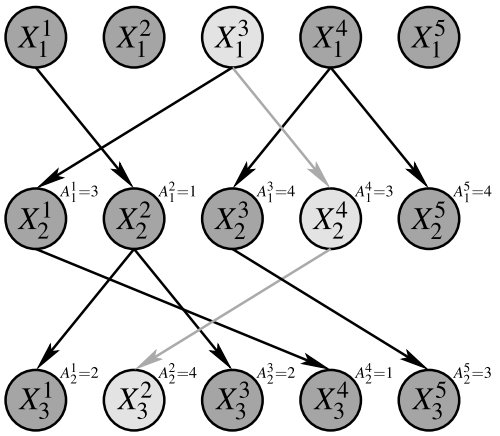


Fig. 1 Example of ancestral lineages generated by an SMC algorithm for $N = 5$ and $T = 3$. The lighter path is $X_{1:3}^2 = (X_1^3, X_2^4, X_3^2)$ and its ancestral lineage is $B_{1:3}^2 = (3, 4, 2)$.

the resampling procedure is thus interpreted here as being the operation by which child particles at time n choose their parent particles at time $n-1$ according to a probability $r(\cdot|\mathbf{W}_{n-1})$ dependent on the parents' weights $\mathbf{W}_{n-1}$, or "fitness." The introduction of the variables $\mathbf{A}_n$ allows us to keep track of the "genealogy" of particles and is necessary to describe precisely one of the algorithms introduced later on (see Section 4). For this purpose, for $i = 1, \ldots, N$ and $n = 1, \ldots, T$ we introduce B_n^i the index the ancestor particle of $\mathbf{X}_T^i$ at generation n had at that time. More formally for $i = 1, \ldots, N$ we define $B_T^i := i$ and for $n = T-1, \ldots, 1$ we have the following backward recursive relation $B_n^i := A_n^{B_{n+1}^i}$. As a result for any $i = 1, \ldots, N$ we have the identity $\mathbf{X}_T^i = (X_1^{B_1^i}, X_2^{B_2^i}, \ldots, X_{T-1}^{B_{T-1}^i}, X_T^{B_T^i})$ and $\mathbf{B}_T^i = (B_1^i, B_2^i, \ldots, B_{T-1}^i, B_T^i = i)$ is the ancestral 'lineage' of a particle. This is illustrated in Figure 1.

This SMC algorithm provides an approximation of $\pi(\mathbf{x})$ and its normalising constant Z given by

$$\hat{\pi}(\mathbf{x}) = \sum_{i=1}^{N} W_T^i \delta_{\mathbf{X}_T^i}(\mathbf{x}) \text{ and } \hat{Z} = \prod_{n=1}^{T} \left[\frac{1}{N} \sum_{i=1}^{N} w_n\left(\mathbf{X}_n^i\right) \right]. \tag{3}$$

The validity of the algorithms presented here relies on a set of very weak assumptions. First we require the importance weight functions $w_n(\mathbf{x}_n)$ to be properly defined; *i.e.* the supports of the proposals cover the supports of the targets. Second it also relies on the following assumptions on the resampling procedure.

Let $O_n^i = \sum_{k=1}^{N} \mathbb{I}\left\{A_n^k = i\right\}$ be the number of offspring of particle i at time n. Then for any $i = 1, \ldots, N$ and $n = 1, \ldots, T$ the resampling scheme must satisfy the following unbiasedness condition

$$\mathbb{E}\left[O_n^i | \mathbf{W}_n\right] = N W_n^i. \tag{4}$$

In fact in practice, for computational efficiency, $\mathbf{O}_n = \left(O_n^1, \ldots, O_n^N\right)$ is typically drawn first (*i.e.* without explicit reference to $\mathbf{A}_n$) according to a probability distribution $s(\cdot|\mathbf{W}_n)$ such that (4) holds and the offspring then matched to their parents. For example, the simplest unbiased resampling algorithm consists of sampling $\mathbf{O}_n$ according to a multinomial distribution of parameters $(N, \mathbf{W}_n)$. More sophisticated schemes such as residual resampling [11] and stratified resampling [9] also satisfy (4). Once $\mathbf{O}_n$ has been sampled, this is followed by a deterministic allocation procedure of the child particles to the parents, which defines a new set of indices *e.g.* the O_n^1 first child particles are associated to the parent particle number 1, *i.e.* $A_n^1 = 1, \ldots, A_n^{O_n^1} = 1$, likewise for the O_n^2 following child particles and the parent particle number 2, *i.e.* $A_n^{O_n^1+1} = 2, \ldots, A_n^{O_n^1+O_n^2} = 2$ etc.

Further on, we will impose the slightly stronger unbiasedness condition

$$r\left(A_n^i = k|\mathbf{W}_n\right) = W_n^k \ . \tag{5}$$

Note that even if (4) holds then (5) is not necessarily satisfied, for example by the standard deterministic allocation procedure, but this property can be easily enforced by the addition of a random permutation of these indices. As we shall see our indexing system makes the writing of the probability distributions underpinning our algorithms extremely simple.

Many sharp convergence results have been established for SMC methods including Lp-bounds, central limit theorems, large deviations results etc.; see [4] for a detailed overview of these results.

3 Particle Independent MH Sampler

The aim of this review is to outline how SMC approximations of $\pi(\mathbf{x})$ can be used as proposal distributions for MCMC algorithms. It is natural to suggest the use of the unconditional distribution of a particle generated by an SMC algorithm targeting $\pi(\mathbf{x})$ as a proposal distribution for an IMH algorithm targeting $\pi(\mathbf{x})$. This is likely to result in a very efficient IMH algorithm as discussed in the previous section. It is easy to sample from this unconditional distribution by running an SMC targeting $\pi(\mathbf{x})$ to obtain $\hat{\pi}(\mathbf{x})$ given in (3) and then sample from $\hat{\pi}(\mathbf{x})$. However, computing the MH acceptance ratio of such a MH update would then require us to be able to evaluate

$$q(\mathbf{x}) = \mathbb{E}\left(\hat{\pi}(\mathbf{x})\right) \ , \tag{6}$$

where the expectation is with respect to all the variables used to generate $\hat{\pi}(\mathbf{x})$: this is practically impossible. We show below how it is possible to bypass this problem. We would like to stress at this point the fact that we do not believe that the PIMH algorithm on its own is a practically relevant alternative to standard SMC approximations of $\pi(\mathbf{x})$. However its pedagogical value should become clear below while one should bear in mind that, as it is the case with standard IMH type updates, such

an update can be of interest when used in conjunction with other MCMC updates. In order to illustrate the simplicity of the implementation of our approach we describe a particular instance of the methodology in order to sample from $\pi(\mathbf{x})$, where $\mathbf{x}$ is updated in one single block.

3.1 Algorithm

In order to sample from $\pi(\mathbf{x})$ the particle IMH (PIMH) sampler proceeds as follows (with the notation of Section 2, in particular (3)):

Particle Independent Metropolis-Hastings Sampler

Initialization, $m = 0$

- Run an SMC algorithm targeting $\pi(\mathbf{x})$, sample $\mathbf{X}(0) \sim \hat{\pi}(\cdot)$ and compute $\hat{Z}(0)$.

At iteration $m \geq 1$

- Run an SMC algorithm targeting $\pi(\mathbf{x})$, sample $\mathbf{X}^* \sim \hat{\pi}(\cdot)$ and compute $\hat{Z}^*$.
- With probability

$$1 \wedge \frac{\hat{Z}^*}{\hat{Z}(m-1)}, \tag{7}$$

set $\mathbf{X}(m) = \mathbf{X}^*$ and $\hat{Z}(m) = \hat{Z}^*$, otherwise set $\mathbf{X}(m) = \mathbf{X}(m-1)$ and $\hat{Z}(m) = \hat{Z}(m-1)$.

The output of the algorithm is the chain $\{\mathbf{X}(m)\}_{m \geq 0}$. Note the interesting property that the acceptance probability (7) converges to 1 as $N \to \infty$ since both $\hat{Z}^*$ and $\hat{Z}(m-1)$ are consistent estimates of the unknown normalising constant Z, under weak assumptions.

3.2 Extended Proposal and Target Distributions

We show here the surprising result that the invariant distribution of the PIMH sampler is $\pi(\mathbf{x})$ *for any* $N \geq 1$. The key to establish this result is to reformulate the PIMH as a standard IMH sampler defined on an extended state-space with a suitable invariant distribution.

Sampling from the proposal $q(\mathbf{x})$ in (6) requires sampling $\hat{\pi}(\mathbf{x})$ then drawing one particle $\mathbf{X}_T$ from $\hat{\pi}(\mathbf{x})$ by setting $\mathbf{X} = \mathbf{X}_T^K$ where $\Pr\left(K = k | \hat{\pi}(\mathbf{x})\right) = W_T^k$. Denoting for $n = 1, \ldots, T$ the set of N simulated $\mathcal{X}$-valued random variables at time n as $\overline{\mathbf{X}}_n := \left(X_n^1, \ldots, X_n^N\right) \in \mathcal{X}^N$, then the joint probability distribution of all the random variables used in the proposal distribution is

$$q(k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}) = w_T^k \, \psi(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}) \tag{8}$$

where w_T^k is a realization of W_T^K and

$$\psi\left(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right) := \left(\prod_{i=1}^{N} q_1\left(x_1^i\right)\right)\prod_{n=2}^{T}\left(r\left(\mathbf{a}_{n-1}|\mathbf{w}_{n-1}\right)\prod_{i=1}^{N} q_n\left(x_n^i\middle|\mathbf{x}_{n-1}^{a_{n-1}^i}\right)\right)$$

is the distribution of all the random variables generated by the SMC sampler described in Section 2, which is defined on $\mathcal{X}^{TN}\times\{1,\ldots,N\}^{(T-1)N+1}$. We now define, on the same space, the following artificial target probability distribution

$$\begin{aligned}&\tilde{\pi}\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right) \qquad (9)\\ &= \frac{\pi\left(\mathbf{x}_T^k\right)}{N^T}\frac{\psi\left(\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)}{q_1(x_1^{b_1^k})\prod_{n=2}^{T} r(b_{n-1}^k|\mathbf{w}_{n-1})q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})}\\ &= \frac{\pi\left(\mathbf{x}_T^k\right)}{N^T}\prod_{i=1,i\neq b_1^k}^{T} q_1(x_1^i)\prod_{n=1}^{T-1} r(\mathbf{a}_{n-1}^{-b_n^k}|\mathbf{w}_{n-1},b_n^k)\prod_{i=1,i\neq b_n^k}^{T} q_n\left(x_n^i\middle|\mathbf{x}_{n-1}^{a_{n-1}^i}\right)\end{aligned}$$

where we have used the notation $\mathbf{a}_{n-1}^{-b_n^k}=\mathbf{a}_{n-1}\backslash\{a_{n-1}^{b_n^k}\}$. By construction, we have $\mathbf{X}_T^K\sim\pi$ under $\tilde{\pi}$ and it is easy to check that

$$\begin{aligned}\frac{\tilde{\pi}\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)}{q\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)} &= \frac{1}{N^T}\frac{\pi\left(\mathbf{x}_T^k\right)}{w_T^k\; q_1(x_1^{b_1^k})\prod_{n=2}^{T} r(b_{n-1}^k|\mathbf{w}_{n-1})q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})}\\ &= \frac{1}{N^T}\frac{\pi\left(\mathbf{x}_T^k\right)}{q_1(x_1^{b_1^k})\prod_{n=2}^{T} q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})\prod_{n=1}^{T} w_n^{b_n^k}}\\ &= \frac{\pi\left(\mathbf{x}_T^k\right)\prod_{n=1}^{T}\left(\frac{1}{N}\sum_{m=1}^{N} w_n\left(\mathbf{x}_n^m\right)\right)}{q_1(x_1^{b_1^k})\prod_{n=2}^{T} q_n(x_n^{b_n^k}|\mathbf{x}_{n-1}^{b_{n-1}^k})\prod_{n=1}^{T} w_n(\mathbf{x}_n^{b_n^k})}\\ &= \frac{\hat{Z}}{Z}.\end{aligned}$$

In the calculations above we have used (5) on the second line whereas the final result is obtained thanks to the definitions of the incremental weights (1)–(2) and of the normalising constant estimate (3). This allows us to conclude that the PIMH sampler is a standard IMH sampler of target distribution $\tilde{\pi}\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)$ and proposal distribution $q\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)$. This indeed follows by the definition of $q\left(k,\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_T,\mathbf{a}_1,\ldots,\mathbf{a}_{T-1}\right)$ and the last calculation above which explains the form of the acceptance probability of the PIMH. This IMH sampler is automatically irreducible and aperiodic as we have made the assumption that the importance weight functions $w_n\left(\mathbf{x}_n\right)$ are properly defined.

3.3 Structure of the Invariant Distribution and Alternative Algorithm

To better understand the structure of the artificial target $\tilde{\pi}$, we explain here how we would sample from it. The algorithm follows straightforwardly from (9).

- Sample uniformly on $\{1,\ldots,N\}^T$ an ancestral lineage $\mathbf{B}_T^K = \left(B_1^K, B_2^K, \ldots, B_T^K\right)$. Recall that we have $B_T^K = K$, $B_n^K := A_n^{B_{n+1}^K}$.
- Sample $\mathbf{X}_T^K = (X_1^{B_1^K}, X_2^{B_2^K}, \ldots, X_{T-1}^{B_{T-1}^K}, X_T^{B_T^K}) \sim \pi$. Obviously we cannot do this, which is why we are using MCMC in the first place.
- Sample all the remaining variables conditional upon $\left(\mathbf{X}_T^K, \mathbf{B}_T^K\right)$ according to their conditional distribution under $\tilde{\pi}$.

Sampling from this conditional distribution under $\tilde{\pi}$ can be achieved using the following conditional SMC algorithm. We recall that $\mathbf{A}_{n-1}^{-B_n^K} = \mathbf{A}_{n-1} \backslash \{A_{n-1}^{B_n^K}\}$.

Conditional Sequential Monte Carlo Algorithm

$\underline{n=1}$

- For $i \neq B_1^K$, sample $\mathbf{X}_1^i \sim q_1(\cdot)$.
- Compute $w_1\left(\mathbf{X}_1^i\right)$ and normalise the weights $W_1^i \propto w_1\left(\mathbf{X}_1^i\right)$.

For $\underline{n=2,\ldots,T}$

- Sample $\mathbf{A}_{n-1}^{-B_n^K} \sim r(\cdot|\mathbf{W}_{n-1}, A_{n-1}^{B_n^K})$.
- For $i \neq B_n^K$, sample $X_n^i \sim q_n(\cdot|\mathbf{X}_{n-1}^{A_{n-1}^i})$ and set $\mathbf{X}_n^i = (\mathbf{X}_{n-1}^{A_{n-1}^i}, X_n^i)$.
- Compute $w_n\left(\mathbf{X}_n^i\right)$ and normalise the weights $W_n^i \propto w_n\left(\mathbf{X}_n^i\right)$.

In the case of multinomial resampling, denoting $\mathcal{B}(a,\mathbf{b})$ the binomial distribution of parameters $(a,\mathbf{b})$, $\mathcal{B}^+(a,\mathbf{b})$ the binomial distribution of similar parameters restricted to $\{1,\ldots,N\}$ and $\mathcal{M}(a,\mathbf{b})$ the multinomial distribution, an efficient approach to sample $\mathbf{A}_{n-1}^{-B_n^K} \sim r(\cdot|\mathbf{W}_{n-1}, A_{n-1}^{B_n^K})$ proceeds as follows.

- Sample $O_{n-1}^{B_n^K} \sim \mathcal{B}^+\left(N, W_{n-1}^{B_n^K}\right)$.
- Allocate randomly $O_{n-1}^{B_n^K} - 1$ parent indexes uniformly in $\{1,\ldots,N\} \backslash \left\{B_n^K\right\}$ and set these parents equal to B_{n-1}^K.
- For $i \neq B_n^K$ compute $\overline{W}_{n-1}^i \propto W_{n-1}^i$ with $\sum_{i=1, i\neq B_n^K}^N \overline{W}_{n-1}^i = 1$ and denote $\overline{\mathbf{W}}_{n-1}$ these $N-1$ weights.
- Sample $\mathbf{O}_{n-1} \backslash \left\{O_{n-1}^{B_n^K}\right\} \sim \mathcal{M}\left(N - O_{n-1}^{B_n^K}, \overline{\mathbf{W}}_{n-1}\right)$.
- Allocate randomly the associated parent indexes uniformly in $\{1,\ldots,N\} \backslash \left\{\text{indexes with parents equal to } B_{n-1}^K\right\}$.

This procedure follows directly from the fact that $\mathbf{O}_{n-1} \sim \mathcal{M}(N, \mathbf{W}_{n-1})$ so the marginal distribution of $O_{n-1}^{B_n^K}$ is $\mathcal{B}\left(N, W_{n-1}^{B_n^K}\right)$ and, conditional upon $O_{n-1}^{B_n^K}$, we have $\mathbf{O}_{n-1} \backslash \left\{O_{n-1}^{B_n^K}\right\} \sim \mathcal{M}\left(N - O_{n-1}^{B_n^K}, \overline{\mathbf{W}}_{n-1}\right)$. Finally conditional upon $O_{n-1}^{B_n^K} \geq 1$ we have $O_{n-1}^{B_n^K} \sim \mathcal{B}^+\left(N, W_{n-1}^{B_n^K}\right)$.

Note that an alternative to the PIMH algorithm to sample from $\pi(\mathbf{x})$ consists of alternating a conditional SMC step to update $\widehat{\pi}(\mathbf{x})$ and a step to sample $\left(\mathbf{X}_T^K, \mathbf{B}_T^K\right)$ from $\widehat{\pi}(\mathbf{x})$. For any $N \geq 1$, this algorithm admits $\pi(\mathbf{x})$ as invariant distribution as it is just a (collapsed) Gibbs sampler of invariant distribution $\widetilde{\pi}(k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1})$. Contrary to the PIMH, it is here necessary to have $N \geq 2$ to ensure irreducibility of this sampler.

3.4 Using All the Particles

The standard estimate of $\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ for M MCMC iterations is $\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{X}(m))$. A possible criticism of the PIMH is that in the implementation above we generate N particles at each iteration m of the MCMC algorithm to decide whether to accept or reject one single candidate. This might appear wasteful. However, it can be shown that the estimate

$$\frac{1}{M}\sum_{m=1}^{M}\left(\sum_{i=1}^{N} W_T^i(m)\, f(\mathbf{X}_T^i(m))\right)$$

converges also towards $\int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ as $M \to \infty$ where $\{W_T^i(m), \mathbf{X}_T^i(m)\}$ corresponds to the set of normalized weights and particles used to compute $\hat{Z}(m)$. Following [8] it is also possible to propose an estimate which recycles the candidate populations of particles rejected by the PIMH; see [1] for details.

4 Particle Marginal MH Sampler and Particle Gibbs Sampler

We now consider the case where we are interested in sampling from a distribution

$$\pi(\theta, \mathbf{x}) = \frac{\gamma(\theta, \mathbf{x})}{Z}$$

with $\gamma : \Theta \times \mathcal{X}^T \to \mathbb{R}^+$ assumed known pointwise and Z a possibly unknown normalising constant, independent of $\theta \in \Theta$. For many statistical models of practical interest $\mathbf{x}$ can be high dimensional (*e.g.* a vector of latent variables of the size of a large dataset) and the conditional distribution $\pi(\mathbf{x}|\theta)$ is non-standard. We have

$$\pi(\mathbf{x}|\theta) = \frac{\gamma(\theta, \mathbf{x})}{\gamma(\theta)}\ ,\ \pi(\theta) = \frac{\gamma(\theta)}{Z}$$

where $\gamma(\theta) = \int_{\mathcal{X}^T} \gamma(\theta, \mathbf{x})\, d\mathbf{x}$ is typically unknown. We propose here two strategies to sample from $\pi(\theta, \mathbf{x})$. The first strategy consists of using a particle approximation of an MH algorithm updating simultaneously θ and $\mathbf{x}$. The second strategy consists of using a particle approximation of the Gibbs sampler sampling from $\pi(\mathbf{x}|\theta)$ and $\pi(\theta|\mathbf{x})$.

Both strategies will rely on the use of an SMC algorithm in order to propose approximate samples from $\pi(\mathbf{x}|\theta)$ and approximately compute its normalising constant $\gamma(\theta)$. Hence we need to consider a family of bridging distributions $\{\pi_n(\mathbf{x}_n|\theta); n = 1, \ldots, T-1\}$ where

$$\pi_n(\mathbf{x}_n|\theta) = \frac{\gamma_n(\theta, \mathbf{x}_n)}{Z_n^\theta} \tag{10}$$

and $\pi_T(\mathbf{x}_T|\theta) = \pi(\mathbf{x}|\theta)$ and a family of proposal distributions $\{q_n^\theta(x_n|\mathbf{x}_{n-1})\}$ that defines sampling of $x_n \in \mathcal{X}$ conditional upon $\mathbf{x}_{n-1} \in \mathcal{X}^{n-1}$ and θ. Note that $Z_T^\theta = \gamma(\theta)$.

4.1 Particle Marginal MH Sampler

Consider a MH algorithm of target distribution $\pi(\theta, \mathbf{x})$. Assume for the time being that sampling from $\pi(\mathbf{x}|\theta)$ for any $\theta \in \Theta$ is feasible and recall the standard decomposition $\pi(\theta, \mathbf{x}) = \pi(\theta)\pi(\mathbf{x}|\theta)$. In such situations it is natural to suggest the following form of proposal distribution for an MH update

$$q\left(\left(\theta^*, \mathbf{x}^*\right)\middle|(\theta, \mathbf{x})\right) = q\left(\theta^*\middle|\theta\right)\pi\left(\mathbf{x}^*|\theta^*\right),$$

for which the proposed $\mathbf{x}^*$ is perfectly "adapted" to the proposed θ^*, and the only degree of freedom of the algorithm is $q(\theta^*|\theta)$, suggesting that the algorithm effectively targets the marginal distribution $\pi(\theta)$ as the MH acceptance ratio is given by

$$1 \wedge \frac{\pi(\theta^*, \mathbf{x}^*)}{\pi(\theta, \mathbf{x})} \frac{q((\theta, \mathbf{x})|(\theta^*, \mathbf{x}^*))}{q((\theta^*, \mathbf{x}^*)|(\theta, \mathbf{x}))} = 1 \wedge \frac{\gamma(\theta^*)}{\gamma(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}. \tag{11}$$

This algorithm is appealing since the difficult problem of sampling from $\pi(\theta, \mathbf{x})$ is reduced to that of sampling from $\pi(\theta)$ which is typically defined on a much smaller space and for which the design of proposal density is usually easier. Unfortunately, as discussed earlier, sampling exactly from $\pi(\mathbf{x}|\theta)$ is rarely feasible and $\gamma(\theta)$ is rarely known analytically, preventing the use of the above "idealized" Marginal MH (MMH) algorithm. It is natural to propose a Particle MMH (PMMH) algorithm which is a particle approximation of this "ideal" MMH algorithm using an SMC approximation of both samples from $\pi(\mathbf{x}|\theta)$ and of its normalising constant $\gamma(\theta)$. The PMMH algorithm proceeds as follows.

Particle Marginal Metropolis-Hastings Sampler

Initialization, $m = 0$

- Set randomly $\theta(0)$.
- Run an SMC algorithm targeting $\pi(\mathbf{x}|\theta(0))$, sample $\mathbf{X}(0) \sim \widehat{\pi}(\cdot|\theta(0))$ and compute $\widehat{\gamma}(\theta(0))$.

At iteration $m \geq 1$

- Sample $\theta^* \sim q(\cdot|\theta(m-1))$.
- Run an SMC algorithm targeting $\pi(\mathbf{x}|\theta^*)$, sample $\mathbf{X}^* \sim \widehat{\pi}(\cdot|\theta^*)$ and compute $\widehat{\gamma}(\theta^*)$.
- With probability
$$1 \wedge \frac{\widehat{\gamma}(\theta^*)}{\widehat{\gamma}(\theta(m-1))} \frac{q(\theta(m-1)|\theta^*)}{q(\theta^*|\theta(m-1))} \tag{12}$$
set $\theta(m) = \theta^*$, $\mathbf{X}(m) = \mathbf{X}^*$, $\widehat{\gamma}(\theta(m)) = \widehat{\gamma}(\theta^*)$, otherwise set $\theta(m) = \theta(m-1)$, $\mathbf{X}(m) = \mathbf{X}(m-1)$, $\widehat{\gamma}(\theta(m)) = \widehat{\gamma}(\theta(m-1))$.

Under very weak assumptions, the acceptance ratio (12) converges to (11) as $N \to \infty$. However more remarkably it can be established, using a reasoning very similar to that used for the PIMH algorithm, that this algorithm admits $\pi(\theta, \mathbf{x})$ as invariant distribution *for any* $N \geq 1$.

4.2 Particle Gibbs Sampler

A popular alternative to the MH algorithm to sample from $\pi(\theta, \mathbf{x})$ consists of using the Gibbs sampler. Numerous implementations rely on the fact that sampling from the conditional distribution $\pi(\theta|\mathbf{x})$ is feasible and thus the potentially tedious design of a proposal for θ can be bypassed. We will assume that this is the case here. Sampling from $\pi(\mathbf{x}|\theta)$ is typically impossible so we propose the following particle approximation.

Particle Gibbs Sampler

Initialization, $m = 0$

- Set randomly $\theta(0)$.
- Run an SMC algorithm targeting $\pi(\mathbf{x}|\theta(0))$, sample $\mathbf{X}(0) \sim \hat{\pi}(\cdot|\theta(0))$ and denote $\mathbf{B}(0)$ its ancestral lineage.

At iteration $m \geq 1$

- Sample $\theta(m) \sim \pi(\cdot|\mathbf{X}(m-1))$.
- Run a conditional SMC algorithm for $\theta(m)$ consistent with $\mathbf{X}(m-1), \mathbf{B}(m-1)$, sample $\mathbf{X}(m) \sim \hat{\pi}(\cdot|\theta(m))$ and denote $\mathbf{B}(m)$ its ancestral lineage.

Under very weak assumptions, the interesting feature of this algorithm is that it admits $\pi(\theta, \mathbf{x})$ as invariant distribution *for any* $N \geq 1$. Contrary to the PIMH and the

PMMH algorithms, it is however necessary to have $N \geq 2$ to ensure irreducibility of the Particle Gibbs (PG) sampler.

5 Extensions and Discussion

For ease of presentation, we have limited our description to one of the simplest SMC algorithms. However numerous more sophisticated algorithms have been proposed in the literature over the past fifteen years to improve on such basic schemes. In particular, in many applications of SMC, the resampling step is only performed when the accuracy of the estimator is poor. Practically, this is assessed by looking at the variability of the weights using the so-called Effective Sample Size (ESS) criterion [11, pp. 35–36] given at time n by

$$ESS = \left(\sum_{i=1}^{N} \left(W_n^i\right)^2 \right)^{-1}.$$

Its interpretation is that inference based on the N weighted samples is approximately equivalent to inference based on ESS perfect samples from the target. The ESS takes values between 1 and N and we resample only when it is below a threshold N_T otherwise we set $W_n^i \propto W_{n-1}^i \; w_n\left(\mathbf{X}_n^i\right)$. We refer to this procedure as dynamic resampling. All the strategies presented in the previous sections can also be applied in this context. The PIMH and PMMH can be implemented in the dynamic resampling context without any modification. However, the PG is more difficult to implement as the conditional SMC step requires simulating a set of $N-1$ particles not only consistent with a "frozen" path but also consistent with the resampling times of the SMC method used to generate the "frozen" path [1].

The PIMH algorithm presented in Section 3 is related to the Configurational-Biased Monte Carlo (CBMC) method which is a very popular method in molecular simulation used to sample long proteins [7]. Similarly to the PIMH sampler, the CBMC algorithm samples N particles and uses resampling steps. However, the resampling step used by the CBMC algorithm is such that a single particle survives, to which a new set of N offspring is then attached. Using our notation, this means that the CBMC algorithm corresponds to the case where $A_n^i = A_n^j$ for all $i, j = 1, \ldots, N$ and $A_n^1 \sim r(\cdot|\mathbf{W}_n)$ *i.e.* at any time n, all the children share the same and unique parent particle. The problem with this approach is that it is somewhat too greedy and that if a "wrong" decision is taken too prematurely then the proposal will be most likely rejected. It can be shown that the acceptance probability of the CBMC algorithm does not converge to 1 for $T > 1$ as $N \to \infty$ contrary to that of the PIMH algorithm. It has been more recently proposed in [3] to improve the CBMC algorithm by propagating forward several particles simultaneously in the spirit of the PIMH algorithm. However, contrary to us, the authors in [3] propose to kill or multiply particles by comparing their weights $w_n\left(\mathbf{X}_n^i\right)$ with respect to some prespecified lower and upper thresholds; *i.e.* the particles are not interacting and their

number is a random variable. In simulations, they found that the performance of this algorithm was very sensitive to the values of these thresholds. Our approach has the great advantage of bypassing the delicate choice of such thresholds. In statistics, a variation of the CBMC algorithm known as the Multiple-Try Method (MTM) has been introduced in the specific case where $T = 1$ in [10]. The key of our methodology is to build efficient proposals using sequential and interacting mechanisms for cases where $T \gg 1$: the sequential structure might be natural for some models (e.g. state-space models) but can also be induced in other scenarios in order to take advantage of the potential improvement brought in by the interacting mechanism [5]. In this respect, both methods do not apply to the same class of problems.

6 Application to Markov Jump Processes

We consider here a discretely observed stochastic kinetic Lotka-Volterra (LV) model. This model is often used to describe biochemical networks which exhibit auto-regulatory behaviour; see [12] for a thorough description of these models and their applications to system biology. Having access to noisy biochemical data, our objective is to perform Bayesian inference for the kinetic rate constants of the LV models

The LV model describes the evolution of two species X_t^1 (prey) and X_t^2 (predator) which are continuous-time non-negative integer-valued processes. In a small time interval $(t, t+dt]$, there are three possible transitions for the Markov Jump Process (MJP) $X_t = \left(X_t^1, X_t^2\right)$

$$\begin{aligned}
&\Pr\left(X_{t+dt}^1 = x_t^1 + 1, X_{t+dt}^2 = x_t^2 \middle| x_t^1, x_t^2\right) = \alpha\, x_t^1 dt + o(dt),\\
&\Pr\left(X_{t+dt}^1 = x_t^1 - 1, X_{t+dt}^2 = x_t^2 + 1 \middle| x_t^1, x_t^2\right) = \beta\, x_t^1\, x_t^2 dt + o(dt),\\
&\Pr\left(X_{t+dt}^1 = x_t^1, X_{t+dt}^2 = x_t^2 - 1 \middle| x_t^1, x_t^2\right) = \gamma\, x_t^2 dt + o(dt),
\end{aligned}$$

corresponding respectively to prey reproduction, predator reproduction and prey death, and predator death. We assume that we only have access to a noisy estimate of the number of preys $Y_n = X_{n\Delta}^1 + W_n$ with $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right)$. We are interested here in making inferences about the kinetic rate constants $\theta = (\alpha, \beta, \gamma)$ which are assumed to be a priori distributed as

$$\alpha \sim \mathcal{G}(1, 10), \quad \beta \sim \mathcal{G}(1, 0.25), \quad \gamma \sim \mathcal{G}(1, 7.5)$$

where $\mathcal{G}$ is the Gamma distribution [12, pp. 188–189]. The initial populations X_0^1, X_0^2 are assumed to be uniformly distributed in the interval $\{20, 21, \ldots, 80\}$.

We are interested in the posterior distribution $p(\mathbf{x}_T, \theta | \mathbf{y}_T)$ where $\mathbf{x}_T = \left(x_0, x_{2\Delta}, \ldots, x_{(T-1)\Delta}\right)$ and $\mathbf{y}_T = (y_0, y_1, \ldots, y_{T-1})$. This inference problem has already been addressed in [2]. In this paper, the authors propose a sophisticated reversible jump MCMC algorithm and a block updating strategy to sample from $p(\mathbf{x}_T, \theta | \mathbf{y}_T)$. The reversible jump MCMC is used to sample the continuous-time

process X_t (and its unknown number of transitions) in the interval $[0,(T-1)\Delta]$ whereas the block updating strategy attempts to update X_t for $t \in [(k-1)\Delta, k\Delta]$ using a sensible proposal. The authors note that both "algorithms suffered significant mixing problems". We use here the PMMH algorithm with $\pi_n(\mathbf{x}_n|\theta) = p(\mathbf{x}_n|\mathbf{y}_n,\theta)$. For the SMC proposals, we simply use the prior of X_t from which it is easy to sample using Gillespie's algorithm [12, pp. 188–189]. For the parameters, we use a Gaussian random walk proposal whose parameters were estimated in a short preliminary run. We could have alternatively used an adaptive MCMC strategy. We generated $T = 50$ observations by simulating the MJP using Gillespie's algorithm with parameters $\alpha = 2$, $\beta = 0.05$, $\gamma = 1.5$, $\Delta = 0.2$, $\sigma^2 = 4$ and $X_0^1 = X_0^2 = 40$; see Figure 2. We ran the algorithms for 100,000 iterations with a burn-in of 20,000. For $N = 1000$, the average acceptance rate of the PMMH sampler was 36%. The results are displayed in Figure 3.

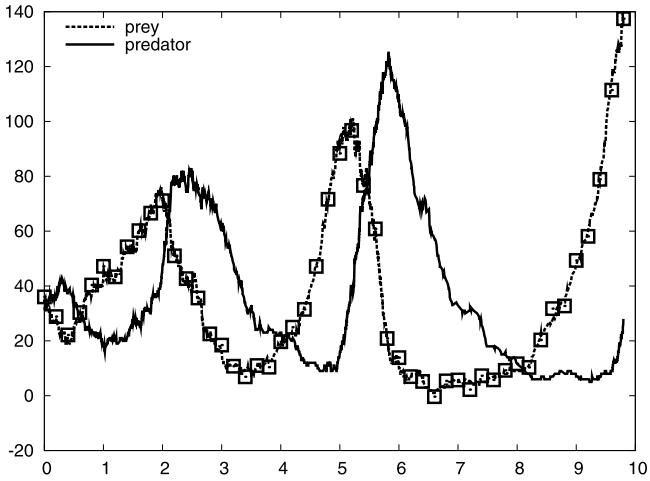


Fig. 2 Lotka-Volterra data. The number of prey X_t^1 and predators X_t^2 are shown in dotted and solid lines, respectively. The squares indicate the observations Y_n.

In Figure 4, we display the autocorrelation function (ACF) for the parameters (α, β) for various N. We can see that $N = 500$ is sufficient in this case for obtaining good performance and that increasing N does not improve the performance of the PMMH algorithm.

7 Conclusion

We have presented a new class of MCMC algorithms which rely on proposal distributions built using SMC methods. One of the major advantages of this approach is

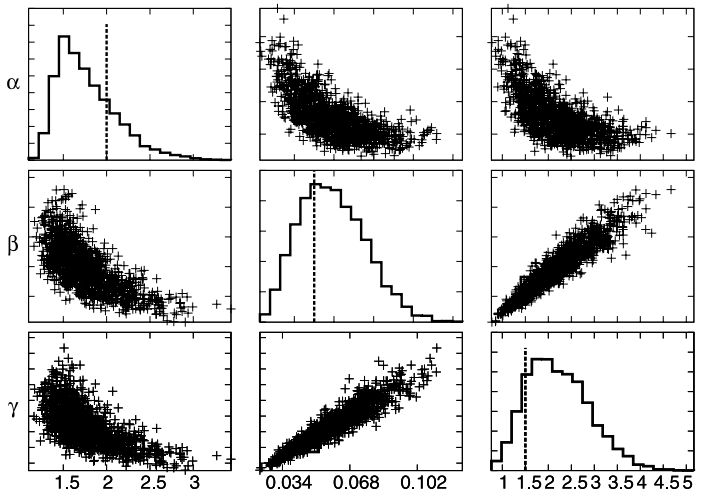
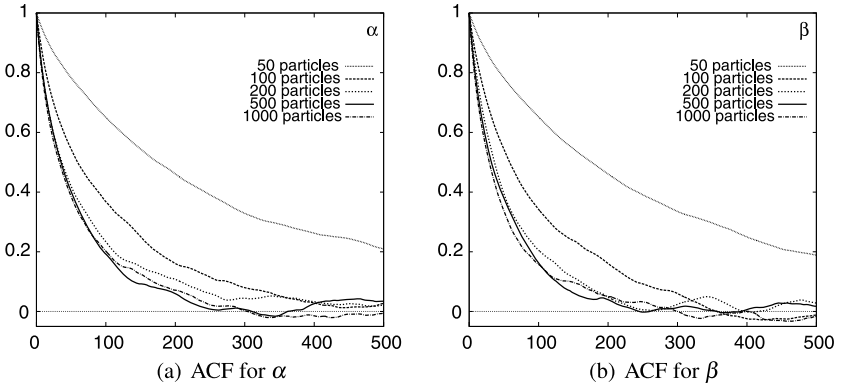


Fig. 3 Histograms and scatter plots of the sampled parameters. The straight lines on histograms represent the true values of the parameters.



(a) ACF for α (b) ACF for β

Fig. 4 Autocorrelation of the parameter α (left) and β (right) for the PMMH sampler for various numbers N of particles.

that it systematically builds high-dimensional proposal distributions whilst requiring the practitioner to design only low-dimensional proposal distributions. It offers the possibility to simultaneously update large vectors of dependent random variables. The lower the variance of the SMC estimates of the normalising constants, the better the performance of these algorithms. This strategy is computationally expensive but to some extent unavoidable and useful in complex scenarios for which standard proposals are likely to fail.

We believe that many problems in statistics where SMC methods have already been used could benefit from PMCMC methods. We have already successfully used

this methodology to fit complex continuous-time Lévy-driven stochastic volatility models and Dirichlet process mixtures [1]. Note that in the former case proposing samples from the prior distribution is the only known approach, which can lead to poor results when using standard MCMC algorithms. The CBMC method, to which our approach is related, is a very popular method in computational chemistry and physics which has been widely used for molecular and polymer simulation [7], and PMCMC algorithms might also prove useful in these areas.

References

1. Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society Series B, to appear.
2. Boys, R.J., Wilkinson, D.J., Kirkwood, T.B.L.: Bayesian inference for a discretely observed stochastic kinetic model. Statistics and Computing **18**, 125–135 (2008)
3. Combe, N., Vlugt, T.J.H., Wolde, P.R., Frenkel, D.: Dynamic pruned-enriched Rosenbluth method. Molecular Physics **101**, 1675–1682 (2003)
4. Del Moral, P.: Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications, Springer-Verlag, New York (2004)
5. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. Journal of the Royal Statistical Society Series B **68**, 411–436 (2006)
6. Doucet, A., Freitas, de J.F.G., Gordon, N.J (eds.): Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
7. Frenkel, D., Smit, B.: Understanding Molecular Simulation. 2nd edition, Academic Press, Orlando (2002)
8. Frenkel, D.: Waste-recycling Monte Carlo. In *Computer simulations in condensed matter: from materials to chemical biology*, Lecture Notes in Physics 703, Springer Berlin, 127–138 (2006)
9. Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. Journal of Computational and Graphical Statistics **5**, 1–25 (1996)
10. Liu, J.S., Liang, F., Wong, W.H.: The use of multiple-try method and local optimization in Metropolis sampling. Journal of the American Statistical Association **95**, 121–134 (2000)
11. Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer Verlag, New York (2001)
12. Wilkinson, D.J.: Stochastic Modelling for Systems Biology. Chapman & Hall/CRC Press, Boca Raton (2006)

Computational Complexity of Metropolis-Hastings Methods in High Dimensions

Alexandros Beskos and Andrew Stuart

Abstract This article contains an overview of the literature concerning the computational complexity of Metropolis-Hastings based MCMC methods for sampling probability measures on $\mathbb{R}^d$, when the dimension d is large. The material is structured in three parts addressing, in turn, the following questions: (i) what are sensible assumptions to make on the family of probability measures indexed by d? (ii) what is known concerning computational complexity for Metropolis-Hastings methods applied to these families? (iii) what remains open in this area?

1 Introduction

Metropolis-Hastings methods [19, 15] form a widely used class of MCMC methods [17, 21] for sampling from complex probability distributions. It is therefore of considerable interest to develop mathematical analyses which explain the structure inherent in these algorithms, especially structure which is pertinent to understanding the computational complexity of the algorithm. In this short article we overview the literature concerning the computational complexity of Metropolis-Hastings based MCMC methods for sampling probability measures on $\mathbb{R}^d$, when the dimension d is large. The presentation will be discursive: theorems will not be given, rather we will outline the essential ideas and give pointers to the relevant literature where the theorems are stated and proved.

Alexandros Beskos
Department of Statistical Science, University College of London, Torrington Place 1-19, London, WC1E 6BT, UK
e-mail: alex@stats.ucl.ac.uk

Andrew Stuart
Mathematics Institute, University of Warwick, Coventry, CV4 7AL, UK
url: http://www.maths.warwick.ac.uk/~stuart

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_4,

The article is organized around three sections. In Section 2 we address the question of how to make sensible assumptions on the family of probability measures indexed by dimension d. In Section 3 we overview what is known concerning computational complexity for Metropolis-Hastings methods applied to these families. Section 4 highlights open questions.

2 Structure of the Target

2.1 Product Target

A pioneering paper in the study of Metropolis methods in high dimensions is [10]; it studied the behaviour of random walk Metropolis methods when applied to target distributions with density

$$\pi_0^d(x) = \Pi_{i=1}^d f(x_i). \tag{1}$$

A similar study was undertaken in [22] for Langevin based Metropolis methods. Whilst these were amongst the first papers to pursue a rigorous study of Metropolis methods in high dimensions, a natural objection to this work is that families of target measures of the form (1) are restrictive from an applied perspective and, in any case, can be tackled by sampling a single one-dimensional target, because of the product structure. Partly in response to this objection, there have been several papers which generalize this work to target measures which retain the product structure inherent in (1), but are no longer i.i.d.. To be precise, we introduce standard deviations $\{\lambda_{i,d}\}_{i=1}^d$ so that

$$\pi_0^d(x) = \Pi_{i=1}^d \lambda_{i,d}^{-1} f(\lambda_{i,d}^{-1} x_i). \tag{2}$$

The papers [2, 23] consider this form of measure in the case where $\lambda_{i,d} = \lambda_i$ only, when the standard deviations do not change with dimension. Similar objections maybe raised concerning applicability of this work, namely that the product structure renders the problem far from most applications.

2.2 Beyond the Product Structure

In [5, 3] a different approach was taken, motivated by an infinite dimensional perspective arising in many applications. The target measure π is defined on a function space and is absolutely continuous with respect to some simpler reference measure π_0:

$$\frac{d\pi}{d\pi_0}(x) \propto \exp\bigl(-\Psi(x)\bigr). \tag{3}$$

For example π and π_0 might be the posterior and prior distributions respectively in the Bayesian formulation for an inverse problem on function space [7], or might

arise from a (possibly conditioned on observations, end-point constraints etc.) SDE via the Girsanov formula [12]. Often π_0 has a product structure when written in an appropriate basis.

Perhaps the simplest context in which to see such a product structure is to consider the case where π_0 is a Gaussian distribution $\mathcal{N}(0, \mathcal{C})$ on a Hilbert space $\mathcal{H}$. The eigenvalue problem

$$\mathcal{C}\phi_i = \lambda_i^2 \phi_i \tag{4}$$

provides a basis $\{\phi_i\}_{i=1}^{\infty}$ in which the operator $\mathcal{C}$ is diagonal and hence may be used to create a coordinate system in which there is a product structure. For the Gaussian measure to be well defined, $\mathcal{C}$ must be a trace-class operator which in turn implies that the λ_i's are square summable [9]. Any function $x \in \mathcal{H}$ may be written as

$$x = \sum_{i=1}^{\infty} x_i \phi_i \,. \tag{5}$$

If $x \sim \mathcal{N}(0, \mathcal{C})$ then the $\{x_i\}$ form a sequence of independent Gaussian random variables on $\mathbb{R}$ with $x_i \sim \mathcal{N}(0, \lambda_i^2)$. Thus we may write

$$x = \sum_{i=1}^{\infty} \xi_i \lambda_i \phi_i \tag{6}$$

where the ξ_i are an i.i.d. sequence of standard unit Gaussians. This shows that any Gaussian measure can be identified with a product measure on $\mathbb{R}^{\infty}$, an important idea which underlies the connections between simple product measures and quite complex measures π given by (3). The representation (6) is known as the *Karhunen-Loéve* expansion.

In the applications cited in [7, 12], the exponent Ψ is shown to satisfy useful properties which can be exploited in the design and analysis of sampling methods. In particular, Ψ can be shown to be bounded from below and above polynomially, and to be Lipschitz, on some Banach space X of full measure under π_0.

Consideration of some finite dimensional approximation of (3) will lead to a target measure π^d of the form

$$\frac{d\pi^d}{d\pi_0^d}(x) \propto \exp\bigl(-\Psi^d(x)\bigr) \tag{7}$$

where π_0^d is given by (2). Such measures are no longer of product form. However, the fact that they arise as approximations of measures on function space which are absolutely continuous with respect to a product measure leads to certain properties of Ψ^d being uniform in d. Furthermore, absolute continuity of π with respect to π_0 means, in rough terms, that if we expand a sample from π and one from π_0 in an orthonormal basis for $\mathcal{H}$, then the expansion coefficients are asymptotically (in the parameter indexing the expansion) identical: indeed absolute continuity sets strict conditions on the rate at which this asymptotic behaviour must occur (the Feldman-Hajek theorem [9]). Intuitively this allows for insight gleaned from the case of prod-

uct measures to be transferred to this more applicable context and explains the importance of the initial work in [10, 22, 23] concerning product measures.

These insights enable proof that, in some contexts, Ψ^d is bounded from below and above polynomially and is Lipschitz, with constants uniform in dimension d, provided appropriate norms are chosen to reflect the underlying infinite dimensional norm on X; see [3].

To give a little more detail on the nature of finite dimensional approximations of (3) we continue with the case where the reference measure is symmetric Gaussian and the Karhunen-Loéve expansion (5). If we denote by P^d the orthogonal projection of $\mathcal{H}$ onto the linear span

$$P^d\mathcal{H} := \text{span}\{\phi_1, \ldots, \phi_d\}$$

then we may define the measure π_{KL} on $\mathcal{H}$ by

$$\frac{d\pi_{\text{KL}}}{d\pi_0}(x) \propto \exp\bigl(-\Psi(P^d x)\bigr). \tag{8}$$

This measure is identical to π_0 on $\mathcal{H}\backslash P^d\mathcal{H}$, i.e. on the orthogonal complement of $P^d\mathcal{H}$.

On $P^d\mathcal{H}$, it provides a measure with the structure (7) and with the reference measure π_0^d given by (2) for $\lambda_{i,d} = \lambda_i$ given by (4); see [3] for details. Further approximation may be necessary, or desirable, as it may not be possible to evaluate Ψ, even on $P^d\mathcal{H}$. In the case of SDE (possibly conditioned on observations, end-point constraints etc.), and finite difference approximations (Euler-Maruyama method) one again obtains a measure of the form (7) with the reference measure π_0^d given by (2), but now the $\lambda_{i,d}$ depend on d and satisfy $\lambda_{i,d} \to \lambda_i$ as $d \to \infty$, for each fixed i; see [3] for details.

In summary, the early foundations of the study of the computational complexity of Metropolis methods in high dimension are based in the study of families of product measures (2); see [23] for an overview. More recently, this has given way to the study of wider classes of problems arising in applications with target measure of the form (3); see [5] for an overview. Whilst product measures might seem unduly restrictive, it turns out that a great deal of intuition can be transferred from this situation to the more applied problems, whenever the underlying reference measure in (3) has a product structure, a situation arising frequently in practice. With this in mind we now turn to the study of complexity.

3 Computational Complexity

We study Metropolis methods applied to the target measure π^d given by (7), and based on approximating (3). We assume that there are constants $0 < C^- \le C^+ < \infty$ and $\kappa \ge 0$ such that, for all indices i and dimensions d,

$$C^- \leq i^\kappa \lambda_{i,d} \leq C^+$$

giving bounds on the standard deviations.

Note that this setup subsumes the simpler cases (1) and (2) – by choosing $\Psi \equiv 0$ for both cases, and $\lambda_{i,d} \equiv 1$ for the first. In real applications a wide range of $\kappa > 0$ are encountered. For SDEs, possibly conditioned on observations, we have $\kappa = 1$. For Gaussian random field priors based on covariance operators which are fractional powers of the Laplacian in spatial dimension 2 (not to be confused with the dimension d of the approximating space) we require $\kappa > \frac{1}{2}$ to obtain almost surely continuous fields; more generally, increasing κ will correspond to random fields with increasing regularity, almost surely.

3.1 The Algorithms

The Metropolis methods we will consider are based on proposals on $\mathbb{R}^d$ with kernel $Q^d(x, dy)$ derived from the following expression in which the parameter $\delta > 0$ and the square root is applied to a positive-definite matrix:

$$\frac{y-x}{\delta} = \alpha \mathcal{A} \nabla \log \pi_0^d(x) + \sqrt{\frac{2\mathcal{A}}{\delta}}\xi, \quad \xi \sim \mathcal{N}(0, I). \tag{9}$$

In the case $\alpha = 0$ we refer to *random walk methods* and for $\alpha = 1$ to *Langevin methods*. We will take $\mathcal{A} = I$ or $\mathcal{A} = \mathcal{C}_d$ where we define the diagonal matrix $\mathcal{C}_d = \text{diag}\{\lambda_{1,d}^2, \cdots, \lambda_{d,d}^2\}$.

In the case where π_0^d is Gaussian we will also be interested in proposals of the form, for $\theta \in [0, 1]$,

$$\frac{y-x}{\delta} = \theta \mathcal{A} \nabla \log \pi_0^d(y) + (1-\theta)\mathcal{A} \nabla \log \pi_0^d(x) + \sqrt{\frac{2\mathcal{A}}{\delta}}\xi \tag{10}$$

for $\xi \sim \mathcal{N}(0, I)$. For both classes of proposal we will refer to δ as the *proposal variance*. All these proposals can be viewed as being derived from Euler-Maruyama-like discretizations of stochastic differential equations (SDEs) which are either π^d-invariant or π_0^d-invariant. Note, for instance, that proposals (9) for $\alpha = 1$ and (10) could be conceived as approximations (the first an explicit, the second an *implicit* one, see [16] for background on numerical approximations of SDEs) of the π_0-invariant SDE:

$$\frac{dx}{dt} = \mathcal{A} \nabla \log \pi_0^d(x) + \sqrt{2\mathcal{A}}\frac{dW}{dt}$$

driven by d-dimensional Brownian motion W. See [1, 14, 13, 11, 24] for more details on this interpretation. In this setting δ is the time-step in the Euler-Maruyama discretization.

The Metropolis-Hastings MCMC method [19, 15] creates a π^d invariant Markov chain $\{x^n\}$ as follows. Let $a(x, y)$ denote the acceptance probability, that is:

$$a(x,y) = 1 \wedge \frac{\pi^d(y)\, Q^d(y,x)}{\pi^d(x)\, Q^d(x,y)}.$$

Given x^n we make a proposal $y^n \sim Q^d(x^n, \cdot)$. With (independent) probability $a(x^n, y^n)$ we set $x^{n+1} = y^n$; otherwise we set $x^{n+1} = x^n$.

3.2 Complexity

Application of the Metropolis-Hastings accept-reject rule to proposals generated by the kernels Q described above gives rise to a π^d-invariant Markov chain $\{x^n\}_{n=0}^{\infty}$ on $\mathbb{R}^d$; we are interested in the computational complexity of running this chain to explore π^d. Let y^n denote the proposed state at step n, calculated from setting $x = x^n$ and $y = y^n$ in (9) or (10). The cost of each update is usually straightforward to compute, as a function of dimension, and thus the question of computational complexity boils down to understanding the number of steps required to explore π^d. Complexity of Metropolis methods on $\mathbb{R}^d$, for d large, is a difficult subject and the work we are about to overview does not provide the kind of complete analysis that is currently available for MCMC methods applied to some combinatorial problems. We will overview results related to optimizing choices of $\mathcal{A}$, α and θ (and δ, as a function of the dimension d) according to four (inter-twined) criteria, which we now describe.

Assume that $x^0 \sim \pi^d$ and that $y \mid x$ is given by one of the proposals (9) or (10) above. The four criteria are:

1. choose proposal parameters to maximize the mean square jump

$$\mathbb{E}\,\|x^{n+1} - x^n\|^2;$$

2. choose proposal parameters to maximize the mean time-step

$$\delta \times \mathbb{E}[\,a(x^n, y^n)\,];$$

3. choose proposal parameters to maximize the proposal variance subject to the constraint that the average acceptance probability is bounded away from zero, uniformly in dimension:

$$\liminf_{d\to\infty} \mathbb{E}[\,a(x^n, y^n)\,] > 0;$$

4. choose proposal parameters to maximize the proposal variance for which there exists a π-invariant diffusion limit for $z^d(t) := x^{\lfloor \delta t \rfloor}$, as $d \to \infty$.

In all four cases we use the rule of thumb that the number of steps $M(d)$ required to sample the invariant measure is given by the expression

$$M(d) \propto \delta^{-1}, \tag{11}$$

where the constant of proportionality is independent of dimension d, but the proposal variance δ depends on d. Later in this section we discuss the theory which justifies this decision. In the final section we will discuss the relations among these criteria and ideal criteria for convergence of Markov chains. For now we proceed on the assumption that the four criteria listed are useful in practice.

In [3] it is shown that, for proposals of the form (9), the optimality criteria 1., 2. and 3. all lead to the same conclusion (in an asymptotic sense, as $d \to \infty$) about optimal scaling of the proposal variance, hence to the same expression for $M(d)$. We summarise the specification of $M(d)$ for the different choices of α and $\mathcal{A}$ in Table 1.

Briefly, for $\alpha = 0$ and $\mathcal{A} = I$ we find that $M(d) = d^{2\kappa+1}$; for $\alpha = 0$ and $\mathcal{A} = \mathcal{C}_d$ we remove the κ-dependence, at the expense of inverting a covariance operator, and find that $M(d) = d$. Similar considerations apply for the case when $\alpha = 1$, only now the corresponding values are $d^{2\kappa+1/3}$ and $d^{1/3}$.

Table 1 Number of steps $M(d)$ to sample the invariant measure for each of the various MCMC algorithms derived via proposals (9) and (10).

Proposal (9), with $\alpha = 0$ and $\mathcal{A} = I$	$M(d) = d^{2\kappa+1}$
Proposal (9), with $\alpha = 0$ and $\mathcal{A} = \mathcal{C}_d$	$M(d) = d$
Proposal (9), with $\alpha = 1$ and $\mathcal{A} = I$	$M(d) = d^{2\kappa+1/3}$
Proposal (9), with $\alpha = 1$ and $\mathcal{A} = \mathcal{C}_d$	$M(d) = d^{1/3}$
Proposal (10), with $\theta = 1/2$ and π_0 Gaussian	$M(d) = \mathcal{O}(1)$

In [4] we show that, by choosing $\theta = \frac{1}{2}$ in proposal (10), it is possible to achieve $M(d) = \mathcal{O}(1)$ when the reference measure is Gaussian. In [4] numerical illustration is given only in the case of SDEs conditioned to start and end at specified points (diffusion bridges); however, [8] shows application of the same algorithmic idea to the problem of data assimilation for the Navier-Stokes equation.

3.3 A Special Result: Diffusion Limit

We now turn to the subject of diffusion limits. This will enable us to connect criterion 4. with criteria 1., 2. and 3., providing substantiation for the use of the heuristic (11) to measure the number of steps required to explore the target distribution in stationarity.

First we consider the simplest case where the target measure has the form (1). In [10] it was shown that, using (9) with $\alpha = 0$ and $\mathcal{A} = I$, and choosing the proposal variance δ to scale as $\delta = \ell^2 d^{-1}$, for some constant $\ell > 0$, leads to an average acceptance probability of order 1. Furthermore, with this choice of scaling, individual components of the resulting Markov chain converge to the solution of an SDE. Analytically, if the Markov chain is started in stationarity, and

$$z^d(t) := x_i^{\lfloor d \cdot t \rfloor}$$

denotes a continuous-time interpolant of the i^{th} component of the Markov chain, then $z^d \Rightarrow z$ as $d \to \infty$ in $C([0,T];\mathbb{R})$, where z solves the SDE

$$\frac{dz}{dt} = h(\ell)\,(\log f)'(z) + \sqrt{2h(\ell)}\,\frac{dW}{dt}. \tag{12}$$

Here $h(\ell)$ is often termed the *speed measure* and simply sets a time-scale for the SDE; it is identified explicitly in [10].

The diffusion limit leads to the interpretation that, started in stationarity, and applied to target measures of the form (1), the random walk Metropolis algorithm will require an order of δ^{-1} steps to explore the invariant measure; it also provides the justification for (11). Furthermore, the existence of a diffusion limit in this case shows that optimality criteria 1., 2., 3. and 4. all coincide. But the diffusion limit contains further information: it can be shown that the value of ℓ which maximizes $h(\ell)$, and therefore maximizes the speed of convergence of the limiting diffusion, leads to a universal acceptance probability, for random walk Metropolis algorithms applied to targets (1), of approximately 0.234. This means that, for the stated class of target distributions and algorithms, optimality can be obtained simply by tuning the algorithm to attain this desired acceptance probability.

These ideas have been generalized to other proposals, such as those based on (9) with $\alpha = 1$ and $\mathcal{A} = I$ in [22]. In this case, the choice $\delta = \ell^2 d^{-1/3}$ leads to a diffusion limit for

$$z^d(t) := x_i^{\lfloor d^{1/3} t \rfloor},$$

again implying that optimality criteria 1., 2., 3. and 4. all coincide. This leads to the interpretation that the algorithm will take time of order $d^{1/3}$ to explore the invariant measure. Furthermore, the choice of ℓ which maximizes the speed of the limiting SDE can be identified and results from an acceptance probability of approximately 0.574.

These papers of Roberts and coworkers concerning i.i.d. product measures are extended to non-i.i.d. products in [2, 23]. The impact of this work has been very high, in part because of the simple criteria for optimality when expressed in terms of the average acceptance probabilities 0.234 and 0.574, and in part because the existence of a diffusion limit provides an important conceptual understanding of the behaviour of MCMC methods in high dimensions. It is therefore natural to wonder if these optimal average acceptance probabilities arise also in the nonproduct case and if diffusion limits can then be found. We finish this section by discussing these two issues.

As mentioned above, [5, 3] study the question of optimal scaling of the proposal variance according to criteria 1., 2. and 3., for proposals (9), with $\alpha \in \{0, 1\}$ and $\mathcal{A} \in \{I, \mathcal{C}_d\}$, for non-product target measures of the form (7). There, it is shown that the mean square jumping distance (criterion 1.) and the mean time-step (criterion 2.) are both maximized by choosing the acceptance probabilities to be 0.234 (for

$\alpha = 0$) or 0.574 (for $\alpha = 1$) as in the i.i.d. product case (1). It is also shown that such a choice corresponds to optimizing with respect to criterion 3.

For target measures of the form (7), individual components of the Metropolis Markov chain cannot be expected to converge to a scalar SDE as happens for (1). However, it is natural to expect convergence of the entire Markov chain to an infinite dimensional continuous time stochastic process. In [13, 14] it is shown that the target measure π given by (3) is invariant for $\mathcal{H}$-valued SDEs (or stochastic PDEs, labelled SPDEs) with the form

$$\frac{dz}{ds} = -z - \mathcal{C}\nabla\Psi(z) + \sqrt{2\mathcal{C}}\frac{dW}{ds}, \tag{13}$$

where W is cylindrical Brownian motion (see [9] for a definition) in $\mathcal{H}$. In [18], we show that for proposal (9) with $\alpha = 0$ and $\mathcal{A} = \mathcal{C}_d$, started in stationarity, and $z^d(t) := x^{\lfloor dt \rfloor}$, $z^d \Rightarrow z$ as $d \to \infty$ in $C([0,T];\mathcal{H})$. This generalizes the work in [10, 22, 23] to the non-product set-up and shows that, in stationarity, the random walk Metropolis algorithm requires $\mathcal{O}(d)$ steps to explore the target distribution.

4 Open Questions

There are, of course, many open questions in the broad area of analyzing and constructing efficient MCMC methods in infinite dimensions. We mention a few interesting avenues in this general area, reflecting our personal tastes.

- *Rigorous complexity estimates.* Perhaps the primary open question concerning the work described herein is whether it can be used as the basis of the proof of a spectral gap for the Markov chain $\{x^n\}_{n=0}^{\infty}$, and determination of how the spectral gap scales with dimension d. A natural approach to this problem would be to use the theory highlighted in [20]. This theory provides a methodology for establishing convergence results of the following form: there are constants $C > 0, \lambda < 1$ and function $V : \mathbb{R}^d \mapsto [1, \infty)$ such that, for every $x^0 \in \mathbb{R}^d$, and every function g with $|g| \le V$,

$$|\mathbb{E}[g(x^n)] - \pi^d(g)| \le C\,V(x^0)\lambda^n.$$

  The distance of the constant λ from 1 can be used to estimate the spectral gap of the Markov chain. In typical proofs, the value of λ reflects both the mixing rate of the Markov chain in the center of the state space (in a *small set*) and the rate of return to the center of the state space. Since the results outlined in the previous section are concerned with behaviour in stationarity, it is likely that they reflect behaviour in the center of the state space. Thus, they do not contain information about travel times from outside the center of the state space; indeed this may lead to optimal scalings of the proposal variance which differ from those in the center of the state space, as shown in [6]. This relates to the burn-in time of the

algorithm, whereas the work described in Section 3 is primarily concerned with behaviour in stationarity.

- *[Number of steps]/[work per step] trade-off.* We have indicated above that the work in [4] demonstrates that, for measures of the form (7) with π_0^d Gaussian, it is possible to construct algorithms which explore the state space in a number of steps independent of dimension. These algorithms use proposals given by (10) with $\theta = \frac{1}{2}$. However, the algorithms require, at each step, either drawing a sample from the Gaussian reference measure $\mathcal{N}(0, \mathcal{C}_d)$ (in the case $\mathcal{A} = \mathcal{C}_d$), or inversion of the operator $I + \frac{\delta}{2}\mathcal{C}_d^{-1}$ (in the case $\mathcal{A} = I$). In contrast, proposal (9) with $\mathcal{A} = I$ is potentially considerably cheaper per step, but does require $\mathcal{O}(d^{2\kappa+1})$ steps to explore the invariant measure. There is, therefore, a trade-off between cost per step, and number of steps, for proposals based on (9) and (10). For probability measures arising from SDEs (possibly conditioned by observations, end-point constraints etc.) the linear algebra associated with proposals of the form (10) is (asymptotically in d) no more expensive than the cost of an update under (9) with $\mathcal{A} = I$, so it is clear that methods based on (10) with $\theta = \frac{1}{2}$ have a significant advantage; this advantage is illustrated numerically in [4]. However, for other classes of problems the trade-off remains to be studied. This poses an interesting avenue for study.
- *Non-Gaussian reference measures.* At the end of Subsection 3.2 we highlighted the fact that certain probability measures can be explored in number of steps *independent of the dimension* d, when started in stationarity. However this relies heavily on the assumption that the reference measure π_0^d in (7) is Gaussian. It remains an open question whether similar ideas to those in [4, 8] can be developed in the case of non-Gaussian reference measures. This is intimately related to the development of π-invariant SPDEs for measures of the form (3) [1, 13].
- *Other proposals.* The proposals we have discussed have been based on the discretization of π- or π_0-reversible SPDEs, leading to the Metropolis and Metropolis-Hastings variants of MCMC methods. However, there are many other proposals known to be effective in practice. In particular, Hybrid Monte Carlo (HMC) methods are widely used by practitioners. These methods double the size of the state space, from d to $2d$, by adding a momentum variable; they then use randomized Hamiltonian mechanics to explore the probability measure. Practical experience indicates that these methods can be very effective and theoretical studies of these proposals, of the type described in this review, would be of interest. More generally, there may be other proposals which yield improved complexity and this area is likely to be fruitful for further development.

Acknowledgements This article describes work overviewed in the second author's plenary lecture at MCQMC08, held in Montreal in July 2008. The authors are grateful to the organizers of this conference for the opportunity to present this material in lectured and written form. The authors are also indebted to all coauthors on papers listed in the bibliography, as well as to Jeff Rosenthal, for helpful discussions relating to the material in this paper. They are grateful to the European Research Council, the Engineering and Physical Sciences Research Council (UK) and the Office of Naval Research (USA) for funding all of their coauthored research referenced in the bibliography.

References

1. Apte, A., Hairer, M., Stuart, A., Voss, J.: Sampling the posterior: an approach to non-Gaussian data assimilation. Physica D **230**, 50–64 (2007)
2. Bédard, M.: Weak convergence of Metropolis algorithms for non-IID target distributions. Ann. Appl. Probab. **17**, 1222–1244 (2007)
3. Beskos, A., Roberts, G., Stuart, A.: Optimal scalings for local Metropolis-Hastings chains on non-product targets in high dimensions. Ann. Appl. Prob. (2009)
4. Beskos, A., Roberts, G., Stuart, A., Voss, J.: MCMC methods for diffusion bridges. Stochastics and Dynamics **8**(3), 319–350 (2008)
5. Beskos, A., Stuart, A.: MCMC methods for sampling function space. In: Proceedings of the International Congress of Industrial and Applied Mathematicians, (Zurich, 2007) (2009)
6. Christensen, O., Roberts, G., Rosenthal, J.: Optimal scaling for the transient phase of local Metropolis-Hastings algorithms. J. Roy. Stat. Soc. B **67**, 253–268 (2005)
7. Cotter, S.L., Dashti, M., Robinson, J.C., Stuart, A.M.: Bayesian inverse problems for functions and applications to fluid mechanics. Inverse Problems, to appear 2010.
8. Cotter, S.L., Dashti, M., Robinson, J.C., Stuart, A.M.: MCMC in function space and applications to fluid mechanics. In preparation
9. Da Prato, G., Zabczyk, J.: Stochastic Equations in Infinite Dimensions, *Encyclopedia of Mathematics and its Applications*, vol. 44. Cambridge University Press (1992)
10. Gelman, A., Gilks, W., Roberts, G.: Weak convergence and optimal scaling of random walk metropolis algorithms. Ann. Appl. Prob. **7**, 110–120 (1997)
11. Hairer, M., Stuart, A., Voss, J.: Sampling conditioned diffusions. In: Von Weizsäcker Proceedings, p. To appear. Cambridge University Press (2009)
12. Hairer, M., Stuart, A., Voss, J.: Signal processing problems on function space: Bayesian formulation, stochastic PDEs and effective MCMC methods. In: Handbook of Nonlinear Filtering. Oxford University Press, Editors D. Crisan and B. Rozovsky (2010)
13. Hairer, M., Stuart, A.M., Voss, J.: Analysis of SPDEs arising in path sampling. II. The nonlinear case. Ann. Appl. Probab. **17**(5-6), 1657–1706 (2007)
14. Hairer, M., Stuart, A.M., Voss, J., Wiberg, P.: Analysis of SPDEs arising in path sampling. I. The Gaussian case. Commun. Math. Sci. **3**(4), 587–603 (2005)
15. Hastings, W.: Monte Carlo sampling methods using markov chains and their applications. Biometrika **57**, 97–109 (1970)
16. Kloeden, P.E., Platen, E.: Numerical solution of stochastic differential equations, *Applications of Mathematics (New York)*, vol. 23. Springer-Verlag, Berlin (1992)
17. Liu, J.: Monte Carlo Strategies in Scientific Computing. Springer Texts in Statistics. Springer-Verlag (2001)
18. Mattingly, J., Pillai, N., Stuart, A.: SPDE limits of the Random Walk Metropolis algorithm in high dimensions (2009)
19. Metropolis, N., Rosenbluth, R., Teller, M., Teller, E.: Equations of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1092 (1953)
20. Meyn, S.P., Tweedie, R.L.: Markov Chains and Stochastic Stability. Communications and Control Engineering Series. Springer-Verlag, London (1993)
21. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer Texts in Statistics. Springer-Verlag (1999)
22. Roberts, G., Rosenthal, J.: Optimal scaling of discrete approximations to Langevin diffusions. J. Roy. Stat. Soc. B **60**, 255–268 (1998)
23. Roberts, G., Rosenthal, J.: Optimal scaling for various Metropolis-Hastings algorithms. Statistical Science **16**, 351–367 (2001)
24. Stuart, A., Voss, J., Wiberg, P.: Conditional path sampling of SDEs and the Langevin MCMC method. Commun. Math. Sci. **2**(4), 685–697 (2004)

On Quasi-Monte Carlo Rules Achieving Higher Order Convergence

Josef Dick

Abstract Quasi-Monte Carlo rules which can achieve arbitrarily high order of convergence have been introduced recently. The construction is based on digital nets and the analysis of the integration error uses Walsh functions. Various approaches have been used to show arbitrarily high convergence. In this paper we explain the ideas behind higher order quasi-Monte Carlo rules by leaving out most of the technical details and focusing on the main ideas.

1 Introduction

In this paper we study the approximation of multivariate integrals of the form

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

by quasi-Monte Carlo rules

$$\frac{1}{N}\sum_{h=0}^{N-1} f(\boldsymbol{x}_h).$$

Whereas the classical theory, see [12, 13, 14], focused on functions with bounded variation (or functions with square integrable partial mixed derivatives up to first order in each variable) or periodic functions, see [21], here we focus on functions which are not periodic and are smooth. The smoothness is a requirement if one wants to achieve convergence rates of order $N^{-\alpha}(\log N)^{c(s,\alpha)}$ with $\alpha > 1$ (here $c(s,\alpha)$ is a function which depends only on the dimension s and the smoothness α), as, for example, by the lower bound by Sharygin [20] we can in general at most get

Josef Dick
School of Mathematics and Statistics, The University of New South Wales, Sydney, NSW 2052, Australia
e-mail: josef.dick@unsw.edu.au

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_5,

$N^{-1}(\log N)^s$ for functions which have only bounded variation but no additional smoothness.

Let us assume our integrand $f : [0,1]^s \to \mathbb{R}$ is smooth. For $s = 1$ we consider the norm $\|\cdot\|_\alpha$ defined by

$$\|f\|_\alpha^2 = \left(\int_0^1 f(x)\,\mathrm{d}x\right)^2 + \cdots + \left(\int_0^1 f^{(\alpha-1)}(x)\,\mathrm{d}x\right)^2 + \int_0^1 |f^{(\alpha)}(x)|^2\,\mathrm{d}x,$$

and the corresponding inner product

$$\begin{aligned}\langle f, g\rangle_\alpha = &\int_0^1 f(x)\,\mathrm{d}x \int_0^1 g(x)\,\mathrm{d}x + \cdots + \int_0^1 f^{(\alpha-1)}(x)\,\mathrm{d}x \int_0^1 g^{(\alpha-1)}(x)\,\mathrm{d}x \\ &+ \int_0^1 f^{(\alpha)}(x) g^{(\alpha)}(x)\,\mathrm{d}x,\end{aligned}$$

where $f^{(\tau)}$ denotes the τth derivative of f for $1 \le \tau \le \alpha$ and where $f^{(0)} = f$. For simplicity we assume throughout the paper that $\alpha \ge 1$ is an integer, although the results can be generalized to include all real numbers $\alpha > 1$, see [4].

In dimensions $s > 1$ we consider the tensor product, but before we can do so we need some additional notation. Let $S = \{1, \ldots, s\}$, $\boldsymbol{x} = (x_1, \ldots, x_s)$ and for $u \subseteq S$ let $\boldsymbol{x}_u = (x_j)_{j \in u}$ denote the vector which only consists of the components x_j of $\boldsymbol{x}$ for which $j \in u$. Further, for $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_s) \in \{0, \ldots, \alpha\}^s$ let $|\boldsymbol{\tau}| = \tau_1 + \cdots + \tau_s$, $f^{(\boldsymbol{\tau})}(\boldsymbol{x}) = \frac{\partial^{|\boldsymbol{\tau}|} f}{\partial x_1^{\tau_1} \cdots \partial x_s^{\tau_s}}(\boldsymbol{x})$ and for $\boldsymbol{\tau} = \boldsymbol{0}$ let $f^{(\boldsymbol{0})}(\boldsymbol{x}) = f(\boldsymbol{x})$.

We define a norm $\|\cdot\|_\alpha$ by

$$\|f\|_\alpha^2 = \sum_{u \subseteq \{1,\ldots,s\}} \sum_{\boldsymbol{\tau}_{S\setminus u} \subseteq \{0,\ldots,\alpha-1\}^{s-|u|}} \int_{[0,1]^{|u|}} \left(\int_{[0,1]^{s-|u|}} f^{(\boldsymbol{\tau}_{S\setminus u}, \boldsymbol{\alpha}_u)}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}_{S\setminus u}\right)^2 \mathrm{d}\boldsymbol{x}_u,$$

where $\boldsymbol{\tau}_{S\setminus u} \in \{0, \ldots, \alpha-1\}^{s-|u|}$ shall denote a vector for which τ_j does not occur for $j \in u$ and otherwise has a value in $\{0, \ldots, \alpha-1\}$, and where $(\boldsymbol{\tau}_{S\setminus u}, \boldsymbol{\alpha}_u)$ is the vector for which the jth component is α for $j \in u$ and τ_j for $j \in S \setminus u$. The corresponding inner product is given by

$$\begin{aligned}\langle f, g\rangle_\alpha = &\sum_{u \subseteq \{1,\ldots,s\}} \sum_{\boldsymbol{\tau}_{S\setminus u} \subseteq \{0,\ldots,\alpha-1\}^{s-|u|}} \\ &\int_{[0,1]^{|u|}} \int_{[0,1]^{s-|u|}} f^{(\boldsymbol{\tau}_{S\setminus u}, \boldsymbol{\alpha}_u)}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}_{S\setminus u} \int_{[0,1]^{s-|u|}} g^{(\boldsymbol{\tau}_{S\setminus u}, \boldsymbol{\alpha}_u)}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}_{S\setminus u}\,\mathrm{d}\boldsymbol{x}_u.\end{aligned}$$

We say that a function f has smoothness α if $\|f\|_\alpha < \infty$. In the papers on higher order quasi-Monte Carlo rules various definitions of smoothness have been used, different from the one just introduced, for technical reasons: In [3] the author considered a Korobov space of periodic functions for which the kth Fourier coefficient is

of order $|k|^{-\alpha}$, i.e., functions in this space have $\|f\|_\alpha < \infty$, but are in addition also periodic. Non-periodic functions were first included in [4], but the results therein were based on a somewhat different norm purely for technical reasons. The results in [4] also include fractional smoothness, i.e., therein $\alpha > 1$ is allowed to be any real number. The function space considered in [4] was based on Walsh series, and it was shown that this space includes all smooth functions, i.e., functions with smoothness $\alpha > 1$. Later, it was shown in [5] that functions f with $\|f\|_\alpha < \infty$ are contained in this Walsh space. A function space with norm as above was finally considered in [1]. Periodic functions on the other hand where considered in [3].

First results on convergence rates faster than $N^{-1}(\log N)^s$ for non-periodic functions using quasi-Monte Carlo rules were obtained in (in chronological order):

- [18, 19], where a convergence rate of $N^{-3/2}(\log N)^{(s-1)/2}$ was shown using scrambled digital nets;
- [10], where a convergence rate of $N^{-2+\delta}$, $\delta > 0$, was shown using randomly shifted lattice rules and the baker's transformation;
- [2], where a convergence of $N^{-2+\delta}$, $\delta > 0$, was shown, also using randomly digitally shifted digital nets and the baker's transformation.

For periodic functions on the other hand, the following results are known:

- it has long been known that we can achieve $N^{-\alpha}(\log N)^{\alpha s}$ by using lattice rules, as explained in Subsection 2.3, although for arbitrary N and s we do not have a priori constructions.
- From Niederreiter [11] it is known that we can achieve $N^{-\alpha+\varepsilon}$ using explicitly constructed Kronecker sequences, a simple example being $(\{n\sqrt{p_1}\}, \ldots, \{n\sqrt{p_s}\})$, $n = 0, 1, \ldots$, with $p_1, \ldots, p_s$ being distinct prime numbers.

The focus in this work is on generalized nets and sequences which can achieve convergence rates of $N^{-\alpha}(\log N)^{\alpha s}$ for smooth non-periodic functions, where $\alpha \geq 1$ can be an arbitrarily large integer (where we assume that the integrand has smoothness at least α).

There are two main hurdles to arrive at quasi-Monte Carlo rules which achieve the optimal order of convergence for functions with smoothness α, where $\alpha \in \mathbb{N}$ can be arbitrarily high.

The first main step towards proving higher order convergence of the integration error (i.e., convergence of $N^{-\alpha}(\log N)^{\alpha s}$ for any $\alpha \geq 1$) is a result concerning the decay of the Walsh coefficients. We will explain the details in Section 4.1. It requires a result on the decay of the Walsh coefficients of smooth functions, first shown explicitly in [4], see also [5].

The second main step is to construct point sets explicitly which can be used in a quasi-Monte Carlo rule. The construction scheme uses digital nets and a quality criterion on the generating matrices of such point sets can be obtained using the result in the first step. The details of this will be explained in Section 4.3.

It is useful to first look at how lattice rules can achieve arbitrarily high order of convergence for smooth periodic functions, as part of the theory for non-periodic functions is similar, albeit much more technical.

2 Higher Order Convergence for Smooth Periodic Functions Using Lattice Rules

In this section we consider numerical integration using lattice rules, which will give us a basic understanding of how the theory on numerical integration works, see also [21] for a particularly nice introduction to this theory.

2.1 Lattice Rules

First let us introduce lattice rules. Assume we want a quasi-Monte Carlo rule with N points. For a real number x let $\{x\} = x - \lfloor x \rfloor$ denote the fractional part of x. Then choose a vector $\boldsymbol{g} \in \{1, \dots, N-1\}^s$ and use the quadrature rule

$$\frac{1}{N} \sum_{\ell=0}^{N-1} f\left(\left\{ \frac{\ell \boldsymbol{g}}{N} \right\} \right).$$

This quadrature rule is called lattice rule.

Such rules work well with periodic functions. Before we can introduce the error analysis we need some understanding of the connection between smooth periodic functions and the decay of the Fourier coefficients.

2.2 Decay of the Fourier Coefficients of Smooth Periodic Functions

Let now $f : [0,1]^s \to \mathbb{R}$ be a smooth periodic function. I.e., for any $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^s$ (here $\{0,1\}$ is the set consisting of the two elements 0 and 1), and any $\boldsymbol{\tau} \in \{0, \dots, \alpha - 1\}^s$ we have $f^{(\boldsymbol{\tau})}(\boldsymbol{x}) = f^{(\boldsymbol{\tau})}(\boldsymbol{y})$. Assume f has square integrable partial mixed derivatives up to order α in each variable, then $\|f\|_\alpha < \infty$. We assume in the following that $\alpha \geq 1$. Let the Fourier series of f be given by

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^s} \widehat{f}(\boldsymbol{k}) \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}},$$

where $\boldsymbol{k} \cdot \boldsymbol{x} = k_1 x_1 + \cdots + k_s x_s$ and $\widehat{f}(\boldsymbol{k})$ is the $\boldsymbol{k}$th Fourier coefficient $\widehat{f}(\boldsymbol{k}) = \int_{[0,1]^s} f(\boldsymbol{x}) \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}} \, \mathrm{d}\boldsymbol{x}$.

Consider the case $s = 1$ for a moment: Then $f(x) = \sum_{k=-\infty}^{\infty} \widehat{f}(k) \mathrm{e}^{2\pi \mathrm{i} k x}$. Assume that f is differentiable and let $\widehat{f'}(k)$ denote the kth Fourier coefficient of f', i.e., $\widehat{f'}(k) = \int_0^1 f'(x) \mathrm{e}^{-2\pi \mathrm{i} k x} \, \mathrm{d}x$. Then by differentiating the Fourier series for f we obtain $2\pi \mathrm{i} k \widehat{f}(k) = \widehat{f'}(k)$, or, for $k \neq 0$, $\widehat{f}(k) = \widehat{f'}(k)/(2\pi \mathrm{i} k)$. Another way of obtaining the last formula for $k \neq 0$ is by using integration by parts:

$$\begin{aligned}\widehat{f}(k) &= \int_0^1 f(x)\mathrm{e}^{-2\pi\mathrm{i}kx}\,\mathrm{d}x \\ &= -\frac{1}{2\pi\mathrm{i}k}[f(x)\mathrm{e}^{-2\pi\mathrm{i}kx}]_{x=0}^1 + \frac{1}{2\pi\mathrm{i}k}\int_0^1 f'(x)\mathrm{e}^{-2\pi\mathrm{i}kx}\,\mathrm{d}x \\ &= \frac{\widehat{f'}(k)}{2\pi\mathrm{i}k},\end{aligned}$$

as $f(0) = f(1)$. If, say, $\int_0^1 |f'(x)|\,\mathrm{d}x < \infty$, then the equation above implies that for $k \neq 0$ we have

$$|\widehat{f}(k)| = \frac{1}{2\pi|k|}\left|\int_0^1 f'(x)\mathrm{e}^{-2\pi\mathrm{i}kx}\,\mathrm{d}x\right| \leq \frac{1}{2\pi|k|}\int_0^1 |f'(x)|\,\mathrm{d}x.$$

Repeated use of the argument above shows that if f is α times differentiable, then $|\widehat{f}(k)| = \mathcal{O}(|k|^{-\alpha})$.

The case $s > 1$ works similarly. We have $|\widehat{f}(\boldsymbol{k})| = \mathcal{O}(|\bar{k}_1 \cdots \bar{k}_s|^{-\alpha})$, where $\bar{k} = k$ for $k \neq 0$ and 1 otherwise. The constant in the bound on the Fourier coefficient depends on the norm of the function, indeed, one can show that $|\widehat{f}(\boldsymbol{k})| \leq C_{\alpha,s}|\bar{k}_1 \cdots \bar{k}_s|^{-\alpha}\|f\|_\alpha$ with some constant $C_{\alpha,s}$ independent of $\boldsymbol{k}$ and f.

2.3 Numerical Integration

The following property is useful in analyzing the integration error of Fourier series when one approximates the integral with a lattice rule (we assume N is a prime number):

$$\frac{1}{N}\sum_{\ell=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}\ell\boldsymbol{k}\cdot\boldsymbol{g}/N} = \begin{cases} 1 & \text{if } \boldsymbol{k}\cdot\boldsymbol{g} \equiv 0 \pmod N, \\ 0 & \text{otherwise.}\end{cases}$$

The set of all $\boldsymbol{k} \in \mathbb{Z}^s$ for which the above sum is 1 is called the dual lattice, i.e., we have

$$\mathcal{L} = \{\boldsymbol{k} \in \mathbb{Z}^s : \boldsymbol{k}\cdot\boldsymbol{g} \equiv 0 \pmod N\}.$$

Using the Fourier series expansion of the function f we obtain

$$\begin{aligned}\left|\int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{N}\sum_{\ell=0}^{N-1} f(\{\ell\boldsymbol{g}/N\})\right| &= \left|\widehat{f}(\boldsymbol{0}) - \sum_{\boldsymbol{k}\in\mathbb{Z}^s}\widehat{f}(\boldsymbol{k})\frac{1}{N}\sum_{\ell=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}\ell\boldsymbol{k}\cdot\boldsymbol{g}/N}\right| \\ &= \left|\sum_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}}\widehat{f}(\boldsymbol{k})\right| \\ &\leq \sum_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}}|\widehat{f}(\boldsymbol{k})|.\end{aligned}$$

We can use the bound on the Fourier coefficients from the previous section to obtain

$$\left| \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{N}\sum_{\ell=0}^{N-1} f(\{\ell \boldsymbol{g}/N\}) \right| \le C_{\alpha,s} \|f\|_\alpha \sum_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}} |\bar{k}_1 \cdots \bar{k}_s|^{-\alpha}.$$

The last sum tells us to choose $\boldsymbol{g}$ such that only those $\boldsymbol{k} \in \mathbb{Z}^s \setminus \{\boldsymbol{0}\}$ should satisfy $\boldsymbol{k}\cdot\boldsymbol{g} \equiv 0 \pmod N$ for which $|\bar{k}_1 \cdots \bar{k}_s|^{-\alpha}$ is small. Indeed, one can show that there are $\boldsymbol{g}$ such that $\sum_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}} |\bar{k}_1 \cdots \bar{k}_s|^{-\alpha} = \mathcal{O}(N^{-\alpha+\delta})$ for any $\delta > 0$.

One way to show the last claim is the following (we do not give the details here, just an outline, see [12, Chapter 5] for more information): Let

$$\rho = \min_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}} |\bar{k}_1 \cdots \bar{k}_s|. \tag{1}$$

We call ρ the figure of merit. Then the largest term in $\sum_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}} |\bar{k}_1 \cdots \bar{k}_s|^{-\alpha}$ is given by $\rho^{-\alpha}$. One can now show that the sum $\sum_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}} |\bar{k}_1 \cdots \bar{k}_s|^{-\alpha}$ is dominated by its largest term. Indeed, there are bounds

$$\rho^{-\alpha} \le \sum_{\boldsymbol{k}\in\mathcal{L}\setminus\{\boldsymbol{0}\}} |\bar{k}_1 \cdots \bar{k}_s|^{-\alpha} \le C'_{\alpha,s}\rho^{-\alpha}(\log\rho)^{\alpha s}, \tag{2}$$

see [12, Chapter 5]. Further there is a result which states that there exists a $\boldsymbol{g} \in \{1,\dots,N-1\}^s$ such that $\rho > \frac{(s-1)!N}{(2\log N)^{s-1}}$. Together with (2) this yields the result.

In the following we use a similar approach for numerical integration using digital nets. Instead of considering Fourier series, we now consider Walsh series and lattice rules are replaced by quasi-Monte Carlo rules based on digital nets. Before we can explain this theory we introduce the necessary concepts in the next section.

3 Preliminaries

In the following we introduce the digital construction scheme and Walsh functions. For simplicity we only consider the case where the base b is a prime.

3.1 The Digital Construction Scheme

The construction of the point set used here is based on the concept of digital nets introduced by Niederreiter, see [12].

Definition 1. Let b be a prime and let $n, m, s \ge 1$ be integers. Let $C_1, \dots, C_s$ be $n \times m$ matrices over the finite field $\mathbb{F}_b$ of order b. Now we construct b^m points in $[0,1)^s$: for $0 \le h \le b^m - 1$ let $h = h_0 + h_1 b + \cdots + h_{m-1}b^{m-1}$ be the b-adic

expansion of h. Identify h with the vector $\mathbf{h} = (h_0, \ldots, h_{m-1})^\top \in \mathbb{F}_b^m$, where $\top$ means the transpose of the vector (note that we write $\mathbf{h}$ for vectors in the finite field $\mathbb{F}_b^m$ and $\boldsymbol{h}$ for vectors of integers or real numbers). For $1 \le j \le s$ multiply the matrix C_j by $\mathbf{h}$, i.e.,

$$C_j \mathbf{h} =: (y_{j,1}(h), \ldots, y_{j,n}(h))^\top \in \mathbb{F}_b^n,$$

and set

$$x_{h,j} := \frac{y_{j,1}(h)}{b} + \cdots + \frac{y_{j,n}(h)}{b^n}.$$

The point set $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{b^m-1}\}$ is called a digital net (over $\mathbb{F}_b$) (with generating matrices $C_1, \ldots, C_s$).

For $n, m = \infty$ we obtain a sequence $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\}$, which is called a digital sequence (over $\mathbb{F}_b$) (with generating matrices $C_1, \ldots, C_s$).

Niederreiter's concept of a digital (t, m, s)-net and a digital (t, s)-sequence will appear as a special case in the subsequent section. Further, the digital nets considered below all satisfy $n \ge m$.

For a digital net with generating matrices $C_1, \ldots, C_s$ let $\mathcal{D} = \mathcal{D}(C_1, \ldots, C_s)$ be the dual net given by

$$\mathcal{D} = \{\boldsymbol{k} \in \mathbb{N}_0^s : C_1^\top \mathbf{k}_1 + \cdots + C_s^\top \mathbf{k}_s = \mathbf{0}\},$$

where for $\boldsymbol{k} = (k_1, \ldots, k_s)$ with $k_j = \kappa_{j,0} + \kappa_{j,1} b + \cdots$ and $\kappa_{j,i} \in \{0, \ldots, b-1\}$ we define $\mathbf{k}_j = (\kappa_{j,0}, \ldots, \kappa_{j,n-1})^\top$. The definition of the dual net is related to the definition of the dual space as defined in [15].

3.2 Walsh Functions

Let the real number $x \in [0, 1)$ have base b representation $x = \frac{x_1}{b} + \frac{x_2}{b^2} + \cdots$, with $0 \le x_i < b$ and where infinitely many x_i are different from $b-1$. For $k \in \mathbb{N}$, $k = \kappa_1 b^{a_1 - 1} + \cdots + \kappa_\nu b^{a_\nu - 1}$, $a_1 > \cdots > a_\nu > 0$ and $0 < \kappa_1, \ldots, \kappa_\nu < b$, we define the kth Walsh function by

$$\mathrm{wal}_k(x) = \omega_b^{\kappa_1 x_{a_1} + \cdots + \kappa_\nu x_{a_\nu}},$$

where $\omega_b = \mathrm{e}^{2\pi \mathrm{i}/b}$. For $k = 0$ we set $\mathrm{wal}_0(x) = 1$.

For a function $f : [0, 1] \to \mathbb{R}$ we define the kth Walsh coefficient of f by

$$\widehat{f}(k) = \int_0^1 f(x) \overline{\mathrm{wal}_k(x)} \, \mathrm{d}x$$

and we can form the Walsh series

$$f(x) \sim \sum_{k=0}^{\infty} \widehat{f}(k) \mathrm{wal}_k(x).$$

Note that throughout the paper Walsh functions and digital nets are defined using the same prime number b.

4 Higher Order Convergence of Smooth Functions Using Generalized Digital Nets

In this section we present the ideas behind higher order quasi-Monte Carlo rules based on generalized digital nets.

4.1 Decay of the Walsh Coefficients of Smooth Functions

We will focus mainly on $s = 1$ in this section, the case $s > 1$ is a natural extension as we consider tensor product spaces of functions. We do not give all the details, but provide an heuristic approach. The simplest exposition of the result presented in this subsection which contains all the details may be found in [5].

We now prove a bound on the Walsh coefficients of smooth functions which are not necessarily periodic (if the functions are periodic, then slightly stronger results can be obtained [5]). Note that we cannot differentiate the Walsh series of a function f, since the Walsh functions are piecewise constant and have therefore jumps. But we can use the second approach based on integration by parts, as was done for Fourier series above. Let $J_k(x) = \int_0^x \overline{\mathrm{wal}_k(t)}\,\mathrm{d}t$, then

$$\begin{aligned}\widehat{f}_{\mathrm{wal}}(k) &= \int_0^1 f(x)\overline{\mathrm{wal}_k(x)}\,\mathrm{d}x \\ &= [f(x)J_k(x)]_{x=0}^1 - \int_0^1 f'(x)J_k(x)\,\mathrm{d}x \\ &= -\int_0^1 f'(x)J_k(x)\,\mathrm{d}x, \end{aligned} \tag{3}$$

as $\int_0^1 \overline{\mathrm{wal}_k(x)}\,\mathrm{d}x = 0$.

As for Fourier series, we would now like to relate the Walsh coefficient $\widehat{f}_{\mathrm{wal}}(k)$ to some Walsh coefficient of f'. For Fourier series this happened naturally, but here we obtain the function J_k. The way to proceed now is to obtain the Walsh series expansion of J_k, which will allow us to relate the kth Walsh coefficient of f to some Walsh coefficients of f'.

We need the following lemma which was first shown in [9] and appeared in many other papers (see for example [4] for a more general version). The following notation will be used throughout the paper: $k' = k - \kappa_1 b^{a_1-1}$, and hence $0 \le k' < b^{a_1-1}$.

Lemma 1. *For $k \in \mathbb{N}$ let $J_k(x) = \int_0^x \overline{\mathrm{wal}_k(t)}\,\mathrm{d}t$. Then*

$$J_k(x) = b^{-a_1}\Bigg((1-\omega_b^{-\kappa_1})^{-1}\overline{\mathrm{wal}_{k'}(x)} + (1/2+(\omega_b^{-\kappa_1}-1)^{-1})\overline{\mathrm{wal}_k(x)} + \sum_{c=1}^{\infty}\sum_{\vartheta=1}^{b-1} b^{-c}(\omega_b^{\vartheta}-1)^{-1}\overline{\mathrm{wal}_{\vartheta b^{a_1+c-1}+k}(x)}\Bigg).$$

For $k=0$, i.e., $J_0(x) = \int_0^x 1\,\mathrm{d}t = x$, we have

$$J_0(x) = 1/2 + \sum_{c=1}^{\infty}\sum_{\vartheta=1}^{b-1} b^{-c}(\omega_b^{\vartheta}-1)^{-1}\overline{\mathrm{wal}_{\vartheta b^{c-1}}(x)}. \tag{4}$$

We also need the following elementary lemma.

Lemma 2. *For any $0 < \kappa < b$ we have*

$$|1-\omega_b^{-\kappa}|^{-1} \le \frac{1}{2\sin\frac{\pi}{b}} \quad \text{and} \quad |1/2+(\omega_b^{-\kappa}-1)^{-1}| \le \frac{1}{2\sin\frac{\pi}{b}}.$$

Let $k \in \mathbb{N}$ with $k = \kappa_1 b^{a_1-1} + \cdots + \kappa_\nu b^{a_\nu-1}$, where $0 < \kappa_1, \ldots, \kappa_\nu < b$ and $a_1 > \cdots > a_\nu > 0$. Further let $k^{(1)} = \kappa_2 b^{a_2-1} + \cdots + \kappa_\nu b^{a_\nu-1}$, $k^{(2)} = \kappa_3 b^{a_3-1} + \cdots + \kappa_\nu b^{a_\nu-1}$, and $k^{(\tau)} = \kappa_{\tau+1} b^{a_{\tau+1}-1} + \cdots + \kappa_\nu b^{a_\nu-1}$ for $0 \le \tau < \nu$ and $k^{(\nu)} = 0$. It is also convenient to define the following function:

$$\mu_\alpha(k) = \begin{cases} a_1 + \cdots + a_{\min(\alpha,\nu)} & \text{for } k > 0, \\ 0 & \text{for } k = 0. \end{cases}$$

Substituting the Walsh series for J_k in (3) we obtain approximately

$$\begin{aligned}\widehat{f}_{\mathrm{wal}}(k) &\approx -b^{-a_1}(1-\omega_b^{-\kappa_1})^{-1}\int_0^1 f'(x)\overline{\mathrm{wal}_{k'}(x)}\,\mathrm{d}x \\ &= -b^{-a_1}(1-\omega_b^{-\kappa_1})^{-1}\widehat{f'}_{\mathrm{wal}}(k^{(1)}).\end{aligned}$$

In actuality we obtain an infinite sum on the right hand side, but the main term is the first one, the remaining terms can be dealt with, see [5] for the details.

We can repeat the last step τ times until either $f^{(\tau)}$ is not differentiable anymore, or $k^{(\tau)} = 0$, that is, we can repeat it $\min(\alpha,\nu)$ times. Hence

$$\begin{aligned}\widehat{f}_{\mathrm{wal}}(k) &\approx b^{-a_1}(\omega_b^{-\kappa_1}-1)^{-1}\widehat{f'}_{\mathrm{wal}}(k^{(1)}) \\ &\approx b^{-a_1-a_2}\prod_{i=1}^{2}(\omega_b^{-\kappa_i}-1)^{-1}\widehat{f''}_{\mathrm{wal}}(k^{(2)}) \\ &\;\;\vdots \\ &\approx b^{-a_1-\cdots-a_{\min(\alpha,\nu)}}\prod_{i=1}^{\min(\alpha,\nu)}(\omega_b^{-\kappa_i}-1)^{-1}\widehat{f}^{(\min(\alpha,\nu))}_{\mathrm{wal}}(k^{(\min(\alpha,\nu))}).\end{aligned}$$

Taking the absolute value and using some estimation we obtain

$$
\begin{aligned}
|\widehat{f}_{\mathrm{wal}}(k)| &\lessapprox b^{-a_1-\cdots-a_{\min(\alpha,\nu)}} \prod_{i=1}^{\min(\alpha,\nu)} |\omega_b^{-\kappa_i} - 1|^{-1} |\widehat{f}_{\mathrm{wal}}^{(\min(\alpha,\nu))}(k^{(\min(\alpha,\nu))})| \\
&\leq \frac{b^{-\mu_\alpha(k)}}{(2\sin\pi/b)^{\min(\alpha,\nu)}} |\widehat{f}_{\mathrm{wal}}^{(\min(\alpha,\nu))}(k^{(\min(\alpha,\nu))})| \\
&\leq \frac{b^{-\mu_\alpha(k)}}{(2\sin\pi/b)^{\min(\alpha,\nu)}} \int_0^1 \left| f^{(\min(\alpha,\nu))}(x) \right| \mathrm{d}x,
\end{aligned}
$$

where we used

$$
\begin{aligned}
|\widehat{f}_{\mathrm{wal}}^{(\min(\alpha,\nu))}(k^{(\min(\alpha,\nu))})| &= \left| \int_0^1 f^{(\min(\alpha,\nu))}(x) \overline{\mathrm{wal}_{k^{(\min(\alpha,\nu))}}(x)} \,\mathrm{d}x \right| \\
&\leq \int_0^1 \left| f^{(\min(\alpha,\nu))}(x) \right| |\overline{\mathrm{wal}_{k^{(\min(\alpha,\nu))}}(x)}| \,\mathrm{d}x \\
&= \int_0^1 \left| f^{(\min(\alpha,\nu))}(x) \right| \mathrm{d}x.
\end{aligned}
$$

Thus if f is α times differentiable, we obtain

$$
|\widehat{f}_{\mathrm{wal}}(k)| \lessapprox C_f b^{-\mu_\alpha(k)}.
$$

By some modification of the above approach, see [5], it can be shown that the constant C_f, which depends on f, can be replaced by a constant which depends only on α and b (but not on f) and the norm of f, i.e., we have

$$
|\widehat{f}_{\mathrm{wal}}(k)| \lessapprox C_{\alpha,b} \|f\|_\alpha b^{-\mu_\alpha(k)}.
$$

The same holds for dimensions $s > 1$, see [1, 4, 5], where the constant additionally depends on the dimension s

$$
|\widehat{f}(\boldsymbol{k})| \lessapprox C_{\alpha,b,s} \|f\|_\alpha b^{-\mu_\alpha(\boldsymbol{k})},
$$

where $\mu_\alpha(\boldsymbol{k}) = \mu_\alpha(k_1) + \cdots + \mu_\alpha(k_s)$ for $\boldsymbol{k} = (k_1, \ldots, k_s)$. For some values of b, this constant $C_{\alpha,b,s}$ goes to 0 exponentially as s increases, see [1, 5].

Thus we have now achieved an analogous result to the decay of the Fourier coefficients of smooth functions and we can now begin to investigate numerical integration.

4.2 Numerical Integration

This section is largely similar to Section 2.3. Note that, as opposed to Section 2.3, we do not assume that the functions here are periodic. Again, we have the property

$$\frac{1}{b^m}\sum_{\ell=0}^{b^m-1}\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}_\ell)=\begin{cases}1 & \text{if } C_1^\top\mathbf{k}_1+\cdots+C_s^\top\mathbf{k}_s=\mathbf{0}\in\mathbb{F}_b^m,\\ 0 & \text{otherwise,}\end{cases}$$

where $\boldsymbol{k}=(k_1,\ldots,k_s)\in\mathbb{N}_0^s$, $k_j=k_{j,0}+k_{j,1}b+\cdots$, and $\mathbf{k}_j=(k_{j,0},\ldots,k_{j,n-1})^\top$.

The set of all $\boldsymbol{k}$ for which the sum above is 1 is called the dual net $\mathcal{D}$, i.e.,

$$\mathcal{D}=\{\boldsymbol{k}\in\mathbb{N}_0^s : C_1^\top\mathbf{k}_1+\cdots+C_s^\top\mathbf{k}_s=\mathbf{0}\in\mathbb{F}_b^m\}.$$

Using the Walsh series expansion of the function f we obtain

$$\begin{aligned}\left|\int_{[0,1]^s}f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}-\frac{1}{b^m}\sum_{\ell=0}^{b^m-1}f(\boldsymbol{x}_\ell)\right| &= \left|\widehat{f}_{\mathrm{wal}}(\mathbf{0})-\sum_{\boldsymbol{k}\in\mathbb{N}_0^s}\widehat{f}(\boldsymbol{k})\frac{1}{b^m}\sum_{\ell=0}^{b^m-1}\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}_\ell)\right|\\ &= \left|\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}\widehat{f}_{\mathrm{wal}}(\boldsymbol{k})\right|\\ &\le \sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}|\widehat{f}_{\mathrm{wal}}(\boldsymbol{k})|.\end{aligned}$$

We can use the bound on the Walsh coefficients of the previous subsection to obtain

$$\left|\int_{[0,1]^s}f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}-\frac{1}{b^m}\sum_{\ell=0}^{b^m-1}f(\boldsymbol{x}_\ell)\right|\le C_{\alpha,b,s}\|f\|_\alpha\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}b^{-\mu_\alpha(\boldsymbol{k})}.$$

The last inequality separates the contribution of the function from the contribution of the quasi-Monte Carlo rule, i.e., $\|f\|_\alpha$ depends only on the function f but not on the quasi-Monte Carlo rule, whereas $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}b^{-\mu_\alpha(\boldsymbol{k})}$ depends only on the generating matrices of the digital net and not on the function itself (only on the smoothness of f; i.e., it is the same for all functions which have smoothness α). Therefore, when considering the integration error we can now focus on the term $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}b^{-\mu_\alpha(\boldsymbol{k})}$, which we do in the following subsection.

4.3 Generalized Digital Nets

The aim is now to find digital nets, i.e., generating matrices $C_1,\ldots,C_s\in\mathbb{F}_b^{n\times m}$ such that $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}b^{-\mu_\alpha(\boldsymbol{k})}=\mathcal{O}(N^{-\alpha}(\log N)^{\alpha s})$, where the number of quadrature points $N=b^m$.

Roughly speaking, the sum $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}b^{-\mu_\alpha(\boldsymbol{k})}$ is dominated by its largest term. To find this largest term, define

$$\mu_\alpha^*(C_1,\ldots,C_s)=\min_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}}\mu_\alpha(\boldsymbol{k}).$$

The dependence on the generating matrices $C_1, \dots, C_s$ on the right hand side of the above equation is via the dual net $\mathcal{D} = \mathcal{D}(C_1, \dots, C_s)$. The largest term in $\sum_{\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}} b^{-\mu_\alpha(\boldsymbol{k})}$ is then $b^{-\mu_\alpha^*(C_1,\dots,C_s)}$.

In order to achieve a convergence of almost $N^{-\alpha} = b^{-\alpha m}$ we must have that the largest term in $\sum_{\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}} b^{-\mu_\alpha(\boldsymbol{k})}$ is also of this order, that is, we must have $\mu_\alpha^*(C_1, \dots, C_s) \approx \alpha m$ (or say $\mu_\alpha^*(C_1, \dots, C_s) > \alpha m - t$ for some constant t independent of m). That this condition is also sufficient is quite technical and was shown in [4, Lemma 5.2]. (The definition of $\mu_\alpha^*(C_1, \dots, C_s)$ is reminiscent of the figure of merit for lattice rules, see (1). For lattice rules an approach of proving the desired order of convergence was described in Subsection 2.3.)

We can use some analogy to find matrices $C_1, \dots, C_s \in \mathbb{F}_b^{n \times m}$ which achieve $\mu_\alpha^*(C_1, \dots, C_s) \approx \alpha m$: The definition of $\mu_\alpha^*(C_1, \dots, C_s)$ is similar to the figure of merit ρ for lattice rules, or more precisely to $\log \rho$, which for classical digital nets is analogous to the strength of the digital net, that is, $m - t$. On the other hand, the classical case corresponds to $\alpha = 1$, hence one can expect a relationship between $\mu_1^*(C_1, \dots, C_s)$ and $m - t$.

Indeed, we have the following: Let $C_j = (\mathbf{c}_{j,1}^\top, \dots, \mathbf{c}_{j,n}^\top)^\top$, i.e., $\mathbf{c}_{j,\ell} \in \mathbb{F}_b^m$ is the ℓth row of C_j. Then the matrices $C_1, \dots, C_s$ generate a classical digital (t, m, s)-net if for all $i_1, \dots, i_s \geq 0$ with $i_1 + \cdots + i_s \leq m - t$, the vectors

$$\mathbf{c}_{1,1}, \dots, \mathbf{c}_{1,i_1}, \dots, \mathbf{c}_{s,1}, \dots, \mathbf{c}_{s,i_s}$$

are linearly independent over $\mathbb{F}_b$.

Now assume $C_1, \dots, C_s$ generate a classical digital (t, m, s)-net and that we are given a $\boldsymbol{k} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}$ with $\mu_1(\boldsymbol{k}) \leq m - t$. Let $i_j = \mu_1(k_j)$ for $j = 1, \dots, s$, then $C_1^\top \mathbf{k}_1 + \cdots + C_s^\top \mathbf{k}_s$ is a linear combination of the vectors $\mathbf{c}_{1,1}, \dots, \mathbf{c}_{1,i_1}, \dots, \mathbf{c}_{s,1}, \dots, \mathbf{c}_{s,i_s}$. As $\boldsymbol{k} \neq \boldsymbol{0}$ and $i_1 + \cdots + i_s \leq m - t$, which implies that $\mathbf{c}_{1,1}, \dots, \mathbf{c}_{1,i_1}, \dots, \mathbf{c}_{s,1}, \dots, \mathbf{c}_{s,i_s}$ are linearly independent, it follows that $C_1^\top \mathbf{k}_1 + \cdots + C_s^\top \mathbf{k}_s \neq \boldsymbol{0} \in \mathbb{F}_b^m$. Thus $\boldsymbol{k} \notin \mathcal{D}$. This shows that if $C_1, \dots, C_s$ generate a classical digital (t, m, s)-net and $\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}$, then $\mu_1(\boldsymbol{k}) > m - t$. This is precisely the type of result described above which we also want to have for $\alpha > 1$.

In the classical case $\alpha = 1$ we had some linear independence condition of the rows of the generating matrices which lead to the desired result. We now want to generalize this linear independence condition to $\alpha > 1$, i.e., we want to have that if $\boldsymbol{k} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}$ with $\mu_\alpha(\boldsymbol{k}) \leq \alpha m - t$, then the generating matrices should have linearly independent rows such that $C_1^\top \mathbf{k}_1 + \cdots + C_s^\top \mathbf{k}_s \neq \boldsymbol{0} \in \mathbb{F}_b^m$. Let $\boldsymbol{k} = (k_1, \dots, k_s)$, where $k_j = \kappa_{j,1} b^{a_{j,1}-1} + \cdots + \kappa_{j,v_j} b^{a_{j,v_j}-1}$, with $a_{j,1} > \cdots > a_{j,v_j} > 0$ and $0 < \kappa_{j,1}, \dots, \kappa_{j,v_j} < b$. First note that if $n < \alpha m - t$, then $\boldsymbol{k} = (b^n, 0, \dots, 0) \in \mathcal{D}$, but $\mu_\alpha(\boldsymbol{k}) = n + 1 \leq \alpha m - t$. In order to avoid this problem we may choose $n = \alpha m$. Hence we may now assume that $a_{j,1} \leq n = \alpha m$ for $j = 1, \dots, s$, as otherwise $\mu_\alpha(\boldsymbol{k}) > \alpha m$ already and no independence condition on the generating matrices is required in this case.

Now $C_1^\top \mathbf{k}_1 + \cdots + C_s^\top \mathbf{k}_s$ is a linear combination of the rows

$$\mathbf{c}_{1,a_{1,1}}, \dots, \mathbf{c}_{1,a_{1,v_1}}, \dots, \mathbf{c}_{s,a_{s,1}}, \dots, \mathbf{c}_{s,a_{s,v_s}}.$$

Thus, if these rows are linearly independent, then $C_1^\top \mathbf{k}_1 + \cdots + C_s^\top \mathbf{k}_s \neq \mathbf{0} \in \mathbb{F}_b^m$, and therefore $\boldsymbol{k} \notin \mathcal{D}$.

Therefore, if $C_1, \ldots, C_s \in \mathbb{F}_b^{n \times m}$ are such that for all choices of $a_{j,1} > \cdots > a_{j,\nu_j} > 0$ for $j = 1, \ldots, s$, with

$$a_{1,1} + \cdots + a_{1,\min(\alpha,\nu_1)} + \cdots + a_{s,1} + \cdots + a_{s,\min(\alpha,\nu_s)} \leq \alpha m - t,$$

the rows

$$\mathbf{c}_{1,a_{1,1}}, \ldots, \mathbf{c}_{1,a_{1,\nu_1}}, \ldots, \mathbf{c}_{s,a_{s,1}}, \ldots, \mathbf{c}_{s,a_{s,\nu_s}}$$

are linearly independent, then $\boldsymbol{k} \in \mathcal{D} \setminus \{\mathbf{0}\}$ implies that $\mu_\alpha(\boldsymbol{k}) > \alpha m - t$. (Note that we also include the case where some $\nu_j = 0$, in which case we just set $a_{j,1} + \cdots + a_{j,\min(\alpha,\nu_j)} = 0$.)

We can now formally define such digital nets for which the generating matrices satisfy such a property. The following definition is a special case of [4, Definition 4.3].

Definition 2. Let $m, \alpha \geq 1$, and $0 \leq t \leq \alpha m$ be integers. Let $\mathbb{F}_b$ be the finite field of prime order b and let $C_1, \ldots, C_s \in \mathbb{F}_b^{\alpha m \times m}$ with $C_j = (\mathbf{c}_{j,1}^\top, \ldots, \mathbf{c}_{j,\alpha m}^\top)^\top$. If for all $0 < a_{j,\nu_j} < \cdots < a_{j,1}$, where $0 \leq \nu_j$ for all $j = 1, \ldots, s$, with

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j,\alpha)} a_{j,l} \leq \alpha m - t$$

the vectors

$$\mathbf{c}_{1,a_{1,\nu_1}}, \ldots, \mathbf{c}_{1,a_{1,1}}, \ldots, \mathbf{c}_{s,a_{s,\nu_s}}, \ldots, \mathbf{c}_{s,a_{s,1}}$$

are linearly independent over $\mathbb{F}_b$, then the digital net with generating matrices $C_1, \ldots, C_s$ is called a digital $(t, \alpha, 1, \alpha m \times m, s)$-net over $\mathbb{F}_b$.

The need for a more general definition in [4] arises as we assume therein that the smoothness α of the integrand is not known, so one cannot choose $n = \alpha m$ in this case.

We have seen so far that a digital $(t, \alpha, 1, \alpha m \times m, s)$-net used as quadrature points in a quasi-Monte Carlo rule will yield a convergence of the integration error of order $N^{-\alpha}(\log N)^{\alpha s}$ for integrands with $\|f\|_\alpha < \infty$.

The remaining question now is: do digital $(t, \alpha, 1, \alpha m \times m, s)$-nets for all given $\alpha, s \geq 1$ and some fixed t (which may depend on α and s but not on m) exist for all $m \in \mathbb{N}$? An affirmative answer to this question will be given in the next subsection.

4.4 Construction of Generalized Digital Nets

In this subsection we present explicit constructions of digital $(t, \alpha, 1, \alpha m \times m, s)$-nets. The basic construction principle appeared first in [3] and was slightly modified in [4]. The construction requires a parameter d, which, in case the smoothness of the

integrand α is known, should be chosen as $d = \alpha$. In this subsection we present this construction and a bound on the t-value, but we assume that α is known explicitly and hence choose $d = \alpha$.

Let $C_1, \dots, C_{s\alpha}$ be the generating matrices of a digital $(t', m, s\alpha)$-net; we recall that many explicit examples of such generating matrices are known, see e.g., [8, 12, 13, 14, 16, 17, 22] and the references therein. As we will see later, the choice of the underlying $(t', m, s\alpha)$-net has a direct impact on the bound on the t-value of the digital $(t, \alpha, 1, \alpha m \times m, s)$-net. Let $C_j = (\mathbf{c}_{j,1}^\top, \dots, \mathbf{c}_{j,m}^\top)^\top$ for $j = 1, \dots, s\alpha$; i.e., $\mathbf{c}_{j,l}$ are the row vectors of C_j. Now let the matrix $C_j^{(\alpha)}$ be made of the first rows of the matrices $C_{(j-1)\alpha+1}, \dots, C_{j\alpha}$, then the second rows of $C_{(j-1)\alpha+1}, \dots, C_{j\alpha}$, and so on. The matrix $C_j^{(\alpha)}$ is then an $\alpha m \times m$ matrix; i.e., $C_j^{(\alpha)} = (\mathbf{c}_{j,1}^{(\alpha)}, \dots, \mathbf{c}_{j,\alpha m}^{(\alpha)})^\top$, where $\mathbf{c}_{j,l}^{(\alpha)} = \mathbf{c}_{u,v}$ with $l = (v-j)\alpha + u$, $1 \le v \le m$, and $(j-1)\alpha < u \le j\alpha$ for $l = 1, \dots, \alpha m$ and $j = 1, \dots, s$.

To give the idea why this construction works we may consider the case $s = 1$. Let $\alpha > 1$. To simplify the notation we drop the j (which denotes the coordinate) from the notation for a moment. Let $C^{(\alpha)}$ be constructed from a classical digital (t', m, α)-net with generating matrices $C_1, \dots, C_\alpha$ as described above. Let $\alpha m \ge a_1 > a_2 > \cdots > a_\nu \ge 1$. Then we need to consider the row vectors $\mathbf{c}_{a_1}^{(\alpha)}, \dots, \mathbf{c}_{a_\nu}^{(\alpha)}$. Now by the construction above, the vector $\mathbf{c}_{a_1}^{(\alpha)}$ may stem from any of the generating matrices $C_1, \dots, C_\alpha$. Without loss of generality assume that $\mathbf{c}_{a_1}^{(\alpha)}$ stems from C_1, i.e., it is the i_1th row of C_1, where $i_1 = \lceil a_1/\alpha \rceil$. Next consider $\mathbf{c}_{a_2}^{(\alpha)}$. This row vector may again stem from any of the matrices $C_1, \dots, C_\alpha$. If $\mathbf{c}_{a_2}^{(\alpha)}$ also stems from C_1, then $\lceil a_2/\alpha \rceil < i_1$. If not, we may w.l.o.g. assume that it stems from C_2. Indeed, it will be the i_2th row of C_2, where $i_2 = \lceil a_2/\alpha \rceil$. We continue in this fashion and define numbers $i_3, i_4, \dots, i_l$, where $1 \le l \le \alpha$. Further we set $i_{l+1} = \cdots = i_\alpha = 0$. Then by the (t', m, α)-net property of $C_1, \dots, C_\alpha$, it follows that $\mathbf{c}_{a_1}^{(\alpha)}, \dots, \mathbf{c}_{a_\nu}^{(\alpha)}$ are linearly independent provided that $i_1 + \cdots + i_\alpha \le m - t'$. Hence, if we choose t such that $a_1 + \cdots + a_{\min(\alpha,\nu)} \le \alpha m - t$ implies that $i_1 + \cdots + i_\alpha \le m - t'$ for all admissible choices of $a_1, \dots, a_\nu$, then the digital $(t, \alpha, 1, \alpha m \times m, 1)$-net property of $C^{(\alpha)}$ follows.

Note that $i_1 = \lceil a_1/\alpha \rceil$ and $i_l \le \lceil a_l/\alpha \rceil$ for $l = 2, \dots, \alpha$. Thus

$$\begin{aligned} i_1 + \cdots + i_\alpha &\le \lceil a_1/\alpha \rceil + \cdots + \lceil a_\alpha/\alpha \rceil \\ &\le (a_1 + \cdots + a_\alpha + \alpha(\alpha-1))/\alpha \\ &= \frac{a_1 + \cdots + a_\alpha}{\alpha} + \alpha - 1 \\ &\le m - t/\alpha + \alpha - 1. \end{aligned}$$

Thus, if we choose t such that $m - t/\alpha + \alpha - 1 \le m - t'$, then the result follows. Simple algebra then shows that

$$t = \alpha t' + \alpha(\alpha - 1)$$

will suffice.

A more general and improved result is given in the following which is a special case of [4, Theorem 4.11], with an improvement for some cases from [6] (a proof of this result can be found in [6, 3, 7]).

Theorem 1. *Let $\alpha \geq 1$ be a natural number and let $C_1, \ldots, C_{s\alpha}$ be the generating matrices of a digital $(t', m, s\alpha)$-net over the finite field $\mathbb{F}_b$ of prime power order b. Let $C_1^{(\alpha)}, \ldots, C_s^{(\alpha)}$ be defined as above. Then the matrices $C_1^{(\alpha)}, \ldots, C_s^{(\alpha)}$ are the generating matrices of a digital $(t, \alpha, 1, \alpha m \times m, s)$-net over $\mathbb{F}_b$ with*

$$t = \alpha \min\left(m, t' + \left\lfloor \frac{s(\alpha-1)}{2} \right\rfloor\right).$$

This shows that digital $(t, \alpha, 1, \alpha m \times m, s)$-nets exist for all $\alpha, m, s \geq 1$ with t bounded independently of m. Indeed, also the dependence of t on α and s is known from [6]: namely $t \asymp \alpha^2 s$.

Geometrical properties of digital $(t, \alpha, 1, \alpha m \times m, s)$-nets and their generalization were shown in [6]. In the following section we show pictures of those properties.

5 Geometrical Properties of Generalized Digital Nets

In this section we describe geometrical properties of generalized digital nets. The generating matrices $C_1^{(2)} \in \mathbb{F}_2^{4\times 8}$ and $C_2^{(2)} \in \mathbb{F}_2^{4\times 8}$ for the digital net over $\mathbb{F}_2$ shown in Figure 1 are obtained from the classical digital $(1, 4, 4)$-net with the following generating matrices:

$$C_1 = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{pmatrix}, C_2 = \begin{pmatrix} 0&0&0&1\\ 0&0&1&0\\ 0&1&0&0\\ 1&0&0&0 \end{pmatrix}, C_3 = \begin{pmatrix} 1&1&1&1\\ 0&1&0&1\\ 0&0&1&1\\ 0&0&0&1 \end{pmatrix}, C_4 = \begin{pmatrix} 0&1&1&0\\ 1&1&0&1\\ 0&0&0&1\\ 0&0&1&0 \end{pmatrix}.$$

Using the construction principle from [3, 4] described above, we obtain

$$C_1^{(2)} = \begin{pmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&1&0\\ 0&1&0&0\\ 0&0&0&1\\ 1&0&0&0 \end{pmatrix} \text{ and } C_2^{(2)} = \begin{pmatrix} 1&1&1&1\\ 0&1&1&0\\ 0&1&0&1\\ 1&1&0&1\\ 0&0&1&1\\ 0&0&0&1\\ 0&0&0&1\\ 0&0&1&0 \end{pmatrix}.$$

Theorem 1 implies that $C_1^{(2)}$, $C_2^{(2)}$ generate a digital $(4, 2, 1, 8 \times 4, 2)$-net. Upon inspection one can see that it is also a $(3, 2, 1, 8 \times 4, 2)$-net, but not a $(2, 2, 1, 8 \times 4, 2)$-net (the first two rows of $C_1^{(2)}$ and $C_2^{(2)}$ are linearly dependent).

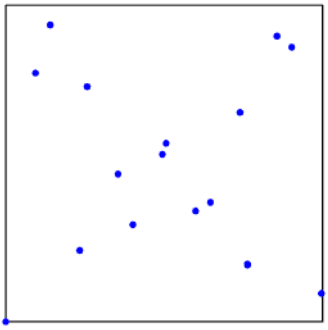


Fig. 1 A digital $(3, 2, 1, 8 \times 4, 2)$-net over $\mathbb{Z}_2$ which is also a classical digital $(1, 4, 2)$-net over $\mathbb{Z}_2$.

Figure 2 shows that the point set is a classical $(1, 4, 2)$-net. Indeed, this is true more generally: Generalized digital nets are also classical digital nets (with the classical t-value worse than the best classical nets known for the chosen parameters, which is understandable as generalized digital nets have some additional structure as we will see below) and are therefore also well distributed. This statement is made precise in the following proposition.

Proposition 1. *Let $\alpha \geq 1$ be a natural number and let $C_1, \dots, C_{s\alpha}$ be the generating matrices of a digital $(t, m, s\alpha)$-net over the finite field $\mathbb{F}_b$ of prime power order b. Let $C_1^{(\alpha)}, \dots, C_s^{(\alpha)}$ be defined as above. Then the matrices $C_1^{(\alpha)}, \dots, C_s^{(\alpha)}$ are the generating matrices of a digital (t, m, s)-net over $\mathbb{F}_b$.*

Proof. Let $d_1, \dots, d_s \geq 0$ be integers such that $d_1 + \cdots + d_s \leq m - t$. Then the first d_j rows of $C_j^{(\alpha)}$ stem from the matrices $C_{(j-1)\alpha+1}, \dots, C_{j\alpha}$. Indeed there are numbers $l_{(j-1)\alpha+1}, \dots, l_{j\alpha} \geq 0$, such that $d_j = l_{(j-1)\alpha+1} + \cdots + l_{j\alpha}$ with the property that the first d_j rows of $C_j^{(\alpha)}$ are exactly the union of the first $l_{(j-1)\alpha+r}$ rows of $C_{(j-1)\alpha+r}$ for $r = 1, \dots \alpha$. Hence the fact that $\sum_{j=1}^{s} d_j = \sum_{j=1}^{s} \sum_{r=1}^{\alpha} l_{(j-1)\alpha+r} \leq m - t$ and the $(t, m, s\alpha)$-net property of $C_1, \dots, C_{s\alpha}$ imply that the union of the first d_j rows of C_j, $j = 1, \dots, s$, are linearly independent. This implies that $C_1^{(\alpha)}, \dots, C_s^{(\alpha)}$ generate a digital (t, m, s)-net.

Remark 1. (i) Proposition 1 yields a better result than what can be obtained from using Theorem 1 and [4, Theorem 4.10 (ii)].

(ii) From the proof of Proposition 1 it is apparent that we do not consider all linear combinations of rows of the generating matrices $C_1, \dots, C_{s\alpha}$ of the original (t, m, s)-net. Hence, for particular choices of the original net, it is possible to

obtain a generalized net which is a classical (t', m, s)-net with $t' < t$, where t is the quality parameter of the original net. For example, if one uses a digital net obtained from a Sobol sequence, than choosing certain (as of now unknown) direction numbers could yield such an improvement. Similar optimizations could also be applied to other digital nets.

(iii) When using the construction of generalized nets from [3, 4] as outlined above, then, as of now, it seems advantageous to choose the original digital net with as small a t-value as possible. For example, using the table from MinT (`http://mint.sbg.ac.at`), we know that there is a digital $(6, 15, 10)$-net. For $\alpha = 2$, the generalized digital net constructed as outlined above using a digital $(8, 15, 20)$-net, is also a digital $(8, 15, 10)$-net, for $\alpha = 3$, we can obtain a generalized digital net which is also a digital $(9, 15, 10)$-net and for $\alpha = 4$ we can obtain a generalized digital net which is also a digital $(10, 15, 10)$-net in base 2 (with the advantage of additional properties of generalized nets).

(iv) An analogue to Proposition 1 for digital sequences also holds with the classical t-value of the generalized sequence being the same as the original digital sequence (which is of course in dimension $s\alpha$).

Figure 3 shows a partition of the square for which each union of the shaded rectangles contains exactly two points. Figures 4 and 5 show that also other partitions of the unit square are possible where each union of shaded rectangles contains the fair number of points. Many other partitions of the square are possible where the point set always contains the fair number of points in each union of rectangles, see [6], but there are too many of them to show them all here. Even in the simple case considered here there are 12 partitions possible, for each of which the point set is fair - this is quite remarkable since the point set itself has only 16 points (we exclude all those partitions for which the fairness would follow already from some other partition, otherwise there would be 34 of them). In the classical case we have 4 such partitions, all of which are shown in Figure 2. (The partitions from the classical case are included in the generalized case; so out of the 12 partitions 4 are shown in Figure 2, one is shown in Figure 3, one is shown in Figure 5 and one is indicated in Figure 4.)

The subsets of $[0, 1)^s$ which form a partition and which each have the fair number of points are of the form:

$$J(\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{d}_{\boldsymbol{\nu}}) = \prod_{j=1}^{s} \bigcup_{d_{j,l}=0}^{b-1} \bigcup_{l \in \{1,\ldots,\alpha m\} \setminus \{a_{j,1},\ldots,a_{j,\nu_j}\}} \left[\frac{d_{j,1}}{b} + \cdots + \frac{d_{j,n}}{b^{\alpha m}}, \frac{d_{j,1}}{b} + \cdots + \frac{d_{j,n}}{b^{\alpha m}} + \frac{1}{b^{\alpha m}} \right),$$

where $b \geq 2$ is the base and where $\sum_{j=1}^{s} \sum_{l=1}^{\nu_j} a_{j,l} \leq \alpha m - t$. For $j = 1, \ldots, s$ we again assume $1 \leq a_{j,\nu_j} < \cdots < a_{j,1} \leq \alpha m$ in case $\nu_j > 0$ and $\{a_{j,1}, \ldots, a_{j,\nu_j}\} = \emptyset$ in case $\nu_j = 0$. Further, we also use the following notation: $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_s)$, $|\boldsymbol{\nu}|_1 = \sum_{j=1}^{s} \nu_j$, $\mathbf{a}_{\boldsymbol{\nu}} = (a_{1,1}, \ldots, a_{1,\nu_1}, \ldots, a_{s,1}, \ldots, a_{s,\nu_s})$, $\mathbf{d}_{\boldsymbol{\nu}} \in \{0, \ldots, b-1\}^{|\boldsymbol{\nu}|_1}$, and $\mathbf{d}_{\boldsymbol{\nu}} =$

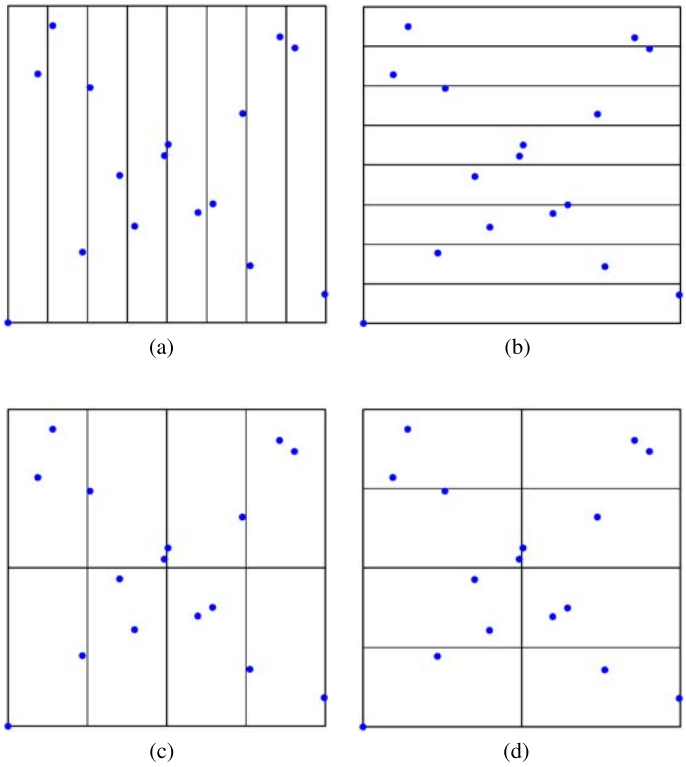


Fig. 2 The digital $(3, 2, 1, 8 \times 4, 2)$-net is also a digital $(1, 4, 2)$-net, as each elementary interval of volume $1/8$ of every partition of the unit square contains exactly two points.

$(d_{1,i_{1,1}}, \dots, d_{1,i_{1,\nu_1}}, \dots, d_{s,i_{s,1}}, \dots, d_{s,i_{s,\nu_s}})$, where the components $a_{j,l}$ and $d_{j,l}$, $l = 1, \dots, \nu_j$, do not appear in the vectors $\mathbf{a}_{\boldsymbol{\nu}}$ and $\mathbf{d}_{\boldsymbol{\nu}}$ in case $\nu_j = 0$.

Figures 2, 3, 4, and 5 give only a few examples of unions of intervals for which each subset of the partition contains the right number of points. As the $J(\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{d}_{\boldsymbol{\nu}})$, for fixed $\boldsymbol{\nu}$ and $\mathbf{a}_{\boldsymbol{\nu}}$ (with $\mathbf{d}_{\boldsymbol{\nu}}$ running through all possibilities) form a partition of $[0, 1)^s$, it is clear that the right number of points in $J(\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{d}_{\boldsymbol{\nu}})$ has to be $b^m \mathrm{Vol}(J(\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{d}_{\boldsymbol{\nu}}))$. For example, the digital net in Figure 3 has 16 points and the partition consists of 8 different subsets $J(\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{d}_{\boldsymbol{\nu}})$, hence each $J(\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{d}_{\boldsymbol{\nu}})$ contains exactly $16/8 = 2$ points. (In general, the volume of $J(\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{d}_{\boldsymbol{\nu}})$ is given by $b^{-|\boldsymbol{\nu}|_1}$, see [6].)

6 Geometrical Numerical Integration

The geometrical properties needed for numerical integration can be illustrated in the one-dimensional case.

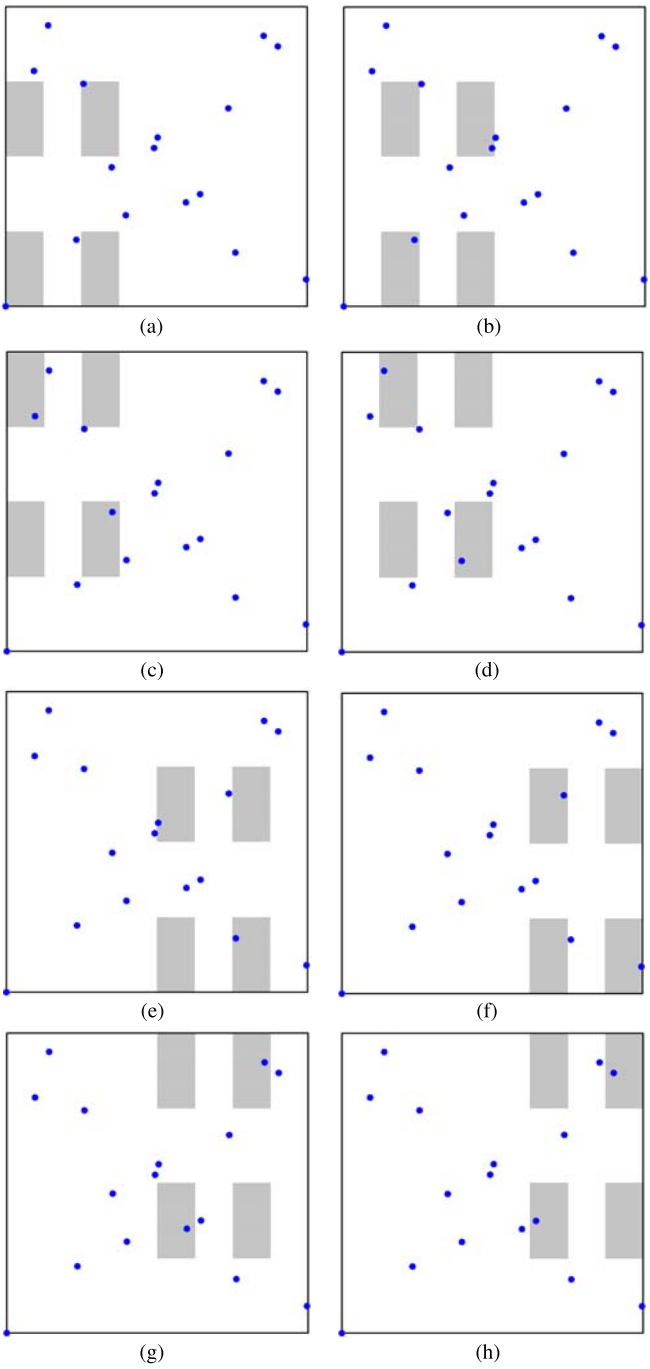


Fig. 3 The digital $(3, 2, 1, 8 \times 4, 2)$-net. The union of the shaded rectangles in each figure from (a) to (h) contains exactly two points.

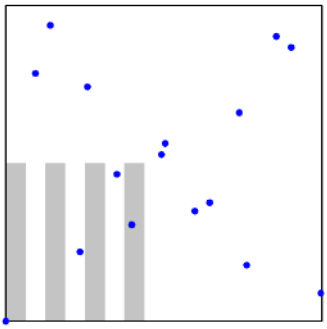


Fig. 4 Digital $(3, 2, 1, 8 \times 4, 2)$-net over $\mathbb{Z}_2$. The union of the shaded rectangles contains two points. As in Figure 3 one can also form a partition of the square with this type of rectangle where each union of rectangles contains two points.

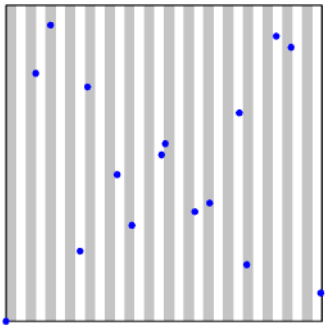


Fig. 5 Digital $(3, 2, 1, 8 \times 4, 2)$-net over $\mathbb{Z}_2$. The union of the shaded rectangles contains half the points.

Assume $f : [0, 1] \to \mathbb{R}$ is twice continuously differentiable. Then

$$f(x) = f(0) + \int_0^x f'(t)\,\mathrm{d}t = f(0) + x f'(0) + \int_0^1 (x-t)_+ f''(t)\,\mathrm{d}t, \tag{5}$$

where $(x-t)_+$ is $x-t$ for $x \geq t$ and 0 otherwise.

Let $x_1, \ldots, x_N \in [0, 1]$, then using (5) we obtain

$$\begin{aligned}
&\frac{1}{N}\sum_{h=1}^{N} f(x_h) - \int_0^1 f(x)\,\mathrm{d}x \\
&= f(0) + f'(0)\frac{1}{N}\sum_{h=1}^{N} x_h + \frac{1}{N}\sum_{h=1}^{N}\int_0^1 (x_h - t)_+ f''(t)\,\mathrm{d}t \\
&\quad - f(0) - f'(0)\int_0^1 x\,\mathrm{d}x - \int_0^1\int_0^1 (x-t)_+ f''(t)\,\mathrm{d}t\,\mathrm{d}x
\end{aligned}$$

$$= f'(0)\left[\frac{1}{N}\sum_{h=1}^{N} x_h - \int_0^1 x\,\mathrm{d}x\right]$$
$$+ \int_0^1 f''(t)\left[\frac{1}{N}\sum_{h=1}^{N}(x_h - t)_+ - \int_0^1 (x-t)_+\,\mathrm{d}x\right]\mathrm{d}t.$$

Taking the absolute value of the integration error we obtain

$$\left|\frac{1}{N}\sum_{h=1}^{N} f(x_h) - \int_0^1 f(x)\,\mathrm{d}x\right| \le \left[|f'(0)| + \int_0^1 |f''(t)|\,\mathrm{d}t\right] \sup_{0\le t\le 1} |\Delta_N(t)|,$$

where

$$\Delta_N(t) = \left|\frac{1}{N}\sum_{h=1}^{N}(x_h - t)_+ - \int_0^1 (x-t)_+\,\mathrm{d}x\right|.$$

The factor $|f'(0)| + \int_0^1 |f''(t)|\,\mathrm{d}t$ is a seminorm of the function f and the factor $\sup_{0\le t\le 1}|\Delta_N(t)|$ measures properties of the quadrature points $x_1, \ldots, x_N$. For example, for $t = 0$ the quadrature rule would numerically integrate the function x and

$$\Delta_N(0) = \frac{1}{N}\sum_{h=1}^{N} x_h - \int_0^1 x\,\mathrm{d}x$$

is the integration error.

In order to obtain a convergence of $N^{-2+\delta}$, $\delta > 0$, our quadrature points should be chosen such that $\Delta_N(t) = \mathcal{O}(N^{-2+\delta})$ for any $0 \le t \le 1$. Any equidistant quadrature points only yield $\sup_{0\le t\le 1}|\Delta_N(t)| = \mathcal{O}(N^{-1})$, hence the points from a digital $(t, \alpha, 1, \alpha m \times m, s)$-net are not equidistant for $\alpha > 1$, but introduce some cancellation effect as we explain in the following.

Consider Figure 6. We assume we want to numerically integrate the function x (then $\Delta(0)$ would be the integration error) using a $(t, 2, 1, 2m \times m, 1)$-net (where $b = 2$). This function is relevant as it appears in the upper bound (the case where $0 < t \le 1$ is similar.) Assume we want to put two points in the interval $[0, 1/2)$, such that one point is in $[0, 1/4)$ and another one is in $[1/4, 1/2)$, as illustrated in Figure 6(a). Then we get some integration error for the point x_1 in $[0, 1/4)$ of the form $e_1 = x_1 - \int_0^{1/4} \mathrm{d}x$ and another integration error for the point x_2 in $[1/4, 1/2)$ of the form $e_2 = x_2 - \int_{1/4}^{1/2} x\,\mathrm{d}x$. The integration error for the interval $[0, 1/2)$ is then the sum of the two errors $e_1 + e_2$. If both e_1 and e_2 have the same sign then the absolute value of error $|e_1 + e_2|$ for the integral $\int_0^{1/2} x\,\mathrm{d}x$ increases, whereas if they have opposite signs then we get some cancellation effect and the absolute value of the error, $|e_1 + e_2|$, decreases.

We can partition each of the intervals $[0, 1/4)$ and $[1/4, 1/2)$ again into two intervals to obtain $[0, 1/8)$ and $[1/8, 1/4)$ on the one hand and $[1/4, 3/8)$ and $[3/8, 1/2)$ on the other hand, see Figure 6(b). Next we put two points in the interval $[0, 1/2)$: In Figure 6(c) one point is in the interval $[0, 1/8)$ and the other one in $[3/8, 1/2)$ and in

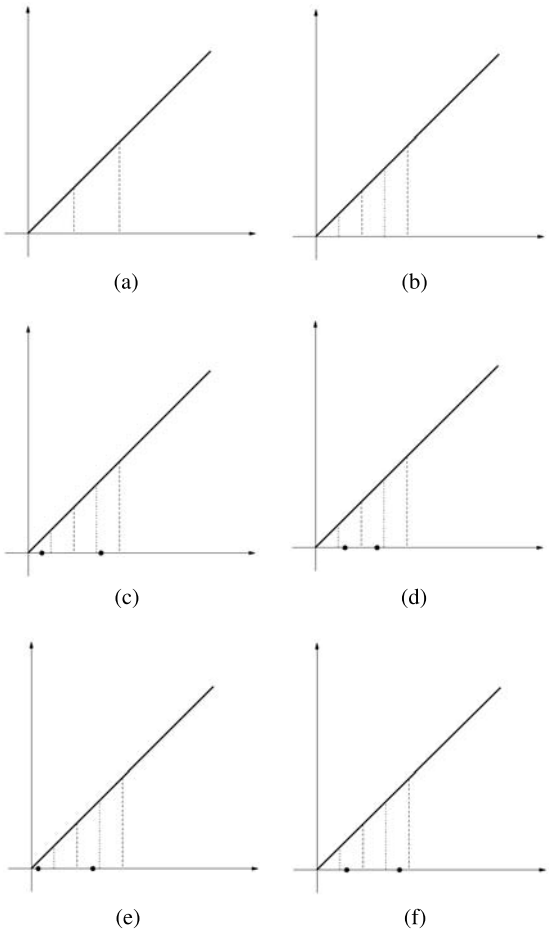


Fig. 6 Geometrical numerical integration.

Figure 6(d) one is in $[1/8, 1/4)$ and one in $[1/4, 3/8)$. In both cases, when considering the integral $\int_0^{1/2} x \, \mathrm{d}x$, we get some cancellation effect: in Figure 6(c) the point in $[0, 1/8)$ underestimates the integral $\int_0^{1/4} x \, \mathrm{d}x$, whereas the point in $[3/8, 1/2)$ overestimates the integral $\int_{1/4}^{1/2} x \, \mathrm{d}x$. Similarly for Figure 6(d). On the other hand, in Figure 6(e) both points underestimate the corresponding integral and in Figure 6(f) both points overestimate the corresponding integral - hence the integration errors add up in this case.

We started out saying that we want to have one point in the black interval in Figure 7(a) and one in the white. But to get some cancellation effect we also want to have that one point is in the black part in Figure 7(b) and one in the white part.

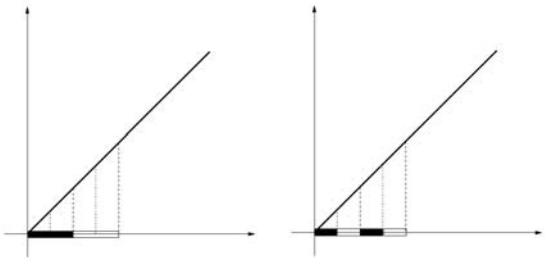


Fig. 7 Geometrical numerical integration.

But such a structure is exhibited by the point set shown in Figure 1. Considering the projection of the point set onto the x-axis, Figure 2(c) shows that the same number of points is in the interval $[0, 1/4)$ as there is in $[1/4, 1/2)$. Figures 3(a) and (b) on the other hand show that the same number of points is in $[0, 1/8) \cup [1/4, 3/8)$ as there is in $[1/8, 1/4) \cup [3/8, 1/2)$. Therefore this point set shows the desired cancellation effect which allows us to obtain a convergence beyond $\mathcal{O}(N^{-1+\delta})$.

Acknowledgements The support of the ARC under its Centre of Excellence program is gratefully acknowledged. The author would also like to thank Friedrich Pillichshammer for producing the figures in this paper and his hospitality during the author's visit at the J.K. Universität Linz. Finally, the author would also like to express his gratitude to Art Owen and the referees for their helpful comments.

References

1. J. Baldeaux and J. Dick, QMC rules of arbitrary high order: reproducing kernel Hilbert space approach. To appear in Constructive Approx., 2010.
2. L.L. Cristea, J. Dick, G. Leobacher and F. Pillichshammer, The tent transformation can improve the convergence rate of quasi-Monte Carlo algorithms using digital nets. Numer. Math., 105, 413–455, 2007.
3. J. Dick, Explicit constructions of quasi-Monte Carlo rules for the numerical integration of high-dimensional periodic functions, SIAM J. Numer. Anal., 45, 2141–2176, 2007.
4. J. Dick, Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrary high order, SIAM J. Numer. Anal., 46, 1519–1553, 2008.
5. J. Dick, The decay of the Walsh coefficients of smooth functions. To appear in Bull. Austral. Math. Soc., 2009.
6. J. Dick and J. Baldeaux, Equidistribution properties of generalized nets and sequences. To appear in: P. L'Ecuyer and A.B. Owen (eds.), Monte Carlo and quasi-Monte Carlo methods 2008, Springer Verlag, to appear 2010.
7. J. Dick and P. Kritzer, Duality theory and propagation rules for generalized digital nets. To appear in Math. Comp., 2010.
8. H. Faure, Discrépance de suites associées à un système de numération (en dimension s), Acta Arith., 41, 337–351, 1982.
9. N.J. Fine, On the Walsh functions, Trans. Amer. Math. Soc., 65, 372–414, 1949.

10. F.J. Hickernell, Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules. In: K.T. Fang, F.J. Hickernell, and H. Niederreiter (eds.), Monte Carlo and quasi-Monte Carlo methods 2000, (Hong Kong), pp. 274–289, Springer, Berlin, 2002.
11. H. Niederreiter, Quasi-Monte Carlo methods and pseudo-random numbers, Bull. Amer. Math. Soc., 84, 957–1041, 1978.
12. H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, CBMS–NSF Series in Applied Mathematics 63, SIAM, Philadelphia, 1992.
13. H. Niederreiter, Constructions of (t, m, s)-nets and (t, s)-sequences, Finite Fields and Their Appl., 11, 578–600, 2005.
14. H. Niederreiter, Nets, (t, s)-sequences and codes. In: A. Keller, S. Heinrich, and H. Niederreiter (eds.), Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 83–100, Springer, Berlin, 2008.
15. H. Niederreiter and G. Pirsic, Duality for digital nets and its applications. Acta Arith., 97, 173–182, 2001.
16. H. Niederreiter and C.P. Xing, Global function fields with many rational places and their applications. In: R.C. Mullin and G.L. Mullen (eds.), Finite fields: theory, applications, and algorithms (Waterloo, ON, 1997), Contemp. Math., Vol. 225, pp.87–111, Amer. Math. Soc., Providence, RI, 1999.
17. H. Niederreiter and C.P. Xing, Rational points on curves over finite fields: theory and applications, London Mathematical Society Lecture Note Series, Vol. 285, Cambridge University Press, Cambridge, 2001.
18. A.B. Owen, Randomly permuted (t, m, s)-nets and (t, s)-sequences. In: H. Niederreiter and P. Jau-Shyong Shiue (eds.), Monte Carlo and quasi-Monte Carlo Methods in Scientific Computing, pp. 299–317, Springer, New York, 1995.
19. A.B. Owen, Scrambled net variance for integrals of smooth functions. Ann. Statist., 25, 1541–1562, 1997.
20. I.F. Sharygin, A lower estimate for the error of quadrature formulas for certain classes of functions, Zh. Vychisl. Mat. i Mat. Fiz., 3, 370–376, 1963.
21. I.H. Sloan and S. Joe, Lattice Methods for Multiple Integration, Oxford University Press, Oxford, 1994.
22. I.M. Sobol, Distribution of points in a cube and approximate evaluation of integrals, Zh. Vychisl. Mat. i Mat. Fiz., 7, 784–802, 1967.

Sensitivity Estimates for Compound Sums

Paul Glasserman and Kyoung-Kuk Kim

Abstract We derive unbiased derivative estimators for expectations of functions of random sums, where differentiation is taken with respect to a parameter of the number of terms in the sum. As the number of terms is integer valued, its derivative is zero wherever it exists. Nevertheless, we present two constructions that make the sum continuous even when the number of terms is not. We present a locally continuous construction that preserves continuity across a single change in the number of terms and a globally continuous construction specific to the compound Poisson case. This problem is motivated by two applications in finance: approximating Lévy-driven models of asset prices and exact sampling of a stochastic volatility process.

1 Introduction

By a compound sum we mean a random variable with the representation

$$X(\lambda) = \sum_{i=1}^{N(\lambda)} \xi_i, \tag{1}$$

in which $N(\lambda)$ is a non-negative integer-valued random variable, and the ξ_i are i.i.d. copies of a positive random variable ξ independent of $N(\lambda)$. The distribution of N (and therefore that of X) depends on the parameter λ; for example, in a compound Poisson sum, $N(\lambda)$ has a Poisson distribution with mean λ. We consider the problem of estimating a derivative of the form of $d\mathbb{E}[\Phi(X(\lambda))]/d\lambda$, the sensitivity of the expectation of some function Φ of $X(\lambda)$.

Paul Glasserman
Columbia Business School, New York, USA

Kyoung-Kuk Kim
Korea Advanced Institute of Science and Technology, South Korea

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_6,

Compound sums arise in many areas of applied probability, including queueing, inventory, and insurance risk, but our motivation arises from two applications in finance. The first is the use of Lévy processes to model asset returns; for purposes of simulation, the jump component of a Lévy process is often approximated by a compound Poisson process. The second application comes from the simulation of the Heston [10] stochastic volatility model. In [8], we have given an exact representation of the transitions of the process suitable for simulation. The representation involves compound sums in which the number of terms N has either a Poisson or Bessel distribution. Both Lévy processes and the Heston model are used in pricing options; sensitivities to model parameters are then essential inputs to option hedging.

Related problems of simulation sensitivity estimation have received extensive study; see, e.g., Chapter VII of Asmussen and Glynn [1] for an overview and references. A difficulty posed by a family of integer-valued random variables $N(\lambda)$ indexed by λ is that $dN(\lambda)/d\lambda$ must be zero anywhere it exists. This suggests that

$$\frac{d}{d\lambda}\Phi(X(\lambda)) = \Phi'(X(\lambda))\frac{d}{d\lambda}X(\lambda) = 0$$

wherever it exists, rendering it useless as an estimator of the sensitivity of the expectation $\mathbb{E}[\Phi(X(\lambda))]$.

The same problem arises when a Poisson process is used to model arrivals to a queue, and this has motivated alternative sample-path methods, such as the phantom method of Brémaud and Vazquez-Abad [4] and its variants. These methods address the discontinuities in $N(\lambda)$ by estimating the impact of one additional (or one fewer) arrival. The issue of sample-path discontinuities can alternatively be avoided using a likelihood ratio method estimator, as in p.222 of [1].

But the case of a compound sum offers possibilities not open to $N(\lambda)$ itself because $X(\lambda)$ need not be integer valued and may change continuously in λ even if $N(\lambda)$ does not. The main contribution of this article is to introduce and analyze two constructions that develop this idea.

For each λ in some parameter domain Λ, (1) determines the distribution of $X(\lambda)$ once the distributions of $N(\lambda)$ and the ξ_i are specified. However, we are free to vary the joint distribution of, say, $X(\lambda)$ and $X(\lambda-\Delta)$, so long as we respect the constraint on the marginals. Different constructions impose different joint distributions and lead to potentially different values of $dX(\lambda)/d\lambda$.

Of our two constructions, one applies only to compound Poisson sums and the other applies more generally. For both, we give conditions ensuring that the resulting pathwise derivative is unbiased, in the sense that

$$\mathbb{E}\left[\Phi'(X(\lambda))\frac{d}{d\lambda}X(\lambda)\right] = \frac{d}{d\lambda}\mathbb{E}[\Phi(X(\lambda))]. \tag{2}$$

The compound Poisson construction is globally continuous, in the sense that, for a fixed interval Λ, $X(\cdot)$ is almost surely continuous on Λ. Our other construction is only locally continuous: the interval over which $X(\cdot)$ is continuous in λ is stochastic.

The locally continuous feature of this construction makes this an interesting example within the broader literature on simulation sensitivity estimation. A general (and generally sound) rule of thumb states that pathwise derivatives are unbiased estimators of derivatives of expectations (as in (2)) for globally continuous constructions. It has been recognized since at least Cao [6] that this is more than what is strictly necessary — it should (nearly) suffice for the probability of a discontinuity in a neighborhood of λ of length Δ to be $o(\Delta)$. Our locally continuous construction relies on this observation; we know of no previous examples in which this type of weaker condition arises naturally. Through an analysis on the mean squared error, we also show how the estimator degrades as the construction becomes nearly discontinuous.

The rest of this article is organized as follows. In Section 2, we present our motivating applications. Section 3 covers the locally continuous construction, and Section 4 demonstrates its application. Section 5 presents the globally continuous construction for compound Poisson sums.

2 Motivating Applications

2.1 Lévy Processes

A wide class of models of asset prices admits a representation of the form

$$S_t = S_0 \exp(at + X_t), \quad t \geq 0,$$

with $X = \{X_t\}_{t\geq 0}$ a Lévy process, $X_0 = 0$. Here, $S = \{S_t\}_{t\geq 0}$ might represent the price process of a stock, for example, and S_0 and a are constants. In the most familiar model of this type, X is simply a Brownian motion. Other examples include the variance gamma (VG) model (Madan et al. [11]) and the normal inverse Gaussian (NIG) model (Barndorff-Nielsen [3]). Simulation of the VG model is studied in Avramidis and L'Ecuyer [2].

A Lévy process X has stationary independent increments and is continuous in probability. Its law is determined by its characteristic function at any time $t > 0$, which, according to the Lévy-Itô decomposition, must have the form

$$\mathbb{E}[\exp(i\omega X_t)] = \exp\left(t\left(i\omega b - \frac{\sigma}{2}\omega^2 + \int_{\mathbb{R}} (e^{i\omega y} - 1 - i\omega y \mathbf{1}_{|y|\leq 1}) q(dy)\right)\right),$$

for some constants $\sigma > 0$ and b and measure q on $\mathbb{R}$. We will suppose that the Lévy measure admits a density, so $q(dy) = q(y)dy$. Loosely speaking, this representation decomposes X into the sum of a drift bt, a Brownian motion $\sigma W(t)$, and a jump term independent of W and described by q. If q has finite mass ν, then the jump term is a compound Poisson process with arrival rate ν and jumps drawn from the probability

density $q(\cdot)/\nu$. One may also write this as the difference of two compound Poisson processes, one recording positive jumps, the other recording negative jumps.

If q has infinite mass, then we may still interpret a finite $q(A)$, $A \subseteq \mathbb{R}$ as the arrival rate of jumps of size in A, but the total arrival rate of jumps is infinite. The VG and NIG models are of this type, and they take $\sigma = 0$, so the Brownian component is absent — these are pure-jump processes. For purposes of simulation, one may truncate q to a finite measure and then simulate a compound Poisson process as an approximation; see, for example, Asmussen and Glynn [1], Chapter XII, for a discussion of this idea.

If the original density q depends on a parameter λ, one may be interested in estimating sensitivities of expectations with respect to λ. In choosing how to approximate the original Lévy process with a compound Poisson process, we have a great deal of flexibility in specifying how the approximation should depend on λ. If, for example, we truncate all jumps of magnitude less then ϵ, for some $\epsilon > 0$, then the arrival rate of jumps in the compound Poisson approximation will, in general, depend on λ, because the mass of $q \equiv q_\lambda$ outside the interval $(-\epsilon, \epsilon)$ may vary with λ. This puts us in the context of (1). Glasserman and Liu [9] develop alternative methods that avoid putting parametric dependence in N by, for example, letting ϵ vary with λ, precisely to get around the problem of discontinuities. The constructions developed here deal directly with the possibility of discontinuities.

2.2 Stochastic Volatility and Squared Bessel Bridges

A δ-dimensional squared Bessel process V_t is defined by the stochastic differential equation

$$dV_t = \delta dt + 2\sqrt{V_t} dW_t, \quad \delta > 0.$$

(See Chapter XI of Revuz and Yor [12] for more details about squared Bessel processes.) This is, after a time and scale transformation, the equation for the variance process in the Heston stochastic volatility model. Through the method of Broadie and Kaya [5], exact simulation of the Heston model can be reduced to sampling from the distribution of the area under a path of the squared Bessel bridge, $\left(\int_0^1 V_t\, dt | V_0 = v_0, V_1 = v_1\right)$. In Glasserman and Kim [8], we derive the representation

$$\left(\int_0^1 V_t\, dt | V_0 = v_0, V_1 = v_1\right) \stackrel{d}{=} Y_1 + Y_2 + Y_3$$

where

$$Y_1 = \sum_{n=1}^{\infty} \frac{2}{\pi^2 n^2} \sum_{j=1}^{N_n(v_0+v_1)} \Gamma_{n,j}(1,1), \quad Y_2 = \sum_{n=1}^{\infty} \frac{2}{\pi^2 n^2} \Gamma_n(\delta/2, 1), \quad Y_3 = \sum_{j=1}^{\eta} Z_j.$$

Here, the $N_n(v_0 + v_1)$ are independent Poisson random variables, each with mean $v_0 + v_1$; $\Gamma_{n,j}(1,1)$, $\Gamma_n(\delta/2, 1)$ are independent gamma random variables with shape

parameters 1 and $\delta/2$, respectively, and scale parameter 1. Also, η has a Bessel distribution with parameters $\nu := \delta/2 - 1$ and $z := \sqrt{v_0 v_1}$, denoted by $BES(\nu, z)$. This is a non-negative integer valued distribution with probability mass function

$$b_n(\nu, z) := \mathbb{P}(\eta = n) = \frac{(z/2)^{2n+\nu}}{I_\nu(z) n! \Gamma(n + \nu + 1)}, \quad n \geq 0$$

where $I_\nu(z)$ is the modified Bessel function of the first kind and $\Gamma(\cdot)$ is the gamma function. Lastly, the random variables Z_j are independent with the following Laplace transform:

$$\mathbb{E}e^{-\theta Z} = \left(\frac{\sqrt{2\theta}}{\sinh \sqrt{2\theta}} \right)^2.$$

In [8], we simulate Y_1 and Y_2 by truncating their series expansions and approximating the remainders with gamma random variables matching the first two moments of the truncated terms. For Y_3, we need to simulate the Z_j and η. To generate the Z_j, we numerically invert their Laplace transform to tabulate their distribution and then sample from the tabulated values. Rejection methods for sampling from the Bessel distribution are investigated in Devroye [7]; however, for our constructions we sample by inversion, recursively calculating the probability mass function using

$$b_{n+1}(\nu, z) = \frac{(z/2)^2}{(n+1)(n+\nu+1)} b_n(\nu, z).$$

This method leaves us with two compound sums — a Poisson sum in Y_1 and a Bessel sum in Y_3. It is significant that the parameters of the Poisson and Bessel random variables depend on the endpoints v_0, v_1. In the stochastic volatility application, hedging volatility risk entails calculating sensitivities with respect to the level of volatility (or variance), and this puts us back in the setting of estimating the sensitivity of a compound sum. We return to this application in Section 4.

3 Locally Continuous Construction

In this section, we construct a family of compound sums $X(\lambda)$, $\lambda \in \Lambda$, to be locally continuous in λ. If $N(\lambda)$ has nontrivial dependence on λ, a sufficiently large change in λ will introduce a discontinuity in $N(\lambda)$. Our construction preserves continuity of $X(\lambda)$ across the first (but only the first) discontinuity in $N(\lambda)$.

We use $p_n(\lambda)$ to denote $\mathbb{P}(N(\lambda) \leq n)$. In this section, we assume that the $p_n(\lambda)$ are all monotone in λ in the same direction; to be concrete, we assume they are all decreasing,

$$p_n(\lambda - \triangle) > p_n(\lambda), \quad \forall n, \lambda, \triangle > 0,$$

as in the case of the Poisson distribution, for which we have

$$\frac{d}{d\lambda} p_n(\lambda) = -\frac{e^{-\lambda}\lambda^n}{n!} < 0.$$

As is customary in the simulation context, our construction starts from an infinite sequence of independent uniform random variables. The key step lies in the generation of the last summand $\xi_{N(\lambda)}$. The construction at a pair of points λ_0 and $\lambda < \lambda_0$ is illustrated in Figure 1. We generate a uniform U and sample $N(\lambda_0)$ by inversion to get

$$N(\lambda_0) = n \quad \text{iff} \quad p_{n-1}(\lambda_0) < U \le p_n(\lambda_0),$$

with the convention that $p_{-1}(\cdot) = 0$. If λ is sufficiently close to λ_0, then we also have $N(\lambda) = N(\lambda_0) = n$. On this event, we construct a conditionally independent uniform random variable

$$V = \frac{U - p_{n-1}(\lambda)}{p_n(\lambda_0) - p_{n-1}(\lambda)}, \tag{3}$$

and we map V to $\xi_{N(\lambda)}$ using the inverse of the distribution F of ξ, as illustrated in the figure. (A different approach using a similar distance from the boundary is reviewed in [1], pp.233-234.) As λ decreases, it eventually reaches a point λ^* at which $U = p_{n-1}(\lambda^*)$ and at which $N(\lambda)$ drops to $N(\lambda_0) - 1$. Just at this point, we have $\xi_{N(\lambda)} = 0$, provided $F(0) = 0$; thus, $X(\lambda)$ remains continuous at the discontinuity in $N(\lambda)$. On the event $\{N(\lambda) < N(\lambda_0)\}$, we then generate $\xi_{N(\lambda)}$ from an independent uniform without using V or the uniform U used to generate $N(\lambda_0)$. We combine these steps in the algorithm below.

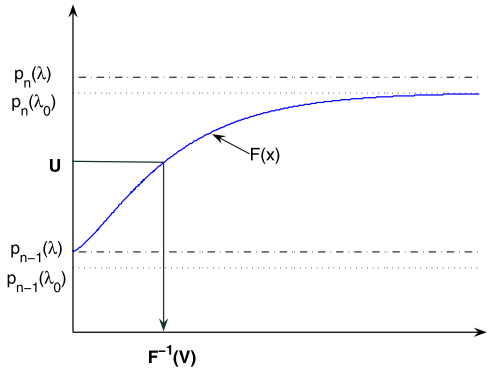


Fig. 1 Illustration of the locally continuous construction.

Algorithm: Locally Continuous Construction

- Generate a uniform random variable U and determine $N(\lambda_0) = n$.
- Generate uniform random variables $U_1, \ldots, U_{n-1}$.

- For $\lambda < \lambda_0$, calculate $N(\lambda)$ using U.
- If $N(\lambda) = n$, then $X = \sum_{i=1}^{n-1} F^{-1}(U_i) + F^{-1}(V)$.
- Otherwise, $X = \sum_{i=1}^{N(\lambda)} F^{-1}(U_i)$.

Example 1. Suppose $N(\lambda)$ is a Poisson random variable with mean λ and the ξ_i are unit-mean exponential random variables. The left panel of Figure 2 shows a sample path of this construction as a function of λ (for fixed U_i) with $\lambda_0 = 12$. As we decrease λ from 12, the first drop in N occurs near 11.5, but X changes continuously across that point. The next discontinuity in N occurs near 10.5 and there X is discontinuous as well. The probability that X has a discontinuity in $(\lambda_0 - \Delta, \lambda_0)$ is the probability of two or more Poisson events in this interval and is therefore $o(\Delta)$. The right panel of the figure shows the average over 10 independent paths and illustrates the smooth dependence on λ in a left neighborhood of $\lambda_0 = 12$.

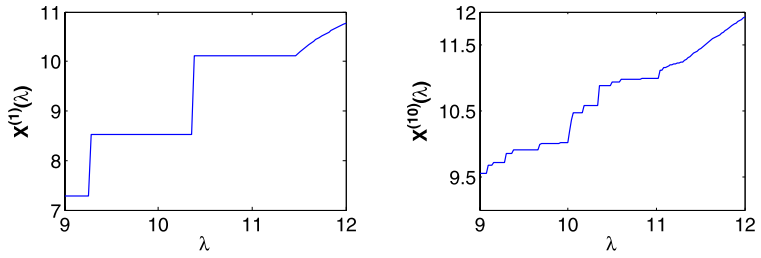


Fig. 2 A single path $X^{(1)}(\lambda)$ (left) and the average of 10 paths $X^{(10)}(\lambda)$ (right) using the locally continuous construction near $\lambda_0 = 12$.

To ensure that this construction leads to unbiased derivative estimates, we need some conditions:

Assumption 1. *All $p_n(\lambda)$ are monotone in λ in the same direction, and $\mathbb{E}N(\lambda) < \infty$.*

Assumption 2. *Each $p_n(\lambda)$ is differentiable in λ and $\sum_{n=0}^{\infty} |(d/d\lambda)p_n(\lambda)|$ is uniformly convergent on compact intervals.*

This assumption about uniform convergence allows for the application of an elementary convergence theorem to conclude that

$$\frac{d}{d\lambda}\mathbb{E}N(\lambda) = \sum_{n=1}^{\infty}\left(-\frac{dp_n(\lambda)}{d\lambda}\right).$$

For the rest of this section, we assume that $(d/d\lambda)p_n(\lambda) \le 0$.

Proposition 1. *Suppose that Assumptions 1–2 hold. Also, suppose that the distribution F of the ξ_i has finite mean, has $F(0) = 0$, and has a density f that is continuous*

and positive on $(0,\infty)$. *Let* Φ *be Lipschitz continuous on* $[0,\infty)$. *Then under the locally continuous construction for* $X(\lambda)$, *we have*

$$\frac{d}{d\lambda}\mathbb{E}\Big[\Phi(X(\lambda))\Big] = \mathbb{E}\Big[\frac{d}{d\lambda}\Phi(X(\lambda))\Big].$$

Proof. There exists $M > 0$ such that $|\Phi(x) - \Phi(y)| \le M \cdot |x - y|$, for all $x, y \ge 0$. Define a sequence of functions $\{g_\alpha\}$ with $\alpha = \lambda_0 - \lambda$ by

$$g_\alpha = \frac{X(\lambda_0) - X(\lambda)}{\lambda_0 - \lambda}$$

which is non-negative by construction. Then,

$$\Big|\frac{\Phi(X(\lambda_0)) - \Phi(X(\lambda))}{\lambda_0 - \lambda}\Big| \le M \cdot g_\alpha.$$

We will prove the result by applying the generalized dominated convergence theorem to the left hand side of this inequality indexed by α. For this, we need to show that $\lim_{\alpha \downarrow 0} g_\alpha = g$ for some function g a.s. and that $\lim_{\alpha \downarrow 0} \mathbb{E}g_\alpha = \mathbb{E}g$.

For almost surely any U, there is a left neighborhood of λ_0 such that $N(\lambda_0) = N(\lambda)$ for all λ in this interval. If $N(\lambda_0) = 0$, then we have nothing to prove. If $N(\lambda_0) = n > 0$, then it is straightforward to see

$$g_\alpha = \frac{F^{-1}(V_0) - F^{-1}(V)}{\lambda_0 - \lambda} \to g$$

as α goes to zero, where

$$V_0 = \frac{U - p_{n-1}(\lambda_0)}{p_n(\lambda_0) - p_{n-1}(\lambda_0)}, \quad V = \frac{U - p_{n-1}(\lambda)}{p_n(\lambda_0) - p_{n-1}(\lambda)}$$

and

$$g = \frac{1 - V_0}{f(F^{-1}(V_0))} \cdot \frac{1}{p_n(\lambda_0) - p_{n-1}(\lambda_0)} \cdot \left(-\frac{dp_{n-1}}{d\lambda}(\lambda_0)\right).$$

To show $\mathbb{E}g_\alpha \to \mathbb{E}g$, we compute as follows:

$$\begin{aligned}\mathbb{E}g_\alpha &= \frac{\mathbb{E}X(\lambda_0) - \mathbb{E}X(\lambda)}{\lambda_0 - \lambda} = \frac{\mathbb{E}N(\lambda_0) - \mathbb{E}N(\lambda)}{\lambda_0 - \lambda} \cdot \mathbb{E}\xi \\ &\to \mathbb{E}\xi \cdot \frac{d\mathbb{E}N(\lambda)}{d\lambda}|_{\lambda=\lambda_0} = \mathbb{E}\xi \cdot \left(-\sum_{j=0}^{\infty} \frac{dp_j}{d\lambda}(\lambda_0)\right).\end{aligned}$$

On the other hand,

$$\mathbb{E}g = \sum_{n=1}^{\infty} \mathbb{E}\left[\frac{1 - V_0}{f(F^{-1}(V_0))}|N(\lambda_0) = n\right]\left(-\frac{dp_{n-1}}{d\lambda}(\lambda_0)\right)$$

$$= \sum_{j=0}^{\infty} \left(-\frac{dp_j}{d\lambda}(\lambda_0) \right) \int_0^{\infty} \frac{1-F(x)}{f(x)} \cdot f(x)dx, \quad (\text{with } x = F^{-1}(V_0))$$

$$-2pt = \lim_{\alpha} \mathbb{E} g_{\alpha}$$

because V_0 is uniform in [0,1] given $N(\lambda_0) = n$. □

The analysis and figures above show how the locally continuous construction smoothes the potential discontinuity in $X(\lambda)$ at the first discontinuity of $N(\lambda)$ by arranging to have $\xi_{N(\lambda)}$ small near the discontinuity. This would not be possible if the support of F were bounded away from zero, which may raise a question about the effectiveness of the construction, because F could be nearly flat near zero, and F^{-1} thus nearly discontinuous near zero. A closer examination suggests that if F is nearly flat near zero, then the variance of $X(\lambda_0) - X(\lambda)$ may be very large. More precisely, we will show this through an analysis on the mean squared difference. For this result, we first need an additional assumption:

Assumption 3. *The continuous and positive density function $f(x)$ is monotone on $(0, x_l) \cup (x_u, \infty)$ for some $0 < x_l < x_u$.*

Proposition 2. *Suppose that $f(x) \sim cx^{1/\beta - 1}$ near zero for $0 < \beta \leq 1$ and that F has finite variance. Then, under Assumptions 1–3, we have*

$$\mathbb{E}\Big[\big(X(\lambda_0) - X(\lambda) \big)^2 \Big] = \sum_{k=2}^{\infty} \Big(k\mathrm{Var}(\xi) + k^2 (\mathbb{E}\xi)^2 - a_k \Big) \mathbb{P}\Big(N(\lambda_0) - N(\lambda) = k \Big)$$
$$+ \sum_{k=0}^{\infty} \psi(b_k) \mathbb{P}\Big(N(\lambda_0) = k+1 \Big)$$

where $a_k \in [0, \mathrm{Var}(\xi) + (2k-1)(\mathbb{E}\xi)^2]$, $b_k = (p_k(\lambda) - p_k(\lambda_0))/(p_{k+1}(\lambda_0) - p_k(\lambda_0))$ and $\psi(b) = O(b^{(1+2\beta)\wedge 2})$.

Proof (a sketch). We first note that the assumption on the behavior of f near zero implies

$$F(x) \sim c\beta x^{1/\beta}, \quad F^{-1}(x) \sim (x/c\beta)^{\beta}.$$

Let us write the mean squared difference of the left side as

$$\mathbb{E}\big[(X(\lambda_0) - X(\lambda))^2 \big] = \sum_{n=2}^{\infty} \mathbb{E}\left[\Big(X(\lambda_0) - X(\lambda) \Big)^2 ;\, N(\lambda_0) - N(\lambda) = n \right]$$
$$+ \mathbb{E}\left[F^{-1}(V_0)^2 ;\, N(\lambda_0) - N(\lambda) = 1 \right]$$
$$+ \mathbb{E}\left[\Big(F^{-1}(V_0) - F^{-1}(V) \Big)^2 ;\, N(\lambda_0) - N(\lambda) = 0 \right]$$

where V_0, V are same as defined in Proposition 1. Let us take a look at each term on the right side.

On the event $\{N(\lambda_0) - N(\lambda) = n,\, N(\lambda) = k\}$, the random variable U is constrained to be uniform in $[p_{n+k-1}(\lambda_0),\, p_{n+k}(\lambda_0)] \cap [p_{k-1}(\lambda),\, p_k(\lambda)]$. Therefore, the

first term equals

$$\sum_{n=2}^{\infty}\mathbb{E}\left[\left(\sum_{i=1}^{n-1}\xi_i+F^{-1}(V_0)\right)^2|N(\lambda_0)-N(\lambda)=n\right]\mathbb{P}(N(\lambda_0)-N(\lambda)=n)$$

and the expectation in the expression is between $\mathbb{E}[(\sum_{i=1}^{n-1}\xi_i)^2]$ and $\mathbb{E}[(\sum_{i=1}^{n}\xi_i)^2]$. From these bounds, it is easy to get the first term in the statement.

As for the second term, straightforward computations yield

$$\begin{aligned}&\mathbb{E}\left[F^{-1}(V_0)^2;\,N(\lambda_0)-N(\lambda)=1\right]\\&\le\sum_{k=0}^{\infty}(p_{k+1}(\lambda_0)-p_k(\lambda_0))\int_0^{1\wedge b_k}F^{-1}(x)^2dx\\&\le\sum_{k=0}^{\infty}(p_{k+1}(\lambda_0)-p_k(\lambda_0))\mathbb{E}(\xi^2)\wedge\left(F^{-1}(b_k)^2b_k\right)\end{aligned}$$

with $b_k=(p_k(\lambda)-p_k(\lambda_0))/(p_{k+1}(\lambda_0)-p_k(\lambda_0))$. Note that $F^{-1}(b)^2b=O(b^{1+2\beta})$.

Similarly, we compute

$$\mathbb{E}\left[\left(F^{-1}(V_0)-F^{-1}(V)\right)^2;\,N(\lambda_0)=N(\lambda)\right]=\sum_{k=0}^{\infty}(p_{k+1}(\lambda_0)-p_k(\lambda_0))\phi(b_k)$$

with $\phi(b)=(1-b)^+\int_0^1\left(F^{-1}(x+(1-x)b)-F^{-1}(x)\right)^2dx$. We derive upper and lower bounds for the integral in $\phi(b)$. If they are $O(b^{(1+2\beta)\wedge 2})$, then the proof is complete.

Let us denote $F(x_l)$, $F(x_u)$ by l and u. Without loss of generality, we assume that $l<1/2$ and that f is monotone on $(0,l+\epsilon)$ for a small positive real number ϵ. Then, for a sufficiently small $0<b\le\epsilon/(1-l)$, we have

$$\begin{aligned}&\int_0^l\left(F^{-1}(x+(1-x)b)-F^{-1}(x)\right)^2dx\\&\le\int_0^b F^{-1}(2b)^2dx+\int_b^l\left(\int_x^{x+(1-x)b}\frac{1}{f(F^{-1}(y))}dy\right)^2dx\\&\le bF^{-1}(2b)^2+\int_b^l\frac{(1-x)^2b^2}{f(F^{-1}(x))^2}dx.\end{aligned}$$

where we used the monotonicity of f. Due to the assumption $f(x)\sim cx^{1/\beta-1}$ near zero, we can find small $\delta,\epsilon'>0$ such that $(1-\epsilon')f(x)<cx^{1/\beta-1}<(1+\epsilon')f(x)$ whenever $x\le\delta$. Therefore, as long as $F^{-1}(b)\le\delta$, the last expression becomes less than

$$O(b^2)+(1+\epsilon')\int_{F^{-1}(b)}^{\delta}\frac{b^2}{cx^{1/\beta-1}}dx=O(b^2)+O(b^{1+2\beta})=O(b^{(1+2\beta)\wedge 2}).$$

In a similar fashion, we can show that there is a lower bound of order $O(b^{(1+2\beta)\wedge 2})$. Note that this bound acts as a lower bound for the entire integral over $(0,1)$ instead of $(0,l)$.

For the integral over $(l,1)$, we observe that

$$\int_l^u \Big(F^{-1}(x+(1-x)b)-F^{-1}(x)\Big)^2 dx = O(b^2)$$

and that

$$\int_u^1 \Big(F^{-1}(x+(1-x)b)-F^{-1}(x)\Big)^2 dx \le \frac{b^2}{(1-b)^3}\int_{x_u}^\infty \frac{\overline{F}(x)^2}{f(x)}dx,$$

which can be shown by a similar argument as above using the monotonicity of f. Also, it is not hard to see that the last integral is finite. □

Proposition 2 gives additional information beyond the unbiasedness of the derivative estimator. For comparison, we examine the mean square difference under a standard construction in which $N(\lambda)$ is generated by inversion (using common random numbers at different values of λ) and the same ξ_j are used at all values of λ. Under this construction, we have

$$\begin{aligned}
&\mathbb{E}\Big[\big(X(\lambda_0)-X(\lambda)\big)^2\Big] \\
&= \sum_{k=2}^\infty \Big(k\mathrm{Var}(\xi)+k^2(\mathbb{E}\xi)^2\Big)\mathbb{P}\Big(N(\lambda_0)-N(\lambda)=k\Big) \\
&\quad + \sum_{k=0}^\infty \mathbb{E}\big[\xi^2\big]\mathbb{P}\Big(N(\lambda)=k|N(\lambda_0)=k+1\Big)\mathbb{P}\Big(N(\lambda_0)=k+1\Big).
\end{aligned}$$

The conditional probability in the second term is only of order $O(b_k)$. Comparing this with the corresponding summation in Proposition 2 term by term, we note that if $F^{-1}(x)\sim x^\beta$ near zero for very small β (F being nearly flat), then $\psi(b_k)$ in Proposition 2 is close to $O(b_k)$. Thus, the smoothness of the locally continuous construction degrades to that of the standard construction as F^{-1} becomes nearly discontinuous at zero.

For example, let us consider the case of Poisson $N(\lambda)$ and $\xi_i = \Gamma_i(k,1)$ having a gamma distribution with shape parameter k and scale parameter 1. Figure 3 shows sample paths of X for $\lambda_0 = 12$, $k = 3, 30$, and the averages of 10 sample paths. We observe that the $k = 30$ case (for which F is very flat near zero) does not benefit from the locally continuous construction as much as the $k = 3$ case.

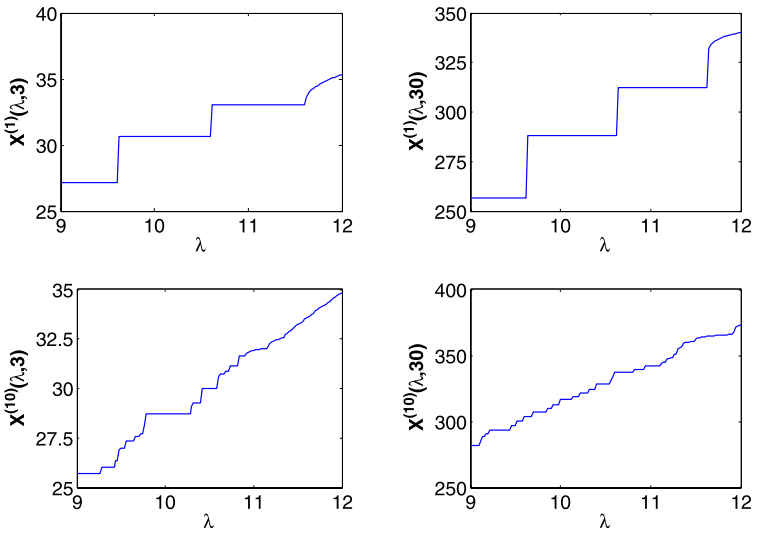


Fig. 3 Locally continuous construction of Poisson sum of $\Gamma(k,1)$ random variables. The figures show individual paths $X^{(1)}(\lambda,k)$ (top) and averages over ten paths $X^{(10)}(\lambda,k)$ (bottom) for $k=3$ (left) and $k=30$ (right).

4 Application to the Squared Bessel Bridge

We now return to the example of Section 2.2 to show that it falls within the scope of our unbiasedness result. In particular, we focus on the compound Bessel sum in Y_3.

Proposition 3. *The Bessel distribution satisfies Assumptions 1–2.*

Proof. As observed by Yuan and Kalbfleisch [13], $\mathbb{E}\eta = zI_{\nu+1}(z)/(2I_\nu(z))$ and this is finite and differentiable; they also provide an expression for $\mathbb{E}\eta^2$. Next, recall the series representation of $I_\nu(z)$ with $\nu > -1$, $z > 0$:

$$I_\nu(z) = \sum_{k=0}^{\infty} \frac{(z/2)^{2k+\nu}}{k!\Gamma(k+1+\nu)}.$$

This is uniformly convergent on compact intervals of ν or z, so we can differentiate $I_\nu(z)$ with respect to either variable and get

$$\frac{(\partial/\partial z)I_\nu(z)}{I_\nu(z)} = \sum_{k=0}^{\infty} \frac{2k+\nu}{z} b_k(\nu,z), \tag{4}$$

$$\frac{(\partial/\partial \nu)I_\nu(z)}{I_\nu(z)} = \sum_{k=0}^{\infty} \left\{ \log\frac{z}{2} - \frac{\Gamma'(k+1+\nu)}{\Gamma(k+1+\nu)} \right\} b_k(\nu,z), \tag{5}$$

because the series in (4)–(5) are uniformly convergent on compact sets.

Direct computations yield

$$\frac{\partial}{\partial z}\mathbb{P}(\eta \le n) = \frac{\partial}{\partial z}\sum_{j=0}^{n} b_j(\nu, z) = \sum_{j=0}^{n}\left\{\frac{2j+\nu}{z} - \frac{(\partial/\partial z)I_\nu(z)}{I_\nu(z)}\right\} b_j(\nu, z)$$
$$= \frac{2}{z}\sum_{j=0}^{n}\sum_{k=n+1}^{\infty}(j-k)b_j(\nu, z)b_k(\nu, z) < 0,$$

where the last equality comes from (4) and $\sum_{j=0}^{n}\sum_{k=0}^{n}(j-k)b_j b_k = 0$. In a similar way, we have

$$\frac{\partial}{\partial \nu}\mathbb{P}(\eta \le n) = \sum_{j=0}^{n}\left\{\log\frac{z}{2} - \frac{(\partial/\partial \nu)I_\nu(z)}{I_\nu(z)} - \frac{\Gamma'(j+1+\nu)}{\Gamma(j+1+\nu)}\right\} b_j(\nu, z)$$
$$= \sum_{j=0}^{n}\sum_{k=n+1}^{\infty}\left\{\frac{\Gamma'(k+1+\nu)}{\Gamma(k+1+\nu)} - \frac{\Gamma'(j+1+\nu)}{\Gamma(j+1+\nu)}\right\} b_j(\nu, z)b_k(\nu, z) > 0.$$

In the last inequality, we used the relationship $\Gamma(z+1) = z\Gamma(z)$ so that

$$\frac{\Gamma'(z+1)}{\Gamma(z+1)} = \frac{1}{z} + \frac{\Gamma'(z)}{\Gamma(z)}.$$

To prove the uniform convergence of $\sum_{n=0}^{\infty}|(\partial/\partial z)\mathbb{P}(\eta \le n)|$, we note that, with $\mu := \mathbb{E}\eta$,

$$\sum_{n=0}^{\infty}\left(-\frac{\partial\mathbb{P}(\eta \le n)}{\partial z}\right) = \frac{2}{z}\sum_{n=0}^{\infty}\sum_{j=0}^{n}\sum_{k=0}^{\infty}(k-j)b_j b_k = \frac{2}{z}\sum_{n=0}^{\infty}\sum_{j=0}^{n} b_j(\mu - j)$$
$$= \frac{2}{z}\sum_{n=0}^{\infty}\left(-\mu\mathbb{P}(\eta > n) + \sum_{j=n+1}^{\infty} j b_j\right) = \frac{2}{z}\left(-\mu^2 + \sum_{j=0}^{\infty} j^2 b_j\right)$$

and the last term converges to $2\mathrm{Var}(\eta)/z$ uniformly. For the other partial derivative, we show that $\sum_{n=N+1}^{\infty}(\partial/\partial \nu)\mathbb{P}(\eta \le n)$ is uniformly bounded as follows:

$$\sum_{n=N+1}^{\infty}\frac{\partial\mathbb{P}(\eta \le n)}{\partial \nu} \le \sum_{n=N+1}^{\infty}\sum_{j=0}^{n}\sum_{k=n+1}^{\infty} k b_j b_k \le \sum_{k=N+2}^{\infty}(k-N-1)k b_k.$$

Here we used $b_j \le 1$ and the last sum can be made arbitrarily small on compact intervals of ν by choosing a sufficiently large N. □

The application of the locally continuous construction to Y_3 is illustrated in Figure 4. The top two figures show averages over 100 and 1000 paths as ν ranges over the interval $[-0.5, 0.5]$. The bottom two figures show averages over the same number of paths but using a standard construction in which separate streams of uniform random variables are used to generate η and the summands Z_j. Under the standard construction, every change in the value of η introduces a discontinuity in Y_3; the figures illustrate the smoothing effect of the locally continuous construction.

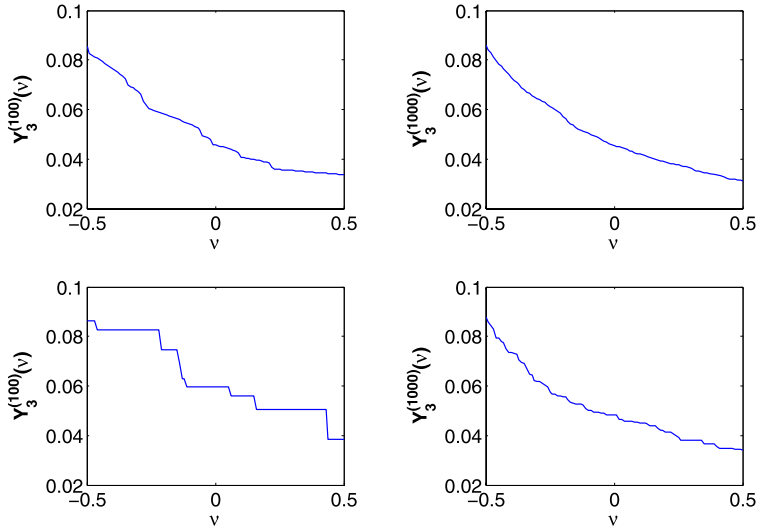


Fig. 4 Averaged paths of Y_3 in $\nu \in [-0.5, 0.5]$ using the locally continuous construction (top) and a standard construction (bottom). The methods are averaged over 100 paths $Y_3^{(100)}(\nu)$ (left) and 1000 paths $Y_3^{(1000)}(\nu)$ (right).

5 Globally Continuous Construction

In this section, we use special properties of the Poisson distribution to develop a *globally* continuous construction of $X(\lambda)$ when $N(\lambda)$ is Poisson with mean λ. The Poisson case is of particular interest in many applications, including the Lévy process simulation discussed in Section 2.1.

In a slight change of notation, we now use N to denote a unit-mean Poisson process. Let $T_1 < T_2 < \cdots$ be the jump times of N, and read $N(\lambda)$ as the number of T_j that fall in $[0, \lambda]$. We will use the spacing between $T_{N(\lambda)}$ and λ to generate $\xi_{N(\lambda)}$ in order to make $\xi_{N(\lambda)} = 0$ at a discontinuity of $N(\lambda)$.

In more detail, on the event $\{N(\lambda) = n\}$, set

$$R_n = T_n/\lambda,\ R_{n-1} = T_{n-1}/T_n,\ \ldots,\ R_1 = T_1/T_2.$$

Then, from standard properties of Poisson processes, it is straightforward to check that the R_j are conditionally independent given $\{N(\lambda) = n\}$ and that they have beta distributions with parameters j and 1, for $j = 1, \ldots, n$. This conditional distribution is simply x^j; we denote it by $F_{j,n}$. Now transform the variables to get

$$\xi_j = F^{-1}(1 - U_j), \quad U_j = F_{j,n}(R_j), \quad j = 1, \ldots, n,$$

and set $X = \sum_{j=1}^{N(\lambda)} \xi_j$. A discontinuity in $N(\lambda)$ occurs when λ crosses some T_n. Just at the point at which $\lambda = T_n$, we get $R_n = 1$ and $U_n = 1$ and therefore $\xi_n = 0$, provided $F^{-1}(0) = 0$.

Algorithm: Globally Continuous Construction

- Set a maximum λ_0
- Generate unit-mean Poisson arrival times $T_1, T_2, \ldots$ until $T_n \le \lambda_0 < T_{n+1}$
- For each $\lambda \le \lambda_0$, determine m such that $T_m \le \lambda < T_{m+1}$ and calculate $R_1, \ldots, R_m$.
- Set $X(\lambda) = \sum_{j=1}^{m} F^{-1}(1 - F_{j,m}(R_j))$

To illustrate, we repeat the example of Figure 3 using the new construction in Figure 5. The global continuity is evident in the figures. In the case $k = 30$ (on the right), the discontinuities are replaced with points of steep increase, again because of the behavior of the distribution F near zero.

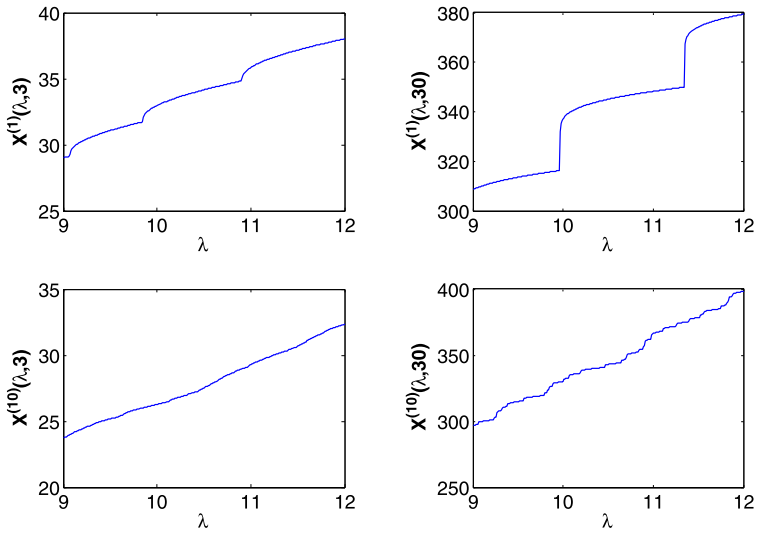


Fig. 5 Globally continuous construction of Poisson sum of $\Gamma(k, 1)$ random variables. The figures show individual paths $X^{(1)}(\lambda, k)$ (top) and averages over ten paths $X^{(10)}(\lambda, k)$ (bottom) for $k = 3$ (left) and $k = 30$ (right).

This construction yields unbiased derivative estimators. The Poisson distribution satisfies Assumptions 1–2.

Proposition 4. *Suppose that the distribution F of the ξ_i has finite mean, has $F(0) = 0$, and has a density f that is continuous and positive on $(0, \infty)$. Let Φ be Lipschitz continuous on $[0, \infty)$. Then, under the globally continuous construction for $X(\lambda)$, we have*

$$\frac{d}{d\lambda}\mathbb{E}\Big[\Phi(X(\lambda))\Big] = \mathbb{E}\Big[\frac{d}{d\lambda}\Phi(X(\lambda))\Big].$$

Proof. We proceed as in Proposition 1 with the obvious changes: $V_0 = 1-(T_n/\lambda_0)^n$ and $V = 1-(T_n/\lambda)^n$ given $\{N(\lambda) = N(\lambda_0) = n\}$. Then, straightforward calculations and the generalized dominated convergence theorem give the result. □

6 Concluding Remarks

We have derived unbiased derivative estimators for random sums through constructions that preserve continuity in the sum across discontinuities in the number of summands. The constructions accomplish this by making the last summand zero at the potential discontinuity. Our first construction provides an interesting example yielding an unbiased pathwise estimator despite being only locally continuous. Our second construction is globally continuous but applies only to compound Poisson sums. The compound Poisson case arises in approximating Lévy processes; a compound Bessel sum arises in exact simulation of the Heston stochastic volatility model.

References

1. Asmussen, S., Glynn, P.: Stochastic Simulation: Algorithms and Analysis. Springer, New York (2007)
2. Avramidis, A.N., L'Ecuyer, P.: Efficient Monte Carlo and quasi-Monte Carlo option pricing under the variance-gamma model. Management Science **52**(12), 1930–1994 (2006)
3. Barndorff-Nielsen, O.E.: Processes of normal inverse gaussian type. Finance and Stochastics **2**, 41–68 (1998)
4. Brémaud, P., Vazquez-Abad, F.: On the pathwise computation of derivatives with respect to the rate of a point process: the phantom RPA method. Queueing Systems **10**, 249–270 (1992)
5. Broadie, M., Kaya, O.: Exact simulation of stochastic volatility and other affine jump diffusion processes. Operations Research **54**, 217–231 (2006)
6. Cao, X.: Convergence of parameter sensitivity estimates in a stochastic experiment. IEEE Transactions on Automatic Control **AC-30**(9), 845–853 (1985)
7. Devroye, L.: Simulating Bessel random variables. Statistics & Probability Letters **57**, 249–257 (2002)
8. Glasserman, P., Kim, K.: Gamma expansion of the Heston stochastic volatility model. Finance and Stochastics (2008). To appear
9. Glasserman, P., Liu, Z.: Estimating greeks in simulating Lévy-driven models (2008). Working Paper, Columbia Business School
10. Heston, S.L.: A closed-form solution for options with stochastic volatility with applications to bond and currency options. Review of Financial Studies **13**, 585–625 (1993)
11. Madan, D., Carr, P., Chang, E.: The variance gamma process and option pricing. European Finance Review **2**, (1998)
12. Revuz, D., Yor, M.: Continuous Martingales and Brownian Motion, 3rd edn. Springer-Verlag, New York (1999)
13. Yuan, L., Kalbfleisch, J.D.: On the Bessel distribution and related problems. Annals of the Institute of Statistical Mathematics **52**, 438–447 (2000)

New Perspectives on $(0, s)$-Sequences

Christiane Lemieux and Henri Faure

Abstract Low-discrepancy sequences that have an optimal value of 0 for their *t-parameter* have always been of interest to both theorists and practitioners. However, in practice the Sobol' sequence often performs better than the original $(0, s)$-sequences in prime bases proposed by Faure in 1982, although the former construction does not have an optimal value of 0 for its t-parameter. In this paper, we introduce new ideas that can be used to find improved constructions for $(0, s)$-sequences in prime bases. To do so, we study them within the framework of *generalized Niederreiter sequences*, which was introduced by Tezuka in 1993. We take a closer look at the structure of the corresponding generating matrices, as this helps us to better understand the differences and analogies between the constructions that we are interested in. This study is then used to guide our search for improved $(0, s)$-sequences, which are shown to perform well on a variety of problems.

1 Introduction

Low-discrepancy sequences are the backbone of quasi-Monte Carlo methods. While several families of sequences have been proven to have good theoretical properties, it is often necessary to carefully choose the parameters that determine a given sequence so that the resulting approximations work well in practice. In particular,

Christiane Lemieux
Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Ontario, Canada
e-mail: clemieux@math.uwaterloo.ca

Henri Faure
Institut de Mathématiques de Luminy
and Université Paul Cézanne
Marseille, France
e-mail: faure@iml.univ-mrs.fr

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_7,

$(0, s)$-sequences are very attractive from a theoretical point of view, because the value of the quality parameter t for this construction is 0, which means their equidistribution is optimal. However, when used in practice—for problems of large dimensions and with a relatively small number of points—they may not work so well if their defining parameters are not chosen carefully, which is the case for Faure sequences [3]. Similarly, for the widely used Sobol' sequence, one needs to choose the so-called *direction numbers*.

Our goal in this paper is to explore a few ideas leading to concrete $(0, s)$-sequence constructions that can be used in practice. We also establish comparisons between different families of constructions, namely the Sobol' sequence, the $(0, s)$-sequences, and generalized Halton sequences. To do so, it is helpful to use the framework of *generalized Niederreiter sequences*, which was proposed by Tezuka in [24].

This paper is organized as follows. In Section 2, we provide background information on low-discrepancy constructions that are based on van der Corput sequences, and describe different families to be studied in this paper. We present generalized Niederreiter sequences in Section 3, but use the language of generating matrices to explain this construction. We believe this offers easier-to-grasp concepts to readers unfamiliar with the mathematical tools used in Tezuka's original definitions. In Section 4, we describe two ideas that each result in concrete recommendations for specific $(0, s)$-sequence constructions. Our parameters can be found on the Internet at [27]. In Section 5, we propose to use the underlying idea at the basis of $(0, s)$-sequences to construct sequences that are extensible in the dimension. We show numerical results on a few different problems in Section 6, and conclude in Section 7 with a summary of our findings.

2 Background Information

We start by describing the *generalized van der Corput sequence* [2], which is obtained by choosing a sequence $\Sigma = (\sigma_r)_{r\geq 0}$ of permutations of $Z_b = \{0, 1, \ldots, b-1\}$. Then, the nth term of the sequence is defined as

$$S_b^{\Sigma}(n) = \sum_{r=0}^{\infty} \sigma_r\big(a_r(n)\big) b^{-r-1},$$

where $a_r(n)$ is the rth digit of the b-adic expansion of $n-1 = \sum_{r=0}^{\infty} a_r(n)\, b^r$. If the same permutation σ is used for all digits, (i.e., if $\sigma_r = \sigma$ for all $r \geq 0$), then we use the notation S_b^{σ} to denote S_b^{Σ}. The van der Corput sequence in base b is obtained by taking $\sigma_r = I$ for all $r \geq 0$, where I stands for the identity permutation over Z_b.

Another way of enriching the van der Corput sequence is by applying a linear transformation to the digits $a_0(n), a_1(n), \ldots$ before outputting a number between 0 and 1 (see in chronological order [22], [3] and [17]). We call this a *linearly scrambled* van der Corput sequence. For a prime base b, it is obtained by choosing a matrix C with elements in $\mathbb{Z}_b$ and an infinite number of rows and columns, and then

defining the nth term of this sequence as

$$S_b^C(n) = \sum_{r=0}^{\infty} \frac{x_{n,r}}{b^{r+1}} \quad \text{in which} \quad x_{n,r} = \sum_{k=0}^{\infty} c_{r+1,k+1} a_k(n), \tag{1}$$

where $c_{r,k}$ is the kth element on the rth row of C. Note that the second summation is finite and performed in $\mathbb{Z}_b$, but the first one can be infinite and is performed in $\mathbb{R}$, with the possibility that $x_{n,r} = b-1$ for all but finitely many r.

An important particular case is the one where C is a nonsingular upper triangular (NUT) matrix, in which case the first summation is finite. Of course, we obtain the original van der Corput sequence S_b^I with the identity matrix.

Finally, a van der Corput sequence in base b—either generalized or linearly scrambled— can be *shifted* by choosing a number $v \in [0,1)$ with base b expansion

$$v = \sum_{r=0}^{\infty} v_r b^{-r-1}$$

and then adding v_r (modulo b) to $\sigma_r\big(a_r(n)\big)$, or $x_{n,r}$ in $S_b^{\Sigma}(n)$, or $S_b^C(n)$, respectively.

To construct multidimensional sequences in the unit hypercube $I^s = [0,1)^s$, we present two classical approaches based on the one-dimensional case above. The first one is to juxtapose van der Corput sequences in different bases. This is precisely the idea behind the *Halton sequence* [10], whose nth term is defined as $X_n = (S_{b_1}(n), \ldots, S_{b_s}(n))$, where the b_j's, for $j = 1, \ldots, s$, are pairwise coprime. That is, the jth coordinate is defined using S_{b_j}, the van der Corput sequence in base b_j.

A *generalized Halton sequence* is defined by choosing s sequences of permutations $\Sigma_j = (\sigma_{j,r})_{r\geq 0}$, $j = 1, \ldots, s$. The sequence's nth point $X_n \in I^s$ is given by

$$X_n = (S_{b_1}^{\Sigma_1}(n), \ldots, S_{b_s}^{\Sigma_s}(n)), \; n \geq 1, \tag{2}$$

where the b_j's are pairwise coprime bases. These b_j's are typically chosen as the first s prime numbers. In this case, we denote the jth base as p_j.

Another approach for defining multidimensional sequences is to choose a base b and also s matrices $C_1, \ldots, C_s$ with elements in $\mathbb{Z}_b$ — called *generating matrices* — and then form a sequence by juxtaposing the s linearly scrambled van der Corput sequences $S_b^{C_1}, \ldots, S_b^{C_s}$. This is a special case of the *digital sequences* proposed by Niederreiter in his general construction principles [17, p. 306 and 313], [18, p. 63 and 72]. In this paper, for simplicity we focus on prime bases.

As it was the case for the choice of bases in the Halton sequences, the matrices C_j cannot be arbitrary and must be carefully chosen in order to obtain *low-discrepancy sequences*, which are sequences for which the star discrepancy satisfies $D^*(P_N) \in O((\log N)^s)$. (Several authors have a $1/N$ factor when defining $D^*(P_N)$, for instance [14, 16], but here we use the convention from Number Theory, as in [3, 17].)

It has been proved that both the generalized Halton sequences and several classes of digital sequences are low-discrepancy sequences.

While it is possible to obtain bounds on $D^*(P_N)$, computing this quantity turns out to be extremely difficult. However, if we replace the sup norm by the L_2-norm, we obtain discrepancy measures that can be computed in practice. More precisely, in what follows we work with the *L_2–discrepancy* $T(P_N)$, introduced by Morokoff and Caflisch [16], whose square can be computed as [16, p.1263–1264]

$$T^2(P_N) = \sum_{i=1}^{N}\sum_{j=1}^{N}\prod_{k=1}^{s}(1-\max(X_{i,k},X_{j,k}))\min(X_{i,k},X_{j,k}) \\ -N2^{-s+1}\sum_{i=1}^{N}\prod_{k=1}^{s}X_{i,k}(1-X_{i,k})+N^2 12^{-s}, \tag{3}$$

where $X_{i,k}$ is the kth coordinate of the ith point $X_i \in I^s$ of P_N. Hence $T(P_N)$ can be computed in $O(N^2 s)$. Note that the L_2-discrepancy considers boxes not necessarily anchored at the origin in its definition, by contrast with the star L_2-discrepancy (T^* in [16], but still denoted T by many authors), for which a formula is given in [26].

We now introduce the fundamental quality parameter t that measures the quality of a given low-discrepancy sequence. This is achieved by the general concept of (t,s)-sequences in base b, originally introduced by Niederreiter in [17] to give a general framework for various constructions of s-dimensional low-discrepancy sequences and to obtain further ones. In this framework, smaller values of the integer $t \geq 0$ give smaller discrepancies. In order to adapt to important new constructions, Tezuka [24] and, a bit later, Niederreiter and Xing [19] have been led to generalize the original definition by using the *truncation operator*, defined as follows (see [19, p.271] and the examples below, after Prop.1, where the truncation is required).

Truncation: Let $x = \sum_{i=1}^{\infty} x_i b^{-i}$ be a b–adic expansion of $x \in [0,1]$, with the possibility that $x_i = b-1$ for all but finitely many i. For every integer $m \geq 1$, define $[x]_{b,m} = \sum_{i=1}^{m} x_i b^{-i}$ (depending on x via its expansion).

An *elementary interval in base b* is an interval of the form $[ab^{-d},(a+1)b^{-d})$ with integers a,d such that $d \geq 0$ and $0 \leq a < b^d$.

We first consider the one-dimensional case, where we say that a sequence $(X_n)_{n\geq 1}$ (with prescribed b-adic expansions for each X_n) is a $(t,1)$-*sequence in base b* if for all integers $l \geq 0$, $m \geq t$, every elementary interval E with $\lambda(E) = b^{t-m}$ contains exactly b^t points of the point set $\{[X_n]_{b,m}\,; lb^m+1 \leq n \leq (l+1)b^m\}$, where the notation $\lambda(\cdot)$ refers to the Lebesgue measure. The original definition of $(t,1)$-sequences was the same with X_n instead of $[X_n]_{b,m}$ in the definition of the point set above. These sequences are sometimes called $(t,1)$-sequences in the *narrow sense* and the others just $(t,1)$-sequences [19, p. 271]. In the following proposition, we deal only with the most interesting case of $(0,1)$-sequences. The proof can be found in [7].

Proposition 1. *The two generalizations of van der Corput sequences defined above, S_b^{Σ} and S_b^{C}—where C is such that every left upper $m \times m$ submatrix is nonsingular for all $m \geq 1$—are $(0,1)$-sequences in base b.*

Here, the truncation is required for sequences S_b^{Σ} when $\sigma_r(0) = b-1$ for all sufficiently large r and for sequences S_b^{C} when the matrix C gives digits $x_{n,r} = b-1$ for all sufficiently large r in (1). See [7] for the details.

A typical way to ensure that the condition in the preceding proposition holds is to take C to be an NUT matrix. The corresponding sequence is called an *NUT digital* $(0,1)$*-sequence*. Note that in this case, it is not necessary to resort to the truncation since all summations are finite in the definition.

For multidimensional sequences in base b, the same idea is applied, but now to elementary intervals of the form $E = [a_1/b^{d_1}, (a_1+1)/b^{d_1}) \times \cdots \times [a_s/b^{d_s}, (a_s+1)/b^{d_s})$. A sequence $(X_n)_{n\geq 1}$ (with prescribed b-adic expansions for each coordinate of X_n) is a (t,s)*–sequence in base* b (in the broad sense) if for all integers $l \geq 0$ and $m \geq t$, every elementary interval E with $\lambda(E) = b^{t-m}$ contains exactly b^t points of the point set $\{[X_n]_{b,m}\ ; lb^m + 1 \leq n \leq (l+1)b^m\}$.

For multidimensional sequences defined over different bases, similar definitions exist (see [4]) with sets of bases $B = (b_1, \ldots, b_s)$ and sets of parameters $T = (t_1, \ldots, t_s)$. But unfortunately, so far the only realization of "low-discrepancy" (T,s)-sequences in bases B are generalized Halton sequences for which $T = (0, \ldots, 0)$ and bases from B are pairwise coprime. A blend of Halton and $(0,s)$-sequences was also proposed in [4] to obtain extensible sequences, but without further investigation. New interesting work in this area has been done by Hofer et al. [11, 12] in a more general setting. In particular, they give conditions under which their new sequences are uniformly distributed.

For digital sequences in a given base b, the first construction that was defined so that $t = 0$ was given by Faure in [3]. It is obtained by choosing a prime base $b \geq s$ and using generating matrices C_j given by the $(j-1)$th power of the (upper triangular) Pascal matrix P_b in $\mathbb{Z}_b$.

As shown by Tezuka in [25], if we take

$$C_j = A_j P_b^{j-1}, \qquad j = 1, \ldots, s \tag{4}$$

where each A_j is a nonsingular lower triangular (NLT) matrix, then we also get a $(0,s)$-sequence. This family of constructions is called *generalized Faure sequences* in [25]. Note that the truncation is required for such generalized Faure sequences.

The Sobol' sequences [22] are digital sequences in base 2 for which t is not necessarily 0. In fact, a necessary condition for having $t = 0$ is that we must have $b \geq s$ (see [17, Cor. 5.17]). This construction is not as simple as the one from Faure and requires arithmetic operations on polynomials. The generating matrices are obtained by choosing a primitive polynomial $p_j(z)$ in the ring $\mathbb{F}_2[z]$ of polynomials over the finite field $\mathbb{F}_2$, given by $p_j(z) = z^{d_j} + a_{j,1}z^{d_j-1} + \cdots + a_{j,d_j-1}z + 1$, where each $a_{j,l} \in \mathbb{F}_2$ and d_j is $p_j(z)$'s degree. We then need d_j *direction numbers* of the form $v_{j,r} = m_{j,r}/2^r$, where $m_{j,r}$ is an odd integer between 1 and 2^r for $r = 1, \ldots, d_j$. Once these d_j direction numbers are chosen, the following ones are obtained through the recurrence

$$v_{j,r} = a_{j,1}v_{j,r-1} \oplus \cdots \oplus a_{j,d_j-1}v_{j,r-d_j+1} \oplus v_{j,r-d_j} \oplus (v_{j,r-d_j}/2^{d_j}), \tag{5}$$

where $\oplus$ represents the addition of vectors with components in $\mathbb{F}_2$.

The rth column of C_j is then formed by the base 2 expansion of $v_{j,r}$. That is, each direction number is assigned to a column of C_j and fills it with its binary rep-

resentation. By the definition of the initial vectors $v_{j,1}, \ldots, v_{j,d_j}$ and the recurrence (5) used to obtain the next ones, one can see that each C_j is an NUT matrix. In turn, based on Proposition 1, this implies that each one-dimensional projection of the Sobol' sequence is a $(0,1)$-sequence (as shown also in [22, Remark 3.5]). This also implies that Sobol' sequences do not need truncation in their definition.

In what follows, we want to find ways of selecting simple matrices A_j in (4) so that the resulting constructions are competitive with the Sobol' sequence based on carefully chosen direction numbers. To do so, it is useful to use the framework of generalized Niederreiter sequences, which include these two families of constructions and can therefore help understanding the analogies between them.

3 Framework of Generalized Niederreiter Sequences

This construction from [24] builds digital sequences in a given base b by first choosing s polynomials $p_1(z), \ldots, p_s(z)$ in $\mathbb{F}_b[z]$, where $e_j = \deg(p_j(z))$. For each $j = 1, \ldots, s$, we also need a sequence $y_{j,1}(z), y_{j,2}(z), \ldots$ of polynomials. These polynomials, when reduced modulo $p_j(z)$, must be independent within each group of size e_j. That is the polynomials $y_{j,1}(z) \bmod p_j(z), y_{j,2}(z) \bmod p_j(z), \ldots, y_{j,e_j}(z) \bmod p_j(z)$ must be linearly independent.

We then consider the coefficients $a^{(j)}(k,l,r)$ in the development of

$$\frac{y_{j,l}(z)}{p_j(z)^k} = \sum_{r=w}^{\infty} a^{(j)}(k,l,r) z^{-r}$$

and use them to construct the generating matrices. More precisely, for $l \le e_j$, the lth row of C_j is given by $a^{(j)}(1,l,1), a^{(j)}(1,l,2), \ldots$. That is, the lth row contains the coefficients from development of $y_{j,l}(z)/p_j(z)$ for $l \le e_j$. Then the next block of e_j rows is based on the coefficients in the development of $y_{j,l}(z)/(p_j(z))^2$ and so on.

As shown in [24], a digital sequence constructed in this way has a quality parameter $t = (e_1 - 1) + \ldots + (e_s - 1)$.

The name *generalized Niederreiter sequences* comes from the fact that this construction builds on the principles proposed by Niederreiter for his construction of digital (t,s)-sequences [17], also known as *Niederreiter sequences*, but with a subtle generalization in the way the polynomials $y_{j,l}(z)$ are chosen.

$(0,s)$-sequences revisited

It is easy to see that the generating matrix resulting from $p_j(z) = z - j + 1$ and $y_{j,l} = 1$ for all j,l is the $(j-1)$th power of the Pascal matrix, P_b^{j-1}. Hence, Faure sequences are generalized Niederreiter sequences with $e_j = 1$ for all j so that we find again that they are $(0,s)$-sequences. However, powers of the Pascal matrix have ones on their diagonal, which in turn implies that points from these sequences will

tend to clump along the main diagonal in $[0,1)^s$ initially, and the larger b is, the more time it takes before this behavior goes away.

An important new step was achieved in [5], where it is shown that the L_2-discrepancy T of NUT digital $(0,1)$-sequences only depend on the diagonal entries of the generating matrices. Hence the discrepancy T of one-dimensional projections of Faure sequences can be improved by multiplying on the left the generating matrices P_b^{j-1} by diagonal matrices. For more details, see [6, Corollary 2 and Section 5]. This theoretical result is at the origin of new scramblings for Halton sequences (see [8]) and Faure sequences (see Sections 4 and 5 below).

The generalization proposed by Tezuka [24] amounts to replacing the $y_{j,l}(z)$ by arbitrary constants in $\mathbb{Z}_b$, i.e., multiply P_b^{j-1} on the left by an NLT matrix, as in (4). It has been widely used under the name GFaure, in the software FinDer.

Sobol' sequences revisited

As mentioned in Section 2, here we take $p_j(z)$ to be the jth primitive polynomial in base 2 (sorted in increasing degree). We also need polynomials $y_{j,1}(z),\ldots,y_{j,e_j}(z)$ to initialize each block of e_j rows. Note that the requirement that the "direction numbers" be odd means that $y_{j,1}(z)$ must be of degree e_j-1, $y_{j,2}(z)$ must be of degree e_j-2, and so on, up to $y_{j,e_j}(z)=1$, which must be of degree 0. Hence, each generating matrix C_j is NUT, as mentioned before.

Various proposals for finding good direction numbers have been proposed in the literature [13, 15]. It should be noted that the naive choice of selecting direction numbers so that the first submatrix is the $e_j\times e_j$ identity I_{e_j} is not good. It causes the same kind of sub-optimal behavior as the one resulting from the fact that we have ones on the diagonal of the Pascal matrices for the original Faure $(0,s)$-sequences. Going further, we can think of the direction numbers as a way of performing a deterministic scrambling of the naive Sobol' sequence, using an NUT block-diagonal matrix as the scrambling matrix. More formally, we have the following result.

Proposition 2. *Let C be a generating matrix for the Sobol' sequence based on a primitive polynomial $p(z)$ of degree e, and initial direction numbers given by an NUT $e\times e$ matrix V. Let G be another generating matrix for the Sobol' sequence based on $p(z)$ but now with $V=I_e$. Then we can write $C=DG$, where*

$$D=\begin{pmatrix} V & 0 & \cdots \\ 0 & V & \ddots \\ \vdots & \ddots & \ddots \end{pmatrix}.$$

Proof. Let $p(z)=z^e+a_1z^{e-1}+\cdots+a_{e-1}z+1$. The key point of the proof is to understand the structure of the matrix C. To do so, it is useful to partition it into blocks of $e\times e$ matrices. Let $B_{l,c}$ be the cth such matrix on the lth block of e rows that form C. We will prove that $B_{l,c}$ has the form $VH_{l,c}$ for some matrix $H_{l,c}$ independent of

V, for all $l, c \geq 1$. To do so, we introduce two $e \times e$ matrices Q and F, defined as

$$Q = \begin{pmatrix} 1 & 0 & \dots & 0 \\ a_{e-1} & 1 & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ a_1 & a_2 & \dots & 1 \end{pmatrix}, \qquad F = (I_e + R_2)(I_e + R_3)\dots(I_e + R_e),$$

where R_k is an $e \times e$ matrix with the first $k-1$ terms of its kth column given by $a_{k-1}, a_{k-2}, \dots, a_1$ and filled with zeros otherwise.

First, by definition of the Sobol' sequence we have $B_{1,1} = V$. Then, we show by induction that $B_{1,c} = V(QF)^{c-1}$ for any $c > 1$. Hence we need to show that $B_{1,c} = B_{1,c-1}(QF)$. To do so, we apply the recurrence (5) in two steps: first, we take the appropriate combinations of columns from $B_{1,c-1}$ explicitly described by (5). This step is performed through the multiplication by Q. Then, the kth column in $B_{1,c}$, for $k = 2, \dots, e$, is recursively updated by adding the appropriate combination of the $k-1$ preceding columns. One can easily verify that this is achieved by multiplying the matrix $B_{1,c-1}Q(I_e + R_2)\dots(I_e + R_{k-1})$ by $(I_e + R_k)$.

For the next rows of C, one can easily verify that the lth block of e rows of C starts with $l-1$ zero matrices. Then, we have that $B_{l,l} = VF^{l-1}$, for any $l > 1$. This comes from the last term $(v_{j,r-d_j}/2^{d_j})$ in (5), which has the effect of pasting $B_{l-1,l-1}$ into $B_{l,l}$. But at the same time, we need to take into account the effect of the recurrence (5) on $B_{l,l}$. This is achieved by successively multiplying by terms of the form $(I_e + R_k)$, for $k = 2, \dots, e$.

For $c > l$, we have that $B_{l,c} = (B_{l,c-1}Q + B_{l-1,c-1})F$, where the first term $B_{l,c-1}QF$ corresponds to the application of (5) but without the $(v_{j,r-d_j}/2^{d_j})$ term, which is handled separately by the second term $B_{l-1,c-1}F$).

Hence for all $l, c \geq 1$, $B_{l,c} = VH_{l,c}$ for some matrix $H_{l,c}$ independent of V. Therefore, the generating matrix G has submatrices of the form $G_{l,c} = H_{l,c}$, and hence C is obtained by multiplying (from the left) each submatrix $G_{l,c}$ in G by V, which amounts to setting $C = DG$. □

We can use this result to draw an interesting analogy with $(0, s)$-sequences. Indeed, for $(0, s)$-sequences, $e = 1$ and so from the point of view of Proposition 2, finding "good" direction numbers amounts to choosing one factor f_j for each dimension j and then use $A_j = f_j I$, where I is the $\mathbb{N} \times \mathbb{N}$ identity matrix. This is precisely the type of simple scrambling that we choose to focus on in this work. Note that the resulting matrix C_j (4) has diagonal entries given by f_j.

4 New Efficient Scramblings of $(0, s)$-Sequences

Here we propose two ways of constructing a $(0, s)$-sequence. Both suggestions amount to carefully choose the matrices A_j in (4). In both cases, as announced above in our interpretation of Proposition 2, we take A_j to be of the form $A_j = f_j I$,

where f_j is an appropriately chosen factor. The choice of this factor is based on the same ideas as those used in [8] to build generalized Halton sequences. For this reason, we first briefly recall our method from [8].

Generalized Halton sequences from [8]

The approach here is to first build a "short list" of at most 32 multipliers based on the criterion $\theta_p^f(1)/\log p$ described in [6], which is related to a bound on the L_2-discrepancy of S_p^{fI}. More precisely, $\theta_p^f(1)$ is defined as [6, Prop. 2]

$$\theta_p^f(1) = \max_{1 \le N \le p} \left(T^2(N, S_p^{fI}) - \frac{N^2}{12p^2} \right),$$

where $T(N, S)$ measures the L_2-discrepancy of the first N points of the sequence S. Multipliers f for which $\theta_p^f(1)$ is small thus give rise to good one-dimensional sequences. While the construction of this short list rests on a nice theoretical result, we use a more pragmatic method to select a multiplier from this list. Our idea is to make sure that two-dimensional projections over nearby indices are well distributed, hence avoiding the most well-known defect of the original Halton sequences [16].

That is, to select a multiplier f_j for the jth coordinate, we use the criterion

$$\tau_j^{W,N_0}(f_1, \ldots, f_{j-1}, f) = \max_{1 \le l \le W} T(N_0, (S_{p_{j-l}}^{f_{j-l}}, S_{p_j}^f)), \tag{6}$$

where W and N_0 have to be chosen, and $f_1 = 1$ since we use $p_1 = 2$. That is, for each candidate f in the short list for p_j, we compute the value T of the L_2-discrepancy for the first N_0 points of the two-dimensional sequence based on the $(j-l)$th and jth coordinates (using the multipliers f_{j-l} chosen for the $(j-l)$th coordinate, and the candidate f under study, respectively), for $l = 1, \ldots, W$, where W is the "window" size of the criterion. Then we keep the worst (largest) of these W values of T as our quality measure for f. The multiplier for p_j is chosen as the one that minimizes $\tau_j^{W,N}(f_1, \ldots, f_{j-1}, f)$ among all candidates in the short list.

Construction 1 for $(0, s)$-sequences: GF1

Our first idea is to choose b to be the smallest prime larger or equal to s—as typically done when building $(0, s)$-sequences—and then let $A_j = f_j \times I$ where the multipliers f_j are chosen similarly as in the approach described above for generalized Halton sequences. More precisely, we build a list $\mathcal{L}$ containing the m best multipliers f according to θ_b^f, where m is approximately equal to $b/2$. This is a valid approach since any NUT matrix C with diagonal entries given by f is such that $T(N, S_p^f) = T(N, S_p^C)$ [5]. We set f_1 equal to the best multiplier in $\mathcal{L}$ and then choose, for every $j = 2, \ldots, s$, the multiplier $f_j = f \in \mathcal{L}$ that minimizes

$$\tau_j(N_0, W) = \max_{1 \le l \le W} T(N_0, (S_b^{C_{j-l}}, S_b^C)),$$

where $C_{j-l} = f_{j-l} P_b^{j-l-1}$ and $C = f P_b^{j-1}$.

The results reported in Section 6 were done with $N_0 = 2500$ and $W = 7$, which are the values used in [8]. As discussed in [8], taking $N_0 = 2500$ is somewhat arbitrary but ensures $N_0 > b$ in our examples and appears to be an appropriate length to discriminate "good" sequences from "bad" sequences. Note that the construction of the "short list" of multipliers is done independently of N_0, as it is based on the criterion $\theta_p^f(1)$, which provides a bound on the discrepancy *for all* N.

Construction 2 for $(0, s)$-sequences: GF2

The second idea starts with a somewhat unusual choice, which is to take the base b to be about twice as large as the dimension s. More precisely, we take b the smallest prime larger than $2s$. A larger base implies a larger choice of multipliers, and so by taking $b \approx 2s$, we can simply set $A_j = f_j I$ as in the first proposal, but with f_j equal to the multiplier f with the jth smallest value for θ_b^f. This is a very simple method, as no search based on two-dimensional projections needs to be performed. One simply needs to order the multipliers according to θ_b^f, which can be done quickly.

5 A Construction Extensible in the Dimension–GF3

Because $(0, s)$-sequences need to be defined in a base $b \ge s$, they do not have the property of being *extensible in the dimension*, which we now define.

Definition 1. A family of constructions is *extensible in the dimension* if, for every $s \ge 1$, an s-dimensional sequence $\{X_i, i \ge 1\}$ in that family can be transformed into an r-dimensional sequence $\{\tilde{X}_i, i \ge 1\}$ in that family, where $r > s$, in such a way that X_i equals the first s coordinates of $\tilde{X}_i$.

From their definition, it is clear that Sobol' and Halton sequences are extensible in the dimension. The problem with $(0, s)$-sequences is that if we choose a base $b \ge s$, then for any $r > b$ we need to choose a new base in order to define a $(0, r)$-sequence, and thus we cannot simply extend each s-dimensional point to an r-dimensional one.

Now, if we weaken the property of $(0, s)$-sequences by introducing another quality parameter to replace t, then we will be able to create a construction that is closely connected to $(0, s)$-sequences, but has the advantage of being extensible in the dimension. So first, we define this new quality parameter, which has similarities with other criteria discussed in, e.g., [14].

Definition 2. For an s-dimensional digital sequence $\{X_1, X_2, \ldots\}$ in base b and integer $k \le s$, its quality parameter t_k, is defined as the smallest value so that each

projection of the form

$$\{(X_{i,j_1}, X_{i,j_2}, \ldots, X_{i,j_r}), i \geq 1\}$$

with $1 \leq r \leq k$, $j_1 < \cdots < j_r$ and $j_r - j_1 < k$, is a (t_k, r)-sequence.

The quality parameter t_k thus focuses on projections over indices $j_1, \ldots, j_r$ that span a range no larger than k. Note that for a (t, s)-sequence, we have $t_s = t$.

The idea is then to fix the base b, and for any dimension $s \geq 1$, construct an s-dimensional digital sequence in base b using the generating matrices

$$C_j = A_j P_b^{j-1}, \qquad j = 1, \ldots, s, \tag{7}$$

where $A_j = f_{(j \bmod m)} I$ and $m \approx b/2$. As for the GF-2 construction, $f_{(j \bmod m)}$ is the multiplier in $\mathbb{Z}_b$ with the $(j \bmod m)$th smallest value of θ_b^f.

Note that the matrices A_j make the period of the sequence $C_1, C_2, \ldots$ longer than that of the sequence $P_b, P_b^1, P_b^2, \ldots$, which equals b since $P_b^j = P_b^{j+lb}$ for any $l \geq 1$. Furthermore, as done in our numerical experiments, we can add a random shift as in Section 2 to these sequences, which completely breaks the periodic behavior of the sequence's coordinates for a given point, much like the use of a shift modulo 1 breaks the periodic behavior of the coordinates of points from a Korobov lattice when the number of points N is smaller than s. We also have the following result, which can be easily proved using the well-known fact that any projection of a $(0, s)$-sequences has a quality parameter $t = 0$.

Proposition 3. *For any $s \geq 1$, the s-dimensional digital sequence in base b based on the generating matrices (7) has a quality parameter $t_b = 0$.*

Note that this construction can handle problems where the dimension is unbounded, because once b and m are chosen and the factors $f \in \mathbb{Z}_b$ are sorted according to θ_b^f, *no extra parameters need to be chosen*. Korobov lattices also have this property, since the only parameter that needs to be chosen is the generator a of the lattice. It should be noted, however, that when $s \geq b$ this construction is not uniformly distributed. Also, because of its periodic behavior and the fact that it has some bad projections, it may give erroneous results when used to integrate certain classes of functions which heavily depend on the subset of variables corresponding to these projections. Hence this construction should be used carefully. A possible class of functions for which it could be proved to do well are those having *finite-order weights*, as defined in [21].

6 Numerical Results

In this section, we compare the performance of different low-discrepancy sequences on a few problems. Our main goal is to illustrate the importance of the parameter choice for each construction. Hence we compare the (i) original Halton sequences

(H); (ii) generalized Halton sequence proposed and recommended in [8] (GH); (iii) the naive Sobol' sequence where the direction numbers are set to the unit vectors (S); (iv) the Sobol' sequence with direction numbers from [15] (GS); (v) the original $(0, s)$-sequence from [3] (F), and (vi) the alternatives GF-1 and GF-2 described in Section 4. We refer to the group GH, GS, GF-1 and GF-2 as the "scrambled constructions", and the remaining ones as the "non-scrambled" ones.

For each integration problem f, we show the estimated variance of the average

$$Q_{N,M} = \frac{1}{NM} \sum_{l=1}^{M} \sum_{i=1}^{N} f(X_i^l),$$

where $X_i^l = (X_{i,1}^l, \ldots, X_{i,s}^l)$ is obtained by shifting each of the s individual (generalized/linearly scrambled) van der Corput sequences forming the construction under study by a random shift as described in Section 2, for $l = 1, \ldots, M$. The number of points N considered are of the form $N = 2000k$ for $k = 1, \ldots, 50$, and we use $M = 25$ i.i.d. random shifts to estimate the variance. For comparison, we show the estimated variance $\hat{\sigma}^2/NM$ of the Monte Carlo estimator based on NM evaluations, where $\hat{\sigma}^2$ is estimated using a grand total of $100000M$ simulations (or computed exactly when possible). Due to lack of space, we do not show comparative results based on the deterministic error or the average absolute error, as in [8].

We first consider a mortgage-backed security problem from [1]. Here the function f represents the discounted cash-flows paid by this security, which in turns come from a pool of mortgages where the mortgagors have the option of prepaying their entire balance at any time, with no penalty. The cash flows and discount factors both depend on a random interest rate process. The integral $I(f)$ corresponds to the theoretical price for this security and cannot be determined exactly.

As can be seen in Figure 1, the scrambled constructions do significantly better than the non-scrambled ones for this problem. In fact, some of the non-scrambled sequences do *worse* than Monte Carlo.

We then consider a *digital option* problem. This example has been used by Papageorgiou in [20] to show that the Brownian bridge technique—popular in financial problems—does not always help quasi-Monte Carlo methods. Hence for problems like this, it is important to have access to low-discrepancy sequences that work well, even in large dimensions. Here also, as seen in Figure 2, the scrambled constructions clearly do better than the non-scrambled ones.

The next three problems look at test-functions defined over I^s that have been used in the literature [8, 23], and are given by

$$g_1(X) = \prod_{j=1}^{96} (1 + 0.25(X_j - 0.5)); \; g_2(X) = \prod_{j=1}^{75} \frac{|4X_j - 2| + (75 - j)^2}{1 + (75 - j)^2};$$

$$g_3(X) = \alpha_{120} \pi^{-60} \cos\left(\sqrt{\frac{1}{2} \sum_{j=1}^{120} [\Phi^{-1}(X_j)]^2} \right),$$

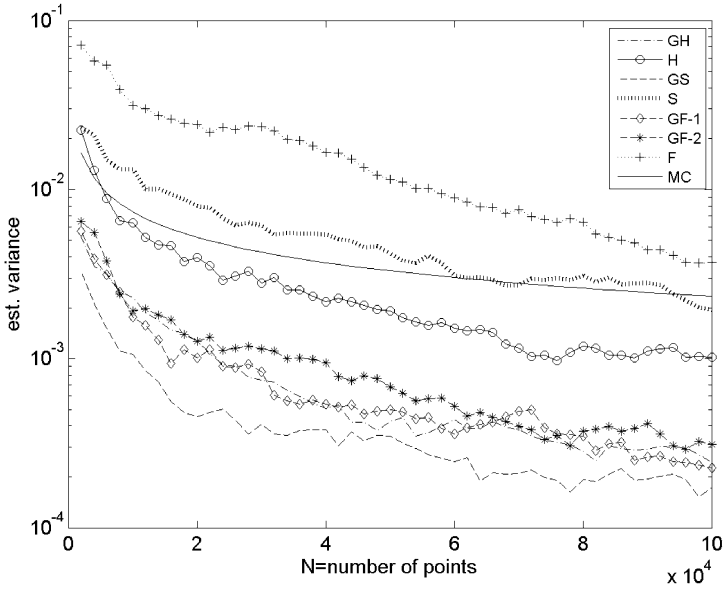


Fig. 1 MBS problem ($s = 360$): the scrambled constructions perform much better than the non-scrambled ones.

where α_{120} is such that g_3 integrates to 1 and Φ is the CDF of a standard normal.

The first function is such that constructions with good low-dimensional projections over all indices should do relatively well on this problem, while constructions whose low-dimensional projections deteriorate as they are defined over larger indices should not perform well. Results are shown on Figure 3.

Indeed, as before the scrambled versions (especially Sobol' and the two GF's) do much better than Monte Carlo and the non-scrambled constructions. Note that the naive Sobol' "S" is clearly worse than Monte Carlo.

The second function is such that the coordinates are in increasing order of importance, which means sequences whose coordinates with higher indices are not so well distributed will do badly on this problem. As we see on Figure 4, the non-scrambled sequences have a variance much larger than for the scrambled sequences. It is worth noting that the GF-2 construction, which does not pay attention to two-dimensional projections as GF-1 does, has a variance significantly larger than the other scrambled sequences, although not as bad as the non-scrambled sequences.

Finally, the results shown on Figure 5 for g_3 confirm once again the superiority of the scrambled sequences, this time on a problem where f is not a product. Table 1 summarizes which methods perform best for each function for $N = 100000$ (variance within a factor of two of the smallest variance). We see GS and GF1 are always among the best. The construction GF2 is less competitive, but we are confident that its performance would improve significantly if multipliers were selected with the

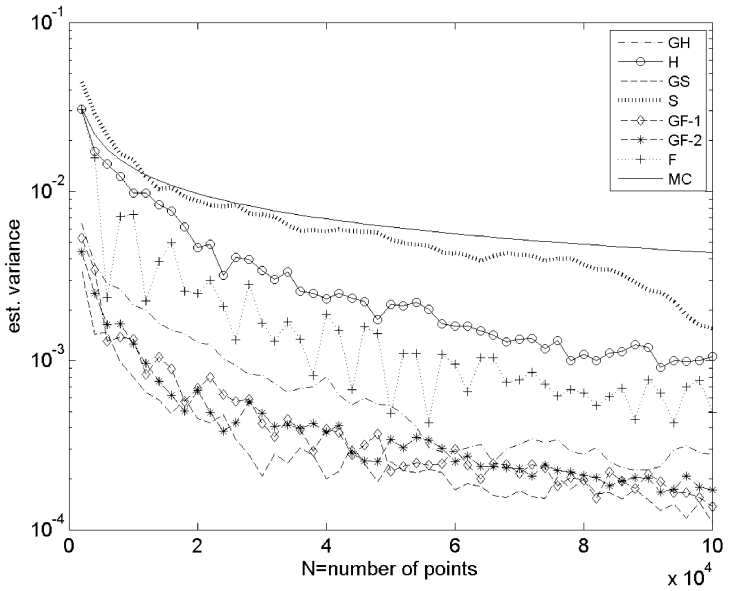


Fig. 2 Digital option with $s = 75$: here also, the scrambled constructions perform much better.

help of two-dimensional projections as in GF1 (but still with the larger base), which is in our plans for the near future.

Table 1 Best methods for each function.

MBS	Dig	g_1	g_2	g_3
GS,GF1,GH	GS,GF1	GS,GF2,GF1	GH,GS,GF1	GS,GF1,GH

We end with some results showing the performance of the construction described in Section 5, which is extensible in the dimension and can therefore handle a problem in any dimension. The problem considered here is a simple queueing system where clients arrive according to a Poisson process with frequency 1/minute, and the service times are exponentially distributed with mean 55 seconds. We wish to determine the expected number of clients, among the first 1000 ones, who wait more than 5 minutes in the queue before being served. Since we need to generate one interarrival time and one service time per client, the dimension s of this problem is 2000. We compare the performance of the GF-3 construction in base $b = 727$ with an *extensible Korobov lattice* taken from [9] (based on the generator $a = 14471$) and the Monte Carlo method. Results are shown on Figure 6. Both low-discrepancy sequences do much better than Monte Carlo on this problem, and the simple GF-3 sequence is competitive with the extensible Korobov one.

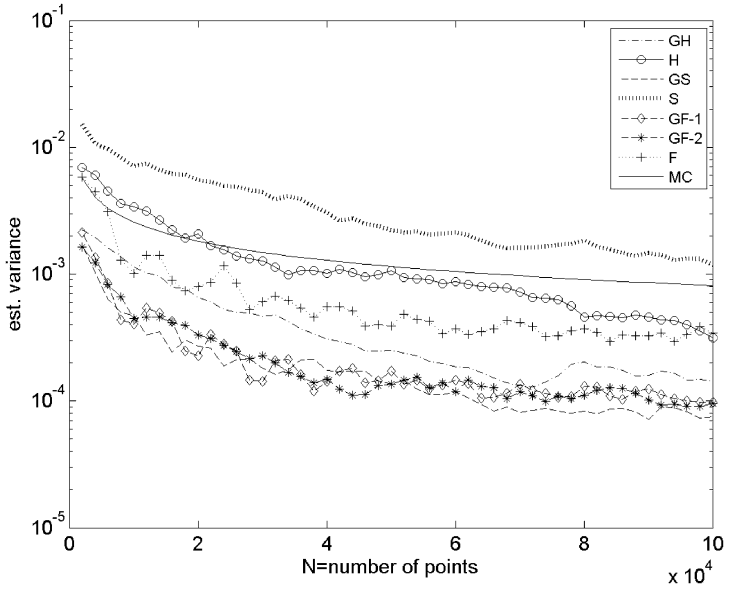


Fig. 3 Function g_1 where $s = 96$: the scrambled sequences GS, GF1 and GF2 are best and some non-scrambled constructions do worse than MC.

7 Conclusion

In this paper, we proposed two new ways of choosing parameters for $(0, s)$-sequences that attempt to correct some of the problems encountered in large dimensions by the original construction of Faure. Our approach shares similarities with other ideas used for the Halton and Sobol' sequence. Numerical results on various problems suggest that for all three families, the constructions based on carefully chosen parameters can perform significantly better than the more naive choices. Furthermore, we saw that the new constructions are competitive with the Sobol' sequence, which is popular among practitioners.

We proposed a new construction based on $(0, s)$-sequences where we fix the base b and allow $s > b$. Hence this construction is extensible in the dimension and can handle problems of unbounded dimension. Our numerical results done with $s = 2000$ suggest it is a promising approach, at least for some classes of problems.

For future research, we plan to extend our search for efficient scramblings, for example by building NLT scrambling matrices A_j—not just diagonal—and also by fine-tuning the multipliers for GF-2 and GF-3 so that the resulting sequences have good low-dimensional projections, as done for the GF-1 construction. We would also like to study further the properties of the GF-3 construction, so as to get a better understanding of the classes of functions for which it can perform well.

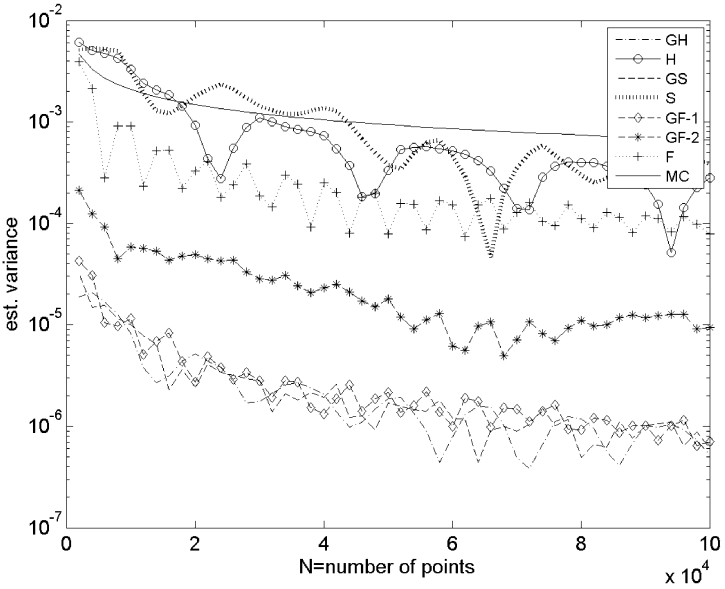


Fig. 4 Function g_2 where $s = 75$: here GF2 does not do as well as GH, GS and GF1, but is still better than the non-scrambled sequences.

Acknowledgements We thank the program committee of MCQMC'08 for giving us the opportunity to present this work in Montréal, and the editor and referees for their useful suggestions. The second author also acknowledges the financial support of NSERC.

References

1. Caflisch, R.E., Morokoff, W., Owen, A.B.: Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. J. Comput. Finance **1**(1), 27–46 (1997)
2. Faure, H.: Discrépance des suites associées à un système de numération (en dimension un). Bull. Soc. Math. France **109**, 143–182 (1981)
3. Faure, H.: Discrépance des suites associées à un système de numération (en dimension s). Acta Arit. **41**, 337–351 (1982)
4. Faure, H.: Multidimensional quasi-Monte Carlo methods. Theoret. Comput. Sci. **123**(1), 131–137 (1994)
5. Faure, H.: Discrepancy and diaphony of digital (0,1)-sequences in prime base. Acta Arit. **117**, 125–148 (2005)
6. Faure, H.: Selection criteria for (random) generation of digital $(0, s)$-sequences. In: H. Niederreiter, D. Talay (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 113–126. Springer New York (2006)
7. Faure, H.: Van der corput sequences towards general $(0, 1)-$sequences in base b. J. Théor. Nombres Bordeaux **19**, 125–140 (2007)
8. Faure, H., Lemieux, C.: Generalized Halton sequences in 2008: A comparative study (2009). To appear in ACM Trans. Model. Comp. Sim.
9. Gill, H.S., Lemieux, C.: Searching for extensible Korobov rules. J. Comp. **23**, 603–613 (2007)

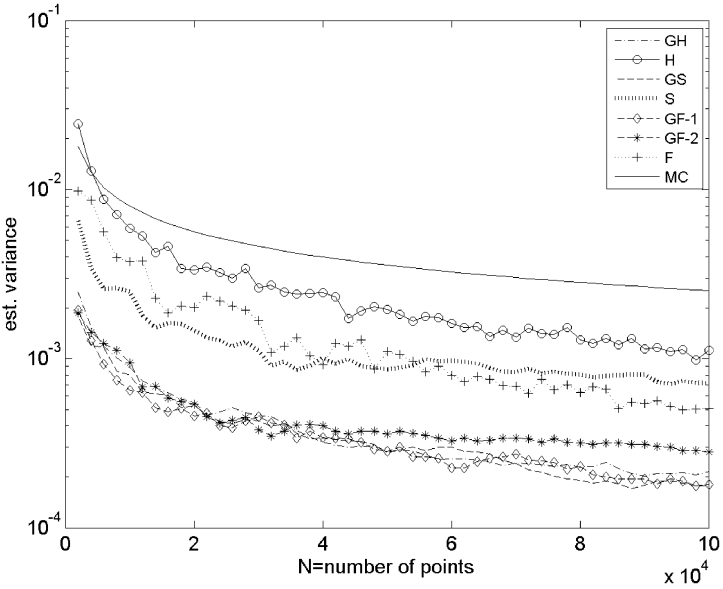


Fig. 5 Function g_3 where $s = 120$: all scrambled sequences do better than the non-scrambled ones, with GF2 slightly less competitive than the others.

10. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Mathem. **2**, 84–90 (1960)
11. Hofer, R.: On the distribution properties of Niederreiter-Halton sequences. J. Number Th. **129**, 451–463 (2009)
12. Hofer, R., Kritzer, P., Larcher, G., Pillichshammer, F.: Distribution properties of generalized van der Corput-Halton sequences and their subsequences. Int. J. Number Th. (2009). In press.
13. Joe, S., Kuo, F.Y.: Constructing Sobol' sequences with better two-dimensional projections. SIAM J. Scient. Comput. **30**, 2635–2654 (2008)
14. Larcher, G.: Digital point sets: Analysis and applications. In: P. Hellekalek, G. Larcher (eds.) Random and Quasi-Random Point Sets, *LNS*, vol. 138, pp. 167–222. Springer New York (1998)
15. Lemieux, C., Cieslak, M., Luttmer, K.: RandQMC user's guide: A package for randomized quasi-Monte Carlo methods in C. Tech. Rep. 2002-712-15, Department of Computer Science, University of Calgary (2002)
16. Morokoff, W.J., Caflisch, R.E.: Quasi-random sequences and their discrepancies. SIAM J. Scient. Comput. **15**, 1251–1279 (1994)
17. Niederreiter, H.: Point sets and sequences with small discrepancy. Monats. Math. **104**, 273–337 (1987)
18. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia (1992)
19. Niederreiter, H., Xing, C.: Quasirandom points and global function fields. In: Finite fields and applications (Glasgow, 1995), *London Math. Soc. Lecture Note Ser.*, vol. 233, pp. 269–296. Cambridge Univ. Press, Cambridge (1996)
20. Papageorgiou, A.: The Brownian bridge does not offer a consistent advantage in quasi-Monte Carlo integration. J. Comp. **18**(1), 171–186 (2002)
21. Sloan, I.H., Wang, X., Woźniakowski, H.: Finite-order weights imply tractability of multivariate integration. J. Comp. **20**, 46–74 (2004)

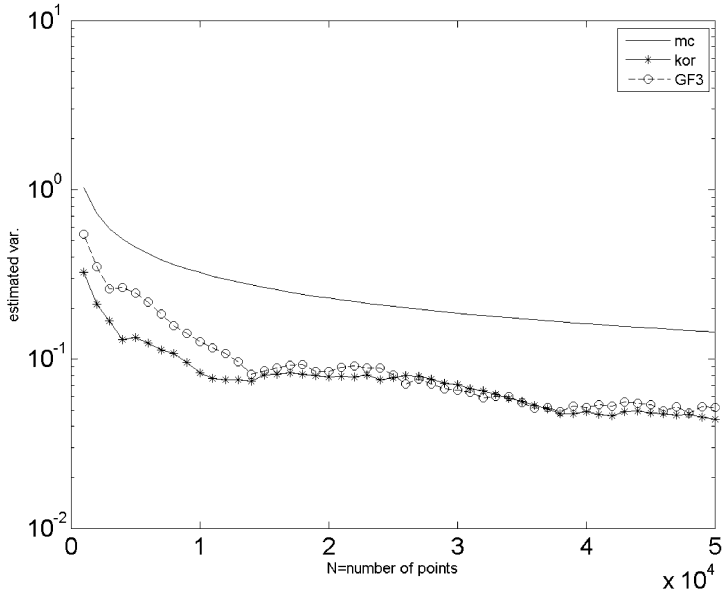


Fig. 6 Queueing problem: $s = 2000$.

22. Sobol', I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput. Math. and Math. Physics **7**, 86–112 (1967)
23. Sobol', I.M., Asotsky, D.I.: One more experiment on estimating high-dimensional integrals by quasi-Monte Carlo methods. Math. Comput. Simul. **62**, 255–263 (2003)
24. Tezuka, S.: Polynomial arithmetic analogue of Halton sequences. ACM Trans. Model. Comp. Sim. **3**, 99–107 (1993)
25. Tezuka, S.: A generalization of Faure sequences and its efficient implementation. Tech. Rep. RT0105, IBM Research, Tokyo Research Laboratory (1994)
26. Warnock, T.: Computational investigations of low discrepancy point sets. In: S.K. Zaremba (ed.) Application de la théorie des nombres à l'analyse numérique, pp. 319–343. Academic Press New York (1972)
27. www.math.uwaterloo.ca/~clemieux/0s.html

Variable Subspace Sampling and Multi-level Algorithms

Thomas Müller-Gronbach and Klaus Ritter

Abstract We survey recent results on numerical integration with respect to measures μ on infinite-dimensional spaces, e.g., Gaussian measures on function spaces or distributions of diffusion processes on the path space. Emphasis is given to the class of multi-level Monte Carlo algorithms and, more generally, to variable subspace sampling and the associated cost model. In particular we investigate integration of Lipschitz functionals. Here we establish a close relation between quadrature by means of randomized algorithms and Kolmogorov widths and quantization numbers of μ. Suitable multi-level algorithms turn out to be almost optimal in the Gaussian case and in the diffusion case.

1 Introduction

Let μ be a Borel probability measure on a Banach space $(\mathfrak{X}, \|\cdot\|_{\mathfrak{X}})$, and let F denote a class of μ-integrable functionals $f : \mathfrak{X} \to \mathbb{R}$. In the corresponding quadrature problem we wish to compute

$$S(f) = \int_{\mathfrak{X}} f(x)\,\mu(dx)$$

for $f \in F$ by means of randomized (Monte Carlo) algorithms that use the values $f(x)$ of the functional f at a finite number of sequentially (adaptively) chosen points $x \in \mathfrak{X}$.

Thomas Müller-Gronbach
Fakultät für Informatik und Mathematik, Universität Passau, 94032 Passau, Germany

Klaus Ritter
Fachbereich Mathematik, Technische Universität Darmstadt, 64289 Darmstadt, Germany
url: http://www.mathematik.tu-darmstadt.de/~ritter

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_8,

The classical instance of this quadrature problem is given by $\mathfrak{X} = \mathbb{R}^d$ and μ being the uniform distribution on $[0,1]^d$, say, or the d-dimensional standard normal distribution. In the present paper we are mainly interested in infinite-dimensional spaces $\mathfrak{X}$. The classical instance of infinite-dimensional quadrature is path integration with respect to the Wiener measure μ on $\mathfrak{X} = C([0,1])$ or, more generally, quadrature with respect to a Gaussian measure μ on a function space $\mathfrak{X}$.

Further important instances of quadrature problems arise for stochastic (partial) differential equations, and here the measure μ is usually given only implicitly, since it depends on the solution process of the equation. We have $\dim(\mathfrak{X}) < \infty$ if μ is a marginal distribution of the solution of an SDE and $\dim(\mathfrak{X}) = \infty$ for quadrature on the path space. For SPDEs, both the marginal and the path dependent case lead to infinite-dimensional quadrature problems.

The present paper is motivated by the following developments. On the one hand a new class of algorithms, namely multi-level Monte Carlo algorithms, has been introduced by Heinrich [18] and Giles [14]. On the other hand infinite-dimensional quadrature problems have been studied from a complexity point of view by Wasilkowski and Woźniakowski [37] and Hickernell and Wang [21]. The purpose of this paper is to illustrate the approach and the results from [5], which provides a link between the two developments and which establishes the concept of approximation of distributions as the basis for integration of Lipschitz functionals f on infinite-dimensional spaces $\mathfrak{X}$. Furthermore, we provide a continuation of the survey paper [30] on strong and weak approximation of SDEs with a new focus on multi-level Monte Carlo algorithms.

The content of this paper is organized as follows. In Section 2 we present multi-level Monte Carlo algorithms in general terms together with the particular case of multi-level Euler Monte Carlo algorithms for SDEs, which serve as a basic example in the sequel.

Section 3 is devoted to the presentation of a reasonable cost model for the analysis of infinite-dimensional quadrature problems. We distinguish between full space sampling, variable subspace sampling, and fixed subspace sampling. In the latter case an algorithm may only evaluate the integrands f at the points in a finite-dimensional subspace $\mathfrak{X}_0 \subset \mathfrak{X}$, which may be chosen arbitrarily but which is fixed for a specific algorithm. We add that fixed subspace sampling is frequently used for infinite-dimensional quadrature problems. In contrast, a multi-level algorithm uses dependent samples in a hierarchy of finite-dimensional subspaces $\mathfrak{X}_1 \subset \mathfrak{X}_2 \subset \ldots \subset \mathfrak{X}$ with only a small proportion taken in high-dimensional spaces. For both variants of subspace sampling the cost per evaluation at $x \in \mathfrak{X}$ is given by the dimension of the (minimal) subspace containing x. Full space sampling permits evaluations anywhere in $\mathfrak{X}$ at cost one, which is perfectly reasonable for finite-dimensional quadrature problems; in the infinite-dimensional case its main purpose is to establish lower bounds.

Section 4 contains an analysis of multi-level algorithms. In the particular case of Lipschitz continuous integrands we provide upper error bounds for these algorithms in terms of average Kolmogorov widths of the measure μ, see Theorem 3.

In Section 5 we introduce the concept of minimal errors, which allows a rigorous comparison of the power of full space sampling, variable subspace sampling, and fixed subspace sampling.

In Section 6 we focus on the Lipschitz case and we present upper and lower bounds for the minimal errors in terms of average Kolmogorov widths and quantization numbers. Since the latter two quantities can equivalently be defined in terms of the Wasserstein distance on the space of Borel probability measures on $\mathfrak{X}$ our error estimates exhibit the tight connection between quadrature by means of randomized algorithms and approximation of the underlying measure μ by means of probability measures with a finite-dimensional or finite support. These results are applied to quadrature with respect to Gaussian measures μ and with respect to the distributions of the solutions of SDEs. Suitable multi-level algorithms turn out to be almost optimal in both cases.

2 Multi-level Algorithms

Multi-level Monte Carlo methods have been introduced by Heinrich [18] and Heinrich and Sindambiwe [20] for computation of global solutions of integral equations and for parametric integration, respectively. Moreover, the authors have shown that suitable multi-level algorithms are (almost) optimal in both cases. See [19] for further results and references. In their work finite-dimensional quadrature problems arise as subproblems and the Monte Carlo methods take values in infinite-dimensional Banach spaces $\mathfrak{Y}$. Here we are interested in the dual situation of infinite-dimensional quadrature with real numbers as outputs of Monte Carlo algorithms, i.e., we have $\mathfrak{Y} = \mathbb{R}$.

In the context of quadrature problems for diffusion processes multilevel algorithms have been introduced by Giles [14], while a two level algorithm has already been considered by Kebaier [22]. Both papers also include numerical examples from computational finance, see also [15, 16].

In order to describe the multi-level approach in general terms it is convenient to assume that μ is the distribution of an $\mathfrak{X}$-valued random element of the form

$$X = \varphi(\widetilde{X})$$

for some random element $\widetilde{X}$ taking values in a Banach space $(\widetilde{\mathfrak{X}}, \|\cdot\|_{\widetilde{\mathfrak{X}}})$ and some measurable mapping

$$\varphi : \widetilde{\mathfrak{X}} \to \mathfrak{X}.$$

As the key assumption we suppose that we have a sequence of measurable mappings

$$\varphi^{(k)} : \widetilde{\mathfrak{X}} \to \mathfrak{X}$$

at hand, which provide approximations

$$X^{(k)} = \varphi^{(k)}(\widetilde{X})$$

to X. Hence

$$\mathrm{E}(f(X^{(k)})) = \mathrm{E}(f(\varphi^{(k)}(\widetilde{X})))$$

may serve as an approximation to

$$S(f) = \mathrm{E}(f(X)) = \mathrm{E}(f(\varphi(\widetilde{X}))).$$

Example 1. A typical example is provided by an SDE

$$dX(t) = a(t, X(t))\,dt + b(t, X(t))\,d\widetilde{X}(t), \qquad t \in [0,1],$$

with initial value

$$X(0) = \xi \in \mathbb{R}^m,$$

drift coefficient

$$a : [0,1] \times \mathbb{R}^m \to \mathbb{R}^m,$$

diffusion coefficient

$$b : [0,1] \times \mathbb{R}^m \to \mathbb{R}^{m \times d},$$

and with a d-dimensional Brownian motion $\widetilde{X}$. In this case $\widetilde{\mathfrak{X}} = C([0,1], \mathbb{R}^d)$ and $\mathfrak{X} = C([0,1], \mathbb{R}^m)$ are the spaces of continuous functions on $[0,1]$ taking values in $\mathbb{R}^d$ and $\mathbb{R}^m$, respectively, and φ maps the driving Brownian motion $\widetilde{X}$ to the solution process X. The mapping $\varphi^{(k)}$ may correspond to the piecewise linear interpolation of the Euler scheme with step size $\delta^{(k)} = 2^{-(k-1)}$. The time discretization is then given by

$$t_i^{(k)} = i\,\delta^{(k)}, \qquad i = 0, \ldots, 2^{k-1},$$

and we have

$$X_0^{(k)} = \xi$$

and

$$X_{i+1}^{(k)} = X_i^{(k)} + a(t_i^{(k)}, X_i^{(k)})\,\delta^{(k)} + b(t_i^{(k)}, X_i^{(k)})\,\big(\widetilde{X}(t_{i+1}^{(k)}) - \widetilde{X}(t_i^{(k)})\big). \tag{1}$$

Finally, the random element $X^{(k)} = \varphi^{(k)}(\widetilde{X})$ is given by the piecewise linear interpolation of $X_0^{(k)}, \ldots, X_{2^{k-1}}^{(k)}$ at the nodes $t_0^{(k)}, \ldots, t_{2^{k-1}}^{(k)}$.

For a Gaussian measure μ on $\mathfrak{X}$ it is reasonable to take $\widetilde{\mathfrak{X}} = \mathfrak{X}$, $\widetilde{X} = X$, and the identity function φ. Metric projections $\varphi^{(k)}$ onto an increasing sequence of finite-dimensional subspaces of $\mathfrak{X}$ may be used for approximation of X.

The classical Monte Carlo approximation to $\mathrm{E}(f(X^{(k)}))$ is based on independent copies $\widetilde{X}_1, \ldots, \widetilde{X}_n$ of $\widetilde{X}$ and given by the random variable

$$A(f) = \frac{1}{n} \sum_{\ell=1}^{n} f(\varphi^{(k)}(\widetilde{X}_\ell)). \tag{2}$$

For its mean square error we clearly have

$$\mathrm{E}(S(f)-A(f))^2=\frac{1}{n}\,\mathrm{Var}(f(X^{(k)}))+b_k^2(f) \tag{3}$$

with the bias

$$b_k(f)=\mathrm{E}(f(X^{(k)}))-S(f).$$

The actual computation of a realization of $A(f)$ requires simulation of the distribution of $X^{(k)}$ and evaluation of f at randomly chosen points from the range of $\varphi^{(k)}$.

Note that

$$\mathrm{E}(f(X^{(k)}))=\mathrm{E}(f(X^{(1)}))+\sum_{j=2}^{k}\mathrm{E}(f(X^{(j)})-f(X^{(j-1)})),$$

where $f(X^{(j)})=f(\varphi^{(j)}(\widetilde{X}))$ and $f(X^{(j-1)})=f(\varphi^{(j-1)}(\widetilde{X}))$ are coupled via $\widetilde{X}$. In the multi-level approach each of the expectations on the right-hand side is approximated separately by means of independent, classical Monte Carlo approximations. With $n_1,\ldots,n_k$ denoting the corresponding numbers of replications and with independent copies

$$\widetilde{X}_{j,1},\ldots,\widetilde{X}_{j,n_j},\qquad j=1,\ldots,k,$$

of $\widetilde{X}$ the multi-level approximation is given by the random variable

$$A_k(f)=A^{(1)}(f)+\sum_{j=2}^{k}A^{(j)}(f) \tag{4}$$

where

$$A^{(1)}(f)=\frac{1}{n_1}\sum_{\ell=1}^{n_1}f(\varphi^{(1)}(\widetilde{X}_{1,\ell})) \tag{5}$$

and

$$A^{(j)}(f)=\frac{1}{n_j}\sum_{\ell=1}^{n_j}(f(\varphi^{(j)}(\widetilde{X}_{j,\ell}))-f(\varphi^{(j-1)}(\widetilde{X}_{j,\ell}))) \tag{6}$$

for $j=2,\ldots,k$. For the mean square error of $A_k(f)$ we get

$$\mathrm{E}(S(f)-A_k(f))^2=\sum_{j=1}^{k}\frac{v_j(f)}{n_j}+b_k^2(f) \tag{7}$$

where

$$v_1(f)=\mathrm{Var}(f(X^{(1)}))$$

and

$$v_j(f)=\mathrm{Var}(f(X^{(j)})-f(X^{(j-1)}))$$

for $j = 2, \dots, k$. The actual computation of a realization of $A_k(f)$ requires simulation of the distribution of $X^{(1)}$ and the joint distribution of $X^{(j)}$ and $X^{(j-1)}$ for $j = 2, \dots, k$. Furthermore, evaluation of f at randomly chosen points from the ranges of $\varphi^{(1)}, \dots, \varphi^{(k)}$ is needed.

Typically the variances $v_j(f)$ and the bias $b_k(f)$ are decreasing with increasing values of j and k, respectively, while the computational cost is increasing. One therefore has to properly balance these effects. A comparison of (3) and (7) reveals that the multi-level approach is a variance reduction technique.

Remark 1. The error formula (7) is a consequence of Bienaymé's equality for real-valued random variables, which does not extend to general Banach spaces. Thus, for the analysis of multi-level algorithms taking values in such a space the so-called Rademacher type of this space plays an important role, see [18, 19, 20].

Example 2. Let us present the details for a multi-level Euler algorithm in the case of an SDE, see Example 1. For notational convenience we consider a scalar equation, i.e., $m = d = 1$. We use $\stackrel{\mathrm{d}}{=}$ to denote equality in distribution of two random elements.

The simulation of $\varphi^{(1)}(\widetilde{X}_{1,\ell})$ and $(\varphi^{(j)}(\widetilde{X}_{j,\ell}), \varphi^{(j-1)}(\widetilde{X}_{j,\ell}))$ in (5) and (6) may be based on i.i.d. standard normally distributed random variables $Z_{i,\ell}^{(j)}$ for $j = 1, \dots, k$, $\ell = 1, \dots, n_j$, and $i = 1, \dots, 2^{j-1}$ as follows. We put

$$U_{0,\ell}^{(j)} = \xi$$

as well as

$$U_{i+1,\ell}^{(j)} = U_{i,\ell}^{(j)} + a(t_i^{(j)}, U_{i,\ell}^{(j)})\,\delta^{(j)} + b(t_i^{(j)}, U_{i,\ell}^{(j)})\,\sqrt{\delta^{(j)}}\,Z_{i+1,\ell}^{(j)}$$

for $i = 0, \dots, 2^{j-1} - 1$, cf. (1). Furthermore, if $j > 1$, we put

$$V_{0,\ell}^{(j)} = \xi$$

as well as

$$\begin{aligned} V_{i+1,\ell}^{(j)} = V_{i,\ell}^{(j)} &+ a(t_i^{(j-1)}, V_{i,\ell}^{(j)})\,\delta^{(j-1)} \\ &+ b(t_i^{(j-1)}, V_{i,\ell}^{(j)})\sqrt{\delta^{(j-1)}/2}\,\bigl(Z_{2i+1,\ell}^{(j)} + Z_{2i+2,\ell}^{(j)}\bigr) \end{aligned}$$

for $i = 0, \dots, 2^{j-2} - 1$.

We stress that the corresponding piecewise linear interpolations $U_\ell^{(j)}$ and $V_\ell^{(j)}$, respectively, are coupled, since both are based on the random vector $(Z_{1,\ell}^{(j)}, \dots, Z_{2^{j-1},\ell}^{(j)})$. On the other hand, $U_\ell^{(j-1)}$ and $V_{\ell'}^{(j)}$ are independent with $U_\ell^{(j-1)} \stackrel{\mathrm{d}}{=} V_{\ell'}^{(j)} \stackrel{\mathrm{d}}{=} X^{(j-1)}$. Altogether we obtain independent random elements

$$\begin{gathered} U_1^{(1)}, \dots, U_{n_1}^{(1)}, \\ (U_1^{(2)}, V_1^{(2)}), \dots, (U_{n_2}^{(2)}, V_{n_2}^{(2)}), \end{gathered}$$

$$\dots$$
$$(U_1^{(k)}, V_1^{(k)}), \dots, (U_{n_k}^{(k)}, V_{n_k}^{(k)})$$

taking values in $C([0,1])$ or $(C([0,1]))^2$, respectively, whose distributions satisfy

$$U_\ell^{(1)} \stackrel{\mathrm{d}}{=} X^{(1)}$$

and

$$(U_\ell^{(j)}, V_\ell^{(j)}) \stackrel{\mathrm{d}}{=} (X^{(j)}, X^{(j-1)}).$$

Consequently

$$A_k(f) \stackrel{\mathrm{d}}{=} \frac{1}{n_1} \sum_{\ell=1}^{n_1} f(U_\ell^{(1)}) + \sum_{j=2}^{k} \frac{1}{n_j} \sum_{\ell=1}^{n_j} (f(U_\ell^{(j)}) - f(V_\ell^{(j)})).$$

We add that scaling of step sizes for the Euler scheme has already been used in a bias reduction technique by means of extrapolation, see [2, 35].

3 A Cost Model for Variable Subspace Sampling

In this section we present a cost model for the analysis of multi-level algorithms and, more generally, for the complexity analysis of infinite-dimensional quadrature problems. See [5] for details.

We assume that algorithms for the approximation of $S(f)$ have access to the functionals $f \in F$ via an oracle (subroutine) that provides values $f(x)$ for points $x \in \mathfrak{X}$ or a subset thereof. The cost per evaluation (oracle call) is modelled by a measurable function

$$c : \mathfrak{X} \to \mathbb{N} \cup \{\infty\}.$$

We define the cost of a computation as the sum of the cost of all oracle calls that are made during the computation. For a randomized algorithm the cost defines a random variable (under mild measurability assumptions), which may also depend on $f \in F$. This random variable is henceforth denoted by $\mathrm{cost}_c(A, f)$.

Let us look at the particular case of a randomized quadrature formula

$$A(f) = \sum_{\ell=1}^{n} a_\ell f(X_\ell) \tag{8}$$

with deterministic weights $a_\ell \in \mathbb{R}$ and random elements X_ℓ taking values in $\mathfrak{X}$. This class of randomized algorithms obviously contains every Monte Carlo method (2) and every multi-level algorithm (4), where $n = n_1 + 2\sum_{j=2}^{k} n_j$ in the latter case. The cost of a randomized quadrature formula is given by

$$\mathrm{cost}_c(A, f) = \sum_{\ell=1}^{n} c(X_\ell).$$

Now we discuss specific choices of c. In the cost model given by

$$c = 1 \tag{9}$$

evaluation of an integrand f is possible at any point $x \in \mathfrak{X}$ at cost one. In this model, which is called *full space sampling*, $\mathrm{cost}_c(A, f)$ is the number of evaluations of the integrand. For finite-dimensional quadrature, i.e., if $\dim(\mathfrak{X}) < \infty$, full space sampling is the common choice in the literature.

However, if $\dim(\mathfrak{X}) = \infty$, then full space sampling seems to be too generous and therefore of limited practical relevance. It is more reasonable and partially motivated by the multi-level construction to consider *variable subspace sampling* instead. In any such model we consider a sequence of finite-dimensional subspaces

$$\{0\} \subsetneq \mathfrak{X}_1 \subset \mathfrak{X}_2 \subset \ldots \subset \mathfrak{X},$$

and we define the cost function c by

$$c(x) = \inf\{\dim(\mathfrak{X}_j) : x \in \mathfrak{X}_j\}. \tag{10}$$

In particular, in the setting of a multi-level algorithm (4),

$$\mathfrak{X}_j = \mathrm{span}\left(\bigcup_{i=1}^{j} \varphi^{(i)}(\widetilde{\mathfrak{X}})\right) \tag{11}$$

is a natural choice, and the cost of this algorithm then satisfies

$$\mathrm{cost}_c(A_k, f) \le n_1 \dim(\mathfrak{X}_1) + \sum_{j=2}^{k} n_j \left(\dim(\mathfrak{X}_j) + \dim(\mathfrak{X}_{j-1})\right) \tag{12}$$

in the corresponding variable subspace model.

We write $x_k \preceq y_k$ for sequences of positive real numbers x_k and y_k, if $x_k \le \gamma\, y_k$ holds for every $k \in \mathbb{N}$ with a constant $\gamma > 0$. Furthermore, $x_k \asymp y_k$ means $x_k \preceq y_k$ and $y_k \preceq x_k$.

Example 3. Consider the spaces $\mathfrak{X}_j$ according to (11) in the setting from Example 2. Here, $\mathfrak{X}_j = \mathrm{span}(\varphi^{(j)}(\widetilde{\mathfrak{X}}))$ is the space of piecewise linear functions in $\mathfrak{X} = C([0,1])$ with equidistant breakpoints $i\,2^{-(j-1)}$ and we have $\dim(\mathfrak{X}_j) = 2^{j-1} + 1$. It follows that $1/n_j \sum_{\ell=1}^{n_j} f(U_\ell^{(j)})$ can be computed at cost $n_j\,(2^{j-1}+1)$, and we get

$$\mathrm{cost}_c(A_k, f) \le n_1\, 2 + \sum_{j=2}^{k} n_j\,(2^{j-1} + 2^{j-2} + 2) \asymp \sum_{j=1}^{k} 2^j\, n_j \tag{13}$$

for the multi-level algorithm (4).

Finally we discuss *fixed subspace sampling*. In this case, evaluations are possible only at points in a finite-dimensional subspace

$$\{0\} \subsetneq \mathfrak{X}_0 \subset \mathfrak{X}.$$

For every such evaluation its cost is given by $\dim(\mathfrak{X}_0)$. Thus

$$c(x) = \begin{cases} \dim(\mathfrak{X}_0), & \text{if } x \in \mathfrak{X}_0, \\ \infty, & \text{otherwise.} \end{cases} \tag{14}$$

Clearly fixed subspace sampling constitutes a particular case of variable subspace sampling.

For both kinds of subspace sampling we think of bases associated to the subspaces, so that $c(x)$ is the (minimal) number of real coefficients needed to represent x and this representation is actually submitted to the oracle.

Example 4. Obviously the multi-level Euler algorithm from Example 2 may also be analyzed in the fixed subspace model defined by $\mathfrak{X}_0 = \operatorname{span}(\varphi^{(k)}(\widetilde{\mathfrak{X}}))$, which leads to

$$\operatorname{cost}_c(A_k, f) \le \sum_{j=1}^{k} n_j\,(2^{k-1}+1) \asymp 2^k \sum_{j=1}^{k} n_j.$$

This analysis, however, would be inadequate, since it does not capture the fact that a large proportion of samples is taken in low-dimensional spaces.

Remark 2. We stress that $\operatorname{cost}_c(A, f)$ is a rough measure of the computational cost for applying the algorithm A to the integrand f, since it only takes into account the information cost, which is caused by the evaluations of f. All further operations needed to compute a realization of $A(f)$ are not considered at all.

In a more detailed analysis it is appropriate to take the real number model of computation as a basis for quadrature problems. See [32, 36] for the definition of this model. Informally, a real number algorithm is like a C-program that carries out exact computations with real numbers. Furthermore, a perfect generator for random numbers from $[0, 1]$ is available and elementary functions like exp, ln, etc. can be evaluated. Finally, algorithms have access to the integrands $f \in F$ via the oracle (subroutine). We think that these assumptions are present at least implicitly in most of the work dealing with quadrature problems.

For simplicity we assume that real number operations as well as calls of the random number generator and evaluations of elementary functions are performed at cost one. Furthermore, in case of μ being the distribution of a diffusion process, function values of its drift and diffusion coefficients are provided at cost one, too. Then the total cost of a computation is given by $\operatorname{cost}_c(A, f)$ plus the total number of real number operations, calls of the random number generator, evaluations of elementary functions, and, eventually, function evaluations of drift and diffusion coefficients.

Example 5. In the analysis according to Remark 2 the right-hand side in (13) still is an upper bound for the cost of the multi-level Euler Monte Carlo algorithm, up to a constant. Indeed, the number of arithmetic operations and calls of the random number generator as well as the number of evaluations of the drift coefficient a and diffusion coefficient b that are needed to compute $1/n_j \sum_{\ell=1}^{n_j} f(U_\ell^{(j)})$ are bounded by the number n_j of replications times the number 2^{j-1} of time steps, up to a constant. Hence $\mathrm{cost}_c(A_k, f)$ properly reflects the computation time in practice.

Example 6. SDEs also give rise to finite-dimensional quadrature problems, where μ is the distribution of the solution X at time $t = 1$, say. Then full space sampling provides the appropriate cost model if only the cost of evaluating the functional f is taken into account, and we get

$$\mathrm{cost}_c(A_k, f) \le n_1 + 2 \sum_{j=2}^{k} n_j \asymp \sum_{j=1}^{k} n_j$$

for the multi-level algorithm according to Example 3. In this way, however, we would ignore the impact of the step size on the computational cost of the Euler scheme. Hence an analysis according to Remark 2 is necessary, and then we once more get the right-hand side of (13) as an upper bound for the cost.

4 Analysis of the Multi-level Algorithm

In the sequel we consider a sequence of mappings $\varphi^{(k)}$ with associated bias and variance functions b_k and v_j, respectively, see Section 2. Furthermore, we consider the corresponding variable subspace model with cost function c, see (10) and (11).

4.1 General Results

Suppose that there exist real numbers $M > 1$ and $\gamma, \rho, \tau > 0$ such that

$$|b_k(f)| \le \gamma\, M^{-k\rho}, \quad \big(v_j(f)\big)^{1/2} \le \gamma\, M^{-j\tau}, \quad \dim(\mathfrak{X}_j) \le \gamma\, M^j. \tag{15}$$

We use A_k to denote the multi-level approximation given by (4) with the numbers of replications defined by

$$n_j = \begin{cases} \lceil M^{k\,2\rho - j\,(1/2+\tau)} \rceil, & \text{if } \tau \ge 1/2, \\ \lceil M^{k\,(2\rho+1/2-\tau) - j\,(1/2+\tau)} \rceil, & \text{if } \tau < 1/2 \end{cases}$$

for $j = 1, \ldots, k$. By $a_+ = \max(a, 0)$ we denote the positive part of $a \in \mathbb{R}$.

The following result is due to Giles, see [14] for the case $\rho \ge 1/2$.

Theorem 1. *Assume that* (15) *holds and put*

$$\widetilde{\rho} = \min(\rho, 1/2)$$

as well as

$$\Gamma_k = \mathrm{cost}_c(A_k, f).$$

Then there exists a constant $\widetilde{\gamma} > 0$*, which may depend on* M, γ, ρ, τ*, such that the multi-level approximation* A_k *satisfies*

$$\left(\mathrm{E}(S(f) - A_k(f))^2\right)^{1/2} \leq \widetilde{\gamma} \begin{cases} \Gamma_k^{-\widetilde{\rho}}, & \text{if } \tau > 1/2, \\ \Gamma_k^{-\widetilde{\rho}} \log(\Gamma_k) & \text{if } \tau = 1/2 \leq \rho, \\ \Gamma_k^{-\widetilde{\rho}} (\log(\Gamma_k))^{1/2} & \text{if } \tau = 1/2 > \rho, \\ \Gamma_k^{-\rho/(1+2(\rho-\tau)_+)} & \text{if } \tau < 1/2. \end{cases}$$

Proof. First assume that $\tau \geq 1/2$. Due to (7) and the definition of n_j,

$$\begin{aligned} \mathrm{E}(S(f) - A_k(f))^2 &\preceq \sum_{j=1}^{k} M^{-j\,2\tau}\, M^{-(k\,2\rho - j\,(1/2+\tau))} + M^{-k\,2\rho} \\ &\preceq M^{-k\,2\rho} \sum_{j=1}^{k} M^{-j\,(\tau-1/2)} + M^{-k\,2\rho} \\ &\preceq \begin{cases} M^{-k\,2\rho}, & \text{if } \tau > 1/2, \\ M^{-k\,2\rho}\, k, & \text{if } \tau = 1/2. \end{cases} \end{aligned}$$

By (12),

$$\begin{aligned} \Gamma_k &\preceq 2 \sum_{j=1}^{k} n_j \dim(\mathfrak{X}_j) \\ &\preceq \sum_{j=1}^{k} \left(1 + M^{k\,2\rho - j\,(1/2+\tau)}\right) M^j \\ &\preceq M^k + M^{k\,2\rho} \sum_{j=1}^{k} M^{j\,(1/2-\tau)} \\ &\preceq M^k + \begin{cases} M^{k\,2\rho}, & \text{if } \tau > 1/2, \\ M^{k\,2\rho}\, k, & \text{if } \tau = 1/2. \end{cases} \end{aligned}$$

Furthermore,

$$\Gamma_k \geq n_1 \dim(\mathfrak{X}_1) \succeq M^{k\,2\rho},$$

which implies

$$\log(\Gamma_k) \succeq k.$$

Finally, use the relations

$$M^{-k\,2\rho} \asymp (M^k + M^{k\,2\rho})^{-2\widetilde{\rho}}$$

and

$$M^{-k\,2\rho}\, k \asymp \begin{cases} (M^k + M^{k\,2\rho}\, k)^{-2\widetilde{\rho}}\, k^2, & \text{if } \rho \geq 1/2, \\ (M^k + M^{k\,2\rho}\, k)^{-2\widetilde{\rho}}\, k, & \text{if } \rho < 1/2 \end{cases}$$

to finish the proof for $\tau \geq 1/2$.

Next, consider the case $\tau < 1/2$. Then

$$\begin{aligned} \mathrm{E}(S(f) - A_k(f))^2 &\preceq \sum_{j=1}^{k} M^{-j\,2\tau}\, M^{-(k\,(2\rho+(1/2-\tau))-j\,(1/2+\tau))} + M^{-k\,2\rho} \\ &\asymp M^{-k\,2\rho} \end{aligned}$$

and

$$\begin{aligned} \Gamma_k &\preceq \sum_{j=1}^{k} \bigl(1 + M^{k\,(2\rho+(1/2-\tau))-j\,(1/2+\tau)}\bigr)\, M^j \\ &\asymp M^k + M^{k\,(2\rho+1-2\tau)} \\ &\asymp M^{k\,(1+2\,(\rho-\tau)_+)}, \end{aligned}$$

which completes the proof.

Remark 3. For finite-dimensional quadrature a variant of Theorem 1 is applicable, if the underlying definition of the computational cost is chosen according to Remark 2. Instead of the bound on $\dim(\mathfrak{X}_j)$ in (15) one has to assume that the cost for simulation of $(f(X^{(j)}), f(X^{(j-1)}))$ is bounded by $\gamma\, M^j$. Actually, this variant is close to the analysis of the multi-level algorithm in [14, Theorem 3.1].

Next, we discuss the performance of the classical Monte Carlo approximation under the assumption (15). Clearly,

$$\bigl(\mathrm{Var}(f(X^{(k)}))\bigr)^{1/2} \leq \sum_{j=1}^{k} \bigl(v_j(f)\bigr)^{1/2},$$

so that (15) implies

$$|b_k(f)| \leq \gamma\, M^{-k\,\rho}, \quad \mathrm{Var}(f(X^{(k)})) \leq \gamma, \quad \dim(\mathfrak{X}_k) \leq \gamma\, M^k \tag{16}$$

with some constant $\gamma > 0$.

Assume that (16) holds. We use A'_k to denote the classical Monte Carlo approximation (2) with the number of replications defined by

$$n = \lceil M^{k\,2\rho} \rceil$$

and we put

$$\Gamma_k' = \mathrm{cost}_c(A_k', f),$$

where c is the cost function given by (14) for the appropriate fixed subspace model with $\mathfrak{X}_0 = \mathfrak{X}_k$. Then it is straightforward to check that there exists a constant $\widetilde{\gamma} > 0$, which may depend on M, γ, ρ, such that

$$\left(\mathrm{E}(S(f) - A_k'(f))^2\right)^{1/2} \leq \widetilde{\gamma}\left(\Gamma_k'\right)^{-\rho/(1+2\rho)}. \tag{17}$$

Remark 4. We compare the multi-level algorithm with the classical Monte Carlo approximation on the basis of the upper error bounds provided by Theorem 1 and (17), respectively.

Up to logarithmic factors, the corresponding orders of convergence of these bounds in terms of powers of the cost are given by

$$\theta^*(\rho, \tau) = \begin{cases} \min(1/2, \rho), & \text{if } \tau \geq 1/2, \\ \rho/(1 + 2(\rho - \tau)_+), & \text{if } \tau < 1/2 \end{cases}$$

for the multi-level algorithm, and

$$\theta(\rho) = \rho/(1 + 2\rho)$$

for the classical approach. Put $\widetilde{\tau} = \min(1/2, \tau)$. We always have

$$1 < \frac{\theta^*(\rho, \tau)}{\theta(\rho)} \leq \frac{\theta^*(\widetilde{\tau}, \tau)}{\theta(\widetilde{\tau})} = 1 + 2\widetilde{\tau} \leq 2$$

and

$$\lim_{\rho \to 0} \frac{\theta^*(\rho, \tau)}{\theta(\rho)} = \lim_{\rho \to \infty} \frac{\theta^*(\rho, \tau)}{\theta(\rho)} = 1.$$

4.2 Lipschitz Continuous Integrands

Now we turn to the particular case of Lipschitz continuous integrands, as we assume that

$$|f(x) - f(y)| \leq \|x - y\|_{\mathfrak{X}}, \qquad x, y \in \mathfrak{X}. \tag{18}$$

Moreover, we put $X^{(0)} = 0$ and

$$\delta_j = \left(\mathrm{E}\|X - X^{(j)}\|_{\mathfrak{X}}^2\right)^{1/2}$$

for $j = 0, \ldots, k$.

We immediately get

$$b_k(f) \leq \delta_k$$

as well as

$$v_j(f) \le \mathrm{E}\|X^{(j)} - X^{(j-1)}\|_{\mathfrak{X}}^2 \le (\delta_j + \delta_{j-1})^2$$

for $j = 1, \dots, k$. The analysis for Lipschitz continuous integrands therefore corresponds to the diagonal case $\rho = \tau$ in Theorem 1.

We select mappings $\varphi^{(j)}$ such that

$$\dim \operatorname{span}((\varphi^{(j)}(\widetilde{\mathfrak{X}}))) \le 2^j, \tag{19}$$

and for any integer $N \ge 16$ we define the parameters of the multi-level algorithm A_N by

$$k = \lfloor \log_2(N/8) \rfloor \tag{20}$$

and

$$n_j = \lceil 2^{k-j}/(3k) \rceil \tag{21}$$

for $j = 1, \dots, k$. See [5, Lemma 3] for the following result.

Theorem 2. *Under the assumptions* (18)–(21) *the multi-level algorithm* A_N *satisfies*

$$\mathrm{cost}_c(A_N, f) \le N$$

and

$$\left(\mathrm{E}(S(f) - A_N(f))^2\right)^{1/2} \le 12\sqrt{2}\left(\frac{\log_2 N}{N} \sum_{j=0}^{k} 2^j \delta_j^2\right)^{1/2}.$$

Proof. Note that (19) implies $\dim(\mathfrak{X}_j) \le 2^{j+1} - 2$ for $\mathfrak{X}_j$ according to (11), and therefore

$$\mathrm{cost}_c(A_N, f) \le 2n_1 + \sum_{j=2}^{k} 3\,2^j n_j \le 2^{k+3} \le N$$

follows from (12), (20), and (21). Moreover

$$\begin{aligned}
\mathrm{E}(S(f) - A_N(f))^2 &\le \sum_{j=1}^{k} \frac{(\delta_j + \delta_{j-1})^2}{n_j} + \delta_k^2 \\
&\le \frac{2}{n_1}\delta_0^2 + \sum_{j=1}^{k-1} 2\left(\frac{1}{n_j} + \frac{1}{n_{j+1}}\right)\delta_j^2 + 2\left(\frac{1}{n_k} + \frac{1}{2}\right)\delta_k^2 \\
&\le \frac{18k}{2^k} \sum_{j=0}^{k} 2^j \delta_j^2,
\end{aligned}$$

which completes the proof.

Example 7. In the situation of Example 3 the estimate (19) is satisfied. Moreover, it is well known that (under standard smoothness conditions for the drift and diffusion coefficient of the SDE)

$$\delta_j \le c_p\, 2^{-j/2}$$

for all spaces $\mathfrak{X} = L_p([0,1])$ with $1 \le p < \infty$ and

$$\delta_j \le c_\infty \, j \, 2^{-j/2}$$

for $\mathfrak{X} = C([0,1])$. Hence

$$\left(\mathrm{E}(S(f) - A_N(f))^2\right)^{1/2} \le 12\sqrt{2}\, c_p \, \frac{\log_2 N}{N^{1/2}}$$

for $\mathfrak{X} = L_p([0,1])$ and

$$\left(\mathrm{E}(S(f) - A_N(f))^2\right)^{1/2} \le 12\sqrt{2}\, c_\infty \, \frac{(\log_2 N)^2}{N^{1/2}}$$

for $\mathfrak{X} = C([0,1])$. Analogous results are valid for systems of SDEs. See, e.g., [29, 30] for results and references.

Note that Asian as well as look-back options lead to Lipschitz-continuous integrands. We refer to [14, 16] for a corresponding analysis and numerical experiments using multi-level Euler Monte Carlo algorithms, while a multi-level Milstein Monte Carlo algorithm is employed in [15].

Recall that δ_j is based on the choice of the mapping $\varphi^{(j)} : \widetilde{\mathfrak{X}} \to \mathfrak{X}$. Minimizing δ_j subject to a constraint

$$\dim(\mathrm{span}(\varphi^{(j)}(\widetilde{\mathfrak{X}}))) \le \kappa$$

with $\kappa \in \mathbb{N}$, leads to the notion of average Kolmogorov widths of order two, which are defined by

$$d_\kappa^{(r)} = \inf_{\dim(\mathfrak{X}_0) \le \kappa} \left(\mathrm{E} \inf_{x_0 \in \mathfrak{X}_0} \|X - x_0\|_{\mathfrak{X}}^r \right)^{1/r} \tag{22}$$

with $r = 2$. Here the infimum with respect to $x_0 \in \mathfrak{X}_0$ corresponds to the best approximation of any realization of X by elements from the subspace $\mathfrak{X}_0$, and $\varphi^{(j)}$ is a metric projection of $\widetilde{\mathfrak{X}}$ onto $\mathfrak{X}_0$. The quality of this subspace is measured by an average distance of X to $\mathfrak{X}_0$, and minimization over all subspaces with dimension at most κ leads to the average Kolmogorov width $d_\kappa^{(r)}$. We add that $\lim_{\kappa \to \infty} d_\kappa^{(r)} = 0$, if $\mathfrak{X}$ is separable and $\mathrm{E}\|X\|_{\mathfrak{X}}^r < \infty$. Average Kolmogorov widths and their relation to further scales of approximation quantities for random elements were studied in [4, 27, 28], see also [34].

We now suppose that the sequence of Kolmogorov widths $d_\kappa^{(2)}$ is regularly varying of index $-\rho \in]-\infty, 0[$, i.e.,

$$d_\kappa^{(2)} = \kappa^{-\rho} \, L(\kappa) \tag{23}$$

with a slowly varying function $L : [1, \infty[\to]0, \infty[$. This means that L satisfies $\lim_{x \to \infty} L(r\,x)/L(x) = 1$ for every $r > 0$. By definition L is almost increasing if

$$\inf_{x_0 \le x < y} L(y)/L(x) > 0$$

for some $x_0 > 0$, see [3].

Theorem 3. *Assume that (23) holds. If*

(i) $\rho \neq 1/2$ *or*
(ii) $\rho = 1/2$ *and* L *is bounded or almost increasing,*

then there exists a constant $\gamma > 0$ *and a sequence of multi-level algorithms* A_N *such that*

$$\mathrm{cost}_c(A_N, f) \leq N$$

and

$$\big(\mathrm{E}(S(f) - A_N(f))^2\big)^{1/2} \leq \gamma \begin{cases} N^{-1/2} (\log_2 N)^{1/2} & \text{if } \rho > 1/2, \\ \max(N^{-1/2}, d_N^{(2)}) \log_2 N & \text{if } \rho = 1/2, \\ d_N^{(2)} \log_2 N & \text{if } \rho < 1/2. \end{cases}$$

Proof. Consider the multi-level algorithm A_N from Theorem 2, where the mappings $\varphi^{(j)}$ are chosen such that

$$\delta_j \leq 2\, d_{2^j}^{(2)}.$$

By assumption,

$$d_{2^j}^{(2)} = 2^{-j\rho}\, L(2^j).$$

Since $\mathrm{cost}_c(A_N, f) \leq N$ and

$$\mathrm{E}(S(f) - A_N(f))^2 \leq 576\, \frac{\log_2 N}{N} \sum_{j=0}^{k} 2^{j(1-2\rho)}\, (L(2^j))^2,$$

it remains to show that

$$\sum_{j=0}^{k} 2^{j(1-2\rho)}\, (L(2^j))^2 \preceq \big(\max\big(1, N^{1-2\rho}\, (L(N))^2\big) \log_2 N\big)^{\widetilde{\rho}} \tag{24}$$

with $\widetilde{\rho} = 0$ if $\rho > 1/2$ and $\widetilde{\rho} = 1$ otherwise.

First assume that $\rho > 1/2$. Then $(1 - 2\rho)/2 < 0$, which implies

$$\lim_{j \to \infty} 2^{j(1-2\rho)/2}\, (L(2^j))^2 = 0,$$

since the function L^2 is slowly varying as well. Consequently,

$$\sum_{j=0}^{k} 2^{j(1-2\rho)}\, (L(2^j))^2 \preceq \sum_{j=0}^{k} 2^{j(1-2\rho)/2} \preceq 1.$$

Next assume that $0 < \rho < 1/2$. Then

$$(L(x))^2 / (L(y))^2 \preceq (y/x)^{1-2\rho}$$

for $1 \le x \le y$, and therefore

$$\sum_{j=0}^{k} 2^{j(1-2\rho)} (L(2^j))^2 \preceq \sum_{j=0}^{k} 2^{j(1-2\rho)} (N/2^j)^{1-2\rho} (L(N))^2$$
$$\preceq N^{1-2\rho} (L(N))^2 \log_2 N.$$

Finally, consider the case $\rho = 1/2$. By assumption L is bounded or almost increasing, and therefore

$$\sum_{j=0}^{k} 2^{j(1-2\rho)} (L(2^j))^2 = \sum_{j=0}^{k} (L(2^j))^2 \preceq \max(1, (L(N))^2) \log_2 N,$$

which completes the proof.

Remark 5. The error bound in Theorem 3 can be slightly improved in the case $\rho < 1/2$ if the slowly varying function L is almost increasing. Then

$$\big(\mathrm{E}(S(f) - A_N(f))^2\big)^{1/2} \le \gamma\, d_N^{(2)} (\log_2 N)^{1/2}$$

for some constant $\gamma > 0$.

Example 8. Consider an SDE, and let $\mathfrak{X} = C([0,1], \mathbb{R}^m)$ or $\mathfrak{X} = L_p([0,1], \mathbb{R}^m)$ with $1 \le p < \infty$. Then (under appropriate smoothness conditions on the coefficients of the SDEs)

$$d_\kappa^{(2)} \asymp \kappa^{-1/2},$$

see [5, Prop. 3]. Hence the estimate from Example 7 can be slightly improved for $\mathfrak{X} = C([0,1], \mathbb{R}^m)$ to an upper bound of order $\log_2 N / N^{1/2}$.

Remark 6. Our proof of Theorem 3 is based on inequality (24), which is equivalent to

$$\sum_{j=0}^{k} (L(2^j))^2 \preceq \max\big(1, (L(2^k))^2\big)\, k \tag{25}$$

in the case $\rho = 1/2$. Note that the latter inequality does not hold without an additional assumption on the slowly varying function L. For example, consider the function

$$L(x) = \exp\big((\log_2 x)^{1/3} \cos((\log_2 x)^{1/3})\big)$$

with $x \ge 1$. Then L is slowly varying and we have

$$\limsup_{k\to\infty} \big(\max\big(1, (L(2^k))^2\big)\, k\big)^{-1} \sum_{j=0}^{k} (L(2^j))^2 = \infty.$$

5 Minimal Errors in Different Cost Models

In order to determine the power of variable subspace sampling, and in particular the power of multi-level algorithms, we consider the worst case errors and cost of randomized algorithms A on a class F of integrands $f : \mathfrak{X} \to \mathbb{R}$. These quantities are defined by

$$e(A) = \sup_{f \in F} \big(\mathrm{E}(S(f) - A(f))^2\big)^{1/2}$$

and

$$\mathrm{cost}_c(A) = \sup_{f \in F} \mathrm{E}\,\mathrm{cost}_c(A, f),$$

if the cost per evaluation of $f \in F$ is modelled by $c : \mathfrak{X} \to \mathbb{N} \cup \{\infty\}$.

Actually we have already used the worst case point of view in the previous section. For instance, with $F = \mathrm{Lip}(1)$ denoting the class of all functionals f that satisfy (18), the error bound from Theorem 2 is equivalent to

$$e(A_N) \le 12\sqrt{2} \left(\frac{\log_2 N}{N}\right)^{1/2} \left(\sum_{j=0}^{k} 2^j \, \delta_j^2\right)^{1/2},$$

and obviously

$$\mathrm{cost}_c(A_N) \le N.$$

We extend our analysis beyond the class of multi-level algorithms, as we consider the class $\mathcal{A}^{\mathrm{ran}}$ of all randomized algorithms. See, e.g., [5] for the formal definition. Here we only mention that $\mathcal{A}^{\mathrm{ran}}$ contains in particular all random variables of the form

$$A(f) = \phi(f(X_1), \ldots, f(X_n))$$

with any choice of a joint distribution of $(X_1, \ldots, X_n)$ on $\mathfrak{X}^n$ and any measurable mapping $\phi : \mathbb{R}^n \to \mathbb{R}$. Note that randomized quadrature formulas are a particular instance thereof, see (8).

For comparing the power of different sampling regimes it does not suffice to establish upper bounds for the error and cost of specific algorithms. Instead, one has to study minimal errors and to establish lower bounds.

Let C_{fix} denote the set of all cost functions given by (14) with any finite-dimensional subspace $\{0\} \subsetneq \mathfrak{X}_0 \subset \mathfrak{X}$, let C_{var} denote the set of all cost functions given by (10) with any increasing sequence of finite-dimensional subspaces $\{0\} \subsetneq \mathfrak{X}_i \subset \mathfrak{X}$, and let C_{full} consist of the constant cost function one, see (9). For

$$\mathrm{samp} \in \{\mathrm{fix}, \mathrm{var}, \mathrm{full}\}$$

and $N \in \mathbb{N}$ we introduce the N-th minimal error

$$e^{\mathrm{ran}}_{N,\mathrm{samp}} = \inf\{e(A) : A \in \mathcal{A}^{\mathrm{ran}},\ \exists\, c \in C_{\mathrm{samp}} : \mathrm{cost}_c(A) \le N\}.$$

According to this definition a most favorable cost model $c \in C_{\text{samp}}$ is used for assessing the quality of an algorithm $A \in \mathcal{A}^{\text{ran}}$. We add that minimal errors are key quantities in information-based complexity, see, e.g., [30, 31, 34, 36].

Clearly

$$e^{\text{ran}}_{N,\text{full}} \le e^{\text{ran}}_{N,\text{var}} \le e^{\text{ran}}_{N,\text{fix}},$$

and these quantities allow us to compare the different sampling regimes. For instance, variable subspace sampling is superior to fixed subspace sampling for a class of integrands F and a measure μ iff the minimal errors $e^{\text{ran}}_{N,\text{var}}$ are significantly smaller than the minimal errors $e^{\text{ran}}_{N,\text{fix}}$. Note that a lower bound for $e^{\text{ran}}_{N,\text{fix}}$ and an upper bound for $e^{\text{ran}}_{N,\text{var}}$ are needed to establish this conclusion. Conversely, a lower bound for $e^{\text{ran}}_{N,\text{var}}$ and an upper bound for $e^{\text{ran}}_{N,\text{fix}}$ are needed to prove that variable subspace sampling is not superior to fixed subspace sampling.

6 Optimal Quadrature of Lipschitz Functionals

Throughout this section we assume that

$$F = \text{Lip}(1).$$

In this case the minimal errors for the quadrature problem can be estimated from above and below in terms of average Kolmogorov widths, see (22), and quantization numbers. A partial result was already formulated in Theorem 3.

The quantization numbers of order $r \ge 1$ are defined by

$$q^{(r)}_n = \inf_{|\mathfrak{X}_0| \le n} \left(\mathrm{E} \min_{x_0 \in \mathfrak{X}_0} \|X - x_0\|^r_{\mathfrak{X}} \right)^{1/r}.$$

Both, average Kolmogorov widths and quantization numbers correspond to a best approximation from optimally chosen subsets $\mathfrak{X}_0 \subseteq \mathfrak{X}$, subject to a constraint on the dimension of the linear subspace $\mathfrak{X}_0$ or the size of the finite set $\mathfrak{X}_0$. Quantization of random elements X that take values in finite-dimensional spaces $\mathfrak{X}$ has been studied since the late 1940's, and we refer to the monograph [17] for an up-to-date account. For random elements X taking values in infinite-dimensional spaces $\mathfrak{X}$, quantization has been studied since about ten years. Results are known for Gaussian processes, see, e.g., [7, 10, 12, 24, 25], and for diffusion processes, see [5, 8, 9, 26].

In the sequel we assume that $q^{(1)}_n < \infty$. Clearly, $\lim_{n\to\infty} q^{(r)}_n = 0$ if $\mathfrak{X}$ is separable and $\mathrm{E}\|X\|^r_{\mathfrak{X}} < \infty$.

In the following theorem the upper bounds on $e^{\text{ran}}_{N,\text{full}}$ and $e^{\text{ran}}_{N,\text{fix}}$ as well as all lower bounds are due to [5]. See Theorem 3 for the upper bound on $e^{\text{ran}}_{N,\text{var}}$.

Theorem 4. *For full space sampling*

$$N^{1/2} \sup_{n \ge 4N} (q^{(1)}_{n-1} - q^{(1)}_n) \preceq e^{\text{ran}}_{N,\text{full}} \preceq N^{-1/2} q^{(2)}_N.$$

For variable subspace sampling

$$\max(e^{\mathrm{ran}}_{N,\mathrm{full}}, d^{(1)}_{2N}) \preceq e^{\mathrm{ran}}_{N,\mathrm{var}},$$

and, under the assumption of Theorem 3,

$$e^{\mathrm{ran}}_{N,\mathrm{var}} \preceq \max(N^{-1/2}, d^{(2)}_N) \log_2 N.$$

For fixed subspace sampling

$$\inf_{\kappa\, n\le N} \max(e^{\mathrm{ran}}_{n,\mathrm{full}}, d^{(1)}_\kappa) \preceq e^{\mathrm{ran}}_{N,\mathrm{fix}} \preceq \inf_{\kappa\, n\le N} (n^{-1/2} + d^{(2)}_\kappa).$$

Remark 7. Clearly $d^{(r)}_\kappa$ and $q^{(r)}_n$ only depend on the distribution μ of X, and they can equivalently be defined in terms of the Wasserstein distance on the space of Borel probability measures on $\mathfrak{X}$. See, e.g., [5] for these facts and for further references. Thus Theorem 4 relates quadrature of Lipschitz functionals by means of randomized algorithms to approximation of μ by distributions with finite support and distributions concentrated on finite-dimensional subspaces, and the latter constraints reflect the restrictions on evaluation of the functionals in the three sampling regimes.

We add that an analogue analysis can be carried out for quadrature of Lipschitz functionals by means of deterministic algorithms only. In the setting of full space sampling it is well known that this quadrature problem is equivalent to the quantization problem in the sense that the corresponding minimal errors satisfy

$$e^{\mathrm{det}}_{N,\mathrm{full}} = q^{(1)}_N. \tag{26}$$

See [5] for details and for further references.

Remark 8. The following algorithms achieve the upper bounds in Theorem 4. For full space sampling we may use quantization for variance reduction, see [5, Thm. 2] for details. For variable subspace sampling we may use the multi-level algorithm according to Theorem 3. For fixed subspace sampling we may choose mappings $\varphi^{(k)}$ such that

$$\dim(\mathrm{span}(\varphi^{(k)}(\widetilde{\mathfrak{X}}))) \le k$$

and

$$\bigl(\mathrm{E}\|X - X^{(k)}\|^2_{\mathfrak{X}}\bigr)^{1/2} \le 2\, d^{(2)}_k$$

and employ the classical Monte Carlo algorithm (2), see [5, Thm. 4].

6.1 Gaussian Measures

In this section we study the case of a zero mean Gaussian measure μ on a separable Banach space $\mathfrak{X}$. In order to apply Theorem 4 we have to know the asymptotic

behaviour of the average Kolmogorov widths and the quantization numbers. To this end we consider the small ball function

$$\psi(\varepsilon) = -\ln \mu(\{x \in \mathfrak{X} : \|x\| \le \varepsilon\}), \quad \varepsilon > 0,$$

of μ, and we assume that there exist constants $\alpha > 0$ and $\beta \in \mathbb{R}$ such that

$$\psi(\varepsilon) \asymp \varepsilon^{-\alpha} (\ln \varepsilon^{-1})^{\beta} \tag{27}$$

as ε tends to zero. This implies

$$q_n^{(r)} \asymp (\ln n)^{-1/\alpha} (\ln \ln n)^{\beta/\alpha},$$

and

$$d_\kappa^{(r)} \asymp \kappa^{-1/\alpha} (\ln \kappa)^{\beta/\alpha},$$

see [7, Thm. 3.1.2] and [4, Cor. 4.7.2], respectively.

Typically, (27) holds for infinite-dimensional spaces $\mathfrak{X}$, see, e.g., [23] for results and further references. For example, if μ is the distribution of a d-dimensional Brownian sheet on $\mathfrak{X} = L_2([0,1]^d)$ then $\alpha = 2$ and $\beta = 2(d-1)$, see [6, 13].

Essentially the following results are a consequence of Theorems 2 and 4, see [5, Sec. 8].

Theorem 5. *For variable subspace sampling the minimal errors are bounded as follows.*
If $\alpha > 2$, then

$$N^{-1/\alpha} (\ln N)^{\beta/\alpha} \preceq e_{N,\mathrm{var}}^{\mathrm{ran}} \preceq N^{-1/\alpha} (\ln N)^{\beta/\alpha + 1/2}.$$

If $\alpha = 2$ and $\beta \neq -1$, then

$$N^{-1/2} (\ln N)^{\beta/2} \preceq e_{N,\mathrm{var}}^{\mathrm{ran}} \preceq N^{-1/2} (\ln N)^{(\beta/2+1/2)_+ + 1/2}.$$

If $\alpha = 2$ and $\beta = -1$, then

$$N^{-1/2} (\ln N)^{-1/2} \preceq e_{N,\mathrm{var}}^{\mathrm{ran}} \preceq N^{-1/2} (\ln N)^{1/2} (\ln \ln N)^{1/2}.$$

If $0 < \alpha < 2$, then

$$e_{N,\mathrm{var}}^{\mathrm{ran}} \preceq N^{-1/2} (\ln N)^{1/2}$$

and

$$\limsup_{N \to \infty} e_{N,\mathrm{var}}^{\mathrm{ran}} N^{1/2} (\ln N)^{1+1/\alpha} (\ln \ln N)^{-\beta/\alpha} > 0.$$

Theorem 5 provides sharp upper and lower bounds on the minimal errors for variable subspace sampling, up to logarithmic factors and up to the fact that one of the lower bounds is established only for an infinite sequence of integers N. The order of the polynomial term $N^{-\gamma_{\mathrm{var}}}$ is

$$\gamma_{\mathrm{var}} = \min(1/2, 1/\alpha).$$

We add that the upper bounds hold for suitable multi-level algorithms, which thus turn out to be almost optimal for variable subspace sampling, see [5].

Theorem 6. *For full space sampling the minimal errors satisfy*

$$e_{N,\mathrm{full}}^{\mathrm{ran}} \preceq N^{-1/2} (\ln N)^{-1/\alpha} (\ln \ln N)^{\beta/\alpha}$$

and

$$\limsup_{N\to\infty} e_{N,\mathrm{full}}^{\mathrm{ran}} N^{1/2} (\ln N)^{1+1/\alpha} (\ln \ln N)^{-\beta/\alpha} > 0.$$

Roughly speaking, Theorem 6 determines the asymptotic behaviour of the minimal errors for full space sampling, and the order of the polynomial term $N^{-\gamma_{\mathrm{full}}}$ is

$$\gamma_{\mathrm{full}} = 1/2.$$

We conclude that variable subspace sampling is as powerful as full subspace sampling iff $\alpha \le 2$ and, consequently, suitable multi-level algorithms are almost optimal even in a much stronger sense in this case. As a specific example we mention any fractional Brownian motion with Hurst parameter $H \in]0,1[$ either on $\mathfrak{X} = C([0,1])$ or on $\mathfrak{X} = L_p([0,1])$ with $1 \le p < \infty$. In all cases we have $\alpha = 1/H$ and therefore $\gamma_{\mathrm{full}} = \gamma_{\mathrm{var}}$ iff $H \ge 1/2$.

Theorem 7. *For fixed subspace sampling the minimal errors satisfy*

$$e_{N,\mathrm{fix}}^{\mathrm{ran}} \preceq N^{-1/(2+\alpha)} (\ln N)^{\beta/(2+\alpha)}$$

and

$$\limsup_{N\to\infty} e_{N,\mathrm{fix}}^{\mathrm{ran}} N^{1/(2+\alpha)} (\ln N)^{(2+2\alpha-\alpha\beta)/(\alpha(2+\alpha))} (\ln \ln N)^{-2\beta/(\alpha(2+\alpha))} > 0.$$

Ignoring again logarithmic factors as well as the shortcoming of the lower bound result, Theorem 7 states that the minimal errors for fixed subspace sampling behave like $N^{-\gamma_{\mathrm{fix}}}$ with order

$$\gamma_{\mathrm{fix}} = 1/(2+\alpha).$$

Clearly, $\gamma_{\mathrm{fix}} < \gamma_{\mathrm{var}}$ for all $\alpha > 0$ so that variable subspace sampling is always superior to fixed subspace sampling, and this superiority is maximal for $\alpha = 2$ when $\gamma_{\mathrm{var}} = 1/2 = 2\gamma_{\mathrm{fix}}$. The dependence of the orders γ_{var}, γ_{full}, and γ_{fix} on the parameter α of the small ball function (27) is illustrated in Figure 6.1, which summarizes the essential content of Theorems 5 to 7.

6.2 Diffusion Processes

In this section we consider the distribution μ of an m-dimensional diffusion process X on the space $\mathfrak{X} = C = C([0,1],\mathbb{R}^m)$ or on a space $\mathfrak{X} = L_p = L_p([0,1],\mathbb{R}^m)$ with $1 \le p < \infty$. More precisely, X is given by

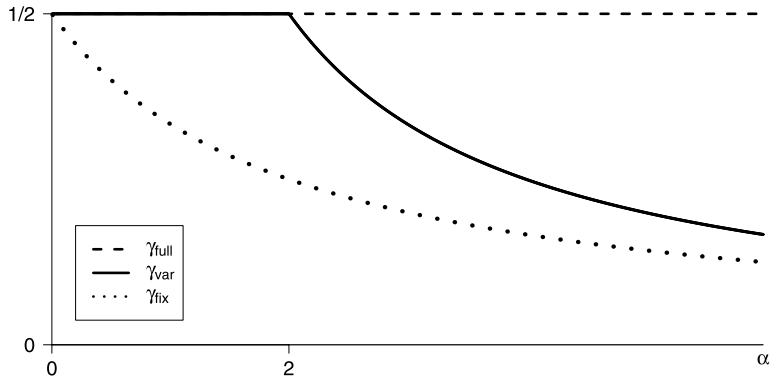


Fig. 1 Dependence of γ_{var}, γ_{full}, γ_{fix} on α.

$$\begin{aligned} dX_t &= a(X_t)\,dt + b(X_t)\,dW_t, \\ X_0 &= u_0 \in \mathbb{R}^m \end{aligned} \tag{28}$$

for $t \in [0,1]$ with an m-dimensional Brownian motion W, and we assume that the following conditions are satisfied:

(i) $a : \mathbb{R}^m \to \mathbb{R}^m$ is Lipschitz continuous
(ii) $b : \mathbb{R}^m \to \mathbb{R}^{m\times m}$ has bounded first and second order partial derivatives and is of class C^∞ in some neighborhood of u_0
(iii) $\det b(u_0) \neq 0$

We first present bounds for the quantization numbers and the average Kolmogorov widths. Let $\mathfrak{X} = C$ or $\mathfrak{X} = L_p$. The quantization numbers $q_n^{(r)}$ satisfy

$$q_n^{(r)} \asymp (\ln n)^{-1/2}$$

for every $r > 0$. The average Kolmogorov widths $d_\kappa^{(r)}$ satisfy

$$d_\kappa^{(r)} \asymp \kappa^{-1/2}$$

for every $r > 0$. See [5, Prop. 3].

The estimates from Theorems 5–7 with $\alpha = 2$ and $\beta = 0$ are valid, too, in the diffusion case, see [5, Sec. 9].

Theorem 8. *Let $\mathfrak{X} = C$ or $\mathfrak{X} = L_p$. For full space sampling the minimal errors satisfy*

$$e_{N,\text{full}}^{\text{ran}} \preceq N^{-1/2}\,(\ln N)^{-1/2}$$

and

$$\limsup_{N\to\infty} e_{N,\text{full}}^{\text{ran}}\, N^{1/2}\,(\ln N)^{3/2} > 0.$$

For fixed subspace sampling the minimal errors satisfy

$$e^{\mathrm{ran}}_{N,\mathrm{fix}} \preceq N^{-1/4}$$

and

$$\limsup_{N\to\infty} e^{\mathrm{ran}}_{N,\mathrm{fix}}\, N^{1/4}\,(\ln N)^{3/4} > 0.$$

For variable subspace sampling the minimal errors satisfy

$$N^{-1/2} \preceq e^{\mathrm{ran}}_{N,\mathrm{var}} \preceq N^{-1/2}\ln N.$$

For full space and fixed subspace sampling the lower bounds from Theorem 8 can be improved in the case $\mathfrak{X} = C$, see [5, Thm. 12].

Theorem 9. *Let $\mathfrak{X} = C$. For full space sampling the minimal errors satisfy*

$$e^{\mathrm{ran}}_{N,\mathrm{full}} \succeq N^{-1/2}\,(\ln N)^{-3/2}.$$

For fixed subspace sampling the minimal errors satisfy

$$e^{\mathrm{ran}}_{N,\mathrm{fix}} \succeq N^{-1/4}\,(\ln N)^{-3/4}.$$

Remark 9. For a Gaussian measure μ on an infinite-dimensional space, as studied in Section 6.1, as well as for μ being the distribution of the solution of an SDE on the path space, the corresponding quantization numbers $q_N^{(1)}$ essentially behave like powers of $\ln N$, asymptotically. Observing (26) we conclude that in both cases quadrature of arbitrary Lipschitz functionals is intractable by means of deterministic algorithms.

7 Concluding Remarks

The majority of results presented in this survey is concerned with Lipschitz continuous integrands f. The multi-level approach, however, is not at all linked to any kind of smoothness assumption on f. Instead, only bias and variance estimates are needed, see Theorem 1, and there are good reasons to consider classes F of integrands that either contain non-Lipschitz functionals or are substantially smaller than Lip(1).

Motivated by applications from computational finance non-continuous integrands are considered in [1] and [16]. These authors establish new results on strong approximation of SDEs, which in turn are used in the multi-level approach. In particular the computation of the expected payoff for digital and barrier options is covered by this work.

For finite-dimensional spaces $\mathfrak{X}$ much smaller classes F of integrands than Lip(1) are studied since long. With a view towards infinite-dimensional integration as a limiting case, tractability results for d-dimensional integration are most

interesting, since they provide bounds on the minimal errors with an explicit dependence on the dimension d. We refer to the recent monograph [33]. Here weighted Hilbert spaces with a reproducing kernel play an important role, and in this setting full space sampling for infinite-dimensional quadrature case has already been analyzed in [21].

As for the class F of functionals, the multi-level approach also does not rely on specific properties of the measure μ. Actually, only suitable subspaces have to be identified and the simulation of two-level couplings of corresponding distributions must be feasible. So far, most of the work on multi-level algorithms is dealing with SDEs that are driven by a Brownian motion, and results for Gaussian measures μ are available as well. Recent progress in a different direction is made in [11], which provides the construction and analysis of a multi-level algorithm for Lévy-driven SDEs.

Acknowledgements We are grateful to the editor, the referees, as well as to Rainer Avikainen and Mike Giles for their valuable remarks. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) within the Priority Program 1324.

References

1. R. Avikainen, On irregular functionals of SDEs and the Euler scheme, Finance and Stochastics **13**, 381–401 (2009).
2. V. Bally, D. Talay, The law of the Euler scheme for stochastic differential equations. I: Convergence rate of the distribution function, Probab. Theory Relat. Fields **104**, 43–60 (1996).
3. N.H. Bingham, C.M. Goldie, J.L. Teugels, Regular Variation, Cambridge Univ. Press, Cambridge (1987).
4. J. Creutzig, Approximation of Gaussian Random Vectors in Banach Spaces, Ph.D. Dissertation, Fakultät für Mathematik und Inf., Friedrich-Schiller Universität Jena (2002).
5. J. Creutzig, S. Dereich, T. Müller-Gronbach, K. Ritter, Infinite-dimensional quadrature and approximation of distributions, Found. Comput. Math. **9**, 391–429 (2009).
6. E. Csáki, On small values of the square integral of a multiparameter Wiener process, in: Statistics and Probability, (J. Mogyorodi, I. Vincze, W. Wertz, eds.), pp. 19–26, Reidel, Dordrecht (1984).
7. S. Dereich, High Resolution Coding of Stochastic Processes and Small Ball Probabilities, Ph.D. Dissertation, Institut für Mathematik, TU Berlin (2003).
8. S. Dereich, The coding complexity of diffusion processes under supremum norm distortion, Stochastic Processes Appl. **118**, 917–937 (2008).
9. S. Dereich, The coding complexity of diffusion processes under $L^p[0,1]$-norm distortion, Stochastic Processes Appl. **118**, 938–951 (2008).
10. S. Dereich, F. Fehringer, A. Matoussi, M. Scheutzow, On the link between small ball probabilities and the quantization problem, J. Theoret. Probab. **16**, 249–265 (2003).
11. S. Dereich, F. Heidenreich, A multilevel Monte Carlo algorithm for Lévy driven stochastic differential equations, work in progress.
12. S. Dereich, M. Scheutzow, High-resolution quantization and entropy coding for fractional Brownian motion, Electron. J. Probab. **11**, 700–722 (2006).
13. J.A. Fill, F. Torcaso, Asymptotic analysis via Mellin transforms for small deviations in L^2-norm of integrated Brownian sheets, Probab.Theory Relat. Fields **130**, 259–288 (2004).
14. M.B. Giles, Multilevel Monte Carlo path simulation, Oper. Res. **56**, 607–617 (2008).

15. M.B. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, in: Monte Carlo and Quasi-Monte Carlo Methods 2006 (A. Keller, S. Heinrich, H. Niederreiter, eds.), pp. 343–358, Springer-Verlag, Berlin (2008).
16. M.B. Giles, D.J. Higham, X. Mao, Analysing multi-level Monte Carlo for options with non-globally Lipschitz payoff, Finance and Stochastics **13**, 403–413 (2009).
17. S. Graf, H. Luschgy, Foundations of Quantization for Probability Distributions, Lect. Notes in Math. **1730**, Springer-Verlag, Berlin (2000).
18. S. Heinrich, Monte Carlo complexity of global solution of integral equations, J. Complexity **14**, 151–175 (1998).
19. S. Heinrich, Multilevel Monte Carlo methods, in: Large Scale Scientific Computing, Lect. Notes in Comp. Sci. **2179** (S. Margenov, J. Wasniewski, P. Yalamov, eds.), pp. 58–67, Springer-Verlag, Berlin (2001).
20. S. Heinrich, E. Sindambiwe, Monte Carlo complexity of parametric integration, J. Complexity **15**, 317–341 (1999).
21. F.J. Hickernell, X. Wang, The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimension, Math.Comp. **71**, 1641–1661 (2001).
22. A. Kebaier, Statistical Romberg extrapolation: a new variance reduction method and applications to option pricing, Ann. Appl. Prob. **15**, 2681–2705 (2005).
23. W.V. Li, Q.-M. Shao, Gaussian processes: inequalities, small ball probabilities and applications, in: Stochastic Processes: Theory and Methods, Handbook of Statist., Vol. 19, (D.N. Shanbhag, C.R. Rao, eds.), pp. 533–597, North-Holland, Amsterdam (2001).
24. H. Luschgy, G. Pagès, Functional quantization of Gaussian processes, J. Funct. Anal. **196**, 486–531 (2002).
25. H. Luschgy, G. Pagès, Sharp asymptotics of the functional quantization problem for Gaussian processes, Ann. Appl. Prob. **32**, 1574–1599 (2004).
26. H. Luschgy, G. Pagès, Functional quantization of a class of Brownian diffusions: a constructive approach, Stochastic Processes Appl. **116**, 310–336 (2006).
27. V.E. Maiorov, Widths and distribution of values of the approximation functional on the Sobolev space with measure, Constr. Approx. **12**, 443–462 (1996).
28. P. Mathé, *s*-numbers in information-based complexity, J. Complexity **6** 41–66 (1990).
29. T. Müller-Gronbach, Optimal uniform approximation of systems of stochastic differential equations, Ann. Appl. Prob. **12**, 664–690 (2002).
30. T. Müller-Gronbach, K. Ritter, Minimal errors for strong and weak approximation of stochastic differential equations, in: Monte Carlo and Quasi-Monte Carlo Methods 2006 (A. Keller, S. Heinrich, H. Niederreiter, eds.), pp. 53–82, Springer-Verlag, Berlin (2008).
31. E. Novak, Deterministic and Stochastic Error Bounds in Numerical Analysis, Lect. Notes in Math. **1349**, Springer-Verlag, Berlin (1988).
32. E. Novak, The real number model in numerical analysis, J. Complexity **11**, 57–73 (1995).
33. E. Novak, H. Woźniakowski, Tractability of Multivariate Problems, Vol. I: Linear Information, European Mathematical Society, Zürich (2008).
34. K. Ritter, Average-Case Analysis of Numerical Problems, Lect. Notes in Math. **1733**, Springer-Verlag, Berlin (2000).
35. D. Talay, L. Tubaro, Expansion of the global error for numerical schemes solving stochastic differential equations, Stochastic Anal. Appl. **8**, 483–509 (1990).
36. J.F. Traub, G.W. Wasilkowski, H. Woźniakowski, Information-Based Complexity, Academic Press, New York (1988).
37. G.W. Wasilkowski, H. Woźniakowski, On tractability of path integration, J. Math. Phys. **37**, 2071–2088 (1996).

Markov Chain Monte Carlo Algorithms: Theory and Practice

Jeffrey S. Rosenthal

Abstract We describe the importance and widespread use of Markov chain Monte Carlo (MCMC) algorithms, with an emphasis on the ways in which theoretical analysis can help with their practical implementation. In particular, we discuss how to achieve rigorous quantitative bounds on convergence to stationarity using the coupling method together with drift and minorisation conditions. We also discuss recent advances in the field of adaptive MCMC, where the computer iteratively selects from among many different MCMC algorithms. Such adaptive MCMC algorithms may fail to converge if implemented naively, but they will converge correctly if certain conditions such as Diminishing Adaptation are satisfied.

1 Introduction

Markov chain Monte Carlo (MCMC) algorithms were first introduced in statistical physics [17], and gradually found their way into image processing [12] and statistical inference [15, 32, 11, 33]. Their main use is to sample from a complicated probability distribution $\pi(\cdot)$ on a state space $\mathcal{X}$ (which is usually high-dimensional, and often continuous, e.g. an open subset of $\mathbf{R}^d$). In particular, MCMC has revolutionized the field of Bayesian statistical inference, where $\pi(\cdot)$ would usually be a posterior distribution which is otherwise intractable but which can (hopefully) be easily sampled using MCMC.

In brief, MCMC proceeds as follows. We define a Markov chain $P(x, \cdot)$ on $\mathcal{X}$ that leaves $\pi(\cdot)$ stationary. We first sample X_0 from some (simple) *initial distribution* on $\mathcal{X}$. We then iteratively sample X_n from $P(X_{n-1}, \cdot)$, for $n = 1, 2, 3, \ldots$. The hope is that for "large enough" n, the distribution of X_n will be approximately equal to $\pi(\cdot)$,

Department of Statistics
University of Toronto
Toronto, Ontario, Canada
url: http://probability.ca/jeff/

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_9,

i.e. $P(X_n \in A) \approx \pi(A)$ for all measurable $A \subseteq \mathcal{X}$. If so, then X_n is approximately a *sample* from $\pi(\cdot)$. And, once we can generate samples from $\pi(\cdot)$, then we can easily use those samples to approximately compute any quantities of interesting involving probabilities or expectations with respect to $\pi(\cdot)$.

Such algorithms have become extremely popular in Bayesian statistics and other areas. At last count, the *MCMC Preprint Service* lists about seven thousand research papers, and the phrase "Markov chain Monte Carlo" elicits over three hundred thousand hits in *Google*. As a result of this popularity, many people are using MCMC algorithms without possessing much knowledge of the theory of Markov chains or probability, and there has been some divorce between theoreticians and practitioners of MCMC.

Despite this, there are a number of ways in which theory has had, and continues to have, important implications for the practical use of MCMC. In this paper, we concentrate on two areas: theoretical bounds on time to stationarity (Section 3), and validity of adaptive MCMC algorithms (Section 4); for additional background see e.g. [24] and the references therein.

2 Asymptotic Convergence

The first and most basic question about MCMC is whether it converges asymptotically, i.e. whether it is true that for sufficiently large n, the distribution of X_n is close to $\pi(\cdot)$. This is a bare minimal requirement for an MCMC algorithm to be "valid".

On a finite state space $\mathcal{X}$, it is well known that if a time-homogeneous Markov chain is irreducible and aperiodic, then it has a unique stationarity distribution $\pi(\cdot)$, to which it will converge in distribution as $n \to \infty$.

In this context, "irreducible" means that for all $x, y \in \mathcal{X}$, y is *accessible* from x, i.e. there is $n \in \mathbf{N}$ such that $P^n(x, \{y\}) \equiv \mathbf{P}(X_n \in \{y\} \,|\, X_0 = x) > 0$. This is clearly impossible on a continuous (uncountable) state space $\mathcal{X}$, since the subset $\{y \in \mathcal{X} : \exists n \in \mathbf{N},\ P^n(x, \{y\}) > 0\}$ is always countable. However, it is possible to weaken the condition "irreducible" to that of *ϕ-irreducible*, meaning there exists a non-zero σ-finite measure ϕ on $\mathcal{X}$ such that for all measurable $A \subseteq \mathcal{X}$ with $\phi(A) > 0$, and all $x \in \mathcal{X}$, there exists $n \in \mathbf{N}$ such that $P^n(x, A) > 0$. It is then well known (see e.g. [18, 33, 24]) that if a Markov chain (on a general countably-generated state space $\mathcal{X}$) is ϕ-irreducible and aperiodic, and possesses an stationarity probability distribution $\pi(\cdot)$ (which is no longer guaranteed), then asymptotic convergence still holds, and in fact

$$\lim_{n \to \infty} \sup_{A \subseteq \mathcal{X}} |P^n(x, A) - \pi(A)| \;=\; 0, \qquad \pi\text{-}a.e.\ x \in \mathcal{X}. \tag{1}$$

For example, if the Markov chain transition probabilities all have positive densities with respect to Lebesgue measure on $\mathbf{R}^d$, then we can simply let $\phi(\cdot)$ be Lebesgue measure, to see that ϕ-irreducibility is satisfied (and aperiodicity follows

immediately as well). More generally, ϕ-irreducibility follows if the n-step transitions $P^n(x,\cdot)$ have positive densities on subsets which expand to $\mathcal{X}$ as $n \to \infty$.

Such considerations are usually sufficient to easily guarantee asymptotic convergence of MCMC algorithms which arise in practice. However, results such as (1) only apply when $n \to \infty$. This leads to numerous questions, such as: How large must n be before $P^n(x,A) \approx \pi(A)$? And, how can the Markov chain be modified to make this convergence faster? Each of these questions can be approached experimentally, through repeated simulation and analysis of output for specific examples. However, they can also be considered theoretically, as we now discuss.

3 Quantitative Convergence Bounds

In this section, we consider the question of how to obtain rigorous, quantitative bounds on the total variation distance to stationarity of a Markov chain $\{X_n\}$ to its stationary distribution $\pi(\cdot)$, i.e. how to bound

$$\|\mathcal{L}(X_n) - \pi\| := \sup_{A \subseteq \mathcal{X}} |\mathbf{P}(X_n \in A) - \pi(A)|.$$

Of course, if the Markov chain is complicated and high dimensional (as we assume here), then $\mathcal{L}(X_n)$ is complicated too, so our task is non-trivial.

While there are many approaches to this problem, the one we shall consider here is based on the *coupling inequality*. Specifically, let $\{X_n\}$ and $\{X'_n\}$ be two different copies of the Markov chain, each marginally following the transition probabilities $P(x,\cdot)$. Assume that $\{X'_n\}$ was started in stationarity, so that $\mathbf{P}(X'_n \in A) = \pi(A)$ for all n and A. Then by writing $\mathbf{P}(X_n \in A) = \mathbf{P}(X_n \in A,\ X_n = X'_n) + \mathbf{P}(X_n \in A,\ X_n \neq X'_n)$, and similarly for X'_n, it follows that

$$\begin{aligned}
\|\mathcal{L}(X_n) - \pi\| &= \sup_{A \subseteq \mathcal{X}} |\mathbf{P}(X_n \in A) - \pi(A)| \\
&= \sup_{A \subseteq \mathcal{X}} |\mathbf{P}(X_n \in A) - \mathbf{P}(X'_n \in A)| \\
&= \sup_{A \subseteq \mathcal{X}} |\mathbf{P}(X_n \in A,\ X_n \neq X'_n) - \mathbf{P}(X'_n \in A,\ X_n \neq X'_n)| \\
&\leq \mathbf{P}(X_n \neq X'_n).
\end{aligned}$$

In other words, to bound $\|\mathcal{L}(X_n) - \pi\|$, it suffices to "force" $X_n = X'_n$ with high probability. However, this presents its own challenges. In particular, if $\mathcal{X}$ is continuous, then if $\{X_n\}$ and $\{X'_n\}$ proceed independently, then we will usually have $\mathbf{P}(X_n = X'_n) = 0$, which is of no help. On the other hand, if we can define $\{X_n\}$ and $\{X'_n\}$ *jointly* in a way that increases $\mathbf{P}(X_n = X'_n)$, then this can help to bound convergence. One way to accomplish this is with *small sets*, as we discuss next.

3.1 Minorisation Conditions (Small Sets)

Suppose we know that $P(x,\cdot) \geq \epsilon\,\nu(\cdot)$, for all $x \in C \subseteq \mathcal{X}$, for some "overlap" probability measure $\nu(\cdot)$. That is,

$$P(x,A) \;\geq\; \epsilon\,\nu(A)\,, \qquad x \in C,\; A \subseteq \mathcal{X}\,. \tag{2}$$

Such inequalities are called *minorisation conditions*, and the subset C is called a *small set*. For background, see e.g. [18, 24].

For example, if $P(x,dy)$ has a density $f(x,y)$ with respect to Lebesgue measure $\lambda(\cdot)$, and $f(x,y) \geq \delta$ for $x \in C$ and $y \in B$, then (2) is satisfied with $\epsilon = \delta\,\lambda(B)$ and $\nu(A) = \lambda(A \cap B)\,/\,\lambda(B)$. In particular, it is often easy enough to verify (2) even if the details of the transitions $P(x,\cdot)$ are quite complicated.

If (2) holds, then whenever $(X_{n-1}, X'_{n-1}) \in C \times C$, we can use *Nummelin splitting* [20, 18, 24] to jointly update X_n and X'_n in such a way that $X_n = X'_n$ with probability at least ϵ. Thus, we have managed to "force" $X_n = X'_n$ with non-zero probability, as desired.

Putting this together, it follows that for any $j \in \mathbf{N}$,

$$\|\mathcal{L}(X_n) - \pi\| \;\leq\; (1-\epsilon)^j + \mathbf{P}(N_{n-1} < j)\,, \tag{3}$$

where $N_{n-1} = \#\{m : 0 \leq m \leq n-1,\; (X_m, X'_m) \in C \times C\}$ is the number of "opportunities" that the two chains have had to couple by time n.

If $C = \mathcal{X}$, then $N_{n-1} = n$, and (3) reduces (with $j = n$) simply to $\|\mathcal{L}(X_n) - \pi\| \leq (1-\epsilon)^n$. This is a very precise and useful inequality, which gives an exponentially-decreasing upper bound on the distance to stationarity, depending only on the value of ϵ from (2).

However, in typical MCMC applications it will not be possible to take $C = \mathcal{X}$ due to the inherently "unbounded" nature of the Markov chain. In this case, we need other methods to control N_n. One idea is through a *drift condition*, as we now discuss.

Remark. Of course, strictly speaking, MCMC algorithms are always run on real computers which are finite-state machines, so in some sense the state space $\mathcal{X}$ is always finite. But it is much more useful to model the state spaces as being truly infinite, rather than try to obtain bounds based on some machine-imposed truncation.

3.2 Drift Conditions

Suppose there is some function $V : \mathcal{X} \to [0,\infty)$, and $\lambda < 1$ and $\Lambda < \infty$, such that

$$\mathbf{E}\Big(V(X_n)\,|\,X_{n-1} = x\Big) \;\leq\; \lambda\,V(x) + \Lambda\,, \qquad x \in \mathcal{X}\,. \tag{4}$$

Such inequalities are called *drift conditions*. Intuitively, (4) means that when the chain is at large values of V, it will tend to "drift" towards smaller V values.

For this to be useful, we need to be able to couple the chains when they are at small values of V. So, suppose further that (2) is satisfied with $C = \{x \in \mathcal{X} : V(x) \leq D\}$ for some $D > 0$, i.e. that

$$P(x, \cdot) \geq \epsilon \nu(\cdot), \qquad \forall x \text{ with } V(x) \leq D. \tag{5}$$

Condition (4) then implies that the pair $\{(X_n, X'_n)\}$ will tend to "drift" towards $C \times C$, so hopefully $\mathbf{P}(N_{n-1} < j)$ will be small, thus making the bound (3) useful.

3.3 An Explicit Convergence Bound

Putting this all together proves the following bound [28, 30]. (For related results and discussion see [19, 27, 10, 8, 5, 24].)

Theorem 1. *If the drift condition (4) and minorisation condition (5) hold, with* $D > \frac{2\Lambda}{1-\lambda}$, *then for any integer* $0 \leq j \leq n$,

$$\|\mathcal{L}(X_n) - \pi\| \leq (1-\epsilon)^j + \alpha^{-n+j-1}\Delta^j \left(1 + \frac{\Lambda}{1-\lambda} + \mathbf{E}\big(V(X_0)\big)\right), \tag{6}$$

where $\alpha = \frac{1+D}{1+2\Lambda+\lambda D} > 1$ *and* $\Delta = 1 + 2(\lambda D + \Lambda)$.

If we set $j = \lfloor cn \rfloor$ in (6) for appropriate small $c > 0$, then this provides a quantitative, exponentially-decreasing upper bound on $\|\mathcal{L}(X_n) - \pi\|$, easily computed in terms of only the quantities ϵ from (5) and λ and Λ from (4).

The question remains whether the bound (6) is useful in genuinely complicated MCMC algorithms. We now consider an example.

3.4 A 20-Dimensional Example

We now consider a specific 20-dimensional MCMC algorithm. It corresponds to a model for a James-Stein shrinkage estimator, and is a version of a Gibbs sampler related to "variance components models" and "random-effects models", as applied to data from baseball hitting percentages; for details see [29] and the references therein.

For present purposes, we need know only that the Markov chain's state space is given by

$$\mathcal{X} = [0, \infty) \times \mathbf{R} \times \mathbf{R}^{18} \subseteq \mathbf{R}^{20},$$

and that if we write the chain's state at time n as $X_n = (A^{(n)}, \mu^{(n)}, \theta_1^{(n)}, \ldots \theta_{18}^{(n)})$, then given X_{n-1}, the chain generates X_n by:

$$A^{(n)} \sim IG\left(\frac{15}{2},\ 2+\frac{1}{2}\sum(\theta_i^{(n-1)}-\overline{\theta}^{(n-1)})^2\right);$$

$$\mu^{(n)} \sim N\left(\overline{\theta}^{(n-1)},\ A^{(n)}/18\right);$$

$$\theta_i^{(n)} \sim N\left(\frac{\mu^{(n)}\beta+Y_iA^{(n)}}{\beta+A^{(n)}},\ \frac{A^{(n)}\beta}{\beta+A^{(n)}}\right), \quad 1\le i\le 18;$$

where β is a known positive constant, $\{Y_i\}$ are the known actual data values, and $\overline{\theta}^{(n)} = \frac{1}{18}\sum_{i=1}^{18}\theta_i^{(n)}$. Here $N(m,v)$ is a normal distribution with mean m and variance v, while $IG(a,b)$ is an inverse-gamma distribution with density proportional to $e^{-b/x}x^{-(a+1)}$. This chain is specifically designed so that it will have a stationary probability distribution $\pi(\cdot)$ equal to the posterior distribution for the particular Bayesian statistical model of interest.

This chain represents a typical statistical application of MCMC. In particular, the state space is high-dimensional, and the transition densities are known but messy functions of data values without any particularly nice structure or symmetry. We know the chain has a stationarity distribution $\pi(\cdot)$, but know little else.

On the positive side, it is easily seen that the transition densities for this chain are positive throughout $\mathcal{X}$. So, the asymptotic convergence (1) must hold. However, quantitative bounds on the time to stationarity are more challenging, and might at first glance appear intractable. However, using Theorem 1, we are able to achieve this.

Our first challenge is to verify the drift condition (4). To do this, we choose the drift function

$$V(A,\mu,\theta_1,\ldots,\theta_{18}) = \sum_{i=1}^{18}(\theta_i-\overline{Y})^2,$$

where $\overline{Y} = \frac{1}{18}\sum_{i=1}^{18}Y_i$. (Intuitively, V measures how far our current vector of values are from the "center" of the given data.) It is then messy but reasonably straightforward to compute [29] that (4) is satisfied with $\lambda = 0.000289$ and $\Lambda = 0.161$.

Our next challenge is to verify the minorisation condition (5). To do this, we take $D = 1$, and compute [29] that (5) is satisfied with $\epsilon = 0.0656$.

We then apply Theorem 1 to conclude that, starting with $\theta_i^{(0)} = \overline{Y}$ for all i (say), and setting $j = n/2$ (for n even, say), we have

$$\|\mathcal{L}(X_n)-\pi\| \le (0.967)^n + (0.935)^n(1.17).$$

This is the precise quantitative bound that we sought. In particular, with $n = 140$, we have that

$$\|\mathcal{L}(X_{140})-\pi(\cdot)\| \le 0.009 < 0.01.$$

In other words, we have proved that the chain will "converge" (to within 1% of stationarity) after at most 140 iterations.

Although this is just an *upper* bound on convergence (and, indeed, convergence is probably actually achieved after 10 or fewer iterations), it is the only known rigorous

bound. And, since it is very quick and easy to run the Markov chain for 140 iterations on a computer, this bound is of clear practical benefit. Similar bounds have been obtained for other practical examples of MCMC, see e.g. [28, 16].

Remark. We refer to this example as 20-dimensional since the state space is an open subset of $\mathbf{R}^{20}$. However, since the θ_i are conditionally independent given A and μ, one could also say that this Gibbs sampler has just three components, A, μ, and θ, where θ happens to live in $\mathbf{R}^{18}$ instead of $\mathbf{R}$.

4 Adaptive MCMC

For a given state space $\mathcal{X}$ and target probability distribution $\pi(\cdot)$, there are many possible MCMC algorithms which will converge asymptotically. An important practical question is, which MCMC choice is "best", or at least good enough to converge after a feasible number of iterations?

Even within a given class of MCMC algorithms, choices of related tuning parameters can be crucial in the algorithms success. A number of recent papers [14, 1, 3, 25, 26, 34, 4, 2] have considered the possibility of having the computer modify the Markov chain transitions while the chain runs, in an effort to seek better convergence. This raises a number of theoretical and practical issues, which we now discuss.

4.1 A Toy Example

Suppose $\pi(\cdot)$ is a simple distribution on the trivial state space $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$, with $\pi(x) > 0$ for all $x \in \mathcal{X}$. (For definiteness, take $\pi(x) = 0$ for $x \notin \mathcal{X}$.) Fix $\gamma \in \mathbf{N}$, e.g. $\gamma = 2$. Consider a "random-walk Metropolis" (RWM) algorithm, defined as follows:

- Given X_n, first propose a state $Y_{n+1} \in \mathbf{Z}$, with

$$Y_{n+1} \sim \text{Uniform}\{X_n - \gamma, \ldots, X_n - 1, X_n + 1, \ldots, X_n + \gamma\}.$$

- Then, with probability $\min[1, \pi(Y_{n+1})/\pi(X_n)]$, *accept* this proposal by setting $X_{n+1} = Y_{n+1}$.
- Otherwise, with probability $1 - \min[1, \pi(Y_{n+1})/\pi(X_n)]$, *reject* this proposal by setting $X_{n+1} = X_n$.

It is easily seen that these transition probabilities have $\pi(\cdot)$ as a stationary distribution, and are irreducible and aperiodic, so we have asymptotic convergence as in (1), for any choice of $\gamma \in \mathbf{N}$. (This example is discussed in [3, 25]; for an interactive display see [31].)

However, this still leaves the question of choice of γ. If $\gamma = 1$, the chain will move at most one unit at each iteration, leading to slow convergence. On the other

hand, if say $\gamma = 50$, then the chain will usually propose values outside of $\mathcal{X}$ which will all be rejected, again leading to slow convergence. Best is a "moderate" value of γ, e.g. $\gamma = 4$.

In a more complicated example, the best choice of a tuning parameter (like γ) will be far less obvious. So, we consider the possibility of automating the choice of γ. As an example, we might adapt γ as follows:

- Start with γ set to $\Gamma_0 = 2$ (say).
- Each time a proposal is accepted, set $\Gamma_{n+1} = \Gamma_n + 1$ (so γ increases, and the acceptance rate decreases).
- Each time a proposal is rejected, set $\Gamma_{n+1} = \max(\Gamma_n - 1, 1)$ (so γ decreases, and the acceptance rate increases).

This appears to be a logical way for the computer to seek out good choices of γ, and in simulations [31] it appears to work well for a while. However, if (say) $\pi\{2\}$ is very small, then the chain will eventually get "stuck" with $X_n = \Gamma_n = 1$ for long stretches of time. This is due to a certain *asymmetry*: for the adaptive chain, *entering* the region $\{X_n = \Gamma_n = 1\}$ is much easier than *leaving* it. In particular, this adaptive chain does not converge to $\pi(\cdot)$ at all, but rather may converge to a different distribution giving far too much weight to the state 1. That is, the adaption – which attempted to *improve* the convergence – actually ruined the convergence entirely.

4.2 An Adaptive MCMC Convergence Theorem

In light of counter-examples like the above, we seek conditions which guarantee that adaptive MCMC schemes will in fact converge. One such result is the following, from [25]; for related results see e.g. [1, 3, 34, 4, 2]. To state it, define the "ϵ convergence time function" $M_\epsilon : \mathcal{X} \times \mathcal{Y} \to \mathbf{N}$ by

$$M_\epsilon(x, \gamma) \;=\; \inf\left\{n \geq 1 : \|P_\gamma^n(x, \cdot) - \pi(\cdot)\| \leq \epsilon\right\}.$$

Theorem 2. *An adaptive scheme $\{(X_n, \Gamma_n)\}$, using transition kernels $\{P_\gamma\}_{\gamma \in \mathcal{Y}}$ will converge, i.e. $\lim_{n\to\infty} \|\mathcal{L}(X_n) - \pi(\cdot)\| = 0$, assuming (i) $\pi(\cdot)$ is stationary for each individual P_γ, and (ii) the "Diminishing Adaptation" property that*

$$\lim_{n\to\infty} \sup_{x \in \mathcal{X}} \|P_{\Gamma_{n+1}}(x, \cdot) - P_{\Gamma_n}(x, \cdot)\| = 0$$

in probability, and (iii) the "Containment" property that for all $\epsilon > 0$, the values $\{M_\epsilon(X_n, \Gamma_n)\}$ remains bounded in probability as $n \to \infty$.

In this theorem, condition (i) is basic to any adaptive MCMC algorithm, and (ii) can be ensured by careful design of the adaption, while (iii) is an unfortunate technical condition though it is nearly always satisfied in practical examples [4]. Furthermore, these same conditions also guarantee central limit theorems (CLTs) for

adaptive MCMC with bounded functionals, though not necessarily with unbounded functionals [34].

In light of this theorem, we see that the toy example of 4.1 satisfies conditions (i) and (iii), but not (ii). However, (ii) will be satisfied, and the chain will converge to $\pi(\cdot)$, if we modify the adaption so that at time n, it only adapts with probability $p(n)$ for some probabilities $p(n) \to 0$, otherwise the value of γ is left unchanged. In particular, we could choose, say, $p(n) = 1/n$, in which case we would still have $\sum_n p(n) = \infty$ and thus still have an infinite amount of adaptation, and yet still guarantee convergence.

4.3 A 100-Dimensional Example

For complicated examples in high dimensions, adaption is not as trivial as for the example of Section 4.1, but it is still quite feasible.

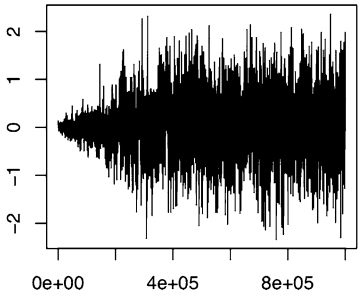
For example, it is known (see [23] and the references therein) that if target distribution $\pi(\cdot)$ is (approximately) a high-dimensional normal distribution with covariance Σ, then the optimal Gaussian proposal distribution for a RWM algorithm is equal to $N(x, (2.38)^2 d^{-1} \Sigma)$.

Now, the target covariance Σ is generally unknown, but it can be approximated by Σ_n, the empirical covariance of the first n iterations of the Markov chain. This suggests [14, 26] an adaptive MCMC algorithm with proposal distribution at the n^{th} iteration given by the mixture distribution

$$Q_n(x, \cdot) \;=\; 0.95\, N\Big(x,\, (2.38)^2 d^{-1}\, \Sigma_n\Big) \;+\; 0.05\, N\Big(x,\; (0.1)^2 d^{-1}\, I_d\Big)$$

[if Σ_n is non-singular, otherwise say $Q_n(x, \cdot) = N\big(x,\; (0.1)^2 d^{-1}\, I_d\big)$]. Such algorithms will generally satisfy condition (ii) of Theorem 2, and furthermore will satisfy condition (iii) provided the tails of $\pi(\cdot)$ are not too heavy [4].

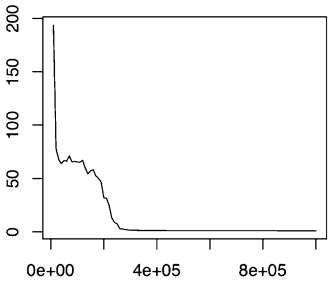
If we run [26] this algorithm on an example in dimension $d = 100$, then a trace plot of the first coordinate (plotted against iteration number) looks as follows:



A close inspection of this plot shows that the first coordinate is initially "stuck" at values very close to 0. Then, after about 300,000 iterations, the empirical Σ_n gets close to the true Σ, so the adaptive algorithm "finds" good proposal distributions and

starts mixing well. At this point, the first coordinate mixes nicely and efficiently over values concentrated between about -1 and 1, corresponding to accurate samples of the first coordinate from the true target distribution $\pi(\cdot)$.

This interpretation can be confirmed by looking at a plot of the sub-optimality factor $b_n \equiv d\left(\sum_{i=1}^{d} \lambda_{in}^{-2}\right) / \left(\sum_{i=1}^{d} \lambda_{in}^{-1}\right)^2$, where $\{\lambda_{in}\}$ are the eigenvalues of the matrix $\Sigma_n^{1/2} \Sigma^{-1/2}$. This quantity b_n is known [23] to measure the convergence slow-down factor of a chain using the covariance estimate Σ_n obtained after n iterations, compared to a chain using the true covariance Σ. The plot clearly shows that the values of b_n are initially very large, and then get close to 1 after about 300,000 iterations:



This further confirms that after about 300,000 iterations, the adaptive scheme "finds" good values of Σ_n which accurately approximate Σ, leading to fast and accurate convergence. And, since the 100×100 covariance matrix Σ involves 5,050 unknown values, it seems clear that this optimisation could not have been done manually, and that the adaptive MCMC scheme really was essential to achieving fast convergence to $\pi(\cdot)$. Similar success has been found in other high-dimensional examples (see e.g. [14, 26]), and we expect that adaptive MCMC will be used more often in the years ahead.

5 Connection with QMC?

In the context of a conference on "MCQMC", it is reasonable to ask about the placement of MCMC algorithms in the Monte Carlo (MC) / Quasi-Monte Carlo (QMC) divide.

For the most part, MCMC algorithms are squarely on the MC side, using pseudorandom number generators to power the iterations according to (approximately) the laws of probability. Furthermore, much of the theoretical analysis, including that discussed herein, uses probability theory and assumes the algorithms follow probabilistic laws. However, it has been observed [21, 6, 22] that it is also possible to power MCMC algorithms using quasi-random sequences.

In principle, QMC is "smarter" than just using (pseudo)random numbers, so should be better. Furthermore, it is known [13, 7, 6] that using e.g. *antithetic* or other not-entirely-random variates can sometimes speed up MCMC convergence.

So, it seems that future MCMC work – both applied and theoretical – might make more use of quasi-randomness and thus make more of a leap towards the QMC world.

However, many of the ideas considered herein – ideas like "irreducible", "coupling", "minorisation", "drift", "Diminishing Adaptation", "Containment", etc. – all use *probabilistic intuition* and it is not clear how to translate them into QMC ideas. Furthermore, in many cases we may not know enough about the (complicated, messy, high-dimensional) target distribution to design QMC effectively, and it might be easier to verify "weak" conditions like minorisation and drift.

Thus, in this paper, we have treated the algorithms as being "truly random", i.e. within the context of traditional Monte Carlo. However, we look forward to more QMC ideas finding their way into MCMC in the future.

6 Summary

The main points of this article may be summarised as follows:

- MCMC algorithms are extremely widely used, in Bayesian statistics and elsewhere.
- *Quantitative convergence bounds* are a very important topic for MCMC, with both practical and theoretical implications.
- An approach using the *coupling inequality*, together with *minorisation* and *drift conditions*, can provide specific, useful bounds (like "140") on the convergence times even of rather complicated Markov chains on continuous, high-dimensional state spaces.
- For a given problem, many different MCMC algorithms are available, and it can be difficult (though very important) to choose among them.
- *Adaptive MCMC* is a promising recent method of getting the computer to help find better MCMC algorithms during the course of a run.
- Naive application of adaptive MCMC may fail to converge to $\pi(\cdot)$.
- However, theorems are available which prove the validity of adaptive MCMC under certain conditions which can often be verified for specific adaptive schemes.
- Adaptive MCMC works well in some high-dimensional statistics-related examples, including an adaptive random-walk Metropolis (RWM) algorithm in dimension 100.
- While MCMC is traditionally on the "MC" side of the MC / QMC divide, we anticipate greater connections between MCMC algorithms and quasi-Monte Carlo ideas in the future.

And more generally:

- Theory informs the applied use of MCMC in many ways, thus providing an excellent arena in which mathematical results can have a genuine and widespread impact on applications of algorithms.

It is to be hoped that many experts in MC and QMC will get more interested in MCMC algorithms, and make further theoretical contributions to this interesting and widely applicable area.

Acknowledgements The author's research is supported in part by NSERC of Canada. All of the author's papers, applets, and software are freely available at his web page probability.ca/jeff.

References

1. C. Andrieu and E. Moulines (2006), On the ergodicity properties of some adaptive Markov Chain Monte Carlo algorithms. Ann. Appl. Prob. **16**, 1462–1505.
2. Y. Atchadé and G. Fort (2008), Limit theorems for some adaptive MCMC algorithms with subgeometric kernels. Preprint.
3. Y.F. Atchadé and J.S. Rosenthal (2005), On Adaptive Markov Chain Monte Carlo Algorithms. Bernoulli **11**, 815–828.
4. Y. Bai, G.O. Roberts, and J.S. Rosenthal (2008), On the Containment Condition for Adaptive Markov Chain Monte Carlo Algorithms. Preprint.
5. P.H. Baxendale (2005), Renewal theory and computable convergence rates for geometrically ergodic Markov chains. Ann. Appl. Prob. **15**, 700–738.
6. R.V. Craiu and C. Lemieux (2007), Acceleration of the Multiple-try Metropolis Algorithm using Antithetic and Stratified sampling. Stat. and Comput. **17(2)**, 109–120.
7. R.V. Craiu and X.-L. Meng (2005), Multi-process parallel antithetic coupling for forward and backward Markov chain Monte Carlo. Ann. Stat. **33(2)**, 661–697.
8. R. Douc, E. Moulines, and J.S. Rosenthal (2002), Quantitative bounds on convergence of time-inhomogeneous Markov Chains. *Annals of Applied Probability* **14**, (2004), 1643–1665.
9. R. Douc, E. Moulines, and P. Soulier (2007), Computable convergence rates for sub-geometric ergodic Markov chains. Bernoulli **13**, 831–848.
10. G. Fort and E. Moulines (2000), Computable Bounds For Subgeometrical And Geometrical Ergodicity. Unpublished manuscript. Available at: http://citeseer.ist.psu.edu/fort00computable.html
11. A.E. Gelfand and A.F.M. Smith (1990), Sampling based approaches to calculating marginal densities. J. Amer. Stat. Assoc. **85**, 398–409.
12. S. Geman and D. Geman (1984), Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans. on pattern analysis and machine intelligence **6**, 721–741.
13. P.J. Green and X.-L. Han (1992), Metropolis methods, Gaussian proposals, and antithetic variables. In Stochastic Models, Statistical Methods and Algorithms in Image Analysis (P. Barone et al., Eds.). Springer, Berlin.
14. H. Haario, E. Saksman, and J. Tamminen (2001), An adaptive Metropolis algorithm. Bernoulli **7**, 223–242.
15. W.K. Hastings (1970), Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97–109.
16. G.L. Jones and J.P. Hobert (2001), Honest exploration of intractable probability distributions via Markov chain Monte Carlo. Statistical Science **16**, 312–334.
17. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953), Equations of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1091.
18. S.P. Meyn and R.L. Tweedie (1993), Markov chains and stochastic stability. Springer-Verlag, London. Available at: http://probability.ca/MT/
19. S.P. Meyn and R.L. Tweedie (1994), Computable bounds for convergence rates of Markov chains. Ann. Appl. Prob. **4**, 981–1011.

20. E. Nummelin (1984), General irreducible Markov chains and non-negative operators. Cambridge University Press.
21. A.B. Owen and S.D. Tribble (2005), A quasi-Monte Carlo Metropolis algorithm. PNAS **102(25)**, 8844–8849.
22. A.B. Owen and S.D. Tribble (2008), Constructions of weakly CUD sequences for MCMC. Elec. J. Stat. **2**, 634–660.
23. G.O. Roberts and J.S. Rosenthal (2001), Optimal scaling for various Metropolis-Hastings algorithms. Stat. Sci. **16**, 351–367.
24. G.O. Roberts and J.S. Rosenthal (2004), General state space Markov chains and MCMC algorithms. Prob. Surv. **1**, 20–71.
25. G.O. Roberts and J.S. Rosenthal (2007), Coupling and Ergodicity of Adaptive MCMC. J. Appl. Prob. **44**, 458–475.
26. G.O. Roberts and J.S. Rosenthal (2006), Examples of Adaptive MCMC. J. Comp. Graph. Stat., to appear.
27. G.O. Roberts and R.L. Tweedie (1999), Bounds on regeneration times and convergence rates for Markov chains. Stoch. Proc. Appl. **80**, 211–229. Corrigendum, Stoch. Proc. Appl. **91** (2001), 337–338.
28. J.S. Rosenthal (1995), Minorization conditions and convergence rates for Markov chain Monte Carlo. J. Amer. Stat. Assoc. **90**, 558–566.
29. J.S. Rosenthal (1996), Convergence of Gibbs sampler for a model related to James-Stein estimators. Stat. and Comput. **6**, 269–275.
30. J.S. Rosenthal (2002), Quantitative convergence rates of Markov chains: A simple account. Elec. Comm. Prob. **7**, No. 13, 123–128.
31. J.S. Rosenthal (2004), Adaptive MCMC Java Applet. Available at: http://probability.ca/jeff/java/adapt.html
32. M.A. Tanner and W.H. Wong (1987), The calculation of posterior distributions by data augmentation (with discussion). J. Amer. Stat. Assoc. **82**, 528–550.
33. L. Tierney (1994), Markov chains for exploring posterior distributions (with discussion). Ann. Stat. **22**, 1701–1762.
34. C. Yang (2008), On The Weak Law of Large Numbers for Unbounded Functionals for Adaptive MCMC. Preprint.

MINT – New Features and New Results

Rudolf Schürer and Wolfgang Ch. Schmid

Abstract (t,m,s)-nets are among the best methods for the construction of low-discrepancy point sets in the s-dimensional unit cube. Various types of constructions and bounds are known today. Additionally there exist many propagation rules connecting nets to other mathematical objects.

The MINT database developed by the authors is one of the most elaborate and convenient tools for accessing information on many aspects of nets. In this article we provide new information about MINT.

We also develop several new constructions by generalizing methods from coding theory and show how these methods can be used for obtaining new (t,m,s)-nets. In many cases the development of these new methods has been guided by MINT.

1 Introduction

The concepts of (t,m,s)-nets and (t,s)-sequences provide powerful methods for the construction of low-discrepancy point sets in the s-dimensional unit cube. A detailed theory was developed in [17] (see also [19, Chapter 4] for a survey or [22] for recent results).

So far an overwhelming variety of different constructions as well as bounds exists. Additionally the existence of nets and sequences is often linked to other mathematical objects, e.g. algebraic function fields, linear codes, orthogonal arrays, or even other nets and sequences. Connections of this type are usually referred to as "propagation rules", and many such rules are known. A series of papers [15, 6, 20, 21] gives an overview of important approaches. There have also been

Rudolf Schürer
url: http://mint.sbg.ac.at/rudi/

Wolfgang Ch. Schmid
Department of Mathematics, University of Salzburg, Austria
url: http://www.mat.sbg.ac.at/~schmidw/

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_10,

attempts to determine the best nets available in a given setting by the publication of tables of net parameters in [15, 6] and, with a slightly different intention, in [2]. However, parts of these tables were outdated before the articles appeared in print.

The most convenient and up-to-date access to such tables is provided by MINT (acronym for "Minimal t"), our web-based database system which is available on the Internet at the address

http://mint.sbg.ac.at/

This system overcomes the former problems of printed tables and provides a number of hitherto unavailable services to the scientific community, many of them described and discussed in [32, 33].

After the introduction of basic notations and concepts we present new features of the MINT database.

Then we describe several new propagation rules for ordered orthogonal arrays and generalized codes. These rules have been found by generalizing existing ones which have been known so far only for orthogonal arrays or linear codes (direct product method, construction X, generalized matrix-product construction). This work has been guided on the one hand by the information in MINT about optimal parameters, on the other hand by the objective to enhance our database and to improve the data in the tables. Furthermore, we show the improvements of these new methods in examples determined by MINT.

2 Basic Notations and Concepts

Nets and Sequences

A (t, m, s)-net in base b is a point set with b^m points in the s-dimensional half-open unit cube $[0, 1)^s$ with certain distribution properties. The concept has been introduced by Niederreiter in [17], generalizing the special cases with $b = 2$ considered by Sobol′ [36] and with $t = 0$ and b prime considered by Faure [9]. An excellent introduction to this area of research can be found in Chapter 4 of Niederreiter's monograph [19].

The definition of (t, m, s)-nets is as follows:

Definition 1. Let $0 \le t \le m$ and $b \ge 2$ be integers. A multiset of $M = b^m$ points in $[0, 1)^s$ is a (t, m, s)-*net in base* b if every interval

$$\prod_{i=1}^{s} \left[\frac{a_i}{b^{d_i}}, \frac{a_i + 1}{b^{d_i}} \right) \tag{1}$$

with non-negative integers $a_i, d_i, a_i < b^{d_i}$ for $i = 1, \ldots, s$, and $\sum_{i=1}^{s} d_i = m - t$ (i.e., volume $1/b^{m-t}$) contains exactly b^t points of the multiset.

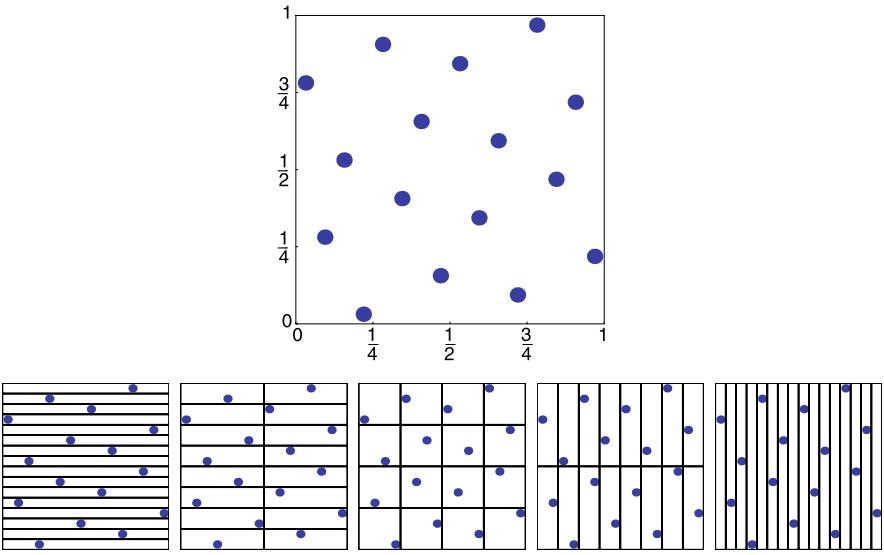


Fig. 1 A $(0, 4, 2)$-net in base $b = 2$ and all intervals of the form (1) with volume $1/2^4$.

Example 1. Figure 1 shows the $2^4 = 16$ points in $[0, 1)^2$ of a $(0, 4, 2)$-net in base $b = 2$. The five small images contain the same point set together with all 80 intervals of the form described in (1) with volume $1/2^4 = 1/16$. It is easy to verify that each of these intervals contains exactly one point of the net.

In [17] it is shown that the star discrepancy of a (t, m, s)-net $\mathcal{N}$ can be bounded by

$$D_M^*(\mathcal{N}) \leq c(s, b)\, b^t \frac{\log^{s-1} M}{M} + \mathcal{O}\left(b^t \log^{s-2} M\right),$$

where $c(s, b)$ is a constant depending only on s and b. Hence, for given b, s, and m, the term b^t controls the star discrepancy of $\mathcal{N}$ and a low star discrepancy can only be expected for small values of t. Thus t is called the *quality parameter* of $\mathcal{N}$ and one is primarily interested in (t, m, s)-nets with t-values being as small as possible. In particular (t, m, s)-nets are low discrepancy point sets provided that the quality parameter t is bounded for $M = b^m \to \infty$ (which turns out to be possible).

The main goal of MINT is to determine for which quadruples (b, t, m, s) a (t, m, s)-net in base b can exist and for which it cannot. An important subclass of nets are *digital nets*:

Definition 2. A (t, m, s)-net in a prime power base b is a *digital* (t, m, s)-net over $\mathbb{F}_b$ if the b-adic digit vectors of the points (interpreted as vectors over $\mathbb{F}_b$ using arbitrary bijections $\{0, \ldots, b-1\} \leftrightarrow \mathbb{F}_b$) form a vector space over $\mathbb{F}_b$.

An important tool for the construction of (digital) nets are (digital) (t, s)-*sequences*, which can be thought of as an infinite nesting of (digital) nets with size $m =$

$t, t+1, \ldots$ (for a formal definition see [17], [19], and [26]). Every (t,s)-sequence in base b yields $(t,m,s+1)$-nets in base b for all $m \geq t$ [17, Theorem 5.15], every digital (t,s)-sequence over $\mathbb{F}_b$ yields digital $(t,m,s+1)$-nets over $\mathbb{F}_b$ for all $m \geq t$ [26, Lemma 1 and 2], and every digital (t,s)-sequence is also a (t,s)-sequence.

Well-known constructions for (t,s)-sequences include the sequences due to Sobol′ [36], Faure [9], Niederreiter [18], and Niederreiter and Xing [25, 26, 27, 37].

Ordered Orthogonal Arrays

Ordered orthogonal arrays (OOAs) were introduced independently in [10] and [16] in an attempt to formalize the underlying combinatorial structure of (t,m,s)-nets:

Definition 3. Let M, s, T, and $k \leq sT$ denote non-negative integers and let S_b denote a set of cardinality $b \geq 2$. Let $S_b^{(s,T)}$ denote the set of sT-tuples over S_b indexed by elements from $\{1,\ldots,s\} \times \{1,\ldots,T\}$. An *ordered orthogonal array* $\mathrm{OOA}(M,s,S_b,T,k)$ is a multiset $\mathcal{A}$ of M elements of $S_b^{(s,T)}$ such that each possible projection on k coordinates indexed by

$$(1,1),\ldots,(1,k_1) \mid \cdots \mid (s,1),\ldots,(s,k_s)$$

with $k_1 + \ldots + k_s = k$ is balanced, i.e., each of the b^k possible k-tuples over S_b appears the same number of times (namely M/b^k times) in such a projection. The parameter T is called the *depth*, k the *strength* of the OOA. The M elements of $\mathcal{A}$ are called *runs* of $\mathcal{A}$.

For a prime power b, an OOA $\mathcal{A}$ is called *linear* over the finite field $\mathbb{F}_b$ if $S_b = \mathbb{F}_b$ and $\mathcal{A}$ is a vector space over $\mathbb{F}_b$.

Note that an $\mathrm{OOA}(M,s,S_b,1,k)$ (i.e., an OOA with depth 1) is an *orthogonal array* $\mathrm{OA}(M,s,S_b,k)$. Thus OOAs are a generalization of orthogonal arrays (OAs). It is easy to see that an OOA with depth $T' < T$ (and otherwise unchanged parameters) can be constructed from an OOA with depth T simply by discarding columns (i,j) with $j > T'$. In particular, an OA can be obtained from every OOA, whereas the embedding of an OA in an OOA with depth $T > 1$ is not always possible.

The connection between nets and OOAs has been established in [10, 16], where it is shown that a (digital) $(m-k,m,s)$-net in base b is equivalent to a (linear) $\mathrm{OOA}(b^m,s,\{0,\ldots,b-1\},m,k)$:

- The OOA $\mathcal{A}$ is obtained from the net $\mathcal{N}$ based on the b-adic expansion of the coordinates of its points. Every point $\mathbf{y} = (y_1,\ldots,y_s) \in \mathcal{N} \subset [0,1)^s$ yields a run $(x^{(i,j)})_{(i,j)\in\{1,\ldots,s\}\times\{1,\ldots,m\}}$ of $\mathcal{A}$ with $x^{(i,j)} = \lfloor b^j y_i \rfloor \bmod b$.
- On the other hand, a net $\mathcal{N}'$ with the same parameters as $\mathcal{N}$ can be recovered from $\mathcal{A}$ using the points

$$\mathbf{y}' = (y_1',\ldots,y_s') \qquad \text{with} \qquad y_i' = \sum_{j=1}^{m} \frac{x^{(i,j)}}{b^j}.$$

Thus OOAs establish a framework for discussing nets (for $T = m$) as well as OAs (for $T = 1$).

Generalized Codes

In the following let b be a prime power. For a linear OA $\mathcal{A} \subseteq \mathbb{F}_b^{(s,1)} \cong \mathbb{F}_b^s$ one can consider the dual

$$\mathcal{A}^\perp := \left\{ \mathbf{x} \in \mathbb{F}_b^s : \langle \mathbf{x}, \mathbf{y} \rangle = 0 \text{ for all } \mathbf{y} \in \mathcal{A} \right\}.$$

It is well-known that if $\mathcal{A}$ has strength k, the Hamming distance of two different vectors in $\mathcal{A}^\perp$ is at least $k+1$. Thus the dual of a linear OA$(b^m, s, \mathbb{F}_b, k)$ is a linear $[s, s-m, k+1]$-code over $\mathbb{F}_b$ and vice versa. This result can be generalized to OOAs in the following way [13, 24]:

Definition 4. Let n, s, T, and $d \le sT + 1$ denote non-negative integers and let $\mathbb{F}_b$ denote the finite field with b elements. Define the weight function $w : \mathbb{F}_b^{(s,T)} \to \mathbb{N}_0$ as

$$w(x^{(1,1)}, \ldots, x^{(1,T)} \mid \cdots \mid x^{(s,1)}, \ldots, x^{(s,T)}) := \sum_{i=1}^{s} \min \left\{ j \in \mathbb{N}_0 : x^{(i,j+1)} = \ldots = x^{(i,T)} = 0 \right\}$$

and the metric d on $\mathbb{F}_b^{(s,T)}$ as $d(\mathbf{x}, \mathbf{y}) := w(\mathbf{x} - \mathbf{y})$.

A non-empty subset $\mathcal{C} \subseteq \mathbb{F}_b^{(s,T)}$ is a *generalized* $((s,T), N, d)$*-code* if $|\mathcal{C}| = N$ and $d(\mathbf{x}, \mathbf{y}) \ge d$ for all $\mathbf{x} \ne \mathbf{y} \in \mathcal{C}$. If $\mathcal{C}$ is a linear subspace of $\mathbb{F}_b^{(s,T)}$, then $\mathcal{C}$ is a *generalized linear* $[(s,T), n, d]$*-code* with $n = \dim \mathcal{C}$, i.e., $n = \log_b N$.

We omit the term "generalized" if the meaning is clear.

Note that for depth $T = 1$ this definition coincides with the usual definition of linear codes. To be precise, every linear $[(s,1), n, d]$-code is a linear $[s, n, d]$-code over the same field and vice versa. The duality between linear OAs and linear codes can be generalized in the following form: the dual of a linear OOA$(b^m, s, \mathbb{F}_b, T, k)$ is a linear $[(s,T), sT - m, k+1]$-code over $\mathbb{F}_b$ and vice versa [13, 24].

Figure 2 sums up the dependencies between all described classes of objects. Arrows indicate that the existence of an object with given parameters b, s, t, and (except for sequences) m implies the existence of another object with the same parameters. To establish non-existence results, implications run in the opposite direction. MINT tracks the existence of all these objects (except of the non-linear (generalized) codes in the bottom row) in order to establish upper and lower bounds on their parameter range. Since March 2007 the information for other objects but nets and sequences is not only calculated in the back-end, but can actually be queried using

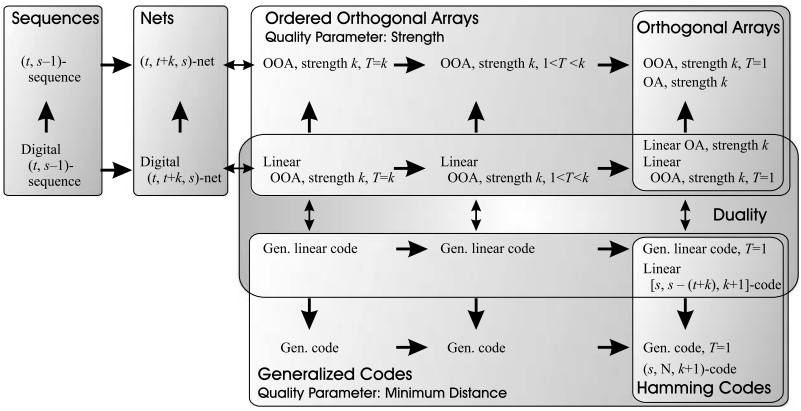


Fig. 2 Classes of objects tracked by MINT.

Table 1 MINT can create tables for the following projections.

Find	depending on	Restriction
t, k	(m, s), (m, n), or (s, n)	none
s, n	(t, m), (t, k), or (m, k)	none
t, m, n	(s, k)	none
m, k	(t, s)	$T = \infty$
t, m, s	(k, n)	$T < \infty$
m, s, k	(t, n)	$T < \infty$

the web front-end (see the following section). MINT as described in [32] was only aware of sequences, nets, and OAs.

3 New Features in MINT

An in-depth discussion of the basic functionality provided by MINT, in particular the generation of parameter tables and construction trees for (t, m, s)-nets, can be found in [32]. Here we focus on more advanced and new features which have been implemented after the publication of [32].

The arguably most important new feature in MINT is querying information about other objects than nets and sequences. It is now possible to obtain information about orthogonal arrays and linear codes, as well as about OOAs and generalized linear codes with depth $T > 1$.

When MINT is used for querying results for (generalized) linear codes, the original set of parameters (code length s, dimension of the dual OOA m, strength of the dual OOA k, and quality parameter of the net t) becomes inappropriate. Therefore, the additional parameters n (for the dimension of the code) and d (for its minimum

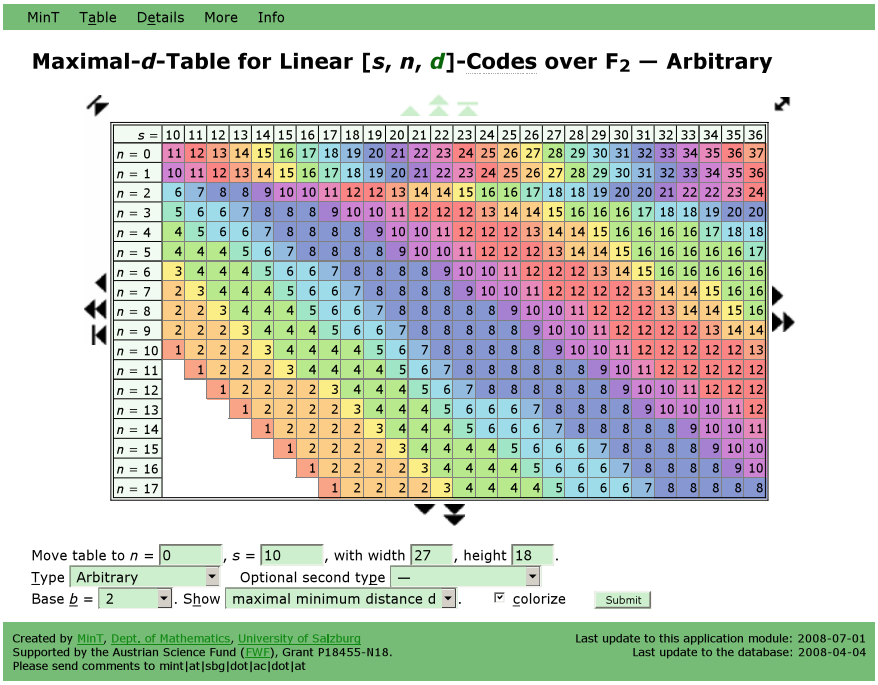


Fig. 3 Screenshot of MINT: a maximal-d-table for linear codes over $\mathbb{F}_2$, showing the largest minimum distance d known for $[s,n,d]_2$-codes with given s and n.

distance) have been introduced. Obviously these parameters depend on each other, satisfying the relations $m = t + k$, $d = k + 1$, and $m + n = Ts$. Even though the full information was contained in the database from the very beginning of MINT, these additional parameters extend the number of use cases significantly. Since for every query the optimal value of one parameter is determined based on two other given parameters, the number of possible queries has multiplied due to the introduction of the additional parameters n and d. Table 1 lists all possible cases for selecting given and dependent parameters. The variable d is not included because it can always be used instead of k, due to the relation $d = k + 1$.

A priori it is not apparent that all 10 query types are well defined. As a matter of fact it turns out that some of them can only be used for certain depths. These restrictions are listed in the third column of Table 1. For details on how these restrictions arise see [33].

Figure 3 contains a screenshot of MINT showing a maximal-d-table for linear codes over $\mathbb{F}_2$. Asking for the largest possible minimal distance d of an $[s,n,d]_b$-code is a very common question in the field of coding theory. MINT can now answer this question, even though it is translated behind the scenes into the questions for the largest strength $k = d - 1$ of a linear OA$(b^{s-n}, s, \mathbb{F}_b, k)$.

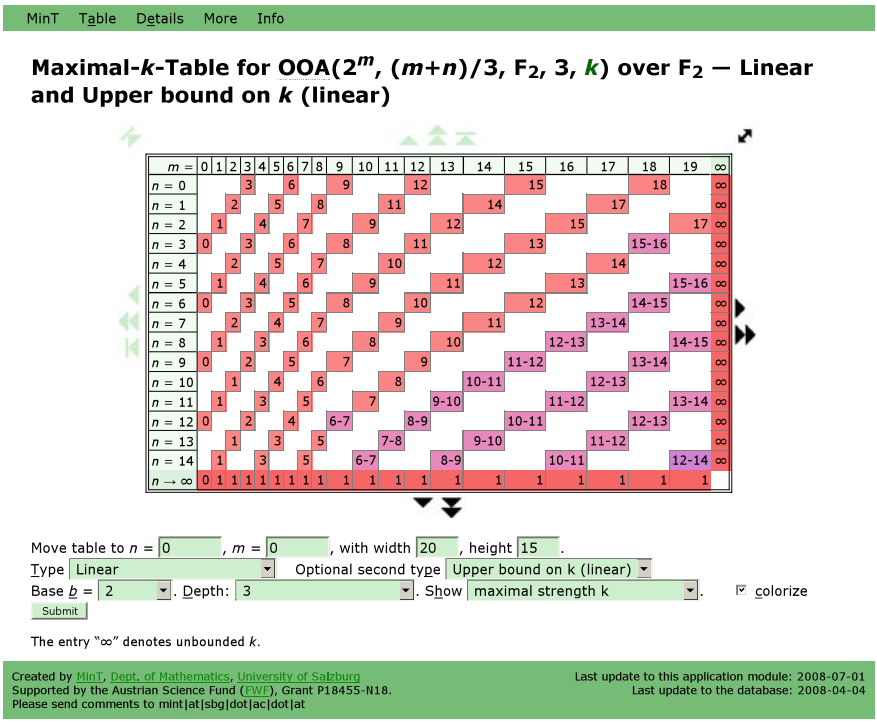


Fig. 4 Screenshot of MINT: a maximal-k-table for linear OOA$(2^m, (m+n)/3, \mathbb{F}_2, 3, k)$, i.e., for linear OOAs with depth $T = 3$ over $\mathbb{F}_2$ with given dimension m and codimension n.

MINT is able to generate parameter tables for all depths and all projections which are mathematically well defined, provided the requested data is in the range included in the database. Some of these combinations yield rather unexpected results, which give new insight into the geometry of the parameter range of OOAs. For example Figure 4 shows a maximal-k-table for linear OOA$(2^m, (m+n)/3, \mathbb{F}_2, 3, k)$, i.e., for linear OOAs with depth $T = 3$ over $\mathbb{F}_2$ with given dimension m and codimension n. This query is only defined if $m+n$ is a multiple of the depth 3, resulting in the banded structure of the table. The limits of k when n or m tends towards infinity exist or diverge, respectively, if they are defined to take only those m and n into account whose sum is a multiple of 3. Furthermore, note that the table shows upper as well as lower bounds on k, and that these bounds coincide for many cases, even though no extensive work on OOAs in depth $T = 3$ has been done.

Figure 5 contains another interesting example. It shows upper bounds on the maximal length s of linear OOAs and generalized linear codes with depth $T = 3$ over $\mathbb{F}_2$ for given code dimension n and quality parameter t. Translated to the language of codes, the net's quality parameter t corresponds to the Singleton defect of the code, i.e., the difference between the minimum distance d and its upper bound $Ts - n + 1$

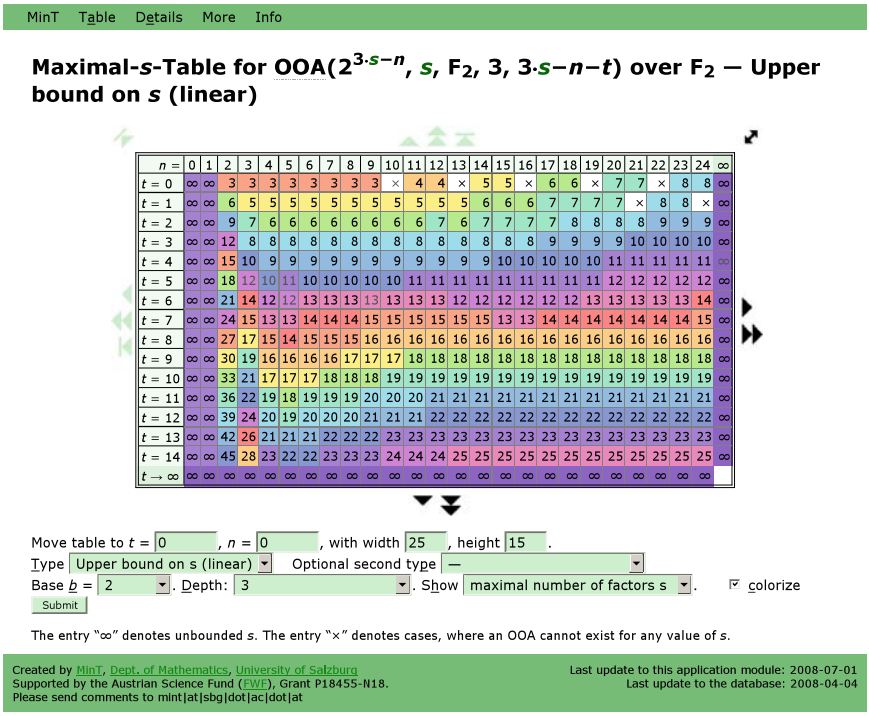


Fig. 5 Screenshot of MINT: a maximal-s-table for linear OOA$(2^{3s-n}, s, \mathbb{F}_2, 3, 3s-n-t)$ or $[(s,3), n, 3s-n-t+1]_2$-codes, i.e., for linear OOAs or generalized linear codes with depth $T=3$ over $\mathbb{F}_2$ with given code dimension n and Singleton defect t.

given by the (generalized) Singleton bound [31]. Codes with Singleton defect $t=0$ are commonly known as *maximum distance separable* or *MDS* codes. Thus the first line of the table gives the maximum length of generalized MDS codes with depth $T=3$ over $\mathbb{F}_2$.

For dimension $n \leq 1$ the length is unbounded due to the existence of trivial codes with arbitrary length. For $n \leq bT$ extended (generalized) Reed–Solomon codes [31] show that a length of $s = b+1$ can be reached; in this example this coincides with the upper bounds established using the linear programming bound (see [12]). For $n > bT$, however, the analogy to ordinary codes breaks: Whereas for $T=1$ the existence of MDS codes with length $s = n+1$ is easily established, it turns out that for $T \geq 3$ there exist dimensions n for which no generalized MDS code can be constructed at all. These cases are marked by the crosses in the table. As n increases, larger values of t are affected, too. More details on generalized MDS codes can be found in [7] (see also [34] and [31]).

One more example is given in Figure 6. It shows details about upper and lower bounds on the optimal t-parameters of $(t, 19, 71)$-nets in base $b=4$. A feature

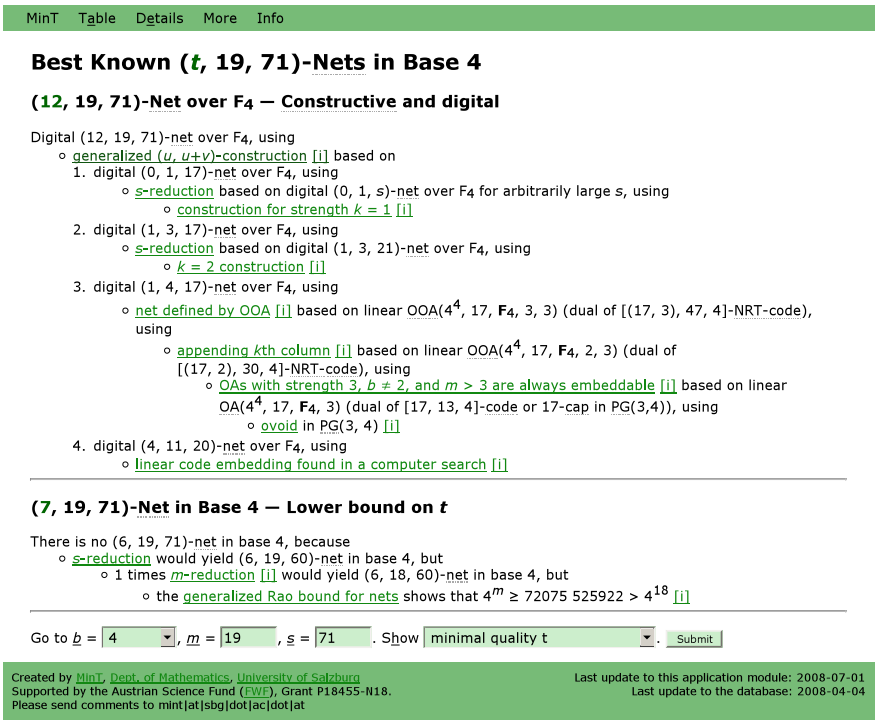


Fig. 6 Screenshot of MINT: details about the optimal t parameter of (digital) $(t, 19, 71)$-nets in base $b = 4$.

which has been added recently is that MINT is now able to handle propagation rules with an arbitrary number of parents. In this example the generalized $(u, u+v)$-construction for nets (Section 5 and [23]) is applied to four different nets in order to construct the digital $(12, 19, 71)$-net over $\mathbb{F}_4$.

Furthermore note that OOAs with depth $1 < T < k$ are used in the construction tree if they occur as intermediate results between the application of otherwise unrelated propagation rules. The $(1, 4, 17)$-net (the third building block of the final $(12, 19, 71)$-net) is constructed based on an $\mathrm{OA}(4^4, 17, \mathbb{F}_4, 3)$, i.e., the depth of the OA has to be increased from $T = 1$ (the OA) to at least $T = k = 3$ (equivalent to the net). Since the embedding from $T = 1$ to $T = 2$ is done with a different propagation rule than from $T = 2$ to $T = 3$, both propagation rules are listed, generating and using the intermediate OOA with depth $T = 2$.

Finally, note that MINT often provides additional information for the applied constructions, propagation rules, and bounds. For example, when the generalized Rao bound [14] is applied for proving the non-existence of a $(6, 18, 60)$-net in base 4, the exact lower bound on the number of points is given.

One new feature in MINT is noteworthy for everyone who has ever struggled with adapting to unfamiliar variable naming conventions: It is now possible to freely configure the variable names used by MINT, e.g. one can rename an "$[s, n, d]_b$-code" (MINT's default notation) to an "$[n, k, d]_q$-code" (the standard naming convention in coding theory) by a single mouse click.

4 New Nets based on OOA Propagation Rules

Since many constructions are known for linear codes, using them as building blocks for nets suggests itself. To this end, the depth of the OOA has to be increased from 1 to k. Whereas decreasing the depth of an OOA (without affecting other parameters) is trivial, increasing the depth is not possible in general. To overcome this problem, the following *coding-theoretic construction* for nets [2] can be used:

1. Start with a linear OOA$(b^m, Ts, \mathbb{F}_b, 1, k)$ with $T = 1$, i.e., an orthogonal array, which is the dual of a linear code.
2. *Fold* it to obtain a linear OOA$(b^m, s, \mathbb{F}_b, T, k)$ (by a simple rearrangement of the basis vectors).
3. A *Gilbert–Varšamov* (GV) argument [2] shows that the depth can be increased to k such that an OOA$(b^m, s, \mathbb{F}_b, k, k)$ is obtained, which is equivalent to an $(m - k, m, s)$-net.

Direct Product of OOAs

Coding-theoretic constructions for nets can be improved by applying a propagation rule for OOAs between Step 2 and 3. Propagation rules are methods for constructing an OOA based on another one. Many propagation rules are known for orthogonal arrays and linear codes. A comprehensive introduction to the theory of codes is given in [11]. We have generalized more than a dozen of these rules to the setting of OOAs. Some of them seem to be particularly useful for the construction of new nets.

Lemma 1. *If $\mathcal{A}$ is a (linear)* OOA(b^m, s, S_b, T, k) *and* $S_b^{(1,T)}$ *is the complete* OOA$(b^T, 1, S_b, T, T)$ *(consisting of all possible runs), then* $\mathcal{A} \times S_b^{(1,T)}$ *is a (linear)* OOA$(b^{m+T}, s + 1, S_b, T, k)$.

The proof is straightforward and has been given in [33]. Based thereupon, a new infinite family of nets can be obtained. These parameters were unknown for $u \geq 6$.

Theorem 1. *A digital* $(3u - 3, 3u + 3, 2^u)$*-net over* $\mathbb{F}_2$ *exists for all* $u \geq 1$.

Proof. Applying Lemma 1 to the OOA$(2^{3u+1}, 2^u - 1, \mathbb{F}_2, 2, 6)$ from [1] give a linear OOA$(2^{3u+3}, 2^u, \mathbb{F}_2, 2, 6)$ for all $u \geq 1$. Based thereupon, a GV-argument (see [2]) shows that the OOA can indeed be embedded in a digital net. □

Many other nets (with previously unknown parameters) can be obtained using Lemma 1 and similar methods:

Corollary 1. *(t, m, s)-nets with the following parameters exist:*

over $\mathbb{F}_2$	*over* $\mathbb{F}_3$	*over* $\mathbb{F}_4$	*over* $\mathbb{F}_5$	*over* $\mathbb{F}_7$	*over* $\mathbb{F}_8$
(15, 23, 40)	(8, 16, 23)	(21, 43, 42)	(5, 12, 26)	(12, 29, 39)	(16, 34, 66)
(17, 23, 128)	(18, 36, 29)	(37, 60, 131)	(26, 55, 56)	(16, 37, 49)	(17, 37, 66)
(103, 189, 66)	(24, 50, 33)	(38, 62, 131)	(61, 100, 253)	(25, 54, 73)	(31, 67, 98)
	(45, 87, 57)	(39, 64, 131)	(62, 102, 253)	(30, 66, 79)	(41, 86, 130)
		(40, 66, 131)	(63, 104, 253)	(36, 77, 97)	(48, 101, 145)
		(41, 68, 131)	(64, 106, 253)		
			(65, 108, 253)		

Proof. We prove only the first result. All other results are established similarly (see MINT for details). From a digital $(21-8, 21, 39)$-net over $\mathbb{F}_2$ constructed in [8] a linear OOA$(2^{21}, 39, \mathbb{F}_2, 2, 8)$ can be obtained by reducing T to 2. Lemma 1 gives a linear OOA$(2^{23}, 40, \mathbb{F}_2, 2, 8)$, which can be embedded in a $(23-8, 23, 40)$-net over $\mathbb{F}_2$ using a GV argument. □

Construction X

The following two results generalize *Construction X* from coding theory [11, Ch. 18, §7] and establish a method for obtaining nested generalized codes from (t, s)-sequences.

Theorem 2. *Let $\mathcal{C}_1$ denote a linear $[(s, T), n_1, d_1]$-code, which is a subcode of a linear $[(s, T), n_2, d_2]$-code $\mathcal{C}_2$, and let $\mathcal{C}_\mathrm{e}$ denote a linear $[(s_\mathrm{e}, T), n_\mathrm{e}, d_\mathrm{e}]$-code, all over the same field. Then a linear $[(s+s_\mathrm{e}, T), n_1+n_\mathrm{e}, d_2+d_\mathrm{e}]$-code can be constructed provided that $n_1+n_\mathrm{e} \le n_2$ and $d_2+d_\mathrm{e} \le d_1$.*

Proof. Let $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_\mathrm{e}$ denote the generator matrices of $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_\mathrm{e}$, respectively, such that the rows of $\mathbf{G}_1$ are a subset of the rows of $\mathbf{G}_2$. Let $\mathbf{G}_2'$ denote the $n_\mathrm{e} \times s$ matrix consisting of n_e rows of $\mathbf{G}_2$ that are not in $\mathbf{G}_1$. Then the new code is defined by the $(n_1+n_\mathrm{e}) \times ((s+s_\mathrm{e}), T)$ generator matrix

$$\mathbf{G} := \begin{pmatrix} \mathbf{G}_1 & \mathbf{0}_{n_1 \times (s_\mathrm{e}, T)} \\ \mathbf{G}_2' & \mathbf{G}_\mathrm{e} \end{pmatrix}.$$

$\mathbf{G}$ obviously generates an $[(s+s_\mathrm{e}, T), n_1+n_\mathrm{e}]$-code $\mathcal{C}$, so it remains to show that the minimum distance of $\mathcal{C}$ (which is given by the minimum weight of all non-zero codewords) is at least d_2+d_e. All code words formed by a non-trivial linear combination of the first n_1 rows of $\mathbf{G}$ have a weight of at least $d_1 \ge d_2+d_e$ because these are essentially the code words of $\mathcal{C}_1$ with $s_\mathrm{e}T$ additional 0's appended. All other non-zero code words have a weight of at least d_2+d_e because they are built using a non-zero code word from $\mathcal{C}_2$ next to a non-zero code word from $\mathcal{C}_\mathrm{e}$. □

Remark 1. Usually $\mathcal{C}_e$ will be chosen such that $n_2 = n_1 + n_e$ and $d_1 = d_2 + d_e$, however in some situations a smaller value of n_e or d_e may also yield good results.

Remark 2. Note that the linear case of Lemma 1 can be derived from Theorem 2 as follows: Let $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{A}^\perp$ and apply Construction X using the auxiliary $[(1,T),0,T+1]$-code $\mathcal{C}_e$. Then take the dual of the resulting code.

Lemma 2. *Given a digital (t,s)-sequence over $\mathbb{F}_b$ and a fixed positive integer $T \geq t/s$, one can construct a chain of linear generalized codes $\mathcal{C}_0 \subset \cdots \subset \mathcal{C}_{sT-t}$ with parameters $[(s,T),n,sT-n-t+1]$ for $n = 0,\ldots,sT-t$.*

Proof. We show the construction of the dual codes. Let $\mathcal{S}$ be a (digital) (t,s)-sequence in base b. We can construct (linear) OOA(b^{t+k},s,S_b,T,k) $\mathcal{A}_{T,k}$ for all $T \geq 1$ and all integers k with $0 \leq k \leq sT-t$ by taking the first b^{t+k} runs of $\mathcal{S}$ (which gives an OOA(b^{t+k},s,S_b,∞,k)) and reducing its depth to T. □

Remark 3. These OOAs are weaker than the (linear) OOA$(b^{t+k},s+1,S_b,T,k)$ obtained by the usual construction of a net from a sequence followed by extracting the embedded OOA from the net. However, the OOAs $\mathcal{A}_{T,k}$ from Lemma 2 have the additional property that $\mathcal{A}_{T,k} \subset \mathcal{A}_{T,k+1}$. In the linear case, $\mathcal{A}_{T,k}$ is a linear subspace of $\mathcal{A}_{T,k+1}$ and $\mathcal{A}_{T,k+1}^\perp \subset \mathcal{A}_{T,k}^\perp$.

Based thereupon many nets with new parameters can be found.

Corollary 2. *The following nets exist:*

over $\mathbb{F}_2$	*over* $\mathbb{F}_3$	*over* $\mathbb{F}_4$	*over* $\mathbb{F}_7$	*over* $\mathbb{F}_8$	
(107, 198, 67)	(47, 92, 57)	(22, 45, 43)	(12, 28, 40)	(18, 39, 68)	(43, 91, 131)
	(48, 94, 58)	(30, 60, 56)	(16, 36, 50)	(18, 40, 67)	(44, 94, 131)
	(48, 95, 57)		(17, 39, 50)	(19, 42, 67)	(49, 104, 145)
	(56, 109, 66)		(26, 57, 74)	(32, 69, 99)	(51, 109, 146)
	(56, 110, 65)	*over* $\mathbb{F}_5$	(37, 80, 97)	(33, 72, 99)	(53, 114, 148)
	(59, 118, 65)	(27, 58, 56)	(39, 85, 99)	(34, 75, 99)	(53, 115, 145)
	(60, 119, 67)			(42, 89, 130)	

Proof. We only show the existence of a $(47,92,57)$-net over $\mathbb{F}_3$. All other results are established similarly (see MINT for details).

An algebraic function field with full constant field $\mathbb{F}_3$, genus 40 and at least 56 rational places exists. Based thereupon a $(40,55)$-sequence over $\mathbb{F}_3$ can be obtained using Niederreiter–Xing's construction [26]. Lemma 2 with $T = 5$ yields a generalized $[(55,5),190,46]$-code $\mathcal{C}_1$ and $[(55,5),198,38]$-code $\mathcal{C}_2$. Theorem 2 applied to $\mathcal{C}_1 \subset \mathcal{C}_2$ with a $[(3,5),8,8]$-code $\mathcal{C}_e$ results in a $[(58,5),198,46]$-code or (after taking the dual) an OOA$(3^{92},58,\mathbb{F}_3,5,45)$. A GV-argument establishes the existence of the $(47,92,57)$-net over $\mathbb{F}_3$. □

5 A Generalized Matrix-Product Construction for Generalized Codes

The following construction is a generalization of the Blokh–Zyablov concatenation [4] (stated in modern terms and including many examples in [5, Section 4.1.9]) to generalized codes with arbitrary depth. We will see that this construction is highly powerful and a large number of well-known constructions can be derived from it as simple corollaries. The name of this construction is due to previous generalizations in [3] and [23] (see our remarks subsequent to Theorem 4).

Theorem 3. *Let*

$$\{\mathbf{0}\} = \mathcal{C}'_0 \subset \mathcal{C}'_1 \subset \cdots \subset \mathcal{C}'_r = \mathbb{F}_b^{(s',T')}$$

denote a chain of linear $[(s',T'), n'_j, d'_j]_b$*-codes (the* inner codes*) and let* $\mathbf{v}_1, \ldots, \mathbf{v}_{s'T'} \in \mathbb{F}_b^{(s',T')}$ *denote vectors such that* $\mathcal{C}'_j$ *is generated by* $\mathbf{v}_1, \ldots, \mathbf{v}_{n'_j}$ *for* $j = 1, \ldots, r$.

Furthermore let $\mathcal{C}_j$ *denote (not necessarily linear)* $((s,T), N_j, d_j)_{b_j}$*-codes with* $b_j = b^{e_j}$ *and* $e_j := n'_j - n'_{j-1}$ *for* $j = 1, \ldots, r$ *(the* outer codes*).*

Then an

$$((ss', TT'), N_1 \cdots N_r, \min_{\substack{1 \le j \le r \\ |\mathcal{C}_j| > 1}} d_j d'_j)_b\text{-code}$$

can be constructed as

$$\mathcal{C} := \left\{ \sum_{j=1}^{r} \varphi_j(\mathbf{x}_j) \;:\; \mathbf{x}_j \in \mathcal{C}_j \text{ for } j = 1, \ldots, r \right\}, \tag{2}$$

where $\varphi_j : \mathbb{F}_{b_j}^{(s,T)} \to \mathbb{F}_b^{(ss',TT')}$ *replaces each symbol (regarded as a vector of length* e_j *over* $\mathbb{F}_b$*) of a codeword from* $\mathcal{C}_j$ *by the corresponding linear combination of the* e_j *vectors* $\mathbf{v}_{n'_{j-1}+1}, \ldots, \mathbf{v}_{n'_j}$*. The elements in the resulting vector are grouped such that column* (a', τ') *(with* $1 \le a' \le s'$ *and* $1 \le \tau' \le T'$*) from* $\mathcal{C}'_j$ *and column* (a, τ) *(with* $1 \le a \le s$ *and* $1 \le \tau \le T$*) from* $\mathcal{C}_j$ *determine column* $((a-1)s' + a', (\tau-1)T' + \tau')$ *in the resulting code word in* $\mathcal{C}$*.*

Remark 4. Using the trivial $[(s,T), 0, sT+1]$-code $\{\mathbf{0}\}$ as $\mathcal{C}_j$ for one or more j's is perfectly valid and can lead to good results in certain cases. Since the minimum distance $sT+1$ does not affect the minimum distance of $\mathcal{C}$ at all, these codes can be excluded in the calculation of the minimum distance of $\mathcal{C}$ (therefore the additional condition "$|\mathcal{C}_j| > 1$" in the range of the minimum).

Remark 5. The construction is linear, i.e., if $\mathcal{C}_1, \ldots, \mathcal{C}_r$ are linear $[(s,T), n_j]$-codes, then $\mathcal{C}$ is a linear $[(ss', TT'), e_1 n_1 + \ldots + e_r n_r]$-code and its generator matrix can be determined easily: Applying φ_j to an $\mathbb{F}_b$-linear generator matrix of $\mathcal{C}_j$ yields $e_j n_j$ additional rows of the generator matrix of $\mathcal{C}$.

Proof (of Theorem 3). The length ss', the depth TT', and the alphabet size b of $\mathcal{C}$ follow from the definition of φ_j. $|\mathcal{C}|$ is given by $N_1 \cdots N_r$ provided that the gener-

ated code words are distinct, which is shown by the following examination of the minimum distance of $\mathcal{C}$.

Consider two arbitrary codewords $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ and let $\mathbf{x}_j, \mathbf{y}_j \in \mathcal{C}_j$ denote the code words in the original codes defining $\mathbf{x}$ and $\mathbf{y}$ (cf. (2)). If $\mathbf{x}_j = \mathbf{y}_j$ for all $j = 1, \ldots, r$, then $\mathbf{x} = \mathbf{y}$ and nothing needs to be shown. Thus, let u denote the smallest integer such that $\mathbf{x}_u \neq \mathbf{y}_u$. We have $1 \leq u \leq r$ and $|\mathcal{C}_u| > 1$. Write $\mathbf{x}_u - \mathbf{y}_u = \left(c^{(k,l)}\right)_{(k,l)\in\{1,\ldots,s\}\times\{1,\ldots,T\}}$.

Since φ_j is linear,

$$\mathbf{x}-\mathbf{y} = \sum_{j=1}^{u-1} \varphi_j(\mathbf{x}_j - \mathbf{y}_j) + \varphi_u(\mathbf{x}_u - \mathbf{y}_u) + \sum_{j=u+1}^{r} \varphi_j(\mathbf{x}_j - \mathbf{y}_j)$$

with the first sum being a concatenation of code words from $\mathcal{C}'_{u-1}$ and the second sum equal to $\mathbf{0}$. $\varphi_u(\mathbf{x}_u - \mathbf{y}_u)$ is a concatenation of code words from $\mathcal{C}'_u \setminus \mathcal{C}'_{u-1}$ (for non-zero $c^{(k,l)}$) and $\mathbf{0} \in \mathcal{C}'_u$ (for all (k,l) with $c^{(k,l)} = 0$). Since the sum of a codeword from $\mathcal{C}'_u \setminus \mathcal{C}'_{u-1}$ and one from $\mathcal{C}'_{u-1}$ is again in $\mathcal{C}'_u \setminus \mathcal{C}'_{u-1}$, there is a corresponding code word of weight at least d'_u in $\mathbf{x}-\mathbf{y}$ for each non-zero $c^{(k,l)}$.

Now let $\varphi(c^{(k,l)})$ denote the code word from $\mathcal{C}'_u$ generated by $c^{(k,l)}$. Write $w(\mathbf{x}_u - \mathbf{y}_u) = \sum_{k=1}^{s} \tau_k$ with $0 \leq \tau_k \leq T$ and $c^{(k,\tau_k)} \neq 0$ for $\tau_k \neq 0$. Similarly, write $w(\varphi(c^{(k,l)})) = \sum_{j=1}^{s'} \tau'_{(k,l),j}$. Then we have

$$\begin{aligned}
d(\mathbf{x},\mathbf{y}) &= w(\mathbf{x}-\mathbf{y}) \\
&= \sum_{k=1}^{s}\sum_{j=1}^{s'} \begin{cases} (\tau_k - 1)T' + \tau'_{(k,\tau_k),j} & \text{if } \tau_k \neq 0 \text{ and } \tau'_{(k,\tau_k),j} \neq 0 \\ 0 & \text{otherwise} \end{cases} \\
&\geq \sum_{\substack{k=1 \\ \tau_k>0}}^{s}\sum_{j=1}^{s'} \tau_k \tau'_{(k,\tau_k),j} = \sum_{\substack{k=1 \\ \tau_k>0}}^{s} \tau_k w(\varphi(c^{(k,\tau_k)})) \\
&\geq w(\mathbf{x}_u - \mathbf{y}_u) d'_u \geq d_u d'_u \,,
\end{aligned}$$

which concludes the proof. □

In the important case that the inner codes are normal codes (i.e., $T' = 1$) a slightly stronger result is possible, which often leads to improved parameters. It allows to use codes with different lengths as outer codes.

Theorem 4. *Let $\mathcal{C}'_0 \subset \ldots \subset \mathcal{C}'_r$ as in Theorem 3, but restricted to $T' = 1$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_{s'}$ be defined as in Theorem 3, but assume (without loss of generality) that the first $i-1$ elements of $\mathbf{v}_i$ are 0. In other words, $(\mathbf{v}_1, \ldots, \mathbf{v}_{s'})^T$ forms an $s' \times s'$ upper triangular matrix.*

Let b_j and e_j be defined as in Theorem 3. For positive integers $s_1 \leq s_2 \leq \cdots \leq s_r$ let $\mathcal{C}_j$ denote a (not necessarily linear) $((s_j, T), N_j, d_j)_{b_j}$-code for $j = 1, \ldots, r$.

Then an

$$((s_1e_1+\ldots+s_re_r,T),N_1\cdots N_r,\min_{\substack{1\le j\le r\\ |\mathcal{C}_j|>1}} d_jd'_j)_b\text{-code}$$

can be constructed.

Proof. The construction is performed in three steps:

1. Let $s := s_r$ and create new codes $\mathcal{D}_j$ by embedding each code $\mathcal{C}_j$ in $\mathbb{F}_{b_j}^{(s,T)}$ by prepending $\mathbf{0} \in \mathbb{F}_{b_j}^{(s-s_j,T)}$ to each code word.
2. Obtain a new code $\mathcal{D}$ by applying Theorem 3 using $\mathcal{C}'_1,\ldots,\mathcal{C}'_r$ as inner codes and $\mathcal{D}_1,\ldots,\mathcal{D}_r$ as outer codes.
3. For $i=1,\ldots,s'$ choose j minimal such that $e_1+\ldots+e_j \ge i$ and construct a code $\mathcal{C}$ from $\mathcal{D}$ by deleting all columns (k,τ) in $\mathcal{D}$ with $k = 0s'+i, 1s'+i,\ldots,(s-s_j-1)s'+i$.
   The total number of deleted blocks is $e_1(s-s_1)+\ldots+e_r(s-s_r)$, thus the length of $\mathcal{C}$ is $s_1e_1+\ldots+s_re_r$. The deleted positions are $\mathbf{0}\in\mathbb{F}_b^T$ for each $\varphi_j(\mathbf{x}_j)$, either due to a 0 in $\mathbf{v}_i$ or due to a $\mathbf{0}$ appended to $\mathbf{x}_j \in \mathcal{C}_j$, thus neither the dimension nor the minimum distance of $\mathcal{D}$ is affected.

□

Remark 6. For $T' > 1$ only weaker results are possible. The reason is that for $T'=1$ the matrix $(\mathbf{v}_1,\ldots,\mathbf{v}_{s'})^T$ is in upper triangular form and can be used for cancelling one additional block in each row. This is not possible for $T' > 1$, because in this setting an additional block can only be cancelled every T' rows.

A large number of well-known constructions turn out to be special cases of Theorem 3 or 4.

- As already stated in the introduction to this section, the restriction of Theorem 3 to normal codes (i.e., $T = T' = 1$) has been given in essentially complete form by Blokh and Zyablov in [4], which predates all other publications by more than two decades. The only other restrictions in this work are that only the case for $b = 2$ and for linear codes is considered.
- The complete result of Theorem 4 for $T = T' = 1$ and linear codes can be found (without proof, but with many examples) in [5].
- The constructions in [28] and [3] are essentially equivalent to $T = T' = 1$, $e_1 = \cdots = e_r = 1$, and $s_1 = \cdots = s_r$.
- The matrix-product construction for digital nets presented in [23] considers the case $e_j = 1$ for all j, for linear codes, and uses upper triangular NSC (non-singular by columns) matrices for obtaining the vectors $\mathbf{v}_i$, which is essentially equivalent to restricting the inner codes to (truncated) Reed–Solomon codes [30].
- For $r = 2$, $\mathbf{v}_1 = (1,1)$, $\mathbf{v}_2 = (0,1)$, and $e_1 = e_2 = 1$ we obtain the well-known $(u, u+v)$-construction, which is due to [29] for depth 1 and $s_1 = s_2$, [35] for depth 1 and $s_1 \le s_2$, and [2] for generalized codes.
- The $(u, u+av, u+v+w)$-construction follows for $r = 3$, $\mathbf{v}_1 = (1,1,1)$, $\mathbf{v}_2 = (0,a,1)$ with $a \notin \{0,1\}$, $\mathbf{v}_3 = (0,0,1)$, and $e_1 = e_2 = e_3 = 1$.

- Concatenation of the inner code $\mathcal{C}'_i$ with the outer code $\mathcal{C}_1$ can be achieved by setting $e_1 = i$ and using a trivial $[s', 0, s'+1]$-code for $j > 1$.
- The trace code from $\mathbb{F}_{b^{s'}}$ to $\mathbb{F}_b$ is obtained by using $\mathbf{v}_i = \mathbf{e}_i$ for $i = 1, \ldots, s'$, $r = 1$, and $e_1 = s'$.

Finally we show that Theorem 4 is applicable to digital nets in its full generality:

Corollary 3. *Let $\mathbf{v}_1, \ldots, \mathbf{v}_{s'}$ and $\mathcal{C}'_j$, b_j, and e_j for $j = 1, \ldots, r$ be defined as in Theorem 4. For positive integers $s_1 \le s_2 \le \cdots \le s_r$ let $\mathcal{N}_j$ denote digital (t_j, m_j, s_j)-nets over $\mathbb{F}_{b_j}$.*

Then a digital (t, m, s)-net over $\mathbb{F}_b$ with

$$s = \sum_{j=1}^{r} e_j s_j,$$

$$m = \sum_{j=1}^{r} e_j m_j,$$

and

$$t \le m + 1 - \min_{1 \le j \le r} (m_j - t_j + 1) d'_j$$

can be constructed.

Proof. Set $d_j := m_j - t_j + 1$ for $j = 1, \ldots, r$ and $T := \max_{j=1,\ldots,r} d_j d'_j - 1$. Each net $\mathcal{N}_j$ defines a linear $\mathrm{OOA}(b_j^{m_j}, s_j, \mathbb{F}_{b_j}, T, d_j - 1)$, which is the dual of a linear $[(s_j, T), Ts_j - m_j, d_j]_{b_j}$-code $\mathcal{C}_j$. Applying Theorem 4 using $\mathcal{C}'_1, \ldots, \mathcal{C}'_r$ as inner codes and $\mathcal{C}_1, \ldots, \mathcal{C}_r$ as outer codes yields an

$$[(s, T), \sum_{j=1}^{r} e_j (Ts_j - m_j), d]_b\text{-code } \mathcal{C}$$

with

$$d = \min_{1 \le j \le r} d_j d'_j = \min_{1 \le j \le r} (m_j - t_j + 1) d'_j.$$

The dual of $\mathcal{C}$ is a linear

$$\mathrm{OOA}(b^m, s, \mathbb{F}_b, T, d - 1)$$

because

$$Ts - \sum_{j=1}^{r} e_j (Ts_j - m_j) = \sum_{j=1}^{r} e_j m_j = m.$$

This OOA can be embedded in a digital $(m + 1 - d, m, s)$-net over $\mathbb{F}_b$ because $T \ge d - 1$. □

Example 2. Choose the Reed–Solomon codes over $\mathbb{F}_4$ with dimensions $0, \ldots, 4$ as $\mathcal{C}'_0 \subset \ldots \subset \mathcal{C}'_4$, i.e., $\mathcal{C}'_j$ is a $[4, j, 5 - j]_4$-code and $e_j = 1$ for all $j = 0, \ldots, 4$. Using a $(0, 1, 17)$-net as $\mathcal{N}_1$, a $(1, 3, 17)$-net as $\mathcal{N}_2$, a $(1, 4, 17)$-net as $\mathcal{N}_3$, and a $(4, 11, 20)$-net as $\mathcal{N}_4$ (all digital over $\mathbb{F}_4$), Corollary 3 yields a digital (t, m, s)-net over $\mathbb{F}_4$ with

$s = 17+17+17+20 = 71$, $m = 1+3+4+11 = 19$, and

$$t \leq 19+1-\min\{2\cdot 4, 3\cdot 3, 4\cdot 2, 8\cdot 1\} = 19+1-8 = 12.$$

No $(t, 19, 71)$-net over $\mathbb{F}_4$ with a lower t value is known, as we can retrieve in MINT (see Figure 6).

Acknowledgements This research has been supported by the Austrian Science Foundation (FWF), project no. P 18455-N18.

References

1. Bierbrauer, J., Edel, Y.: Families of nets of low and medium strength. Integers. Electronic Journal of Combinatorial Number Theory **5**(3), #A03, 13 pages (2005). (electronic)
2. Bierbrauer, J., Edel, Y., Schmid, W.Ch.: Coding-theoretic constructions for (t, m, s)-nets and ordered orthogonal arrays. Journal of Combinatorial Designs **10**(6), 403–418 (2002)
3. Blackmore, T., Norton, G.H.: Matrix-product codes over $\mathbb{F}_q$. Applicable Algebra in Engineering, Communication and Computing **12**(6), 477–500 (2001)
4. Blokh, E.L., Zyablov, V.V.: Coding of generalized concatenated codes. Problems of Information Transmission **10**, 218–222 (1974)
5. Brouwer, A.E.: Bounds on the size of linear codes. In: V.S. Pless, W.C. Huffman (eds.) Handbook of Coding Theory, vol. 1, pp. 295–461. Elsevier Science (1998)
6. Clayman, A.T., Lawrence, K.M., Mullen, G.L., Niederreiter, H., Sloane, N.J.A.: Updated tables of parameters of (t, m, s)-nets. Journal of Combinatorial Designs **7**(5), 381–393 (1999)
7. Dougherty, S.T., Skriganov, M.M.: Maximum distance separable codes in the ρ metric over arbitrary alphabets. J. Algebraic Combin. **16**(1), 71–81 (2002)
8. Edel, Y.: Generator matrices of some linear binary OOA. Available at `http://www.mathi.uni-heidelberg.de/~yves/Matritzen/OOAs/q=2/M2(39,7,21,8).html`
9. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arithmetica **41**, 337–351 (1982)
10. Lawrence, K.M.: A combinatorial characterization of (t, m, s)-nets in base b. Journal of Combinatorial Designs **4**(4), 275–293 (1996)
11. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
12. Martin, W.J.: Linear programming bounds for ordered orthogonal arrays and (t, m, s)-nets. In: H. Niederreiter, J. Spanier (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 368–376. Springer-Verlag (2000)
13. Martin, W.J., Stinson, D.R.: Association schemes for ordered orthogonal arrays and (t, m, s)-nets. Canadian Journal of Mathematics **51**(2), 326–346 (1999)
14. Martin, W.J., Stinson, D.R.: A generalized Rao bound for ordered orthogonal arrays and (t, m, s)-nets. Canadian Mathematical Bulletin **42**(3), 359–370 (1999)
15. Mullen, G.L., Mahalanabis, A., Niederreiter, H.: Tables of (t, m, s)-net and (t, s)-sequence parameters. In: H. Niederreiter, P.J.S. Shiue (eds.) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, *Lecture Notes in Statistics*, vol. 106, pp. 58–86. Springer-Verlag (1995)
16. Mullen, G.L., Schmid, W.Ch.: An equivalence between (t, m, s)-nets and strongly orthogonal hypercubes. Journal of Combinatorial Theory, Series A **76**(1), 164–174 (1996)
17. Niederreiter, H.: Point sets and sequences with small discrepancy. Monatshefte für Mathematik **104**(4), 273–337 (1987)

18. Niederreiter, H.: Low-discrepancy and low-dispersion sequences. Journal of Number Theory **30**(1), 51–70 (1988)
19. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992)
20. Niederreiter, H.: Constructions of (t, m, s)-nets. In: H. Niederreiter, J. Spanier (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 70–85. Springer-Verlag (2000)
21. Niederreiter, H.: Constructions of (t, m, s)-nets and (t, s)-sequences. Finite Fields and Their Applications **11**(3), 578–600 (2005)
22. Niederreiter, H.: Nets, (t, s)-sequences, and codes. In: A. Keller et al. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 83–100. Springer-Verlag (2008)
23. Niederreiter, H., Özbudak, F.: Matrix-product constructions of digital nets. Finite Fields and Their Applications **10**(3), 464–479 (2004)
24. Niederreiter, H., Pirsic, G.: Duality for digital nets and its applications. Acta Arithmetica **97**(2), 173–182 (2001)
25. Niederreiter, H., Xing, C.P.: Low-discrepancy sequences obtained from algebraic function fields over finite fields. Acta Arithmetica **72**(3), 281–298 (1995)
26. Niederreiter, H., Xing, C.P.: Low-discrepancy sequences and global function fields with many rational places. Finite Fields and Their Applications **2**(3), 241–273 (1996)
27. Niederreiter, H., Xing, C.P.: Quasirandom points and global function fields. In: S. Cohen, H. Niederreiter (eds.) Finite Fields and Applications, *Lect. Note Series of the London Math. Soc.*, vol. 233, pp. 269–296. Cambridge University Press (1996)
28. Özbudak, F., Stichtenoth, H.: Note on Niederreiter–Xing's propagation rule for linear codes. Applicable Algebra in Engineering, Communication and Computing **13**(1), 53–56 (2002)
29. Plotkin, M.: Binary codes with specified minimum distance. IEEE Transactions on Information Theory **6**(4), 445–450 (1960)
30. Reed, I.S., Solomon, G.: Polynomial codes over certain finite fields. Journal of the Society for Industrial and Applied Mathematics **8**(2), 300–304 (1960)
31. Rosenbloom, M.Y., Tsfasman, M.A.: Codes for the m-metric. Problems of Information Transmission **33**, 55–63 (1997)
32. Schürer, R., Schmid, W. Ch.: MinT: A database for optimal net parameters. In: H. Niederreiter, D. Talay (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 457–469. Springer-Verlag (2006)
33. Schürer, R., Schmid, W.Ch.: MinT – architecture and applications of the (t, m, s)-net database. Mathematics and Computers in Simulation (to be published)
34. Skriganov, M.M.: Coding theory and uniform distributions. St. Petersburg Math. J. **13**(2), 301–337 (2002). Translated from Algebra i Analiz 13 (2001), 191–231
35. Sloane, N.J.A., Whitehead, D.S.: A new family of single-error correcting codes. IEEE Transactions on Information Theory **16**(6), 717–719 (1970)
36. Sobol', I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. U. S. S. R. Computational Mathematics and Mathematical Physics **7**(4), 86–112 (1967)
37. Xing, C.P., Niederreiter, H.: A construction of low-discrepancy sequences using global function fields. Acta Arithmetica **73**(1), 87–102 (1995)

Part III
Contributed Articles

Recursive Computation of Value-at-Risk and Conditional Value-at-Risk using MC and QMC

Olivier Bardou, Noufel Frikha, and Gilles Pagès

Abstract Value-at-Risk (VaR) and Conditional-Value-at-Risk (CVaR) are two widely-used measures in risk management. This paper deals with the problem of estimating both VaR and CVaR using stochastic approximation (with decreasing steps): we propose a first Robbins-Monro (RM) procedure based on Rockafellar-Uryasev's identity for the CVaR. The estimator provided by the algorithm satisfies a Gaussian Central Limit Theorem. As a second step, in order to speed up the initial procedure, we propose a recursive and adaptive importance sampling (IS) procedure which induces a significant variance reduction of both VaR and CVaR procedures. This idea, which has been investigated by many authors, follows a new approach introduced in Lemaire and Pagès [20]. Finally, to speed up the initialization phase of the IS algorithm, we replace the original confidence level of the VaR by a deterministic moving risk level. We prove that the weak convergence rate of the resulting procedure is ruled by a Central Limit Theorem with minimal variance and we illustrate its efficiency by considering typical energy portfolios.

O. Bardou
Laboratoire de Probabilités et Modèles aléatoires France and GDF Suez, Research and Innovation Department
e-mail: olivier-aj.bardou@gdfsuez.com

N. Frikha
Laboratoire de Probabilités et Modèles aléatoires, Université Pierre et Marie Curie, UMR 7599 and GDF Suez, Research and Innovation Department, France
e-mail: noufel-externe.frikha@gdfsuez.com

G. Pagès
Laboratoire de Probabilités et Modèles aléatoires, UMR 7599, Université Pierre et Marie Curie, France
e-mail: gilles.pages@upmc.fr

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_11,

1 Introduction

Following financial institutions, energy companies are developing a risk management framework to face the new price and volatility risks associated to the growth of energy markets. Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) are certainly the best known and the most common risk measures used in this context, especially for the evaluation of extreme losses potentially faced by traders. Naturally related to rare events, the estimation of these risk measures is a numerical challenge. The Monte Carlo method, which is often the only available numerical device in such a general framework, must preferably be associated to efficient variance reduction techniques to remedy its slow convergence rate.

By definition, the VaR_α of a given portfolio at a specified level $\alpha \in (0,1)$ is the lowest amount not exceeded by the loss with probability α. The CVaR_α is the conditional expectation of the portfolio's losses above the VaR_α. Compared to VaR, CVaR is known to have better properties. It is a coherent risk measure in the sense of Artzner, Delbaen, Eber and Heath, see [2]. The most commonly used method to compute VaR is the inversion of the simulated empirical loss distribution function using Monte Carlo or historical simulation tools. Another well-known method relies on linear or quadratic expansion of the distribution of the loss see e.g. [6], [7], [14], [15] and [24]. However, such approximations are no longer acceptable when considering portfolios over a long time interval as it is often the case in energy markets (1 year up to 10 years) or when the loss is a functional of a general path-dependent Stochastic Differential Equation (SDE).

In the context of hedging or optimizing a portfolio of financial instruments by reducing the Conditional Value-at-Risk, it is shown in [23] that it is possible to compute both VaR and CVaR (actually calculate VaR and optimize CVaR) by solving a convex optimization problem with a linear programming approach. It consists in generating loss scenarios and then in introducing constraints in the linear programming problem. Although a different problem is addressed in this paper, the method described in [23] can be used to compute both VaR and CVaR. The advantage of such an approach is that it is possible to estimate both VaR and CVaR simultaneously and without assuming that the market prices have a specific distribution like normal or log-normal. The main drawback is that the dimension (number of constraints) of the linear programming problem to be solved is equal to the number of simulated scenarios. In our approach, we are not constrained by the number of generated sample paths used in the estimation.

The idea to compute both VaR and CVaR with one procedure comes from the fact that they appear as the solutions and the value of the same convex optimisation problem (see Proposition 1) as demonstrated in [23]. Moreover, the convex objective function of the minimization problem reads as an expectation and its gradient too, so that a method to estimate both quantities is to devise a stochastic gradient algorithm and an averaging procedure. Thus, we derive a global recursive procedure which is an alternative method compared to the basic two step procedure which consists in first estimating the VaR using the inversion of the empirical function method and then estimating the CVaR by averaging. This optimization approach provides

a convex Lyapunov function (the gradient of the objective function) for the system which allows to derive the almost sure ($a.s.$) convergence of the VaR procedure. Moreover, the implementation of the algorithm is very easy. From a practical point of view, there is no reason to believe that this procedure behaves better than this alternative method. However, the proposed algorithm is just a building block of a recursive and adaptive IS procedure. As a matter of fact, basically in this kind of problem we are interested by events that are observed with a very small probability (usually less that 5%, 1% or even 0.1%) thus we obtain few significant replications to update our estimates. When α is close to 1 (otherwise it is not a numerical challenge), VaR and CVaR are fundamentally related to rare events thus as a necessary improvement, we also introduce a recursive variance reduction method. To compute more accurate estimates of both quantities of interest, it is necessary to generate more samples in the tail of the loss distribution, the area of interest. A general tool used in this situation is IS. The basic principle of IS is to modify the distribution of the loss by an equivalent change of measure to obtain more "interesting" samples that will lead to better estimates of the VaR and CVaR. The main issue is to find the right change of measure (among a parametrized family) that will induce a significant variance reduction. In [10], the VaR_α is estimated by using a quantile based on a weighted empirical distribution function and combined with a projected IS algorithm. This kind of algorithm is known to converge after a long stabilization phase and provided that the sequence of compact sets has been specified appropriately. Specifying adequately the sequence of compact sets is a significant challenge. Our IS parameters are optimized by an adaptive unconstrained (i.e., without projections) RM algorithm which is combined with our VaR-CVaR procedure. The fact that our estimates for both VaR and CVaR are recursive makes the algorithm well suited for the recursive IS procedure.

One major issue that arises when combining the VaR-CVaR algorithm with the recursive IS procedure is to ensure importance sampling parameters do move appropriately toward the critical risk area. They may remain "stuck" at the very beginning of the IS procedure. To circumvent this problem, we make the confidence level α slowly increase from a low level (say 50%) to the true value of α by introducing a deterministic sequence α_n that converges to α at a prespecified rate. This kind of incremental threshold increase has been proposed previously in [25] in a different framework (use of cross entropy). We finally discuss the possibility of plugging low-discrepancy sequences instead of pseudo-random numbers in the VaR-CVaR algorithm.

The paper is organized as follows. In Section 2 we briefly present the VaR-CVaR Robbins Monro algorithm in its first and naive version. It's a building block that is necessary in order to combine it with adaptive IS. Then, we introduce and study the adaptive variance reduction procedure and present how it modifies the asymptotic variance of our first CLT in Section 2.2. Numerical illustrations are given in Section 3.

2 Design of the VaR-CVaR Stochastic Approximation Algorithm

2.1 Devise of a VaR-CVaR Procedure (First Phase)

2.1.1 Definition of the VaR and the CVaR

We consider that the loss of the portfolio over the considered time horizon can be written as a function of a structural finite dimensional random vector, i.e., $L = \varphi(X)$ where X is a $\mathbb{R}^d$-valued random vector defined on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and $\varphi : \mathbb{R}^d \to \mathbb{R}$ is a Borel function. Thus, φ is the function describing the composition of the portfolio which remains fixed and X is a structural d-dimensional random vector used to model the market prices over a given time interval. We only rely on the fact it is possible to sample from the distribution of X. For instance, in a Black-Scholes framework, X is generally a vector of Brownian increments related to the Euler scheme of a diffusion. The VaR at level $\alpha \in (0,1)$ is the lowest α-quantile of the distribution $\varphi(X)$:

$$\mathrm{VaR}_\alpha(\varphi(X)) := \inf\{\xi \mid \mathbb{P}(\varphi(X) \le \xi) \ge \alpha\}.$$

Since $\lim_{\xi \to +\infty} \mathbb{P}(\varphi(X) \le \xi) = 1$ and $\lim_{\xi \to -\infty} \mathbb{P}(\varphi(X) \le \xi) = 0$, the VaR always exists. We assume that the distribution function of $\varphi(X)$ is continuous (i.e., without atoms) thus the VaR_α is the lowest solution of the equation:

$$\mathbb{P}(\varphi(X) \le \xi) = \alpha.$$

Another risk measure generally used to provide information about the tail of the distribution of $\varphi(X)$ is the *Conditional Value at Risk* (CVaR) (at level α). Assuming that $\varphi(X) \in L^1(\mathbb{P})$, the CVaR is defined by:

$$\mathrm{CVaR}_\alpha(\varphi(X)) := \mathbb{E}[\varphi(X) | \varphi(X) \ge \mathrm{VaR}_\alpha(\varphi(X))].$$

2.1.2 Stochastic Gradient and Averaging Procedure: A Naive Approach

In this paragraph, we present the framework and the first building block of the VaR-CVaR algorithm. Obviously, there is no reason to believe that this first and naive version can do better than others method, like the inversion of the empirical distribution function. However, our quantile estimate has the advantage to be recursive thus it can be combined later with a recursive IS algorithm in a suitable way.

Proposition 1. *Let V be the function defined on $\mathbb{R}$ by: $\xi \mapsto \xi + \frac{1}{1-\alpha}\mathbb{E}[(\varphi(X) - \xi)_+)]$. Suppose that the distribution function of $\varphi(X)$ is continuous. Then, the function V is convex, differentiable and the $\mathrm{VaR}_\alpha(\varphi(X))$ is any point of the set:*

$$\arg\min\ V = \left\{\xi \in \mathbb{R} \mid V'(\xi) = 0\right\} = \{\xi \mid \mathbb{P}(\varphi(X) \le \xi) = \alpha\}$$

where V' is the derivative defined of V. Moreover, for every $\xi \in \mathbb{R}$, $V'(\xi) = \mathbb{E}[H_1(\xi, X)]$ where

$$H_1(\xi, x) := 1 - \frac{1}{1-\alpha} \mathbf{1}_{\{\varphi(x) \geq \xi\}}.$$

Furthermore, $CVaR_\alpha(\varphi(X)) = \min_{\xi \in \mathbb{R}} V(\xi)$.

Proof. Since the function $\xi \mapsto (\varphi(x) - \xi)_+$, $x \in \mathbb{R}^d$, is convex, the function V is convex. $\mathbb{P}(dw)$-*a.s.*, $H_1(\xi, X(w))$ exists at every $\xi \in \mathbb{R}$ and

$$\mathbb{P}(dw)\text{-}a.s., \quad |H_1(\xi, X(w))| \leq 1 \vee \frac{\alpha}{1-\alpha}.$$

Thanks to Lebesgue Dominated Convergence Theorem, one can interchange differentiation and expectation, so that V is differentiable with derivative $V'(\xi) = \mathbb{E}[H_1(\xi, X)] = 1 - \frac{1}{1-\alpha}\mathbb{P}(\varphi(X) > \xi)$ and reaches its absolute minimum at any ξ^* satisfying $\mathbb{P}(\varphi(X) > \xi^*) = 1 - \alpha$, i.e., $\mathbb{P}(\varphi(X) \leq \xi^*) = \alpha$. Moreover,

$$V(\xi^*) = \frac{\xi^* \mathbb{E}[\mathbf{1}_{\varphi(X) > \xi^*}] + \mathbb{E}[(\varphi(X) - \xi^*)_+]}{\mathbb{P}(\varphi(X) > \xi^*)} = \mathbb{E}\big[\varphi(X) | \varphi(X) > \xi^*\big].$$

This completes the proof. □

Since we are looking for ξ for which $\mathbb{E}[H_1(\xi, X)] = 0$, we implement a stochastic gradient descent derived from the Lyapunov function V to approximate $\xi^* := \mathrm{VaR}_\alpha(\varphi(X))$, i.e., we use the RM algorithm:

$$\xi_n = \xi_{n-1} - \gamma_n H_1(\xi_{n-1}, X_n), \quad n \geq 1 \tag{1}$$

where $(X_n)_{n \geq 1}$ is an i.i.d. sequence of random variables with the same distribution as X, independent of ξ_0, with $\mathbb{E}[|\xi_0|] < \infty$ and $(\gamma_n)_{n \geq 1}$ is a positive deterministic step sequence (decreasing to 0) satisfying

$$\sum_{n \geq 1} \gamma_n = +\infty \quad \text{and} \quad \sum_{n \geq 1} \gamma_n^2 < +\infty. \tag{2}$$

A natural idea in order to estimate $C^* := \mathrm{CVaR}_\alpha$ is to devise an averaging procedure of the above quantile search algorithm, namely C_0 and for $n = 1, 2, \ldots$,

$$C_n = \frac{1}{n} \sum_{k=0}^{n-1} \xi_k + \frac{1}{1-\alpha} (\varphi(X_{k+1}) - \xi_k)_+ = C_{n-1} - \frac{1}{n} H_2(\xi_{n-1}, C_{n-1}, X_n). \tag{3}$$

where $H_2(\xi, c, x) := c - \xi - \frac{1}{1-\alpha} (\varphi(x) - \xi)_+$. The resulting algorithm reads for $n \geq 1$:

$$\begin{cases} \xi_n = \xi_{n-1} - \gamma_n H_1(\xi_{n-1}, X_n), & \xi_0 \in \mathbb{R} \\ C_n = C_{n-1} - \frac{1}{n} H_2(\xi_{n-1}, C_{n-1}, X_n), & C_0 = 0. \end{cases} \tag{4}$$

At this stage, we are facing a two-time scale algorithm in (4), i.e., with different steps γ_n and $1/n$. Since H_1 is free of C_n, we know that the stepsize sequence that provides

the best convergence rate (see e.g. [9]) is $\gamma_n = \gamma_1/n$ where γ_1 has to be chosen adequately. So we verify a posteriori that the resulting algorithm could theoretically be reduced to a one-time-scale procedure. Our numerical experiments indicate that the one-time scale procedure provides less variance during the first iterations than others procedures with different steps size. A slight modification consists in using both procedures with the same step size $(\gamma_n)_{n\geq 1}$ satisfying condition (2) (for more details about the different possible choice we refer to [3]). The resulting algorithm can be written as for $n \geq 1$

$$\begin{cases} \xi_n = \xi_{n-1} - \gamma_n H_1(\xi_{n-1}, X_n), \quad \xi_0 \in \mathbb{R}, \\ C_n = C_{n-1} - \gamma_n H_2(\xi_{n-1}, C_{n-1}, X_n), \quad C_0 = 0. \end{cases} \tag{5}$$

The recurrence for (ξ_n) does not involve (C_n) so its $a.s.$ convergence is ensured by the classical RM Theorem (see e.g. [9]). Then, it is possible to show that C_n converges $a.s.$ toward C^* provided that the distribution function of $\varphi(X)$ is continuous and increasing and that $\varphi(X)$ is square integrable (we refer to [3] for a proof). To achieve the best convergence rate, we are led to introduce the Ruppert and Polyak's averaging principle (see [17] and [26]). If we set $\gamma_n = cn^{-p}$, with $\frac{1}{2} < p < 1$, $c > 0$ in (5) and compute the Césaro means of both components

$$\begin{cases} \overline{\xi}_n := \frac{1}{n}\sum_{k=1}^n \xi_k = \overline{\xi}_n - \frac{1}{n+1}(\overline{\xi}_n - \xi_n) \\ \overline{C}_n := \frac{1}{n}\sum_{k=1}^n C_k = \overline{C}_n - \frac{1}{n+1}(\overline{C}_n - C_n) \end{cases} \tag{6}$$

where (ξ_k, C_k), $k \geq 0$ is defined by (5) then, provided that

$$\mathbb{E}\left[|\varphi(X)|^{2a}\right] < +\infty \text{ for some } a > 1, \tag{7}$$

and that the distribution of $\varphi(X)$ has a positive probability density $f_{\varphi(X)}$ on its support, we obtain asymptotically efficient estimators which satisfy the Gaussian CLT:

$$\sqrt{n}\begin{pmatrix} \overline{\xi}_n - \xi^* \\ \overline{C}_n - C^* \end{pmatrix} \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Sigma) \tag{8}$$

where the asymptotic covariance matrix Σ is given by

$$\Sigma = \begin{pmatrix} \frac{\alpha(1-\alpha)}{f_{\varphi(X)}(\xi^*)} & \frac{\alpha}{(1-\alpha) f_{\varphi(X)}(\xi^*)}\mathbb{E}\left[(\varphi(X) - \xi^*)_+\right] \\ \frac{\alpha}{(1-\alpha) f_{\varphi(X)}(\xi^*)}\mathbb{E}\left[(\varphi(X) - \xi^*)_+\right] & \frac{1}{(1-\alpha)^2}\mathrm{Var}\,(\varphi(X) - \xi^*)_+ \end{pmatrix}. \tag{9}$$

Remark 1. The result above is not surprising. The asymptotic variance of the quantile estimate based on the inversion of the empirical distribution function is the same than the one of our procedure $\overline{\xi}_n$, see for example [27], page 75.

The bottleneck in using the above algorithm lies in its very slow convergence owing to the fact that $\mathbb{P}(\varphi(X) > \xi^*) = 1 - \alpha$ is close to 0 in practical implementations, so that we observe few significant replications to update our estimates. Moreover,

in the bank and energy sectors, practitioners usually deal with huge portfolio composed by hundreds or thousands of risk factors and options. The evaluation step of $\varphi(X)$ may require a lot of computational time. Consequently, to achieve accurate estimates of both VaR_α and CVaR_α with reasonable computational effort, the above algorithm (6) needs to be speeded up by an IS procedure to recenter simulation "where things do happen", i.e., scenarios for which $\varphi(X)$ exceeds ξ.

2.2 Variance Reduction Using Adaptive Recursive Importance Sampling (Final Phase)

In this section, we investigate the IS by translation. We show how the IS algorithm investigated in [20] can be combined adaptively with our first algorithm. Consequently, every new sample is used to dynamically optimize the IS change of measure and the estimate of both VaR and CVaR.

2.2.1 Some Background on Recursive IS

Suppose that $F(X)$ is a square integrable random variable such that $\mathbb{P}(F(X) \neq 0) > 0$ and where X is a random vector with density function p over $\mathbb{R}^d$. The main idea of IS by translation, applied to the computation of $\mathbb{E}[F(X)]$, is to use the invariance of the Lebesgue measure by translation: it follows that for every $\theta \in \mathbb{R}^d$,

$$\mathbb{E}[F(X)] = \int_{\mathbb{R}^d} F(x)p(x)dx = \int_{\mathbb{R}^d} F(x+\theta)p(x+\theta)dx = \mathbb{E}\left[F(X+\theta)\frac{p(X+\theta)}{p(X)}\right]. \tag{10}$$

Among all these random vectors with the same expectation, we want to select the one with the lowest variance, i.e., the one with lowest quadratic norm

$$Q(\theta) := \mathbb{E}\left[F^2(X+\theta)\frac{p^2(X+\theta)}{p^2(X)}\right], \quad \theta \in \mathbb{R}^d.$$

A reverse change of variable shows that:

$$Q(\theta) = \mathbb{E}\left[F^2(X)\frac{p(X)}{p(X-\theta)}\right], \quad \theta \in \mathbb{R}^d. \tag{11}$$

Now if the density function p of X satisfies

$$(i) \;\; p \text{ is log-}concave \quad \text{and} \quad (ii) \lim_{\|x\|\to+\infty} p(x) = 0 \tag{12}$$

where $\|.\|$ denotes the Euclidean norm, and

$$Q(\theta) < +\infty, \; \forall \theta \in \mathbb{R}^d \tag{13}$$

then, one shows that the function Q is finite, convex and goes to infinity at infinity, thus $\arg\min_\theta Q = \left\{\theta \in \mathbb{R}^d \mid \nabla Q(\theta) = 0\right\}$ is non empty, where ∇Q is the gradient of Q (for a proof, we refer to [20]). If ∇Q admits a representation as an expectation, then it is possible to devise a recursive RM procedure to approximate the optimal parameter θ^*, namely

$$\theta_n = \theta_{n-1} - \gamma_n K(\theta_{n-1}, X_n),\ n \geq 1 \tag{14}$$

where K is naturally defined by the formal differentiation of Q, for every $x \in \mathbb{R}^d$:

$$K(\theta, x) = F^2(x)\frac{p(x)}{p^2(x-\theta)}\nabla p(x-\theta). \tag{15}$$

Since we have no knowledge about the regularity of F and do not wish to have any, we differentiate the second representation of Q in (11) and not the first one. IS using stochastic approximation algorithms has been investigated by several authors, see e.g. [16], [11] and [13] in order to "optimize" or "improve" the change of measure in IS by a RM procedure. It has recently been studied in the Gaussian framework in [1] where (15) is used to design a stochastic gradient algorithm. However, the regular RM procedure (14) suffers from an instability issue coming from the fact that the classical sub-linear growth assumption in quadratic mean in the Robbins-Monro Theorem

$$\forall \theta \in \mathbb{R}^d,\ \ \mathbb{E}\left[K(\theta, X)^2\right]^{\frac{1}{2}} \leq C\,(1 + \|\theta\|) \tag{16}$$

is only fulfilled when F is constant, due to the behavior of the annoying term $p(x)/p(x-\theta)$ as θ goes to infinity. Consequently, θ_n can escape at infinity at almost every implementation as pointed out in [1]. To circumvent this problem, a "projected version" of the procedure based on repeated reinitializations when the algorithms exits from an increasing sequence of compact sets (while the step γ_n keeps going to 0) was used. This approach is known as the projection "à la Chen". It forces the stability of the algorithm and prevents explosion. Recently, IS using stochastic algorithm was deeply revisited in [20] to remove the constraints introduced by the original algorithm. Moreover, this construction is extended to a large class of probability distributions and to diffusion process. Thanks to another translation of the variable θ, it is possible to plug back the parameter θ "into" $F(X)$, the function F having in common applications a known behavior at infinity.

2.2.2 Unconstrained Adaptive Importance Sampling Algorithm Applied to the VaR-CVaR Procedure

Applied to the problem we are dealing with, the main idea is to twist (by translation) the distribution of X in order to minimize the asymptotic variance of the two components in the CLT (8): the asymptotic variances of the VaR_α and CVaR_α algorithm

$$\frac{\mathrm{Var}\left(\mathbf{1}_{\varphi(X)\geq\xi^*}\right)}{f_{\varphi(X)}(\xi^*)} = \frac{\alpha(1-\alpha)}{f_{\varphi(X)}(\xi^*)} \quad \text{and} \quad \frac{\mathrm{Var}\left((\varphi(X)-\xi^*)_+\right)}{(1-\alpha)^2}.$$

By importance sampling, it is not possible to modify the quantity $f_{\varphi(X)}(\xi^*)$ since it is an intrinsic constant which appears in the CLT (8) through the Jacobian matrix of h, where $h(\xi, C) := \mathbb{E}[H(\xi, C, X)]$ and $H(\xi, C, X) := (H_1(\xi, C, X), H_2(\xi, C, X))$. Consequently, we are led to find the parameters θ^* and μ^* minimizing the two functionals:

$$Q_1(\theta, \xi^*) := \mathbb{E}\left[\mathbf{1}_{\{\varphi(X)\geq\xi^*\}}\frac{p(X)}{p(X-\theta)}\right], \quad Q_2(\mu, \xi^*) := \mathbb{E}\left[\left(\varphi(X)-\xi^*\right)_+^2\frac{p(X)}{p(X-\mu)}\right] \tag{17}$$

under the conditions that for every $(\xi, \theta) \in \mathbb{R}\times\mathbb{R}^d$,

$$\mathbb{E}\left[\left(1+(\varphi(X)-\xi)_+^2\right)\frac{p(X)}{p(X-\theta)}\right] < +\infty \tag{18}$$

and

$$\forall \xi \in \arg\min \; V, \;\; \mathbb{P}(\varphi(X) > \xi) > 0. \tag{19}$$

By differentiation we easily prove that

$$\nabla_\theta Q_1(\theta, \xi^*) = \mathbb{E}\left[\mathbf{1}_{\varphi(X)\geq\xi^*}\frac{p(X)}{p^2(X-\theta)}\nabla p(X-\theta)\right], \tag{20}$$

$$\nabla_\mu Q_2(\mu, \xi^*) = \mathbb{E}\left[\left(\varphi(X)-\xi^*\right)_+^2\frac{p(X)}{p^2(X-\mu)}\nabla p(X-\mu)\right]. \tag{21}$$

The key idea is to introduce a third change of probability in order to control the annoying terms $p(X)/p(X-\theta)$, $p(X)/p(X-\mu)$ by plugging back the parameters θ and μ into $\mathbf{1}_{\{\varphi(X)\geq\xi^*\}}$ and $\varphi(X)$ respectively. Now we follow [20] to design a regular unconstrained Robbins-Monro algorithm which converges $a.s.$ to the optimal parameters θ^* and μ^* (without risk of explosion) provided the growth of $x \mapsto \varphi(x)$ at infinity can be explicitly controlled. From now on, we assume that there exist two positive constants $a,\ C\ > 0$ such that

$$\forall x \in \mathbb{R}^d, \;\; |\varphi(x)|^2 \leq C\, e^{a\|x\|}. \tag{22}$$

We make the following assumption on the probability density p of X

$$\exists b \in [1,2] \text{ such that } \begin{cases} (i) & \frac{\|\nabla p(x)\|}{p(x)} = O(||x||^{b-1}) \quad \text{as} \quad \|x\| \to \infty \\ (ii) & \exists\, \rho > 0 \text{ such that } \log(p(x)) + \rho\,\|x\|^b \;\; \text{is convex.} \end{cases} \tag{23}$$

Then, one shows that under the conditions (18), (19), (22), (23), Q_1 and Q_2 are both finite and differentiable on $\mathbb{R}^d$ with gradients given by

$$\nabla Q_1(\theta,\xi^*) := \mathbb{E}\left[\mathbf{1}_{\varphi(X-\theta)\geq\xi^*}\underbrace{\frac{p^2(X-\theta)}{p(X)p(X-2\theta)}\frac{\nabla p(X-2\theta)}{p(X-2\theta)}}_{W_1(\theta,X)}\right], \tag{24}$$

$$\nabla Q_2(\mu,\xi^*) := \mathbb{E}\left[\left(\varphi(X-\mu)-\xi^*\right)_+^2\underbrace{\frac{p^2(X-\mu)}{p(X)p(X-2\mu)}\frac{\nabla p(X-2\mu)}{p(X-2\mu)}}_{W_2(\mu,X)}\right]. \tag{25}$$

We refer to [20] for a proof. The two last expressions may look complicated at first glance but, in fact, the weight term of the expectation involving the probability density can be easily controlled by a deterministic function of θ. For instance, when $X \overset{d}{=} \mathcal{N}(0;1)$,

$$W_1(\theta,X) = e^{\theta^2}(2\theta - X)$$

and more generally, under conditions (18) and (23), there exist two constants A and B such that

$$|W_1(\theta,X)| \leq e^{2\rho|\theta|^b}\left(A\,\|x\|^{b-1} + A\,\|\theta\|^{b-1} + B\right) \tag{26}$$

so that this weight can always be controlled by a deterministic function of θ (for more details, one can refer to [20]). Then by setting,

$$\widetilde{W}_1(\theta,X) := e^{-2\rho|\theta|^b}W_1(\theta,X), \quad \widetilde{W}_2(\mu,X) := e^{-2\rho|\theta|^b - a(\|\mu\|^2+1)}W_2(\mu,X),$$

we can define K_1 and K_2 by

$$K_1(\xi^*,\theta,x) := \mathbf{1}_{\varphi(X-\theta)\geq\xi^*}\widetilde{W}_1(\theta,x), \quad K_2(\xi^*,\mu,x) := \left(\varphi(X-\mu)-\xi^*\right)_+^2\widetilde{W}_2(\mu,x)$$

so that they satisfy the linear growth assumptions (16) and

$$\left\{\theta\in\mathbb{R}^d \mid \mathbb{E}\left[K_1(\xi^*,\theta,X)\right]=0\right\} = \left\{\theta\in\mathbb{R}^d \mid \nabla Q_1(\theta,\xi^*)=0\right\},$$
$$\left\{\mu\in\mathbb{R}^d \mid \mathbb{E}\left[K_2(\xi^*,\mu,X)\right]=0\right\} = \left\{\mu\in\mathbb{R}^d \mid \nabla Q_2(\mu,\xi^*)=0\right\}.$$

Now, since we do not know either ξ^* and C^* (the VaR_α and the CVaR_α) respectively we make the whole procedure adaptive by replacing at step n, these unknown parameters by their running approximation at step $n-1$. This finally justifies to introduce the following global procedure. One defines the state variable $Z_n := (\xi_n, C_n, \theta_n, \mu_n)$ where ξ_n, C_n denote the VaR_α and the CVaR_α approximations, θ_n, μ_n denote the variance reducers for the VaR and the CVaR procedures. We update this state variable recursively by

$$Z_n = Z_{n-1} - \gamma_n L(Z_{n-1}, X_n), \quad n\geq 1 \tag{27}$$

where $(X_n)_{n\geq 1}$ is an i.i.d. sequence with the same distribution as X, and L is defined as follow $L(z,x) = (L_1(\xi,\theta,x),\ L_2(\xi,C,\mu,x),\ K_1(\xi,\theta,x),\ K_2(\xi,\mu,x))$ with

$$L_1(\xi,\theta,x) := e^{-\rho\|\theta\|^b}\left(1-\frac{1}{1-\alpha}\,\mathbf{1}_{\{\varphi(x+\theta)\geq\xi\}}\frac{p(x+\theta)}{p(x)}\right) \tag{28}$$

$$L_2(\xi,C,\mu,x) := C-\left(\xi+\frac{1}{1-\alpha}(\varphi(x+\mu)-\xi)_+\frac{p(x+\mu)}{p(x)}\right). \tag{29}$$

Theorem 1. *(a) Convergence. Suppose* $\mathbb{E}\left[\varphi(X)^2\right] < +\infty$. *Assume that conditions* (18), (19), (22), (23) *are fulfilled and that the step sequence* $(\gamma_n)_{n\geq 1}$ *satisfies* (2). *Then,*

$$Z_n \xrightarrow{a.s.} z^* := (\xi^*, C^*, \theta_\alpha^*, \mu_\alpha^*)$$

where $(Z_n)_{n\geq 0}$ *is the recursive sequence defined by* (27), *where* $C^* = CVaR_\alpha$ *and*

$$(\xi^*,\theta_\alpha^*,\mu_\alpha^*) \in \mathcal{T}^* := \Big\{(\xi,\theta,\mu)\in\mathbb{R}\times\mathbb{R}^d\times\mathbb{R}^d :\ \xi\in\arg\min V,\ \nabla Q_1(\theta,\xi)=\nabla Q_2(\mu,\xi)=0\Big\}.$$

(b) Ruppert and Polyak CLT. Suppose that the assumptions of (a) hold and that the density $f_{\varphi(X)}$ *of* $\varphi(X)$ *is positive on its support, differentiable, and that* (7) *hold. Let* $(\overline{\xi}_n,\overline{C}_n)_{n\geq 1}$ *be the sequence defined by:*

$$\overline{\xi}_n := \frac{\xi_0+\ldots+\xi_{n-1}}{n},\quad \overline{C}_n := \frac{C_0+\ldots+C_{n-1}}{n}. \tag{30}$$

This sequence satisfies the following CLT:

$$\sqrt{n}\begin{pmatrix}\overline{\xi}_n-\xi^*\\ \overline{C}_n-C^*\end{pmatrix} \xrightarrow{\mathcal{L}} \mathcal{N}(0,\Sigma^*) \tag{31}$$

where the elements of Σ^* *are given by*

$$\Sigma^*_{1,1} = \frac{1}{f_{\varphi(X)}(\xi^*)}\ \mathrm{Var}\left(\mathbf{1}_{\{\varphi(X+\theta_\alpha^*)\geq\xi^*\}}\frac{p(X+\theta_\alpha^*)}{p(X)}\right),$$

$$\Sigma^*_{1,2} = \Sigma^*_{2,1} = \frac{\mathrm{Cov}\left((\varphi(X+\mu_\alpha^*)-\xi^*)_+\frac{p(X+\mu_\alpha^*)}{p(X)}, \mathbf{1}_{\{\varphi(X+\theta_\alpha^*)\geq\xi^*\}}\frac{p(X+\theta_\alpha^*)}{p(X)}\right)}{(1-\alpha)\,f_{\varphi(X)}(\xi^*)},$$

$$\Sigma^*_{2,2} = \mathrm{Var}\left((\varphi(X+\mu_\alpha^*)-\xi^*)_+\frac{p(X+\mu_\alpha^*)}{p(X)}\right).$$

Proof. We give a summary of the arguments. (a) First one shows that the sequence (ξ_n,θ_n,μ_n) converges $a.s.$ using the classical Robbins-Monro Theorem (see e.g. [9] or [12]). Then, the $a.s.$ convergence of $(C_n)_{n\geq 1}$ follows. (b) One applies the Ruppert and Polyak's Averaging Principle following a version established in [9] (p.169). □

Remark 2. There exists a CLT for the sequence $(Z_n)_{n\geq 1}$ and for its empirical mean $(\overline{Z}_n)_{n\geq 1}$ thanks to the Ruppert and Polyak averaging principle. We only stated the result for the components of interest (the ones which converge to VaR and CVaR).

Now, let us point out an important issue. The IS procedure raises an important problem that can be noticed if we consider (27) and the definition of the two functions K_1 and K_2 in (29). It is due to the fact that basically, we are dealing with rare events to update the VaR and the IS procedures. Somehow, we have two RM procedures $(\xi_n)_{n\geq 1}$ and $(\theta_n, \mu_n)_{n\geq 1}$ that are in competitive conditions, i.e., on one hand, we added an IS procedure to $(\xi_n)_{n\geq 1}$ to improve the convergence toward ξ^*, and on the other hand, the adjustment of the parameters (θ_n, μ_n) are based on samples X_{n+1} satisfying $\varphi(X_{n+1}-\theta_n) > \xi_n$ and $\varphi(X_{n+1}-\mu_n) > \xi_n$ which tend to become rare events. Somehow, we "postponed" the problem resulting from rare events on the IS procedure itself which may "freeze". To circumvent this problem, we are led to slightly modify the IS procedure.

2.2.3 How to Control the Move to the Critical Risk Area

In order to control the growth of θ_n and μ_n at the beginning of the algorithm, since we have no idea on how to twist the distribution of $\varphi(X)$, we can move slowly toward the critical risk area at level α in which $\varphi(X)$ exceeds ξ by replacing α by a deterministic non decreasing sequence α_n that converges to α in (27) and (29). This kind of incremental threshold increase has been proposed previously in [25]. By doing so, we only modify the VaR computation procedure ξ_n. Our aim is to devise an artificial VaR *companion procedure* which will be dedicated to the optimization of the IS parameters and not to the computation of VaR-CVaR. This VaR algorithm will move slowly to the tail distribution of $\varphi(X)$ and hence will drive the IS parameters. In practice, we decide to plug a deterministic stepwise constant sequence, i.e., we set $\alpha_n = 50\%$ for the first 3000-5000 first iterations then we set $\alpha_n = 80\%$ for 3000-5000 first iterations and finally set $\alpha_n = \alpha$ when the sequence (θ_n, μ_n) has almost converged. Numerically speaking, this kind of stepwise growth leads to a stable IS procedure. The function L_1 in (27) now depends of the current step n of the procedure, namely $L_{1,n}(\xi_{n-1}, X_n) = 1 - \frac{1}{1-\alpha_n}\,\mathbf{1}_{\{\varphi(X_n)\geq \xi_{n-1}\}}$ and the VaR_α algorithm ξ_n becomes

$$\hat{\xi}_n = \hat{\xi}_{n-1} - \gamma_n L_{1,n}(\hat{\xi}_{n-1}, X_n), \quad n \geq 1, \quad \hat{\xi}_0 \in L^1(\mathbb{P}). \tag{32}$$

If we replace ξ_n by $\hat{\xi}_n$ into the procedure devised in (27), we obtain a new IS algorithm defined by for $n \geq 1$

$$\begin{cases} \hat{\theta}_n = \hat{\theta}_{n-1} - \gamma_n L_3\left(\hat{\xi}_{n-1}, \hat{\theta}_{n-1}, X_n\right), \quad \theta_0 \in \mathbb{R}^d, \\ \hat{\mu}_n = \hat{\mu}_{n-1} - \gamma_n L_4\left(\hat{\xi}_{n-1}, \hat{\mu}_{n-1}, X_n\right), \quad \mu_0 \in \mathbb{R}^d. \end{cases} \tag{33}$$

To establish the convergence of this new procedure, we rely on the Robbins-Monro Theorem with remainder sequence (see e.g. [12]). Finally, we use (27) to estimate the couple $(\xi^*,\ C^*)$ in which $(\theta_n,\ \mu_n)$ is replaced by $(\hat{\theta}_n,\ \hat{\mu}_n)$. One shows that this new sequence $(\xi_n,\ C_n,\ \hat{\theta}_n,\ \hat{\mu}_n)$ satisfies the same CLT.

Remark 3. When dealing with loss distribution depending on a path-dependent SDE X, one can also replace the IS based on mean translation in a finite dimensional setting by its equivalent based on a Girsanov transformation ([12] and [20]).

2.3 Quasi-Stochastic Approximation

It is rather natural to plug quasi-random numbers instead of pseudo-random numbers in a recursive stochastic approximation. In this framework, the loss distribution $\varphi(X)$ is replaced by $\Psi(U)$ where $U \stackrel{d}{=} U([0,1]^q)$ and $\varphi(X) \stackrel{d}{=} \Psi(U)$. Such an idea goes back to [19] in a one dimensional setting. We denote by F the loss distribution function. We make the following assumption on Ψ,

$$\Psi \text{ is continuous and } \forall\, u \in \mathbb{R},\ \mathbb{P}(\Psi(U) = u) = 0. \tag{34}$$

Let $x := (x_n)_{n\geq 1}$ be a uniformly distributed sequence over $[0,1]^q$ with low discrepancy, i.e., the star discrepancy of the first n terms (see [21]) is $O\left(n^{-1}(\log(n)^q)\right)$. From a theoretical point of view, the convergence of (ξ_n, C_n) can be derived from the weak convergence (see e.g. [5]) of $F_n^{\Psi} := \frac{1}{n}\sum_{k=1}^n \delta_{\Psi(x_k)}$ to F. For any subset A of $\mathbb{R}$, δ_x denotes the *unit mass at* x defined by $\delta_x(A) = \mathbf{1}_A(x)$. We write $D_n^*(x,\Psi)$ for the star discrepancy of the first n terms of x associated to the system and defined naturally by

$$D_n^*(x,\Psi) := \sup_{u\in\mathbb{R}} \left| \frac{1}{n}\sum_{k=1}^n \mathbf{1}_{\Psi(x_k)\leq u} - F(u) \right|.$$

Suppose that (34) is satisfied. Suppose moreover that Ψ is Lipschitz and that the probability density function f of $\Psi(U)$ is bounded. Using theoretical results about Jordan discrepancy (see [21], page 17), one can show that

$$l_n(x) := \max_{1\leq k\leq n} k D_k^*(x,\Psi) = o\left(n^{1-\frac{1}{q}+\epsilon}\right),\quad \forall\, \epsilon > 0.$$

Theorem 2. *Let* $H_1 : \mathbb{R}\times[0,1]^q \to \mathbb{R}$ *and* $H_2 : \mathbb{R}^2\times[0,1]^q \to \mathbb{R}$ *defined by:*

$$H_1(\xi,x) := 1 - \frac{1}{1-\alpha}\mathbf{1}_{\Psi(x)\geq\xi} \quad \text{and } H_2(\xi,C,x) := C - \left(\xi + \frac{1}{1-\alpha}(\Psi(x)-\xi)_+\right).$$

Suppose that condition (34) *is satisfied. Suppose that* Ψ *is Lipschitz and* f *is bounded. Let* $\gamma := (\gamma_n)_{n\geq 1}$ *be a non-increasing deterministic sequence of gain parameters satisfying*

$$\sum \gamma_n = +\infty,\ \ \gamma_n l_n \to 0 \ \text{ and } \ \sum_{n\geq 1} \max\left(\gamma_n - \gamma_{n+1}, \gamma_n^2\right) l_n < +\infty. \tag{35}$$

Then, the recursive procedure defined for every $n \geq 1$ *by*

$$\begin{cases} \xi_n = \xi_{n-1} - \gamma_n H_1(\xi_{n-1}, x_n), \\ C_n = C_{n-1} - \gamma_n H_2(\xi_{n-1}, C_{n-1}, x_n), \end{cases} \tag{36}$$

satisfies:

$$\xi_n \to \xi^* \quad \text{and} \quad C_n \to C^* \quad \text{as } n \to \infty.$$

Proof. We give a summary of the arguments. Let L be the continuously differentiable Lyapunov function defined for every $x \in \mathbb{R}$ by $L(x) = \sqrt{1+(x-\xi^*)^2}$. A Taylor expansion of L at ξ_n leads to

$$L(\xi_{n+1}) \leq L(\xi_n) - \gamma_{n+1} L'(\xi_n) H_1(\xi^*, x) + C\gamma_{n+1}^2$$

for some positive constant C. Using (35) and successive Abel transforms, one deduces the convergence of $L(\xi_n)$ and then the convergence of ξ_n toward ξ^*. Then, the convergence of C_n can be deduced easily. □

Although, we do not have the rate of convergence of (36), in [19] some *a priori* error bounds (for some specific RM algorithms) show that using low-discrepancy sequences instead of pseudo-random numbers may significantly accelerate the convergence.

3 Numerical Illustrations

The assets are modeled as geometric Brownian Motions. We assume an annual risk free interest rate of $r = 5\%$ and volatility $\sigma = 20\%$.

Example 1. We consider a portfolio composed of a short position in a power plant that produces electricity from gas, day by day with a maturity of $T = 1$ month and 30 long positions in calls on electricity day-ahead prices all with the same strike $K = 60$. Electricity and gas initial spot prices are $S_0^e = 40$ \$/MWh and $S_0^g = 3$ \$/MMBtu (British Thermal Unit) with a Heat Rate equals $h_R = 10$ Btu/kWh and generation costs $C = 5$\$/MWh. The loss can be written

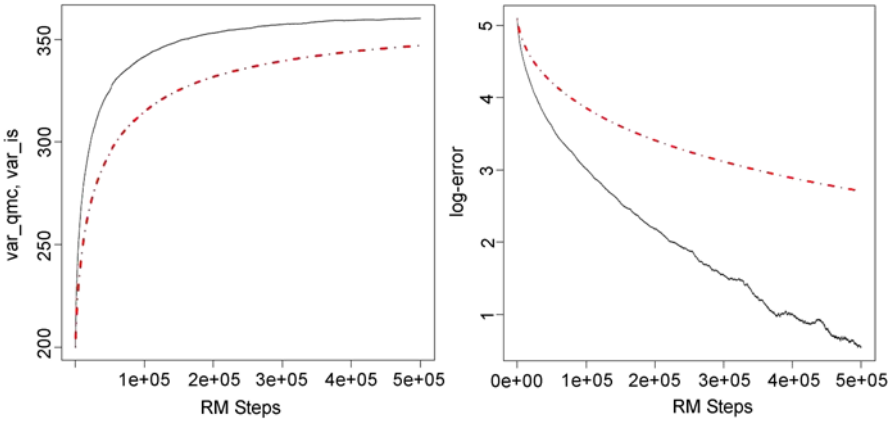
$$\sum_{k=1}^{30} e^{r(T-t_k)} \left(\left(S_{t_k}^e - h_R S_{t_k}^g - C \right)_+ - \left(S_{t_k}^e - K \right)_+ \right) + e^{rT}(C_0 - P_0^c)$$

where $P_0^c \approx 149.9$ is an estimate of the price of the option on the power plant (obtained by Monte Carlo using 500 000 samples) and C_0 is the price of the call options which is equal to 3.8. We use four different values of the confidence level $\alpha = 95\%,\ 99\%,\ 99.5\%$, and 99.9% for this example. In the RM procedure (27), we set the step sequence $\gamma_n = \frac{1}{n^p+100}$, where $p = 3/4$. Choosing the stepsize sequence is an important challenge for the analyst who wishes to apply this method. The numerical results are reported in Table 1. The first column denotes the number of steps used in the Robbins-Monro procedure in (27). The columns VaR and CVaR corresponds to the estimations of the VaR_α and the CVaR_α using (30). The columns VR_{VaR} and VR_{CVaR} denote variance reduction ratios estimations for both VaR and CVaR procedures. Variance Ratios (VR) corresponds to an estimate of the asymptotic variance using (6) divided by an estimate of the asymptotic variance using (27).

Example 2. We consider a portfolio composed of short positions in 10 calls and 10 puts on each of 5 underlying assets, all options having the same maturity of 0.25 years. The strikes are set to 130 for calls, to 110 for puts and the initial spot prices to 120. The underlying assets are assumed to be uncorrelated. We only consider the confidence level $\alpha = 95\%$. We compare the performance of *quasi-stochastic approximation* using a Sobol sequence (see e.g. [21]) and stochastic approximation for the computation of the VaR_α with IS using pseudo-random numbers. The dimension d of the structural vector X is equal to 5. Note that Ψ is not continuous in this example. In the RM procedure, we merely set $\gamma_n = \frac{1}{n+100}$ in (36). The numerical results are summarized in the figure below. The graph on the left depicts the VaR estimations using the procedure (27) with pseudo-random numbers (dotted line) and using (36) (normal line) for several number of RM steps. A good approximation of VaR_α is 362. The graph on the right depicts the log-distance of the VaR estimates to this approximation using pseudo-random numbers (dotted line) and using a Sobol sequence (normal line).

Table 1 Example 1. VaR, CVaR using IS procedure. VR_{VaR}, VR_{CVaR}: variance reduction ratios.

Number of steps	α	VaR	CVaR	VR_{VaR}	VR_{CVaR}
10 000	95 %	115.7	150.5	3.4	6.8
	99 %	169.4	196.0	8.4	12.9
	99.5%	186.3	213.2	13.5	20.3
	99.9%	190.2	219.3	15.3	32.1
100 000	95 %	118.7	150.5	4.5	8.7
	99 %	169.4	195.4	12.6	17.5
	99.5%	188.8	212.9	15.6	29.5
	99.9%	197.4	217.4	21.3	35.5
500 000	95 %	119.2	150.4	5.0	9.2
	99 %	169.8	195.7	13.1	18.6
	99.5%	188.7	212.8	17.0	29.0
	99.9%	198.8	216.8	24.8	46.8



References

1. Arouna, B.: *Adaptive Monte Carlo method, a variance reduction technique*. Monte Carlo Methods and Applications, **10**(1), 1–24 (2004)
2. Artzner, P., Delbaen, F., : Eber, J-M., Heath, D.: *Coherent measures of risk*. Mathematical Finance, **9**(3), 203–228 (1999)
3. Bardou, O., Frikha, N., Pagès, G.: Computation of VaR and CVaR using Stochastic Approximations and Adaptive Importance Sampling. Preprint, LPMA-1263 (2008)
4. Benveniste, A., Metivier, M., Priouret, P.: Adaptative Algorithms and Stochastic Approximations. Springer, Berlin 1990
5. Billingsley, P.: Convergence of Probability Measures. Wiley-Interscience, 1999, p. 296
6. Britten-Jones, M., Schaefer, S.M.: *Non linear Value-at-Risk*. European Finance Review, **5**, 161–187 (1999)
7. Duffie, D., Pan, J.: *An Overview of Value-at-Risk*. Journal of Derivatives, **4**, 7–49 (1997)
8. Duflo, M.: Iterative Random Models. Springer-Verlag, 1997
9. Duflo, M.: Algorithmes Stochastiques. Springer, Berlin, 1996
10. Egloff, D., Leippold, M.: Quantile estimation with adaptive importance sampling, Electronic copy: http://ssrn.com/abstract=1002631, (2007)
11. Dufresne, D., Vázquez-Abad, F.J.: Accelerated simulation for pricing Asian options, Proceedings of the 1998 Winter Simulation Conference, 1493–1500 (1998)
12. Frikha, N., Phd Thesis, In progress
13. Fu, M.C., Su, Y.: *Optimal importance sampling in securities pricing*. Journal of Computational Finance, **5**(4), 27–50 (2000)
14. Glasserman, P., Heidelberger, P., Shahabuddin, P.: Portfolio Value-at-Risk with Heavy-Tailed Risk Factors, Mathematical Finance, **12**, 239–270 (2002)
15. Glasserman, P., Heidelberger, P., Shahabuddin, P.: Variance Reduction Techniques for Estimating Value-at-Risk, SIAM Journal on Numerical Analysis, **47**, 1349–1364 (2000)
16. Glynn, P.W., Iglehart, D.L.: *Importance sampling for stochastic simulations*. Management Science, **35**, 1367–1389 (1989)
17. Juditsky, A.B., Polyak, B.T.: Acceleration of stochastic approximation by averaging, SIAM Journal on Control and Optimization, **30**(4), 838–855 (1992)
18. Kawai, R.: *Optimal importance sampling parameter search for Lévy Processes via stochastic approximation*. SIAM Journal on Numerical Analysis, **47**(1), 293–307 (2008)
19. Lapeyre, B., Pagès, G., Sab, K.: Sequences with Low Discrepancy. Generalization and application to Robbins-Monro algorithm. Statistics, **21**(2) 251–272 (1990)
20. Lemaire, V., Pagès, G.: Unconstrained Recursive Importance Sampling. Preprint LPMA-1231, To appear in Annals of Applied Probability (2008)
21. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, **63**, SIAM, Philadelphia, PA, 1992
22. Pflug, G.Ch.: Some remarks on the Value-at-Risk and the Conditional Value-at-Risk. Kluwer Academic Publishers, Dordrecht, 2000
23. Rockafellar, R.T., Uryasev, S.: *Optimization of Conditional Value-At-Risk*. The Journal of Risk, **2**(3), 21–41 (2000)
24. Rouvinez, C.: *Going Greek with VaR*. Risk, **10**(2), 57–65 (1997)
25. Rubinstein, R.Y.: Simulation-based optimization with rare events. European Journal of Operations Reasearch, 89–112 (1997)
26. Ruppert, D.: Stochastic Approximation. B.K. Ghosh and P.K. Sen, eds, New York, 1991
27. Serfling, R.J.: Approximation Theorems for Mathematical Statistics, Wiley, New York, 1980

Adaptive Monte Carlo Algorithms Applied to Heterogeneous Transport Problems

Katherine Bhan, Rong Kong, and Jerome Spanier

Abstract We apply three generations of geometrically convergent adaptive Monte Carlo algorithms to solve a model transport problem with severe heterogeneities in energy. In the first generation algorithms an arbitrarily precise solution of the transport equation is sought pointwise. In the second generation algorithms the solution is represented more economically as a vector of regionwise averages over a fixed uniform phase space decomposition. The economy of this representation provides geometric reduction in error to a precision limited by the granularity of the imposed phase space decomposition. With the third generation algorithms we address the question of how the second generation uniform phase space subdivision should be refined in order to achieve additional geometric learning. A refinement strategy is proposed based on an information density function that combines information from the transport equation and its dual.

1 Introduction

The radiative transport equation (RTE) is used to model a wide assortment of phenomena, including the operation of nuclear reactors [5], the use of lasers to analyze normal and diseased states in tissue and organs [11], the remote identification of oil and gas deposits [20] and the detection of land mines [17], [18]. The RTE de-

Katherine Bhan, Jerome Spanier
Beckman Laser Institute
University of California, Irvine
Irvine, California
url: http://www.bli.uci.edu

Rong Kong, Jerome Spanier
Claremont Graduate University,
Claremont, California
url: http://www.cgu.edu

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_12,

scribes the movement of the radiation (in the form of neutrons, photons, electrons, positrons) through the physical system (nuclear reactor, animal or human tissue, geological formations) and their interactions with the atomic nuclei, molecules, atoms and cells of the environment according to basic laws that characterize these interactions.

The general RTE can be written in integral form as

$$\Psi(\mathbf{P}) = \int_{\Gamma} K(\mathbf{P}, \mathbf{P}')\Psi(\mathbf{P}')d\mathbf{P}' + S(\mathbf{P}) \tag{1}$$

where the source term, S, describes the distribution of initial interactions, or collisions, throughout the physical phase space Γ and the kernel K describes how particles are absorbed, scattered and transported from state $\mathbf{P}'$ to state $\mathbf{P}$ of the phase space Γ. The space, Γ, consists of vectors $\mathbf{P}$ that describe the location, kinetic energy or velocity of the particle, and the unit direction of its motion through the system. The RTE solution $\Psi(\mathbf{P})$ describes the density of radiation at each state $\mathbf{P}$ of Γ.

Monte Carlo (MC) is preferred as the most accurate, and, for many realistic problems, *the only* method of solving the RTE. However, the convergence rate of conventional MC (CMC) is slow: as dictated by the central limit theorem, CMC variance decreases at a rate proportional to $W^{-\frac{1}{2}}$, where W is the number of independent samples. In the context of solving (1) with MC, the samples are random walks initiated by the source of radiation (nuclear fission, fiber-optic laser sources), transported to their initial collision states $\mathbf{P_0}$ described by the function $S(\mathbf{P})$ and thereby moved from state $\mathbf{P_i}$ to state $\mathbf{P_{i+1}}$ as described by the transport kernel $K(\mathbf{P_{i+1}}, \mathbf{P_i})$. The (infinite-dimensional) sample space of all such random walks is constructed as described in detail in [22].

The economist Milton Friedman and the statistician Samuel S. Wilks are usually credited with developing the statistical sampling procedure known as sequential sampling during the 1940's. Their classified work at Columbia University was unified and described in the seminal book [23]. Similar ideas were developed for accelerating the convergence of Monte Carlo solutions of matrix problems by John Halton in [7], [8] and [10]. Halton also discusses the application of these ideas to the solution of nonlinear systems in [9]. Interest in extending Halton's methods to continuous transport problems at Los Alamos National Laboratory and at Claremont Graduate University led to publications [3, 4, 14, 16] in which new, geometrically convergent algorithms were pioneered to overcome the slow convergence of CMC for RTE solutions. These first generation (G1) algorithms achieved this by representing the RTE solution as an infinite series of basis functions and estimating a finite number of the expansion coefficients. Using a sequential strategy, these coefficients are estimated in successively improving batches or stages, for which the information collected in the previous stages was used to guide the random walks in the next stage. When this is done properly, the statistical error after stage s is reduced (with probability 1) through multiplication by a factor $\lambda < 1$ [15]. This geometric convergence can then produce exponential decreases in statistical error, and thus exponential increases in computational efficiency, provided that the cost of simulation in each adaptive stage can be controlled.

Most realistic problems modeled with the RTE are heterogeneous; that is, the material properties vary greatly across the physical system. These variations are captured by the coefficients of the RTE, or cross sections[1], that, together with the scattering phase function that describes directional changes, characterize the kernel K in equation (1). For instance, when neutrons move about in a nuclear reactor, or when photons of light propagate through human tissue, both scattering and absorption cross sections are frequently modelled as discontinuous, regionwise constant functions of the particle's position in space. Another classic example relevant to nuclear reactor calculations is the occurrence of a severe heterogeneity in the energy variable as seen, e.g., by the rapid changes of the total cross section of certain isotopes, such as iron, over a narrow energy interval. Developing geometrically convergent algorithms for transport problems that are described by cross sections with steep gradients and/or discontinuities is particularly important, as solving such problems with high accuracy using CMC is impractical.

An essential first step in evaluating the potential of any adaptive algorithm for achieving geometric convergence for such challenging real problems is to test them on model transport problems. It is desirable that such model problems represent *realistic* heterogeneities present in each independent variable of the RTE, such as space, angle and energy. In [12] we introduced the new algorithms and applied them to a simple one dimensional homogeneous problem modeling interactions of light with tissue. In [15] we established rigorously the geometric convergence of the G1 method for quite general transport problems. In this paper we address heterogeneity in the energy variable.

In Section 2 of this paper we describe the model transport problems and their governing equations drawn from [4] on which our algorithms were tested and mention briefly the deterministic solutions used to validate our new MC algorithms. In Section 3 we outline the first generation (G1) adaptive methods that are based on global expansions of the solution in Legendre polynomials and report the difficulties encountered with the G1 solution for the problems studied here. In Section 4 we discuss the rationale for the development of the second generation (G2) adaptive algorithms introduced in [12] that are based on histogram fits to the solution for a given phase space decomposition. We comment on advantages of the G2 versus the G1 strategy and observe that the G2 solution converges rapidly to regionwise averages of the RTE solution. Following the G1/G2/G3 algorithmic framework introduced in [12], in Section 5 we describe how the geometric learning achieved by the G2 algorithm can be extended by intelligent refinement of the phase space decomposition. For the G3 adaptive algorithm we propose a strategy for such intelligent mesh refinement based on simulations of both forward- and backward-moving random walks. In Section 6 we summarize our conclusions from the study and outline a continuing research plan for future work.

[1] Cross sections are defined to be the probabilities of interaction per unit distance travelled. For details see discussion in Section 2.

2 The Model Problem and Equations

The problems described in [4] model energy-dependent neutron transport that is coupled with energy-independent photon transport in one spatial dimension. In an operating nuclear reactor, neutrons are created as a by-product of nuclear fission at high energies and they undergo random walks throughout the reactor as they collide with the atomic nuclei present. As they collide, they lose energy. Eventually, a fraction of the high energy (fast) neutrons survive to much lower energies (less than 1 eV) where they are in thermal equilibrium with their environment. Such "slow" neutrons may then interact with atoms of fissile material to create new high energy neutrons for the next "generation". Occasionally, photons are produced as a by-product of other neutron interaction events and the photons then are transported throughout the reactor until they are also absorbed or leave the physical system. Attention may focus in any such experiment on portions of the distribution of neutrons, in energy, location and direction, that affect the operation of the system, or that of the neutrons or photons at positions distant from their origin, as these may then pose radiation hazards to the immediate environment.

The transport of either neutrons or photons is described by fundamental quantities called cross sections, denoted σ, that provide the coefficients for the RTE. These cross sections - so-named because at many energies they are comparable to the physical cross section of the atomic nucleus with which they interact - depend on both the kinetic energy of the particle and its position, but not its direction. The cross sections serve to quantify the various probabilities of interaction of the random-walking neutrons or photons with the various nuclei. For our purposes, the cross sections characterize the probability density functions used to carry out the Monte Carlo simulation. Thus, if $\sigma(E)$ is the (total) neutron cross section (assumed here for simplicity to be independent of position in the system being modeled), the distance s between successive collisions for neutrons is sampled from the probability density function

$$T(s, E) = \sigma(E) e^{-\sigma(E)s},\ 0 < s \leq \infty. \tag{2}$$

The cross sections $\sigma(E)$ are experimentally determined, often very complicated functions of the energy (and, in general, the position) of the particle as it travels through the medium. For example, the neutron cross section for iron drops by a factor of 100 in the energy range [0.025 MeV, 0.027 MeV] (1 MeV = 1 million electron volts, which is roughly the lower extremity of the energy released in nuclear fission). According to the formula (2), when the cross section falls by two orders of magnitude, the distances between successive collisions are greatly magnified. This means that in a problem in which attention is focused on radiation far from the source, nearly all of the penetrating particles must have scattered into this narrow energy range at some time during their random walks. So the twin characteristics of a very challenging reactor problem, but nevertheless one that arises in practice, are both a rapidly changing cross section and a focus on deeply penetrating particles. These are the characteristics that Booth embodied in the model problems he studied in [4], which we describe next.

Booth assumes that neutrons originate with energy 1 MeV at the origin of a semi-infinite line $0 \leq x < \infty$ (see Fig. 1) and move only in the direction of increasing x. While this problem greatly simplifies the representation of physical reality, even the motion along a line serves to illustrate the real complexity to be found in fully three dimensional reactor systems. For example, a more realistic forward and backward (and even sideways, as in three dimensions) motion of neutrons would delay contributions from photons that reach great distances from the radiation source, but the same effect can be achieved in Booth's model problem by allowing x to reach very large values since it is not restricted at all. At each collision, the neutron may be absorbed with constant probability p_a or it may be scattered forward with probability $p_s = 1 - p_a$. These probabilities will, in a general transport problem, depend on position but are treated as constant here in order to focus on the unique challenge posed by the severe energy heterogeneity for $\sigma(E)$. If a neutron scattering collision occurs at energy E', a new energy E is sampled from

$$p(E' \to E) = \begin{cases} \frac{1}{E'}, & 0 \leq E \leq E' \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

which is an assumption often adopted in the fast neutron energy range. The neutron's position x thus steadily increases and its energy E steadily decreases until it is absorbed. When the neutron is absorbed, a photon is assumed to be emitted in the positive x direction. The cross section for photons is assumed (for simplicity) constant, γ, independent of position and energy and Booth further assumes that every photon is absorbed on its first collision. The problem identified in [4] is to estimate the number of photons reaching a point $x = z$ arbitrarily distant from the origin. In the simulation itself, the code does not actually transport the photons when each neutron is absorbed at a position x, but rather calculates the expected contribution, $\exp[-\gamma(z-x)]$, to the number of photons reaching z from a neutron absorption at x.

We point out that our adaptive methods are based upon the use of correlated sampling, while Booth's adaptive methods are based on the use of importance sampling. Thus, the importance function, $J(\mathbf{P})$, is introduced in [4]. This function satisfies a transport equation that is dual to the RTE (1):

$$J(\mathbf{P}) = \int_\Gamma K^*(\mathbf{P}, \mathbf{P}') J(\mathbf{P}') d\mathbf{P}' + S^*(\mathbf{P}), \tag{4}$$

where $K^*(\mathbf{P}, \mathbf{P}') = K(\mathbf{P}', \mathbf{P})$ and $S^*(\mathbf{P})$ is a detector function. The importance function may be interpreted as the expected value of the estimating random variable, or tally, from a particle originating at $\mathbf{P}$. From the duality between equations (1) and (4) it follows easily (see also [22]) that reciprocity

$$\int_\Gamma \Psi(\mathbf{P}) S^*(\mathbf{P}) d\mathbf{P} = \int_\Gamma J(\mathbf{P}) S(\mathbf{P}) d\mathbf{P} \tag{5}$$

is satisfied for the solutions of the original and the dual RTE. Therefore, estimating by Monte Carlo the linear functional $\int_\Gamma \Psi(\mathbf{P}) S^*(\mathbf{P}) d\mathbf{P}$ of the RTE solution $\Psi(\mathbf{P})$

may also be accomplished by estimating the linear functional $\int_\Gamma J(\mathbf{P})S(\mathbf{P})d\mathbf{P}$ of the dual RTE solution, $J(\mathbf{P})$.

For the coupled neutron/photon problem described above and illustrated in Fig. 1, Booth arrives at the importance function equation

$$J(x,E) = \int_0^{z-x} T(s,E) \left\{ p_a \exp(-\gamma[z-(x+s)]) + \int_0^E p_s\, p(E \to E')J(x+s,E')dE' \right\} ds \tag{6}$$

that expresses the expected photon tally at z from a neutron at position $x < z$ with energy E. To derive this equation, Booth states that for a neutron at position x with

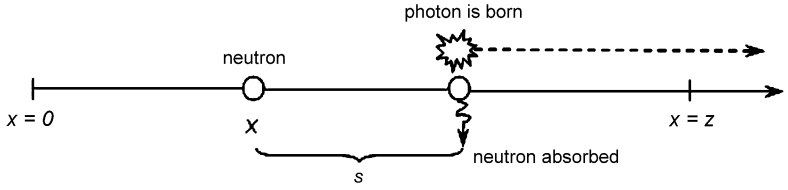


Fig. 1 Schematic of the model problem physics.

energy E, the factor $T(s,E)$ is the probability of colliding after moving a distance s beyond x. On the right hand side of equation (6), this factor is multiplied by the sum of the probability that this collision will result in the neutron's absorption at $x+s$ together with the expected (photon) tally from all other events produced by a neutron scattering event at $x+s$ (and a reduction in energy there from E to E'). Booth then argues that the function $J(0,E)$, whose estimates we seek, factors into a function $J(E)$ of energy alone with an exponential scaling factor:

$$J(0,E) = \exp(-\gamma z)J(E).$$

He then arrives at the equation for the importance function:

$$J(E) = S(E) + \int_0^1 K(E,E')J(E')dE', \tag{7}$$

where the source is

$$S(E) = p_a \frac{\sigma(E)}{\sigma(E)-\gamma} \tag{8}$$

and the kernel is given by

$$K(E,E') = p_s \frac{\sigma(E)}{\sigma(E)-\gamma} p^*(E' \to E), \tag{9}$$

where

$$p^*(E' \to E) = p(E \to E') = \begin{cases} \frac{1}{E}, & 0 \le E' < E \le 1 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Notice that the condition $\gamma < \sigma(E)$ is required in order to guarantee that the source $S(E)$ is positive. The constant γ, which is the cross section for photons, may be adjusted to vary the difficulty of the problem. Small values of γ produce near-certainty that each neutron absorbed will generate a photon that reaches large distances z from the origin, while values of γ close to $\sigma(E)$ produce much shorter photon flights, thereby reducing photon tallies at large z.

As an importance function, $J(E)$ is the solution to the transport equation that is dual, or adjoint to the one that describes the physics of the problem. This can be appreciated by comparing equations (10) and (3): scattering in the adjoint equation (10) from E' to E is exactly the reverse of scattering in equation (3) that describes the physics. Thus, while neutrons lose energy when scattered, the scattering events in the importance equation result in energy increases. It is the integral transport equation (7) that provides the starting point for our research.

Booth identified in [4] two test problems, Problem 1 and Problem 2, to portray, respectively, steep gradients and discontinuities in the energy total cross section, $\sigma(E)$. Following [4], for Problem 1 we model a cross section "notch" by subtracting a Gaussian distribution from a unit cross section. The formula employed is

$$\sigma(E) = 1 - 0.9\exp[-u^2(E)/2]$$

where $u(E) = \frac{E-\mu}{\sqrt{v}}$ and $\mu = 0.4$ and $\sqrt{v} = 0.1$. Such a notch models the cross section behavior typical of that seen near so-called "resonance" energies. At a resonance energy E, nuclides such as iron appear to present greatly magnified "targets" for a nuclear interaction with a neutron of energy E, while at an energy only slightly removed in either direction, the apparent target size is reduced by factors of 100 or more compared with the physical size of the nucleus.

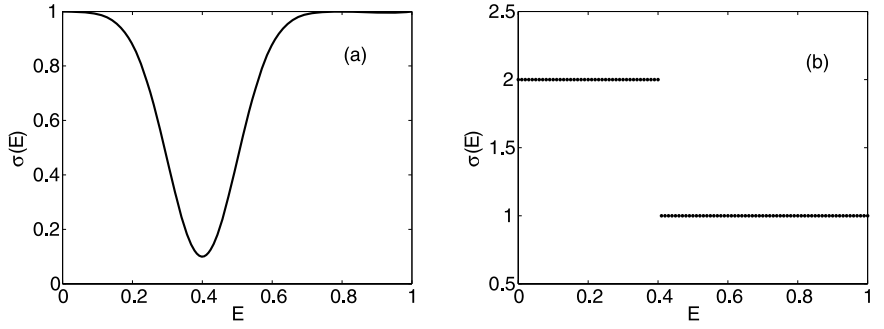


Fig. 2 Cross section $\sigma(E)$ dependence on E: (a) rapidly varying for Problem 1 and (b) discontinuous for Problem 2.

In the second test problem, Problem 2, the cross section is modeled as

$$\sigma(E) = 1 + H(0.4 - E)$$

where $H(\cdot)$ is the Heaviside function. Plots of $\sigma(E)$ are shown in Fig. 2. While such Problem 2 step functions are non-physical in terms of energy treatment, they are commonplace in modeling the spatial behavior of particle cross sections, where the abrupt changes model physically distinct reactor materials (e.g., water and iron) that are immediately adjacent to each other in the reactor assembly. Here, and in Booth's work, the interest is in observing whether such actual discontinuities pose special challenges for the adaptive algorithms or not.

From equation (2) it follows that the average distance between successive collisions for neutrons at energy E is $\frac{1}{\sigma(E)}$. Also, for a photon to be detected, the neutron that generated it must travel very far from $x = 0$; hence, such a neutron must have acquired one or more energies for which $\sigma(E)$ is small. We observe that since the energy in equation (7) can only increase because of (10), the value of J at some $E = E_{test}$ is affected by the behavior of $J(E)$ for *all* values of $E \leq E_{test}$. That is why we have selected several different values of test points in energy, E_{test}, to test the accuracy of our algorithms, with $E_{test} = 1$ presenting the greatest computational challenge.

We implemented two deterministic methods with which to compare our MC solution:

A. a recursion method that generates the first $N + 1$ terms in the Neumann series for $J(E)$;

B. numerical integration based on an adaptive Simpson's rule strategy [19] of an ordinary differential equation equivalent to (7) (the number of integration grid points is set to N).

Both of these methods produced a reasonably good approximation to $J(E)$ and served as a useful reference in debugging our initial Monte Carlo code. For Problem 2 integration of the equivalent ordinary differential equation yielded a solution in closed form for which $J(1) = 1.175656679$. However, evaluation of the deterministic solutions for Problem 1 had to be carried out numerically, hence their accuracy depends on the number of iterations, N.

In Table 1 we provide the estimates of $J(1)$ for Problem 1 produced by each deterministic method. As one method converges from above and the other one from below, we can only conclude that $1.7107 < J(1) < 1.7260$. Ultimately we decided that our G3 algorithm produced the most accurate estimate, 1.17142, of $J(1)$ for Problem 1 (See Section 5).

Table 1 Deterministic estimates of $J(1)$. N = number of iterations.

N	A	B
10^2	1.9849	1.6394
10^3	1.8094	1.6874
10^4	1.7478	1.7046
10^5	1.7260	1.7107

3 G1 Solution

Generation 1 adaptive algorithms strive for knowledge of the RTE solution pointwise with arbitrary precision everywhere. The solution is represented by a truncated series in each independent variable of the RTE in terms of orthogonal basis functions, such as Legendre polynomials. The expansion coefficients are estimated by Monte Carlo in batches, or stages. At stage 0 random walks are sampled as in conventional Monte Carlo in order to obtain the initial estimates of the coefficients. That is, referring to equation (1), the position, energy and direction at an initial collision state $\mathbf{P}$ inside the phase space Γ is generated by sampling from the source function $S(\mathbf{P})$, and the kernel K is then used to model transitions from the initial collision state $\mathbf{P_0}$ to its next one, $\mathbf{P_1}$, and to all subsequent states $\mathbf{P_k}$ until the particle is either absorbed or leaves the physical system of interest.

In each subsequent adaptive stage the sampling of each random walk is altered according to a learning mechanism that enables the variance in the estimates to reduce geometrically. When the sequential use of correlated sampling is employed, for example, such a learning mechanism is provided by a "reduced" source. In sequential correlated sampling, transitions from state to state continue to be modeled by using the RTE kernel K, but instead of the physical source $S(\mathbf{P})$, a reduced source is used to generate initial collision states. The reduced source for the adaptive stage $k+1$ in terms of the reduced source at stage k is given by

$$S^{(k+1)}(\mathbf{P}) \equiv S^{(k)}(\mathbf{P}) + \int_{\Gamma} K(\mathbf{P},\mathbf{P}')\tilde{\psi}^{(k)}(\mathbf{P}')d\mathbf{P}' - \tilde{\psi}^{(k)}(\mathbf{P}), \quad S^{(0)}(\mathbf{P}) = S(\mathbf{P}) \quad (11)$$

where $\tilde{\psi}^{(k)}(\mathbf{P})$ is the correction obtained in stage k to the truncated series representing the solution. By adding together corrections from each stage s, $\tilde{\psi}^{(s)}(\mathbf{P})$, the approximate RTE solution is reconstructed

$$\tilde{\Psi}^{(k)}(\mathbf{P}) = \tilde{\psi}^{(0)}(\mathbf{P}) + \tilde{\psi}^{(1)}(\mathbf{P}) + \cdots + \tilde{\psi}^{(k)}(\mathbf{P})$$

and the convergence of $\tilde{\psi}^{(k)}(\mathbf{P})$ to zero thus controls the convergence of the approximate RTE solution $\tilde{\Psi}^{(k)}(\mathbf{P})$ to the truncated infinite series expansion of the exact RTE solution, $\Psi_M(\mathbf{P})$, where M is the number of expansion coefficients used to represent the exact solution $\Psi(\mathbf{P})$. The reduced source is also called the residual, or equation error, associated with the solution expansion at stage k: it describes the extent to which the $k-$th stage solution *fails* to satisfy the original RTE integral

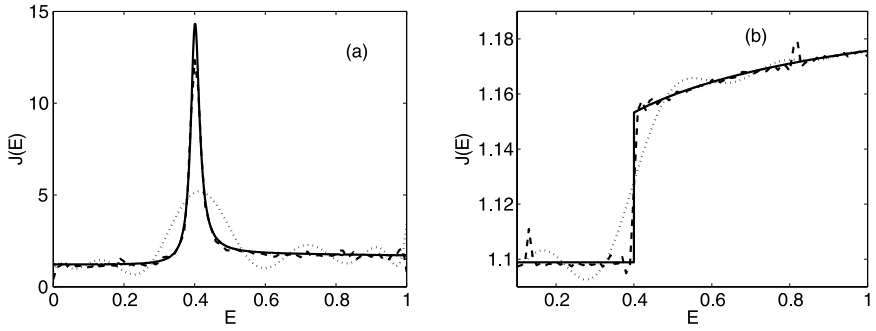


Fig. 3 Comparison of the 12-th and 100-th order Legendre polynomial G1 solution for $J(E)$ with: (a) more accurate G3 solution for Problem 1 and (b) analytical solution for Problem 2.

equation. Hence, the residual can be used to determine the corrections needed at each stage to improve the solution estimate. As the reduced source converges to zero, the series expansion converges to the transport equation solution.

Although, as shown in [13, 15], theoretically an unlimited precision is possible with G1 methods, we found that achieving decent accuracy in practice comes at a high computational cost even for the one dimensional model problems we studied. When solving Problem 1 with the G1 algorithm, for example, we had to use at least 200 coefficients in the Legendre series for $J(E)$ so that the series adequately captures the highly non-polynomial shape of the solution (see Fig. 3 (a)). Using this representation and 10^6 random walks in each of the 7 adaptive stages we obtained 4 decimal digits of accuracy in the estimate of the solution at its peak near $E = 0.4$ and it took about 8 hours to converge. We obtained similar results for Problem 2 (see Fig. 3 (b)). Each dotted and dashed line in Fig. 3 represents the 12- and 100-th order Legendre series G1 solution, respectively. The solid lines represent the most accurate solution for each problem.

Various shortcomings of the G1 solution, such as the appearance of polynomial artifacts, high computational cost and the need for a separate expansion in each independent variable of the RTE, preclude the current G1 strategy to be used to solve realistic transport problems in several variables efficiently. As alluded to in [12], unless the proper basis functions are derived based on knowledge of the spectral properties of the transport operator, no single orthogonal basis can adequately represent all RTE solutions. These considerations led us to the second generation algorithms described in the next section.

4 G2 Solution

The motivation behind the G2 method is that, in practice, one rarely needs to know the solution of the transport equation with perfect accuracy at every point of the phase space. Instead, MC estimates of the integrals of the solution weighted by

detector function(s) measuring certain reaction rates are required. For example, for non-invasive detection of tissue abnormalities using light, the measurements of light reflected from the tissue are taken by fiber optic detectors placed at the tissue surface. In order to solve such problems well, it is only neccessary to achieve accuracy that is consistent with the limits of precision obtained by the actual measuring devices. Thus, the perfection sought by the G1 algorithm is not neccessary.

Suppose, then, that interest focuses *only* on estimating one or more weighted integrals over the phase space Γ such as $\int_\Gamma \Psi(\mathbf{P})S^*(\mathbf{P})d\mathbf{P}$ of the solution Ψ with *high precision*, where $S^*(\mathbf{P})$ is a detector function. We can model the algorithm after the one described in Section 3 by replacing the global Legendre polynomial approximation by a local one defined over each subregion. For region i the G2 algorithm finds an approximation $\tilde{\Psi}_{a,i}$ of the average value of Ψ over Γ_i,

$$\Psi_{a,i}(\mathbf{P}) = \begin{cases} \frac{1}{vol(\Gamma_i)} \int_{\Gamma_i} \Psi(\mathbf{P}')d\mathbf{P}', & \text{if } \mathbf{P} \in \Gamma_i \\ 0, & \text{if } \mathbf{P} \notin \Gamma_i \end{cases} \tag{12}$$

Now let $\tilde{\Psi}^{(0)}_{a,i}(\mathbf{P})$ denote an initial estimate of $\Psi_{a,i}(\mathbf{P})$ obtained from a conventional Monte Carlo simulation. Replacement of the continuous function $\tilde{\Psi}$ by the piecewise constant function $\tilde{\Psi}^{(0)}_{a,i}$ in equation (11) and iteration produces an appropriate reduced source for the new G2 adaptive algorithm:

$$S^{(k+1)}(\mathbf{P}) \equiv S^{(k)}(\mathbf{P}) - \tilde{\psi}^{(k)}_a(\mathbf{P}) + \int_\Gamma K(\mathbf{P}, \mathbf{P}')\tilde{\psi}^{(k)}_a(\mathbf{P}')d\mathbf{P}', S^{(0)}(\mathbf{P}) = S(\mathbf{P}) \tag{13}$$

where the function $\tilde{\psi}^{(k)}_a$ is the correction from stage k to the approximate solution from previous stages. The G2 adaptive algorithm is described more fully in [2, 12].

We have implemented this algorithm and observe that it achieves *very rapid convergence* to estimates of the solution averages in only a few adaptive stages. The geometric learning ceases when the locally constant approximation $\tilde{\Psi}_a(P)$ of the transport solution Ψ has been stabilized. Because such an approximate solution, which is discontinuous, cannot satisfy the original RTE pointwise (except in the trivial case that the latter is globally constant), the precision achievable is limited by the variation of the solution over each subregion of the decomposition.

When this G2 algorithm was applied to the energy problem described in Section 2, we found that it can produce quite good quality maps of the solution of the J equation. Fig. 4 displays these maps for crude and fine energy meshes, consisting of, respectively, 20 and 100 subintervals in E. Solid lines represent the G3 solution. With 10^3 random walks per subinterval and 10 adaptive stages the run time corresponding to the crude mesh results was less than 1 sec, while for a fine mesh it took about 10 sec. In the next pair of plots we examine the error of the G2 solution carefully. In Fig. 5 we plot on a $\log_{10}$ scale the variance corresponding to the G2 solution with the fine mesh and two choices of the test point: (a) $E_{test} = 0.1$ and (b) $E_{test} = 0.4$. These plots reveal that the G2 algorithm achieves stable convergence for an "easy" $E_{test} = 0.1$, while for a "hard" point $E_{test} = 0.4$, although G2 converges

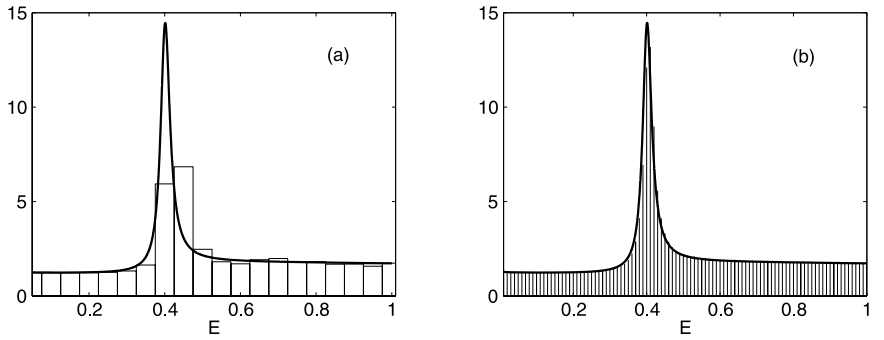


Fig. 4 G2 solution, J(E), for Problem 1 with: (a) crude phase space subdivision (20 subintervals) and (b) fine phase space subdivision (100 subintervals).

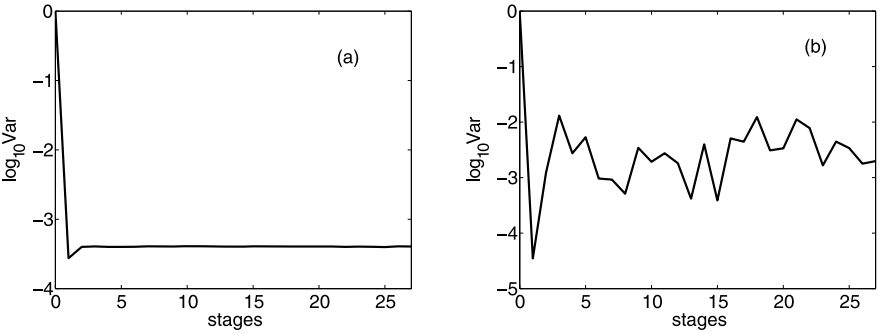


Fig. 5 Variance in G2 solution on a $\log_{10}$ scale, Problem 1: (a) $E_{test} = 0.1$ and (b) $E_{test} = 0.4$.

geometrically, it cannot achieve the same accuracy as for the "easy" test point. As expected, we observe similar results for all E beyond the peak value at $E = 0.4$.

The geometric learning power of this G2 algorithm alone should make possible the accurate solution of many RTE problems not currently accessible by conventional MC. However, we would like to be able to increase this accuracy when it is required. To do this, we need to be able to refine an initial decomposition of the phase space Γ in an intelligent way to achieve the accuracy needed. In other words, in case the precision reached when the G2 geometric learning stops is insufficient, we want to be able to extend it by an *appropriate* refinement of the phase space. What is needed is an *automated* strategy for determining which subregions are most important to refine, and by how much. Such a strategy is described in the following section.

5 G3 Solution

The mechanism we propose to exhibit how to refine any phase space decomposition intelligently is to combine information collected from particle trajectories con-

structed according to the original RTE with information collected from trajectories sampled according to an adjoint RTE. Let $\Psi(\mathbf{P})$ denote the RTE solution and $\Psi^*(\mathbf{P})$ denote the adjoint RTE solution. Because the product function $I(\mathbf{P}) = \Psi(\mathbf{P}) \cdot \Psi^*(\mathbf{P})$ combines the *intensity* of radiation at $\mathbf{P}$ with the likelihood that radiation at $\mathbf{P}$ *will actually reach* the detector, this function quantifies the data on which intelligent grid refinement should be based.

We call the $I(\mathbf{P})$ function the *information density function* (IDF) and it can be applied quite generally to RTE problems involving full spatial, angular, energy and time dependence. In [1, 6, 21, 24], this function is sometimes called a *response* or *contributon function*, and it was used in the early literature to study RTE problems, mainly of nuclear radiation shielding type − problems characterized by their focus on events with low probability outcomes in a simulation.

The detailed knowledge of the function I *pointwise* throughout the phase space poses a daunting problem, even more so than capturing the RTE solution Ψ everywhere since I obeys a complicated RTE that involves *both* Ψ and Ψ^*. However, we can quite easily estimate *integrals* of I over an arbitrary decomposition of Γ by combining information from two G2 algorithm applications, one to obtain an approximate Ψ solution and the other an approximate Ψ^* solution. These regionwise constant approximations, Ψ_a and Ψ_a^*, can then be multiplied together in each subregion and the resulting approximation I_a can be integrated easily to produce the required approximate integrals of I.

We apply these general ideas to the specific problems described in Section 2. Suppose then that the initial decomposition of the energy interval $[0, 1]$ consists of R subintervals: $[0, 1] = \cup_{i=1}^{R-1}[E_{i-1}, E_i) \cup [E_{R-1}, E_R]$ and denote, as before, the importance function by $J(E)$ and the solution of the equation dual to the equation (7) by $J^*(E)$. Suppose that we set the accurate estimation of $J(1)$ to be our goal. Since

$$J(1) = \int_0^1 J(E)\delta(E-1)dE \tag{14}$$

our detector function, $S^*(E) = \delta(E-1)$. This detector function is the source term in the equation dual to (7):

$$\begin{aligned} J^*(E) &= S^*(E) + \int_0^1 K^*(E, E')J^*(E')dE' \\ &= \delta(E-1) + \int_0^1 K(E', E)J^*(E')dE'. \end{aligned} \tag{15}$$

We remark here in passing that since $J(E)$ satisfies the adjoint RTE, $J^*(E)$, as its dual, satisfies the RTE that describes the physics in the coupled neutron-photon problem.

Let $\mathbf{J} = (J_1, J_2, ..., J_R)$ and $\mathbf{J}^* = (J_1^*, J_2^*, ..., J_R^*)$ denote the vector solutions obtained by applying the G2 algorithm to the transport equations for $J(E)$ and $J^*(E)$, respectively. Thus, J_n and J_n^* are approximations to the average values of, respectively, $\mathbf{J}$ and $\mathbf{J}^*$ over the $n-$th subinterval, $[E_{n-1}, E_n)$. The product $I_n = J_n J_n^*$ may

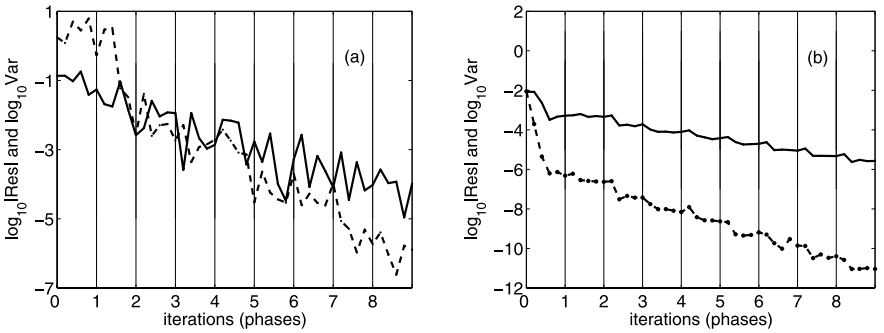


Fig. 6 Error in G3 solution on a $\log_{10}$ scale: (a) Problem 1 and (b) Problem 2.

then be integrated over $[E_{n-1}, E_n)$ and interpreted as an estimate of the total information value of that region.

Our mesh refinement algorithm based on this idea does the following:

1. Choose $I_{\min} = \min_{1\le n\le R}\{I_n\}$.
2. Determine the integer part of $I_n/I_{\min}$, $[I_n/I_{\min}]$.
3. If $[I_n/I_{\min}] = i_n$, partition the $n-$th subinterval into i_n equal subintervals.
4. Redefine $\mathbf{J}, \mathbf{J}^*$ over the new energy mesh and apply the G2 algorithms to find estimates of the averages of $J(E)$ and $J^*(E)$ over the refined energy mesh.
5. Repeat these steps until the values of I_n are approximately independent of n.

The convergence of the G3 method for Problem 1 and Problem 2 is indicated in Fig. 6. We plot, on a $\log_{10}$ scale, the absolute value of the residual (equation (13)) with a dotted line and the variance with a solid line.

For Problem 1 at stage 0 we start the algorithm with a crude mesh of 20 subintervals, 100 random walks per subinterval and run 5 stages of G2 on a fixed mesh. Having completed the G2 learning we proceed to the next phase. In each new phase the energy mesh is refined as described above. At the end of this run we have obtained an energy mesh consisting of highly non-uniform 12257 subintervals.

Table 2 Comparison of relative efficiencies, Problem 1.

Alg.	Stages	Est $J(1)$	|Res.|	Var	Time, sec.	Rel. Eff.
determ.	-	[1.7107; 1.7260]	-	-	-	-
CMC	1	1.6333	7.9×10^{-2}	4.0	1	1
G2	10	1.7134	8.1×10^{-4}	1.1×10^{-3}	33	109
G3	45	1.7142	1.1×10^{-5}	1.3×10^{-6}	20277	155

In Tables 2 and 3 we compare the performances of the G2, G3 algorithms for Problem 1 and Problem 2, respectively, and contrast it with conventional Monte Carlo (CMC). We use relative efficiency to assess the performance of each algorithm. If T_A is the time it takes for an algorithm to achieve a specified variance,

Table 3 Comparison of relative efficiencies, Problem 2.

Alg.	Stages	Est $J(1)$	|Res.|	Var	Time, sec.	Rel. Eff.
exact	-	1.175656	-	-	-	-
CMC	1	1.1781	2.4×10^{-2}	1.5×10^{-2}	2	1
G2	5	1.1755	2.4×10^{-5}	3.2×10^{-6}	11	856
G3	35	1.1756	3.0×10^{-6}	1.3×10^{-8}	186	12079

we can use the central limit theorem to extrapolate to the time it would take for the CMC, T_{CMC}, to achieve the same variance. The relative efficiency is defined to be the ratio of these times:

$$\text{Rel. Eff.} = \frac{T_{CMC}}{T_A}. \tag{16}$$

Traditionally, the figure of merit, F_M,

$$F_M = \frac{1}{\text{Var} \times \text{Run Time}}, \tag{17}$$

is used to compare different CMC algorithms since the decrease in variance (denoted Var) in non-adaptive MC is roughly inversly proportional to run time, so (17) is a measure of algorithm performance that is roughly independent of the number of samples. We use (16) in place of (17) since the variance in our new algorithms decreases at a rate that is faster than linear with the number of samples used.

6 Summary

We have seen that G2 algorithms are capable of providing both good histogram fits to RTE solutions and tally estimates with relatively low computational expenditure. It is conceivable that many transport problems of only average complexity can be analyzed with the help of the G2 method alone based on reasonably fine meshes. For those problems for which the G2 errors are still too large, we have presented a new G3 algorithm capable of extending the geometric convergence of G2, based on intelligently refined crude meshes. We have used IDF integrals over the mesh subdivisions to guide the mesh refinement strategy. The combined G2-G3 technique should significantly reduce the computational cost of many more transport problems than with G2 used alone.

For the problems described in [4] and studied here we have shown that gains over conventional costs by several orders of magnitude are achieved when these methods are applied. We have also learned that the presence of discontinuities in the cross section data, such as would appear in all realistic transport problems composed of materials of different physical composition, do not seem to cause any degradation in the quality of G3 learning. This may be the case because the G2-G3 method depends on the combined behavior in each region of *both* the solution *and* the adjoint solution

of the RTE, not just on the behavior of either alone. That is, the bad behavior of one function in some region might be largely negated by the good behavior of the other solution in the same region, and conversely. Our methods actually produce larger gains for Problem 2 than for Problem 1, perhaps because in Problem 2 the behavior is very tame for *both* solutions over most of the energy range.

We believe that the significance of the accuracy achieved with the G2 and G3 methods is that it is obtainable with relatively simple algorithms that incorporate features "tuned" to each specific RTE problem. Thus, while it is certainly to be expected that the number of subdivisions of the phase space will increase with the dimensionality of the phase space (which is six for the most general steady-state transport problems), this increase is *not* controlled by purely geometric factors in our algorithm design. That is, the growth is governed by the variations in the RTE and adjoint RTE solutions over the phase space, which does not, in general, lead to a product space decomposition of the phase space, as would be the case for multidimensional quadrature. The simple G3 refinement strategy described above that maintains a fixed number of random walks in each subregion can also be improved significantly, we believe, in order to control computational costs.

Our recommendation based on this study is that similar investigations are in order for transport problems that exhibit prototype computational challenges in other phase space variables and in higher dimensions. For example, angular variation is important in streaming problems and spatial heterogeneities are important in nearly all real transport problems. If the methods perform well in these problems, we believe it would be appropriate to add such capability to major production Monte Carlo codes such as MCNP [5] at LANL for general reactor problems and the Virtual Tissue System [11] being developed at UCI for tissue problems.

Acknowledgements This research has been partially supported by Laser Microbeam and Medical Program, LAMMP, NIH P-41-RR-O1192, University of California, Office of the President, UCOP-41730 and National Science Foundation grant NSF/DMS 0712853.

References

1. Aboughantous, C.H.: A Contributon Monte Carlo Method. Nucl. Sci. Eng. **108**(3), 160–177 (1994)
2. Bhan, K., Spanier, J.: Advanced Monte Carlo Methods for Exponential Convergence, UCOP 41730. Tech. rep., UCI (2008)
3. Booth, T.E.: Exponential Convergence on a Continuous Transport Problem. Nucl. Sci. Eng. **127**(3), 338–345 (1997)
4. Booth, T.E.: Adaptive Importance Sampling with Rapidly Varying Importance Function. Nucl. Sci. Eng. **136**(3), 399–408 (2000)
5. Breismeister, E.J.: MCNP4C, a Monte Carlo N-particle Transport Code. Tech. rep., LANL (1999)
6. Cramer, S.N.: Forward-Adjoint Monte Carlo Coupling with No Statistical Error Propagation. Nucl. Sci. Eng. **124**(3), 398–416 (1996)
7. Halton, J.: Sequential Monte Carlo. Proc. Comb. Phil. Soc. **58**, 57–58 (1962)

8. Halton, J.: Sequential Monte Carlo techniques for the solution of linear systems. J. Sci. Comp. **9**, 213–257 (1994)
9. Halton, J.: Sequential Monte Carlo techniques for solving non-linear systems. Monte Carlo Methods and Applications **12**, 113–141 (2006)
10. Halton, J.: Sequential Monte Carlo techniques for linear systems - a practical summary. Monte Carlo Methods and Applications **14**, 1–27 (2008)
11. Hayakawa, C.K., Spanier, J., Venugopalan, V.: Computational Engine for a Virtual Tissue Simulator. In: Monte Carlo and Quasi-Monte Carlo Methods 2006, A. Keller, S. Heinrich and H. Niederreiter (Eds.), pp. 431–444. Springer-Verlag (2007)
12. Kong, R., Ambrose, M., Spanier, J.: Efficient, automated Monte Carlo methods for radiation transport. J. Comp. Phys. **227**(22), 9463–9476 (2008)
13. Kong, R., Spanier, J.: Error analysis of sequential Monte Carlo methods for transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, H. Niederreiter and J. Spanier, (Eds.). Springer-Verlag (1999)
14. Kong, R., Spanier, J.: Sequential correlated sampling methods for some transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, H. Niederreiter and J. Spanier, (Eds.). Springer-Verlag (1999)
15. Kong, R., Spanier, J.: A new proof of geometric convergence for general transport problems based on sequential correlated sampling methods. J. Comp. Phys. **227**(23), 9762–9777 (2008)
16. Lai, Y., Spanier, J.: The adaptive importance sampling algorithm for particle transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, H. Niederreiter and J. Spanier, (Eds.). Springer-Verlag (1999)
17. Maučec, M., Meijer, de R.J.: Monte Carlo simulations as a feasibility tool for non-metallic land-mine detection by thermal neutron backscatteting. Appl. Rad. Isotopes **56**, 837–846 (2002)
18. Maučec, M., Rigollet, C.: Monte Carlo simulations to advance characterisation of landmines by pulsed fast/thermal neutron analysis. Appl. Rad. Isotopes **61**, 35–42 (2004)
19. McKeeman, W.M.: Algorithm 145: Adaptive numerical integration by Simpson's rule. Commun. ACM **5**(12), 604 (1962)
20. Mosher, S., Maučec, M., Spanier, J., Badruzzaman, A., Chedester, C., Evans, M.: Expected-Value Techniques for Monte Carlo Modeling of Nuclear Well-logging Problems. in preparation
21. Serov, I.V., John, T.M., Hoogenboom, J.E.: A new effective Monte Carlo midway coupling method in MCNP applied to a well logging problem. Appl. Radiat. Isot. **49**(12), 1737–1744 (1998)
22. Spanier, J., Gelbard, E.M.: Monte Carlo Principles and Neutron Transport Problems. Addison-Wesley Pub. Co. (1969)
23. Wald, A.: Sequential Analysis. John Wiley and Sons, New York (1947)
24. Williams, M.L.: Generalized Contributon Response Theory. Nucl. Sci. Eng. **108**, 355–383 (1991)

Efficient Simulation of Light-Tailed Sums: an Old-Folk Song Sung to a Faster New Tune...

Jose H. Blanchet, Kevin Leder, and Peter W. Glynn

Abstract We revisit a classical problem in rare-event simulation, namely, efficient estimation of the probability that the sample mean of n independent identically distributed light tailed (i.e. with finite moment generating function in a neighborhood of the origin) random variables lies in a sufficiently regular closed convex set that does not contain their mean. It is well known that the optimal exponential tilting (OET), although logarithmically efficient, is not strongly efficient (typically, the squared coefficient of variation of the estimator grows at rate $n^{1/2}$). After discussing some important differences between the optimal change of measure and OET (for instance, in the one dimensional case the size of the overshoot is bounded for the optimal importance sampler and of order $O(n^{1/2})$ for OET) that indicate why OET is not strongly efficient, we provide a state-dependent importance sampling that can be proved to be strongly efficient. Our procedure is obtained based on computing the optimal tilting at each step, which corresponds to the solution of the Isaacs equation studied recently by Dupuis and Wang [8].

Jose H. Blanchet
Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027
e-mail: jose.blanchet@columbia.edu

Kevin Leder
Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027
e-mail: kl2457@columbia.edu

Peter W. Glynn
Department of Management Science and Engineering, Stanford University, Stanford, California 94305-4026
e-mail: glynn@stanford.edu

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_13,

1 Introduction

Let $X, X_1, X_2, \ldots$ be a sequence of mean zero independent and identically distributed (iid) d-dimensional random variables (rv's). Assume that A is a sufficiently regular (see Section 5) convex set for which $0 \notin A$. We further assume that A satisfies a technical condition, which is detailed in Section 5. We revisit a fundamental problem in the theory of rare-event simulation, namely that of computing $\alpha_n = P(S_n/n \in A)$ for n large, where $S_n = X_1 + X_2 + \ldots + X_n$. In particular, we consider the setting in which X is light-tailed. The purpose of this paper is to provide the first simulation estimator for which it can be proven that the number of simulation runs needed to compute α_n to a given relative accuracy remains bounded as a function of the parameter n.

To fix ideas, consider the one dimensional case in which $A = (\beta, \infty)$ for $\beta > 0$. It is well known that the use of importance sampling, as implemented through optimal exponential tilting (OET), provides an estimator that is "logarithmically efficient" as $n \nearrow \infty$ (in the sense that the squared coefficient of variation grows subexponentially) [12]. Recall that OET involves an importance distribution in which each of the summands is independently sampled from that member of the natural exponential family having mean β, see e.g., [12, 13]. In fact, it can be further shown that OET provides the only iid importance sampling algorithm that achieves logarithmic efficiency [2]. This might not be surprising, given that asymptotically, as $n \to \infty$, OET agrees with the conditional distribution of the S_k's ($k < n$) given $\{S_n > n\beta\}$ (see Proposition 2 below). It is a simple calculation to check that the conditional distribution is in fact the ideal importance sampling change of measure since it creates an unbiased estimator with zero-variance. We will commonly refer to it as the zero-variance change of measure.

However, it turns out that the squared coefficient of variation [ratio of second moment of estimator to probability of interest squared, see Eq. (5)] associated with OET does increase as $n \nearrow \infty$, so that the number of samples required to compute α_n to a given relative accuracy increases as a function of n. In fact, in Section 2, we prove that under mild conditions, the squared coefficient of variation grows at rate $O\left(n^{1/2}\right)$. The reason that OET becomes less efficient with the growth of n has to do with the fact that OET fails to agree with the conditional distribution (zero-variance change of measure) at scales finer than that of the Law of Large Numbers (LLN). As one illustration of this phenomenon, Proposition 3 establishes that the "overshoot" $S_n - n\beta > 0$ is asymptotically exponentially distributed as $n \nearrow \infty$ under the conditional distribution. In particular the moments of the overshoot stay bounded as n grows. However, due to the Central Limit Theorem (CLT), the OET produces an overshoot of order $n^{1/2}$. In other words, the OET tends to bias the increments excessively when the random walk is relatively close to reaching the boundary β, thereby inducing a large overshoot.

An algorithm having the property that the sample size required to compute α_n to a given relative accuracy is bounded as a function of n is called a "strongly efficient" algorithm (equivalently estimator) [2, 10]. To produce a strongly efficient estimator, we use "optimal state-dependent exponential tilting" (OSDET). In the one dimen-

sional case, this corresponds to dynamically updating the OET at each step in the algorithm based on the current position S_k of the random walk $(S_k : k \geq 0)$. We apply OSDET up until we basically reach the boundary (at which point we turn-off importance sampling) or the distance to the target is sufficiently large relative to the remaining time horizon (at which point we simply apply OET computed from such position until the end of the time horizon). It certainly seems intuitively clear that OSDET will likely reduce the growth of the coefficient of variation as $n \nearrow \infty$ relative to simply applying OET, but there is no reason to expect that one would obtain a bounded coefficient of variation. What plays a crucial role is the fact that (under the assumption that the target set A has a twice continuously differentiable boundary) the dynamic tilting induced by OSDET induces a twice continuously differentiable mapping given by the large deviations rate function. An additional polynomial decay rate of order $O\left(n^{-1/2}\right)$ necessary to control the behavior of the squared coefficient of variation arises thanks to the fact that the conditional expected value of a second order term in the Taylor expansion of this mapping has exactly the right behavior to control (after combining the contributions of all time steps) the previous polynomial decay rate. This result is stated in Lemma 2 and used in Proposition 4.

Let us briefly connect our work with the game-theoretic approach introduced by Dupuis and Wang in [8]. It turns out that the OSDET corresponds to applying importance sampling according to the solution to the associated Isaacs equation (Section 3.4 of [8]). They prove that if such a solution is continuously differentiable then one has a logarithmically efficient estimator. In our setting the solution to the Isaacs equation we work with is in fact twice continuously differentiable in the interior. By applying OSDET in a region where the large deviations scaling is applicable (i.e. before we basically reach the target level or the distance to the level is very large relative to the remaining time horizon), we obtain strong efficiency. It is important to note, however, that our sampler uses a small layer at the boundary to avoid sampling at the point where the solution to the Isaacs equation fails to be twice continuously differentiable. Our work then suggests a connection between the degree of smoothness of the solution to the associated Isaacs equation and the efficiency strength of the corresponding importance sampling estimator.

In previous works on importance sampling and large deviations settings for sample means, proofs of strong efficiency have been limited to systems with heavy tailed characteristics [5] or Gaussian increments [4]. The contribution of our paper is to construct an importance sampling algorithm that can be used in a general class of light tailed distributions that is also provably strongly efficient.

Our proof of strong efficiency relies on the analysis of several martingales that arise naturally from the description of the algorithm, and is of independent interest. Given that OSDET achieves bounded relative error, it may not be surprising then that OSDET also induces a bounded overshoot as $n \nearrow \infty$.

The rest of the paper is organized as follows. Sections 2 to 4 concentrate on the one dimensional case. Section 2 describes explicitly our assumptions and collects some needed results from the theory of large deviations. Section 3 introduces the algorithm explicitly and provides a heuristic analysis behind its efficiency. The rig-

orous details are given in Section 4, where we also show that the overshoot under OSDET remains bounded as $n \nearrow \infty$. Section 5 treats the multidimensional case.

2 Large Deviations Results for Light Tailed Sums

In this section, we concentrate on the one dimensional case and present some auxiliary results from the theory of large deviations that will be useful for the description and analysis of our algorithm.

We start with listing the assumptions underlying our development in Sections 2 to 4.

i) $EX = 0$ and $\mathrm{Var}(X) = \sigma^2$

ii) The log-moment generating function $(\psi(\theta) : \theta \in \mathbb{R})$, defined as $\psi(\theta) = \log E \exp(\theta X)$, is assumed to be *steep* to the right in the sense that for each $w > 0$ there exists $\theta_w > 0$ such that $\psi'(\theta_w) = w$.

iii) We assume that $\inf_{\theta \geq 0} \psi''(\theta) > 0$.

iv) The random variable (rv) X is nonlattice (i.e. the characteristic function has modulus strictly less than one except at the origin).

Assumption i) is obviously introduced without loss of generality. The *steepness* assumption is standard in the large deviations literature and it is useful to rule out distributions with extremely light tails (in particular with compact support). Assumption iii) although satisfied by most models of practical interest beyond light-tailed random variables with finite support (in particular, the condition allows Gaussian, gamma random variables and mixtures thereof) is more technical and is applied only in Lemma 2. Nevertheless, such a condition certainly rules out tails that are lighter than Gaussian. The last assumption is again common in the development of exact asymptotics in large deviations, which are required in our setting because we are concerned with strong efficiency rather than logarithmic efficiency.

We are now ready to describe some results from large deviations that will be useful in our development. The so-called rate function plays a crucial role in the theory of large deviations. In our context, we work with a variant of the standard rate function, $J(\cdot)$, defined for $w \geq 0$ by

$$J(w) \triangleq \max_{\theta \geq 0} [\theta w - \psi(\theta)]. \tag{1}$$

The standard rate function is defined by optimizing $\theta \in (-\infty, \infty)$. Both $J(\cdot)$ and the standard rate function agree on the positive real line. In particular, note that we have for $w \geq 0$

$$J(w) \triangleq w\theta_w - \psi(\theta_w) \text{ and } J'(w) = \psi'^{-1}(w) = \theta_w. \tag{2}$$

For $w < 0$ we have that $J(w) \equiv 0$ and that $J(\cdot)$ is continuously differentiable at zero. The algorithmic implication of defining $J(\cdot)$ in this way, as we shall see, is that no importance sampling is applied when one reaches the level above $n\beta$. Note, however, that $J(\cdot)$ is not twice continuously differentiable at zero.

Finally, the natural exponential family $(F_\theta : \theta \in \mathbb{R})$ generated by the distribution $F(\cdot) = P(X \leq \cdot)$ is defined via

$$dF_\theta = \exp(\theta x - \psi(\theta))\, dF. \tag{3}$$

The distribution F_θ is also said to be "exponentially tilted" by the parameter θ. Let $P_\theta(\cdot)$ be the product probability measure generated by F_θ (for $\theta \in \mathbb{R}$) under which the X_i's are iid and let $E_\theta(\cdot)$ be the corresponding expectation operator associated with $P_\theta(\cdot)$. We often use the notation $E_{J'(w)}(\cdot)$, which just means $E_{\theta_w}(\cdot)$.

We shall need the following elementary properties of the rate function.

Proposition 1. *If Assumption ii) is in force, then*

$$J(w) = \sigma^2 w^2/2 + O\left(w^3\right)$$

as $w \searrow 0$. *Moreover, for each* $w \in (0, \infty)$ *we have*

$$J(w+h) = J(w) + \theta_w h + O\left(h^2\right)$$

as $h \longrightarrow 0$ *(uniformly over* $w \in [\varepsilon, 1/\varepsilon]$ *for fixed* $\varepsilon > 0$*). In fact, for each* $w > 0$*, the function* $J(w+\cdot)$ *is infinitely differentiable at zero and its Taylor series converges in a neighborhood of the origin.*

Proof. First it is clear from the formula (2) that on the positive line $J(\cdot)$ inherits the smoothness properties of $\psi(\cdot)$, this gives the last two results of the above proposition. The first two results follow from a Taylor expansion of the function $J(\cdot)$ and therefore by necessity a Taylor expansion of θ_w which is obtained using the inverse function theorem. □

Large deviations theory is intended to both address the question of how to compute asymptotics for rare event probabilities and to describe the conditional behavior of the underlying system given the occurrence of the rare event. The following result is a celebrated large deviations asymptotic approximation due to Bahadur and Rao [3] that will be useful in our development.

Theorem 1. *Under assumptions i), ii) and iv) above,*

$$P(S_n > n\beta) = \frac{\exp(-nJ(\beta))}{\theta_\beta \sqrt{2\pi n \psi''\left(\theta_\beta\right)}} (1 + o(1))$$

as $n \nearrow \infty$ *for fixed* $\beta > 0$.

The following proposition provides an asymptotic description of the conditional behavior of the process $(S_k : 0 \leq k \leq n)$ given that $S_n > \beta n$ (as $n \nearrow \infty$) and provides rigorous support for the claim that the asymptotic conditional distribution of the increments given $\{S_n > n\beta\}$ is $P_{\theta_\beta}(\cdot)$, for a proof see, e.g., [6].

Proposition 2. *Suppose that i), ii) and iv) are in force. Then, for any positive integers* $k_1 < k_2 < ... < k_m < \infty$ *and for each* $x_{k_1}, ..., x_{k_m}$, *continuity points of* $F(\cdot)$,

$$P\left(X_{k_1} \leq x_{k_1}, ..., X_{k_m} \leq x_{k_m} \middle| S_n > n\beta\right) \longrightarrow F_{\theta_\beta}\left(x_{k_1}\right) ... F_{\theta_\beta}\left(x_{k_m}\right)$$

as $n \nearrow \infty$.

While the above result describes the behavior of a typical increment under the conditioning, the proposition below provides an asymptotic description of the limiting overshoot $S_n - n\beta > 0$.

Proposition 3. *Assume that i), ii) and iv) hold and put* $\beta > 0$. *Then, for all* $x > 0$

$$\lim_{n\to\infty} P\left(S_n - n\beta > x \middle| S_n > n\beta\right) = \exp\left(-\theta_\beta x\right)$$

Proof. Note that from Theorem 1 we have that

$$\lim_{n\to\infty} \frac{P(S_n > n\beta + x)}{P(S_n > n\beta)} = \lim_{n\to\infty} P(S_n > n\beta + x)\, e^{nJ(\beta)} \theta_\beta \sqrt{2\pi n \psi''(\theta_\beta)}.$$

Thus it remains to show that

$$\lim_{n\to\infty} P(S_n > n\beta + x)\, e^{nJ(\beta)} \theta_\beta \sqrt{2\pi n \psi''(\theta_\beta)} = e^{-\theta_\beta x}.$$

Following the notation of [7], Theorem 3.7.4, define the following

$$Y_i = \frac{X_i - \beta}{\sqrt{\psi''(\theta_\beta)}},\ \psi_n = \theta_\beta \sqrt{n\psi''(\theta_\beta)},\ W_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} Y_i \text{ and } F_n(y) = P(W_n \leq y).$$

Then a simple calculation (see [7], page 111, for details) gives

$$P(S_n > n\beta + x)\, e^{nJ(\beta)} \theta_\beta \sqrt{2\pi n \psi''(\theta_\beta)} = \psi_n \sqrt{2\pi} \int_{\theta_\beta x/\psi_n}^{\infty} e^{-\psi_n y} dF_n(y).$$

One can now follow nearly the same procedure as in the proof of Theorem 3.7.4 of [7], the only difference being the lower limit of integration. Due to the fact that our lower limit of integration is $\theta_\beta x/\psi_n$ the limit of the previous display is $e^{-\theta_\beta x}$, as desired. □

The previous result implies that $P_{\theta_\beta}(\cdot)$ does not accurately describe the behavior of the random walk, conditioned on $\{S_n > n\beta\}$, at time n. In particular under $P_{\theta_\beta}(\cdot)$, the CLT implies that $n^{-1/2}(S_n - n\beta) \overset{D}{\approx} N\left(0, \psi''\left(\theta_\beta\right)\right)$ and thus the overshoot is $(S_n - n\beta) = O\left(n^{1/2}\right)$ in distribution. On the other hand, Proposition 3 indicates that the conditional overshoot is of order $O(1)$ (in distribution). Therefore, $P_{\theta_\beta}(\cdot)$ may provide a poor description of the conditional distribution of the random

walk at scales that are finer than linear (for instance at scales of order $n^{1/2}$). As a consequence, it is not surprising that the performance of $P_{\theta_\beta}(\cdot)$ as an importance sampling distribution degrades when measured at a fine enough scale. In particular, the estimator induced by $P_{\theta_\beta}(\cdot)$, namely

$$L = \exp\left(-\theta_\beta S_n + n\psi\left(\theta_\beta\right)\right) I\left(S_n > \beta n\right), \tag{4}$$

is not strongly efficient (i.e., the squared coefficient of variation of the estimator is unbounded as $n \nearrow \infty$). More precisely, it follows that if $cv_n(L)$ denotes the coefficient of variation of L, then by definition

$$\left(cv_n(L)\right)^2 \triangleq \frac{\mathrm{Var}_{\theta_\beta}(L)}{\left(E_{\theta_\beta}(L)\right)^2} = \frac{E_{\theta_\beta}\left(L^2\right)}{P\left(S_n > n\beta\right)^2} - 1. \tag{5}$$

Under Assumptions i), ii) and iv) we have for a positive constant C_β

$$\begin{aligned} E_{\theta_\beta}\left(L^2\right) &= E_{\theta_\beta}\left[e^{2n\psi(\theta_\beta)-2\theta_\beta S_n} I\left(S_n > n\beta\right)\right] = E\left[e^{n\psi(\theta_\beta)-\theta_\beta S_n} I\left(S_n > n\beta\right)\right] \\ &= e^{-nJ(\beta)} P(S_n > n\beta) E\left\{\exp\left[-\theta_\beta(S_n - n\beta)\right] | S_n > n\beta\right\} \\ &\sim \frac{C_\beta}{\sqrt{n}} e^{-2nJ(\beta)} E\left\{\exp\left[-\theta_\beta(S_n - n\beta)\right] | S_n > n\beta\right\} = \frac{C_\beta}{2\sqrt{n}} e^{-2nJ(\beta)}. \end{aligned}$$

where we use Theorem 1 for the approximation, and Proposition 3 for the final equality. Using Theorem 1 once more we see that as $n \to \infty$, $(cv_n(L))^2 = O\left(n^{1/2}\right)$.

In our next section, we examine the form of the optimal change of measure and propose an importance sampling distribution that improves upon P_{θ_β} by achieving a bounded squared coefficient of variation.

3 A Proposed Algorithm and Intuitive Analysis

The basic idea of our algorithm is that at each discrete time step (provided the random walk is inside a compact set to be described later) one recomputes the OET change of measure. There are two stopping criteria that must be introduced and that we shall discuss in more detail.

The algorithm is explicitly defined as follows. The constant λ below can be chosen arbitrarily as long as $\lambda > 2\beta$, (this is required in the proof of Proposition 5 below).

Algorithm 1

Set $w = \beta > 1/n^{1/2}$, $L = 1$, $s = 0$, $\bar{s} = 0$, $k = 0$, and $\lambda > 2\beta$.

Repeat STEP 1 until $n = k$ OR $w \leq 1/(n-k)^{1/2}$ OR $w > \lambda$.

STEP 1: Sample X from F_{θ_w} [defined by equations (2) and (3)] and set

$$L \longleftarrow \exp\left(-\theta_w X + \psi\left(\theta_w\right)\right) L,$$

$$s \longleftarrow s + X,$$
$$k \longleftarrow k + 1,$$
$$w \longleftarrow (n\beta - s)/(n-k).$$

STEP 2: If $k < n$ sample $X_{k+1}, ..., X_n$ iid rv's from F_{θ_w} and set

$$\bar{s} \longleftarrow X_{k+1} + ... + X_n,$$
$$L \longleftarrow \exp\left(-\theta_w \bar{s} + (n-k)\,\psi\left(\theta_w\right)\right) L.$$

STEP 3: Output $Y_n = L \times I\left(s + \bar{s} > n\beta\right)$

The intuition behind the stopping conditions indicated in STEP 2, namely $w \leq 1/(n-k)^{1/2}$ or $w > \lambda$, is the following. First, when $w \leq 1/(n-k)^{1/2}$ there is no need for applying importance sampling sequentially until the end as the event of interest is not rare any more (we have reached the Central Limit Theorem region). It seems intuitive that if one replaces $w \leq 1/(n-k)^{1/2}$ simply by $w \leq 0$ (i.e. stop if we reach the boundary) then one still should obtain strong efficiency. Our analysis, however, requires a stopping criterion that is slightly removed from the origin, such as the one that we impose here. This criterion is used in the proof of Lemma 2 below and basically is imposed to deal with the fact that $J(\cdot)$ is not twice continuously differentiable at the origin. Now, whenever we have that $w > \lambda$, for some large constant λ, then we are approaching a scaling for which the large deviations asymptotics (which motivate the design of the algorithm) are no longer applicable (i.e. a situation where the distance to the target is no longer linearly related to the time to go). At that point, we simply apply the tilting once and for all up until the end of the time horizon.

The estimator Y_n obtained from the algorithm above can be expressed as follows. First, define

$$W_j = (n\beta - S_j)/(n-j) \tag{6}$$

for $0 \leq j \leq n-1$, and $W_n \triangleq 0$. Next define the following stopping times $\tau_1^{(n)} = \inf\{0 \leq k < n : W_k > \lambda\}$, $\tau_0^{(n)} = \inf\{k \geq 0 : n\beta - S_k \leq (n-k)^{-1/2}\}$, and

$$\tau^{(n)} = \tau_0^{(n)} \wedge \tau_1^{(n)} \wedge n. \tag{7}$$

We define (allowing ourselves a slight abuse of notation) the change of measure used at step j to be

$$\theta_j \triangleq \theta_{W_j} = J'\left(W_j\right). \tag{8}$$

Let us write

$$Z_{1,n} = \exp\left(-\sum_{j=0}^{\tau^{(n)}-1} \left(\theta_j X_{j+1} - \psi\left(\theta_j\right)\right)\right),$$

$$Z_{2,n} = \exp\left(-\theta_{\tau^{(n)}}\left(S_n - S_{\tau^{(n)}}\right) + (n-\tau^{(n)})\psi\left(\theta_{\tau^{(n)}}\right)\right).$$

We can now define the OSDET (optimal state-dependent exponential tilting) estimator resulting from Algorithm 1 as

$$Y_n = Z_{1,n} Z_{2,n} I\left(S_n > n\beta\right) \tag{9}$$

where X_j follows the distribution F_{θ_j} for $1 \le j \le \tau^{(n)}$ and X_j is sampled according to the distribution $F_{\theta_{\tau^{(n)}}}$ for $\tau^{(n)} + 1 \le j \le n$.

The next section is devoted to the rigorous efficiency analysis of Y_n. However, before we provide the full details behind such analysis we will spend the rest of this section explaining the main intuitive steps. It turns out that the most important contribution comes from term $Z_{1,n}$, so this object will be the focus of our discussion here. A substantial portion of the technical development in the next section is dedicated to showing that for any $p \ge 1$ $\sup_{n\ge 1} \widetilde{E}\left((n-\tau^{(n)})^p\right) < \infty$, where $\widetilde{E}(\cdot)$ is used throughout the rest of the paper to denote the expectation operator induced by the importance sampling strategy described in Algorithm 1 (see Proposition 5 below). This in turn is used to argue that the sum in the exponent in $Z_{1,n}$ has basically $n-m$ terms (where m is a constant).

The next results allows us to express the exponent in $Z_{1,n}$ in terms of a telescopic sum involving the function $J(\cdot)$.

Lemma 1. *For $0 \le j \le n-2$ let $w_{j+1}(x) = (n\beta - s - x)/(n-j-1)$ and $w_j = (n\beta - s)/(n-j)$, if $(n-j)^{-1/2} < w_j < \lambda$ then*

$$\begin{aligned}
&(n-j-1)J\left(w_{j+1}(x)\right) - (n-j)J\left(w_j\right) \\
&= -J'\left(w_j\right)x + \psi\left(\theta_j\right) \\
&+ \frac{(x-w_j)^2}{(n-j-1)}\int_0^1\int_0^1 J''\left(w_j + vu(w_{j+1}(x) - w_j)\right)ududv.
\end{aligned}$$

In addition,

$$J''\left(w_j\right)^{-1} = E_{J'(w_j)}\left(X_{j+1} - w_j\right)^2 = Var_{J'(w_j)}\left(X_{j+1}\right).$$

Remark: A convenient representation that we will use in the future is

$$\begin{aligned}
&\int_0^1\int_0^1 J''\left(w_j + vu(w_{j+1}(x) - w_j)\right)ududv \\
&= E\left(J''\left(w_j + VU(w_{j+1}(x) - w_j)\right)U\right)
\end{aligned} \tag{10}$$

where U and V are independent uniformly distributed random variables over $[0,1]$.

Proof. The result is shown by looking at a Taylor expansion of $J\left(w_{j+1}(x)\right)$ about the point w_j. Recall that by definition $J(\cdot)$ is twice differentiable on $\mathbb{R}\backslash\{0\}$ and differentiable on $\mathbb{R}$. Note that

$$w_{j+1}(x) = w_j + \frac{1}{n-j-1}\left(w_j - x\right). \tag{11}$$

On the other hand,

$$J\left(w_{j+1}(x)\right) - J(w_j) = \int_0^1 J'(w_j + u(w_{j+1}(x) - w_j))(w_{j+1}(x) - w_j)du \tag{12}$$

and

$$\begin{aligned} &J'(w_j + u(w_{j+1}(x) - w_j)) - J'(w_j) \\ &= \int_0^1 J''\left(w_j + vu(w_{j+1}(x) - w_j)\right)u(w_{j+1}(x) - w_j)dv. \end{aligned} \tag{13}$$

Note that the integral representation in the previous display is valid for any values of w_j, $w_{j+1}(x)$ and u because $J''(\cdot)$ is continuous except at the origin. Combining (11), (12) and (13) we obtain that

$$\begin{aligned} J\left(w_{j+1}(x)\right) = J(w_j) &+ \frac{1}{n-j-1}(w_j - x)J'(w_j) \\ &+ \frac{(w_j - x)^2}{(n-j-1)^2}\int_0^1\int_0^1 J''\left(w_j + vu(w_{j+1}(x) - w_j)\right)ududv. \end{aligned}$$

The second statement follows from the relationship between ψ and J. □

We now provide an intuitive analysis of $Z_{n,1}$. Using the previous result, the definition of θ_j in (8), and assuming

$$\int_0^1\int_0^1 J''\left(w_j + vu(w_{j+1}(x) - w_j)\right)ududv \approx J''(w_j)/2$$

we obtain (using formally $\tau^{(n)} \approx n - m$ for some positive integer m)

$$\begin{aligned} \log Z_{n,1} &= -\sum_{j=0}^{\tau^{(n)}-1}\left(\theta_j X_{j+1} - \psi\left(\theta_j\right)\right) \\ &\approx \sum_{j=0}^{n-m-1}\left((n-j-1)J\left(W_{j+1}\right) - (n-j)J\left(W_j\right)\right) \\ &\quad - \frac{1}{2}\sum_{j=0}^{n-m-1}\frac{J''\left(W_j\right)}{(n-j-1)}(X_{j+1} - W_j)^2 \\ &= -nJ(\beta) - mJ(W_{n-m}) - \frac{1}{2}\sum_{j=0}^{n-m-1}\frac{J''\left(W_j\right)(X_{j+1} - W_j)^2}{(n-j-1)}. \end{aligned}$$

Under the sampler we have that $S_{n-m} \approx (n-m)\beta$ with high probability and therefore $mJ(W_{n-m}) \approx mJ(\beta)$. One then arrives at the following plausible upper bound

(for some constant $c \in (0, \infty)$)

$$\widetilde{E} Z_{n,1}^2 \leq c \exp\left(-2nJ\left(\beta\right)\right) n^{-1} \times \widetilde{E} \exp\left(- \sum_{j=0}^{n-m-1} \left(\frac{J''\left(W_j\right)\left(X_{j+1} - W_j\right)^2}{(n-j-1)} - \frac{1}{n-j-1} \right) \right).$$

The main issue then becomes understanding the behavior of the expectation

$$\widetilde{E} \exp\left(- \sum_{j=0}^{n-m-1} \left(\frac{J''\left(W_j\right)\left(X_{j+1} - W_j\right)^2}{(n-j-1)} - \frac{1}{n-j-1} \right) \right). \tag{14}$$

The crucial observation is that $W_j \in ((n-j)^{-1/2}, \lambda)$ throughout the course of the algorithm and that the random variables

$$\frac{J''\left(W_j\right)\left(X_{j+1} - W_j\right)^2}{(n-j-1)} - \frac{1}{n-j-1}$$

are martingale differences with conditional variance of order $O\left(1/(n-j-1)^2\right)$. So, working backwards in time, we shall argue that (14) remains bounded as $n \nearrow \infty$, thereby concluding that $\widetilde{E} Z_{n,1}^2 \leq c P\left(S_n > n\beta\right)^2$ for some constant $c \in (0, \infty)$.

It is important to note that the fact that $J\left(\cdot\right)$ is twice continuously differentiable on $(0, \infty)$ seems crucial for the development.

The next section is devoted to the proof of the following result.

Theorem 2. *For each* $p > 1$,

$$\sup_{n \geq 1} \frac{\widetilde{E} Y_n^p}{P\left(S_n > n\beta\right)^p} < \infty.$$

Note that the result stated in Theorem 2 is in fact stronger than just bounded relative error for the estimator, since the result is stated for arbitrary $p > 1$. For a discussion on the benefits of establishing this stronger result see [11].

4 Rigorous Efficiency Analysis

In order to provide the proof of Theorem 2 we need the following result which is a companion to Lemma 1.

Lemma 2. *In the context of Lemma 1 and equation (10), assume that* $0 \leq j \leq n-2$, $(n-j)^{-1/2} < w_j \leq \lambda$. *Let* U *and* V *be independent, uniformly distributed random variables also independent of* X_{j+1} *given* w_j. *Set*

$$\eta_{j+1}\left(X_{j+1}\right) = VU\left(w_{j+1}\left(X_{j+1}\right) - w_j\right) = VU\frac{w_j - X_{j+1}}{n-j-1}.$$

Then, there exists a constant $c(\lambda) \in (0,\infty)$ *such that*

$$\left| E_{J'(w_j)}\left(\frac{J''\left(w_j+\eta_{j+1}\left(X_{j+1}\right)\right)(X_{j+1}-w_j)^2 U}{(n-j-1)} \right) - \frac{1}{2(n-j-1)} \right| \leq \frac{c(\lambda)}{(n-j)^2}.$$

Proof. Define $\tilde{\eta}\left(X_{j+1}, w_j\right) \doteq w_j + \eta_{j+1}(X_{j+1})$. Then note that

$$\begin{aligned}
&\left| E_{J'(w_j)}\left(J''\left(\tilde{\eta}\left(X_{j+1}, w_j\right)\right)(X_{j+1}-w_j)^2 U\right) - \frac{1}{2} \right| \\
&= \left| E_{J'(w_j)}\left(\left(J''\left(\tilde{\eta}\left(X_{j+1}, w_j\right)\right) - J''\left(w_j\right)\right) U \left(X_{j+1}-w_j\right)^2 \right) \right| \\
&\leq E_{J'(w_j)}\left(\left| J''\left(\tilde{\eta}\left(X_{j+1}, w_j\right)\right) - J''\left(w_j\right) \right| \left(X_{j+1}-w_j\right)^2 \right).
\end{aligned}$$

Let $\kappa > 0$ fixed (to be chosen later) and write

$$\begin{aligned}
&E_{J'(w_j)}\left(\left| J''\left(\tilde{\eta}\left(X_{j+1}, w_j\right)\right) - J''\left(w_j\right) \right| \left(X_{j+1}-w_j\right)^2 \right) \\
&= E_{J'(w_j)}\left(\left| J''\left(\tilde{\eta}\left(X_{j+1}, w_j\right)\right) - J''\left(w_j\right) \right| \left(X_{j+1}-w_j\right)^2 ; \tilde{\eta}\left(X_{j+1}, w_j\right) \leq 0 \right) \\
&+ E_{J'(w_j)}\left(\left| J''\left(\tilde{\eta}\left(X_{j+1}, w_j\right)\right) - J''\left(w_j\right) \right| \left(X_{j+1}-w_j\right)^2 ; \tilde{\eta}\left(X_{j+1}, w_j\right) \in (0,\kappa) \right) \\
&+ E_{J'(w_j)}\left(\left| J''\left(\tilde{\eta}\left(X_{j+1}, w_j\right)\right) - J''\left(w_j\right) \right| \left(X_{j+1}-w_j\right)^2 ; \tilde{\eta}\left(X_{j+1}, w_j\right) > \kappa \right).
\end{aligned}$$

Let us write I_1, I_2 and I_3 for the last three expectations in the previous display respectively. We have for any positive even integer m, that on $w_j \in ((n-j)^{-1/2}, \lambda)$

$$\begin{aligned}
I_1 &= J''\left(w_j\right) E_{J'(w_j)}\left(\left(X_{j+1}-w_j\right)^2 ; w_j + VU\frac{w_j - X_{j+1}}{n-j-1} \leq 0 \right) \\
&= J''\left(w_j\right) E_{J'(w_j)}\left(\left(X_{j+1}-w_j\right)^2 ; VU\frac{X_{j+1}-w_j}{n-j-1} \geq w_j \right) \\
&\leq J''\left(w_j\right) E_{J'(w_j)}\left(\left(X_{j+1}-w_j\right)^2 ; VU\frac{X_{j+1}-w_j}{n-j-1} \geq \frac{1}{(n-j)^{1/2}} \right) \\
&= J''\left(w_j\right) E_{J'(w_j)}\left(\left(X_{j+1}-w_j\right)^2 ; VU\left(X_{j+1}-w_j\right) \geq \frac{(n-j-1)}{(n-j)^{1/2}} \right) \\
&\leq \frac{J''\left(w_j\right)(n-j)^{m/2}}{(n-j-1)^m} E_{J'(w_j)}\left(\left(X_{j+1}-w_j\right)^{m+2} \right) \leq \frac{c(\lambda)}{(n-j)^{m/2}}.
\end{aligned}$$

In the penultimate inequality we have used $E\left(Z^2; Z/a > 1\right) \leq E\left(Z^{2+m}/a^m\right)$ for $a > 0$ any random variable Z, and positive, even integer m, and in the last line we use the fact that $w_j \in ((n-j)^{-1/2}, \lambda)$. Next, we have that

$$I_2 = E_{J'(w_j)}\left(\left|J''\left(\tilde{\eta}\left(X_{j+1},w_j\right)\right)-J''\left(w_j\right)\right|\left(X_{j+1}-w_j\right)^2;\tilde{\eta}\left(X_{j+1},w_j\right)\in(0,\kappa)\right)$$
$$\leq \sup_{0<s<\kappa}\left|J'''(s)\right| E_{J'(w_j)}\left(\left|\eta_{j+1}(X_{j+1})\right|\left(X_{j+1}-w_j\right)^2;\tilde{\eta}\left(X_{j+1},w_j\right)\in(0,\kappa)\right)$$
$$\leq \sup_{0<s<\kappa}\left|J'''(s)\right| E_{J'(w_j)}\left(\frac{\left|X_{j+1}-w_j\right|^3}{n-j-1};\tilde{\eta}\left(X_{j+1},w_j\right)\in(0,\kappa)\right)\leq\frac{c(\lambda)}{n-j}.$$

for some constant $c(\lambda)\in(0,\infty)$ (this follows because $J(\cdot)$ is smooth on $(0,\kappa)$). Finally, we can use the relationship $J''(\theta)=1/\psi''\left(J'(\theta)\right)$ to see that if $\kappa>\lambda$

$$I_3 = E_{J'(w_j)}\left(\left|J''\left(\tilde{\eta}\left(X_{j+1},w_j\right)\right)-J''\left(w_j\right)\right|\left(X_{j+1}-w_j\right)^2;\tilde{\eta}\left(X_{j+1},w_j\right)>\kappa\right)$$
$$\leq \frac{1}{\inf_{\theta\geq 0}\psi''(\theta)} E_{J'(w_j)}\left(\left(w_j-X_{j+1}\right)^2;VU\frac{w_j-X_{j+1}}{n-j-1}>\kappa-w_j\right)$$
$$\leq \frac{1}{\inf_{\theta\geq 0}\psi''(\theta)} E_{J'(w_j)}\left(\left(w_j-X_{j+1}\right)^2;VU\frac{w_j-X_{j+1}}{n-j-1}>\kappa-\lambda\right)$$
$$\leq \frac{1}{\inf_{\theta\geq 0}\psi''(\theta)}\frac{E_{J'(w_j)}\left(\left(w_j-X_{j+1}\right)^{2+m}\right)}{(\kappa-\lambda)^m(n-j-1)^m}\leq\frac{c(\lambda)}{(n-j)^m},$$

where the previous inequality follows from Assumption 2 just as we did for the previous to last line in the analysis of I_1. Combining our estimates for I_1, I_2 and I_3 we obtain the result. □

We will now use the previous result to analyze the second moment of Y_n under the law induced by the importance sampling distribution $\widetilde{P}$ associated with Algorithm 1. First we need to define the following terms. For $0\leq j\leq n-2$, define,

$$D_{j+1}\left(X_{j+1},w_j\right)=E\left[J''\left(w_j+\eta_{j+1}\left(X_{j+1}\right)\right)(X_{j+1}-w_j)^2U|X_{j+1}\right].$$

Next, for $0\leq j\leq n-2$, $d_{j+1}\left(w_j\right)=E_{J'(w_j)}\left(D_{j+1}\left(X_{j+1},w_j\right)\right)$. Finally, write

$$\overline{D}_{j+1}\left(X_{j+1},w_j\right)=\left(D_{j+1}\left(X_{j+1},w_j\right)-d_{j+1}\left(w_j\right)\right)I\left(\tau^{(n)}>j\right)\qquad(15)$$

and note that the $\overline{D}_{j+1}\left(X_{j+1},W_j\right)$'s form a sequence of martingale differences. We then have the following bound.

Proposition 4. *There exists a constant $m(\lambda)\in(0,\infty)$ such that*

$$Y_n^p\leq m(\lambda)\frac{\exp\left(-pnJ(\beta)\right)\left(n-\tau^{(n)}+1\right)^{p/2}}{n^{p/2}}\exp\left(-\sum_{j=0}^{n-2}\frac{p\overline{D}_{j+1}\left(X_{j+1},W_j\right)}{2(n-j)}\right).$$

Proof. First note that Lemma 1 guarantees that given $W_j=w_j$, for $j\leq n-2$

$$-\theta_j X_{j+1}+\psi\left(\theta_j\right)=(n-j-1)J\left(w_{j+1}\left(X_{j+1}\right)\right)-(n-j)J\left(w_j\right)$$

$$-\frac{D_{j+1}\left(X_{j+1},w_j\right)}{2(n-j-1)}.$$

Therefore, on $\tau^{(n)}=n$ we have that

$$-\sum_{j=0}^{\tau^{(n)}-1}\left(\theta_j X_{j+1}-\psi\left(\theta_j\right)\right)=-\sum_{j=0}^{n-2}\left(\theta_j X_{j+1}-\psi\left(\theta_j\right)\right)-\left(\theta_{n-1}X_n-\psi\left(\theta_{n-1}\right)\right)$$

$$=-nJ(\beta)-\sum_{j=0}^{n-2}\frac{D_{j+1}\left(X_{j+1},w_j\right)}{2(n-j-1)}+J\left(W_{n-1}\right)-\left(\theta_{n-1}X_n-\psi\left(\theta_{n-1}\right)\right).$$

On the other hand,

$$J\left(W_{n-1}\right)-\left(X_n\theta_{n-1}-\psi\left(\theta_{n-1}\right)\right)=\theta_{n-1}\left(W_{n-1}-X_n\right)=-\theta_{n-1}\left(S_n-n\beta\right).$$

Therefore,

$$Y_n I\left(\tau^{(n)}=n\right)\le\exp\left(-nJ(\beta)-\sum_{j=0}^{\left(\tau^{(n)}-1\right)\wedge(n-2)}\frac{D_{j+1}\left(X_{j+1},w_j\right)}{2(n-j)}\right)I\left(S_n>n\beta\right).$$

Similarly, on $\tau^{(n)}<n$ we have

$$-\sum_{j=0}^{\tau^{(n)}-1}\left(\theta_j X_{j+1}-\psi\left(\theta_j\right)\right)=-\sum_{j=0}^{\left(\tau^{(n)}-1\right)\wedge(n-2)}\left(\theta_j X_{j+1}-\psi\left(\theta_j\right)\right)\qquad(16)$$

$$=-nJ(\beta)+\left(n-\tau^{(n)}\right)J\left(W_{\tau^{(n)}}\right)-\sum_{j=0}^{\left(\tau^{(n)}-1\right)\wedge(n-2)}\frac{D_{j+1}\left(X_{j+1},W_j\right)}{2(n-j)}.$$

On the other hand, we can use the definition of J and θ to get the following equality on $\tau^{(n)}<n$

$$(n-\tau^{(n)})\psi\left(\theta_{\tau^{(n)}}\right)-\theta_{\tau^{(n)}}\left(S_n-S_{\tau^{(n)}}\right)=-\left((n-\tau^{(n)})J\left(W_{\tau^{(n)}}\right)+\theta_{\tau^{(n)}}\left(S_n-n\beta\right)\right).$$

Recalling the definition of our estimator, Y_n we obtain that

$$Y_n I\left(\tau^{(n)}<n\right)\le\exp\left(-nJ(\beta)-\sum_{j=0}^{\left(\tau^{(n)}-1\right)\wedge(n-2)}\frac{D_{j+1}\left(X_{j+1},w_j\right)}{2(n-j)}\right)I\left(S_n>n\beta\right).$$

Therefore, we obtain that

$$Y_n\le\exp\left(-nJ(\beta)-\sum_{j=0}^{\left(\tau^{(n)}-1\right)\wedge(n-2)}\frac{D_{j+1}\left(X_{j+1},w_j\right)}{2(n-j)}\right)I\left(S_n>n\beta\right).\qquad(17)$$

On the other hand, one can use the fact that $\overline{D}_j = 0$ for $j > \tau^{(n)}$ to see,

$$-nJ(\beta) - \sum_{j=0}^{(\tau^{(n)}-1)\wedge(n-2)} \frac{D_{j+1}\left(X_{j+1}, w_j\right)}{2(n-j)} = -nJ(\beta) - \sum_{j=0}^{n-2} \frac{\overline{D}_{j+1}\left(X_{j+1}, W_j\right)}{2(n-j)}$$
$$- \sum_{j=0}^{(\tau^{(n)}-1)\wedge(n-2)} \left(\frac{d_{j+1}\left(W_j\right)}{2(n-j)} - \frac{1}{2(n-j)} \right) - \sum_{j=0}^{(\tau^{(n)}-1)\wedge(n-2)} \frac{1}{2(n-j)}.$$

We now use Lemma 2 to bound the penultimate term in the previous display,

$$\left| \sum_{j=0}^{(\tau^{(n)}-1)\wedge(n-2)} \left(\frac{d_{j+1}\left(W_j\right)}{2(n-j)} - \frac{1}{2(n-j)} \right) \right| \leq c(\lambda) \sum_{j=1}^{\infty} j^{-2} < \infty. \tag{18}$$

Next using standard bounds on harmonic numbers we have the following

$$\sum_{j=0}^{(\tau^{(n)}-1)\wedge(n-2)} \frac{1}{n-j} \geq \sum_{j=1}^{n} \frac{1}{j} - \sum_{j=1}^{n-\tau^{(n)}+1} \frac{1}{j} \geq \log(n) - \log\left(n - \tau^{(n)} + 1\right) - \frac{1}{2}. \tag{19}$$

Putting the estimates from bounds (16), (18), (19) together into (17) we see that there exists a constant $m(\lambda) \in (0, \infty)$ such that

$$Y_n \leq m(\lambda)\, I(S_n \geq n\beta) \exp\left[-nJ(\beta) - \sum_{j=0}^{n-2} \frac{\overline{D}_{j+1}\left(X_{j+1}, W_j\right)}{2(n-j)} \right] \left(\frac{n - \tau^{(n)} + 1}{n} \right)^{1/2}.$$

The result then follows. □

Recall that we use $\widetilde{E}(\cdot)$ to denote the change of measure induced by Algorithm 1. The previous proposition indicates that

$$\widetilde{E} Y_n^p \leq \frac{m(\lambda)\, e^{-pnJ(\beta)}}{n^{p/2}} \widetilde{E} \left(\exp\left(-\sum_{j=0}^{n-2} \frac{p\overline{D}_{j+1}\left(X_{j+1}, W_j\right)}{2(n-j)} \right) \left(n - \tau^{(n)} + 1\right)^{p/2} \right).$$

Using the Cauchy-Schwarz inequality, we obtain

$$\left(\widetilde{E} Y_n^p\right)^2 \leq \frac{m(\lambda)\, e^{-2pnJ(\beta)}}{n^p} \widetilde{E} \left(n - \tau^{(n)} + 1\right)^p \widetilde{E} \exp\left(-\sum_{j=0}^{n-2} \frac{p\overline{D}_{j+1}\left(X_{j+1}, W_j\right)}{(n-j)} \right). \tag{20}$$

In order to verify strong efficiency of the algorithm it suffices to show that

$$\widetilde{E} \left(\exp\left(-\sum_{j=0}^{n-2} \frac{p\overline{D}_{j+1}\left(X_{j+1}, W_j\right)}{(n-j)} \right) \right) = O(1),$$

$$\widetilde{E}\left(\left(n-\tau^{(n)}+1\right)^p\right)=O(1)$$

as $n \nearrow \infty$. We first establish the required property for $\widetilde{E}\left(\left(n-\tau^{(n)}+1\right)^p\right)$.

Proposition 5. *For any $p \in (1,\infty)$ we have that*

$$\sup_{n\geq 1}\widetilde{E}\left(\left(n-\tau^{(n)}\right)^p\right)<\infty.$$

Proof. By definition $\widetilde{E}[(n-\tau^{(n)})^p]=\sum_{k=1}^{n-1}(n-k)^p\widetilde{P}(\tau^{(n)}=k)$ and

$$\widetilde{P}\left(\tau^{(n)}=k\right)\leq \widetilde{P}\left(\tau^{(n)}>k-1, W_k\geq\lambda\right)+\widetilde{P}\left(\tau^{(n)}>k-1, W_k\leq 1/(n-k)^{1/2}\right).$$

Now, define the martingale difference $\tilde{D}_j=\left(W_j-W_{j-1}\right)\times I\left(\tau^{(n)}>j-1\right)$ (for $1\leq j\leq n-1$) and note that (recall that $W_0=\beta$)

$$\widetilde{P}\left(\tau^{(n)}>k-1, W_k\geq\lambda\right)=\widetilde{P}\left(\tau^{(n)}>k-1, \sum_{j=1}^{k}\tilde{D}_j\geq\lambda-\beta\right),$$

$$\widetilde{P}\left(\tau^{(n)}>k-1, W_k\leq(n-k)^{-1/2}\right)=\widetilde{P}\left(\tau^{(n)}>k-1, \sum_{j=1}^{k}\tilde{D}_j\leq(n-k)^{-1/2}-\beta\right).$$

We will show that there exists a constant $m_1\in(0,\infty)$ such that

$$\widetilde{P}\left(\sum_{j=1}^{k}\tilde{D}_j\geq\lambda-\beta\right)\leq m_1\exp\left(-(n-k)^{1/3}\right). \tag{21}$$

To see this, note that given $W_{k-1}=w_{k-1}$ we can write

$$\tilde{D}_k=\left(\frac{w_{k-1}-X_k}{n-k}\right)I(\tau^{(n)}>k-1).$$

Thus, if $\eta\in(0,\infty)$ then $\widetilde{E}\left(\exp\left(\eta\tilde{D}_k\right)\middle|\,\tilde{D}_1,...,\tilde{D}_{k-1}\right)=\exp\left(\chi_k\left(\frac{\eta}{n-k}\right)\right)$, where

$$\chi_k\left(\frac{\eta}{n-k}\right)=\frac{\eta w_{k-1}I\left(\tau^{(n)}>k-1\right)}{n-k}+\psi\left(\frac{-\eta I\left(\tau^{(n)}>k-1\right)}{n-k}+\theta_{k-1}\right)-\psi\left(\theta_{k-1}\right).$$

If $\eta=(n-k)^{1/3}$, then because $\tau^{(n)}>k-1$ (which implies $W_{k-1}\in(1/(n-k)^{1/2},\lambda)$), the smoothness of ψ, and $\psi'(\theta_{k-1})=w_{k-1}$, we can use a Taylor expansion with remainder term to see that there exists a constant $m_2(\lambda)\in(0,\infty)$ such

that

$$\chi_k\left(\frac{\eta}{n-k}\right) \leq \frac{m_2(\lambda)}{(n-k)^{4/3}}.$$

Applying the previous considerations subsequently for $j = k-1, k-2, ..., 1$ we obtain that

$$\widetilde{E}\left(\exp\left((n-k)^{1/3}\sum_{j=1}^{k} D'_j\right)\right) \leq \exp\left(\sum_{j=1}^{k}\frac{m_2(\lambda)}{(n-j)^{4/3}}\right) = m_1.$$

Chebyshev's inequality then yields inequality (21) as indicated. A completely analogous estimate can be obtained for $\widetilde{P}\left(\sum_{j=1}^{k}\tilde{D}_j \leq (n-k)^{-1/2} - \beta\right)$.
Therefore we conclude that

$$\widetilde{E}\left(\left(n-\tau^{(n)}\right)^p\right) = \sum_{k=1}^{n-1}(n-k)^p\widetilde{P}\left(\tau^{(n)}=k\right) \leq 2\sum_{k=1}^{n-1}(n-k)^p m_1 \exp\left(-(n-k)^{1/3}\right),$$

which is clearly bounded as $n \nearrow \infty$. □

Finally, we turn to the remaining result required to establish strong efficiency.

Proposition 6. *For each $\eta > 0$ and $p > 1$ we have that*

$$\sup_{n\geq 1}\widetilde{E}\left(\exp\left(-p\sum_{j=0}^{n-2}\frac{\eta\overline{D}_{j+1}\left(X_{j+1}, W_j\right)}{(n-j)}\right)\right) < \infty.$$

Proof. We have that for any $\eta > 0$, given $W_j = w_j$, on $\tau^{(n)} > j$ and $0 \leq j \leq n-2$

$$\widetilde{E}\left(e^{-\frac{p\eta\overline{D}_{j+1}\left(X_{j+1},w_j\right)}{(n-j)}}\middle| X_1, ..., X_j\right) = \exp\left(\xi_j\left(-\frac{p}{n-j}\eta I\left(\tau^{(n)} > j\right), w_j\right)\right),$$

where $\xi_j\left(\theta, w_j\right) = \log\widetilde{E}\exp\left(\theta\overline{D}_{j+1}\left(X_{j+1}, w_j\right)\right)$.

From the definition of $\overline{D}_{j+1}$, and the convexity of $J(\cdot)$ observe that $\xi_j\left(\theta, w_j\right) < \infty$ for $\theta < 0$. Moreover, we have that $\xi'_j\left(0, w_j\right) = 0$ and therefore on $\tau^{(n)} > j$ (which implies that $(n-j)^{-1/2} \leq w_j \leq \lambda$) there exists $m_3(\lambda) \in (0, \infty)$ such that

$$\xi_j\left(\frac{-p\eta}{n-j}\right) \leq \frac{p^2\eta^2}{2(n-j)^2}\sup_{-\eta/(n-j)\leq\theta\leq 0}\xi''_j(\theta) \leq \frac{p^2\eta^2 m_3(\lambda)}{2(n-j)^2}.$$

Iterating the previous calculations for $j = n-2, n-3, ..., 0$ we obtain that

$$\widetilde{E}\left(\exp\left(-p\sum_{j=0}^{n-2}\frac{\eta\overline{D}_{j+1}\left(X_{j+1}, W_j\right)}{(n-j)}\right)\right) \leq \exp\left(\sum_{j=1}^{n-1}\frac{\eta^2 m_3(\lambda)}{2(n-j)^2}\right) = O(1)$$

as $n \nearrow \infty$, which yields the result. □

Theorem 2 follows easily from the previous two propositions by recalling the bound in display (20).

In the Introduction we emphasized the distinction between OET and the zero-variance change of measure in the sense that the overshoot is controlled as $n \nearrow \infty$ under the zero-variance change of measure but grows under OET. As the next result shows, under our sampler the overshoot stays bounded in expectation.

Proposition 7.

$$\sup_{n\geq 1} \widetilde{E}\left(|S_n - n\beta|\right) < \infty.$$

Proof. We first write $\widetilde{E}\,|S_n - n\beta| \leq \widetilde{E}|S_n - S_{\tau^{(n)}}| + \widetilde{E}|S_{\tau^{(n)}} - n\beta|$. We analyze the latter term first, therefore note that

$$\begin{aligned} |S_{\tau^{(n)}} - n\beta| &\leq |S_{\tau^{(n)}-1} - (\tau^{(n)} - 1)\beta| + |X_{\tau^{(n)}} - (n - \tau^{(n)} + 1)\beta| \\ &\leq 2\lambda(n - \tau^{(n)} + 1) + |X_{\tau^{(n)}}|, \end{aligned}$$

where the second inequality follows from the definition of $\tau^{(n)}$ and that $\lambda > \beta$. Of course the expected value of $n - \tau^{(n)}$ stays bounded as $n \nearrow \infty$ thanks to Proposition 5. Therefore it remains to look at the expected value of $X_{\tau^{(n)}}$ based on the value of $\tau^{(n)}$. In particular, we have that

$$\begin{aligned} \widetilde{E}|X_{\tau^{(n)}}| &= \sum_{j=0}^{n-1} \widetilde{E}\Big(|X_{j+1}|; \tau^{(n)} = j+1, \tau^{(n)} > j\Big) \\ &\leq \sum_{j=0}^{n-1} \widetilde{E}\Big(|X_{j+1}|^2; \tau^{(n)} > j\Big)^{1/2} \widetilde{P}\Big(\tau^{(n)} = j+1\Big)^{1/2}. \end{aligned}$$

It follows from steepness and the fact that $\tau^{(n)} > j$ that there exists a constant $c_0(\lambda) \in (0, \infty)$ such that $\widetilde{E}(|X_{j+1}|^2 \ |\tau^{(n)} > j) \leq c_0(\lambda)$.

As we established in the proof of Proposition 5, it follows that $\widetilde{P}(\tau^{(n)} = j+1) \leq m_1 \exp\left(-(n-j)^{1/3}\right)$ for some constant $m_1 \in (0, \infty)$. Therefore, we obtain that

$$\sup_{n\geq 1} \widetilde{E}|X_{\tau^{(n)}}| \leq c_0(\lambda)^{1/2} m_1^{1/2} \sum_{j=1}^{\infty} \exp\Big(-j^{1/3}/2\Big) < \infty \tag{22}$$

and thus, $\sup_{n\geq 1} E\left[|S_{\tau^{(n)}} - n\beta|\right] < \infty$.

The proof will be completed once we show that $\widetilde{E}\left[|S_n - S_{\tau^{(n)}}|\right]$ stays bounded with n. First, note that

$$\widetilde{E}\left(|S_n - S_{\tau^{(n)}}|; \tau_0^{(n)} \leq \tau_1^{(n)}\right) \leq (E|X_1|)\,\widetilde{E}(n - \tau_0^{(n)}).$$

Observe $E|X_1|$ appears because from time $\tau_0^{(n)} + 1$ up to n the sampling is done under the original / nominal distribution. Again we can use Proposition 5 to bound the expectation of $n - \tau^{(n)}$. Thus it suffices to consider

$$\widetilde{E}\left(|S_n - S_{\tau^{(n)}}|; \tau_0^{(n)} > \tau_1^{(n)}\right).$$

Let us define,

$$\mu(W_{\tau_1^{(n)}}) = \widetilde{E}\left(\left|X_{\tau_1^{(n)}+1}\right| \,|\, W_0, ..., W_{\tau_1^{(n)}}, \tau_1^{(n)}\right) = \int_{-\infty}^{\infty} |x| \exp\left(\theta_{\tau_1^{(n)}} x - \psi\left(\theta_{\tau_1^{(n)}}\right)\right) dF(x).$$

Using the triangle inequality, conditioning and Cauchy-Schwarz we get the following,

$$\begin{aligned}
\widetilde{E}\left(|S_n - S_{\tau^{(n)}}|; \tau_1^{(n)} < \tau_0^{(n)}\right) &\le \widetilde{E}\left(\sum_{j=\tau_1^{(n)}+1}^{n} \left|X_j\right|; \tau_1^{(n)} < \tau_0^{(n)}\right) \\
&\le \sum_{k=1}^{n-1} \sum_{j=k+1}^{n} \widetilde{E}\left(\left|X_j\right|; \tau_1^{(n)} = k\right) \le \sum_{k=1}^{n-1} (n-k) \widetilde{E}\left(\mu(W_{\tau_1^{(n)}}); \tau_1^{(n)} = k\right) \\
&\le \widetilde{E}\left(\mu(W_{\tau_1^{(n)}})^2\right)^{1/2} \sum_{k=1}^{n-1} (n-k) \left(\widetilde{P}\left(\tau_1^{(n)} = k\right)\right)^{1/2}.
\end{aligned}$$

As we noted before, from the proof of Proposition 5, it follows that

$$\sup_{n\ge 1} \sum_{k=1}^{n-1} (n-k) \widetilde{P}\left(\tau_1^{(n)} = k\right)^{1/2} < \infty.$$

It now remains to show that $\widetilde{E}(\mu(W_{\tau_1^{(n)}})^2)$ stays bounded as n goes to infinity. Notice that $0 \le W_{\tau_1^{(n)}} \le 2\lambda + |X_{\tau_1^{(n)}}|$. A similar analysis behind Eq. (22) then allows us to conclude

$$\sup_{n\ge 1} \widetilde{E}(W_{\tau_1^{(n)}})^2 < \infty. \tag{23}$$

Observe that $W_{\tau_1^{(n)}} = \int_{-\infty}^{\infty} x \exp[\theta_{\tau_1^{(n)}} x - \psi(\theta_{\tau_1^{(n)}})] dF(x)$ therefore

$$\begin{aligned}
\mu\left(W_{\tau_1^{(n)}}\right) &= W_{\tau_1^{(n)}} + 2\int_{-\infty}^{0} |x| \exp\left(\theta_{\tau_1^{(n)}} x - \psi\left(\theta_{\tau_1^{(n)}}\right)\right) dF(x) \\
&\le W_{\tau_1^{(n)}} + 2\int_{-\infty}^{0} |x| \exp\left(-\psi\left(\theta_{\tau_1^{(n)}}\right)\right) dF(x).
\end{aligned}$$

Due to strict convexity, the fact that $\psi(0) = 0$, and $\psi'(0) = 0$ we have that $\psi(\theta) \ge 0$ for $\theta \ge 0$, thus

$$\mu(W_{\tau_1^{(n)}}) \le W_{\tau_1^{(n)}} + 2E|X_1|.$$

The proof is completed by combining the bound in the previous display with the result from (23). □

5 The Multidimensional Case

A vector $x \in R^d$ is always assumed to be a column vector, and we denote its transpose by x^T. Therefore the inner product of two vectors x, y is denoted by $x^T y$. The Hessian matrix of a function $f : \mathbb{R}^d \to \mathbb{R}$ is denoted by $D^2 f$. In this section we impose the following assumptions:

[i] $\left(\tilde{X}_j : j \geq 1\right)$ is a sequence of iid d-dimensional random vectors with mean zero and continuous distribution.

[ii] Let A be a closed convex set for which $0 \notin A$.

[iii] Given $\phi \in \mathbb{R}^d$ define $\varrho(\phi) = \log E \exp\left(\phi^T \tilde{X}_j\right)$, put $I(z) = \max_{\phi \in \mathbb{R}^d} (\phi^T z - \varrho(\phi))$ and suppose that there exists $\xi_* \in A$ and $\phi_* \in \mathbb{R}^d$ such that

$$I(\xi_*) = \phi_*^T \xi_* - \varrho(\phi_*) = \inf_{z \in A} I(z). \tag{24}$$

[iv] Assume there is a local change of coordinates $T : \mathbb{R}^{d-1} \supset U \to \partial A$ (where U is an open set) so that the Hessian of $I \circ T$ is well defined and positive definite at $T^{-1}(\xi_*)$, see [1] for details.

[v] Define $X_j = \phi_*^T \tilde{X}_j$ and put $\psi(\theta) = \log E \exp\left(\theta X_j\right)$ for $\theta \in \mathbb{R}$. Suppose that $\psi(\cdot)$ satisfies Assumptions ii) and iii) from Section 2.

Assumption [iv] in particular requires ∂A to be twice continuously differentiable at ξ_*. The geometric interpretation, explained in [9] and [1], is that the boundary of ∂A must be more flat than the level curve of I corresponding to the value $I(\xi_*)$ at ξ_*. If assumption [iv] is violated then our algorithm is still logarithmically efficient. However, the relative error will grow at a polynomial rate which can be shown to be not larger than that of OET.

Analogous to the one-dimensional setting we define the exponential family $(\widetilde{F}_\phi : \phi \in \mathbb{R}^d)$ generated by the distribution $\widetilde{F}(\cdot) = P(\widetilde{X} \leq \cdot)$ (inequality is taken componentwise)

$$d\widetilde{F}_\phi = \exp\left(\phi^T x - \varrho(\phi)\right) d\widetilde{F}.$$

Note that by the definition of ϕ_* and ξ_*,

$$\xi_* = \frac{E\left[\tilde{X} \exp\left(\phi_*^T \tilde{X}\right)\right]}{E\left[\exp\left(\phi_*^T \tilde{X}\right)\right]} \text{ and thus } \phi_*^T \xi_* = \frac{E\left[\phi_*^T \tilde{X} \exp\left(\phi_*^T \tilde{X}\right)\right]}{E\left[\exp\left(\phi_*^T \tilde{X}\right)\right]}.$$

We define $\beta = \phi_*^T \xi_*$ and use exactly the same notation as in Section 2 in the context of equations (3), (1) and (2). So, we see that $\theta_\beta = 1$ and $J(\beta) = I(\xi_*)$. Moreover, since the analysis of the estimator will be reduced to the one dimensional setting taking advantage of the random variables X_j's defined in Assumption [iv], we also refer the reader to the definitions of W_j, the associated stopping time $\tau^{(n)}$, the change of measure θ_j and the likelihood ratio in equations (11), (7), (8) and (9).

Under Assumptions [i] to [v] we shall develop a strongly efficient estimator for computing $P\left(\tilde{S}_n / n \in A\right)$ as $n \nearrow \infty$ where $\tilde{S}_n = \tilde{X}_1 + \ldots + \tilde{X}_n$. First, let us recall the following result from [9].

Theorem 3. *Under Assumptions [i] to [iv] there exists a constant $c(A)$ such that*

$$P\left(\tilde{S}_n/n \in A\right) \sim \frac{c(A)}{n^{1/2}} \exp\left(-nJ(\beta)\right) \tag{25}$$

as $n \nearrow \infty$.

The previous result allows us to reduce, under assumptions [i] to [v], the multi-dimensional case problem to the one dimensional case studied in Sections 3 and 4. Note that Assumptions [ii] and [iv] are particularly important because they ensures that the premultiplying constant in (25) is $c(A)/n^{1/2}$ (i.e. the same order as in the one dimensional case). The premultiplying factor can in fact take the form $c(A)n^{\gamma}$ for $-\infty < \gamma \leq (d-2)/2$. Only with Assumptions [ii] and [iv] are we assured that $\gamma = -1/2$, [9] addresses the issue of identifying γ for smooth Borel subsets of R^d. If $\gamma = 1/2$ then modulo constants $P(\tilde{S}_n \in nA)$ behaves like $P(S_n \geq n\beta)$. In order for that to occur one must ensure that the boundary of A does not curve away too sharply from the level set of I at the dominating point ξ_*, Assumption [iv] ensures that the curvature of A with respect to I is sufficiently small.

We now provide an explicit description of the proposed algorithm.

Algorithm 2

Set $w = \beta = \phi_*^T \xi_* > 0$, $L = 1$, $s = 0$, $\bar{s} = 0$, $k = 0$, and λ a large positive constant. Repeat STEP 1 until $n = k$ OR $w \leq (n-k)^{-1/2}$ OR $w \geq \lambda$.

STEP 1: Sample $\tilde{X}$ from $\widetilde{F}_{\theta_w \phi_*}$ and set

$$\begin{aligned} L &\longleftarrow \exp\left(-\theta_w \phi_*^T \tilde{X} + \psi(\theta_w)\right) L, \\ s &\longleftarrow s + X, \\ k &\longleftarrow k + 1, \\ w &\longleftarrow \left(n\beta - \phi_*^T s\right)/(n-k). \end{aligned}$$

STEP 2: If $k < n$ sample $\tilde{X}_{k+1}, \ldots, \tilde{X}_n$ iid rv's from $\widetilde{F}_{\theta_w \phi_*}$ and set

$$\begin{aligned} \bar{s} &\longleftarrow \tilde{X}_{k+1} + \ldots + \tilde{X}_n, \\ L &\longleftarrow \exp\left(-\theta_w \phi_*^T \bar{s} + (n-k)\psi(\theta_w)\right) L. \end{aligned}$$

STEP 3: Output $Z_n = L \times I(s + \bar{s} \in nA)$.

Theorem 4. *Let $\widetilde{E}(\cdot)$ be the expectation operator associated with the change of measure described by Algorithm 2. Then, for each $p > 1$ we have*

$$\sup_{n \geq 1} \frac{\widetilde{E}\left(Z_n^p\right)}{P\left(\tilde{S}_n/n \in A\right)^p} < \infty.$$

Proof. Since $I\left(\tilde{S}_n \in nA\right) \leq I(S_n \geq n\beta) = I\left(S_n \geq n\phi_*^T \xi_*\right)$ we obtain that the estimator obtained by running Algorithm 2 is bounded by

$$Y_n = \exp\left(-\sum_{j=0}^{\tau^{(n)}-1}\left(\theta_j X_{j+1} - \psi\left(\theta_j\right)\right)\right)$$
$$\times I\left(S_n \geq n\beta\right)\exp\left(-\theta_{\tau^{(n)}}\left(S_n - S_{\tau^{(n)}}\right) + (n-\tau^{(n)})\psi\left(\theta_{\tau^{(n)}}\right)\right).$$

Therefore

$$\sup_{n\geq 1}\frac{\tilde{E}Z_n^p}{P(\tilde{S}\in nA)^p} \leq \sup_{n\geq 1}\frac{\tilde{E}Y_n^p}{P(S_n\geq n\beta)^p}\sup_{n\geq 1}\frac{P(S_n\geq n\beta)^p}{P(\tilde{S}\in nA)^p}.$$

The proof is completed by using Theorems 1 and 3 because $\sup_{n\geq 1}\frac{P(S_n\geq n\beta)}{P(\tilde{S}_n\in nA)} < \infty$. □

Acknowledgements The work of Jose Blanchet and Kevin Leder was supported by NSF grants NSF-0846816 and DMS-0806145.

References

1. Andriani, C., Baldi, P.: Sharp estimates of deviations of the sample mean in many dimensions. Ann. Inst. Henri Poincare **33**, 371–385 (1997)
2. Asmussen, S., Glynn, P.: Stochastic Simulation: Algorithms and Analysis. Springer-Verlag, New York, NY, USA (2008)
3. Bahadur, R., Rao, R.R.: On deviations of the sample mean. Ann. Math. Stat. (1960)
4. Blanchet, J., Glynn, P.: Strongly efficient estimators for light-tailed sums. In: valuetools '06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools, p. 18. ACM, New York, NY, USA (2006). DOI http://doi.acm.org/10.1145/1190095.1190118
5. Blanchet, J., Liu, J.C.: State-dependent importance sampling for regularly varying random walks. Adv. in Appl. Probab. **40**, 1104–1128 (2008)
6. Borovkov, A.A.: On the limit conditional distributions connected with large deviations. Siberian Mathematical Journal **37**, 635–646 (1996)
7. Dembo, A., Zeitouni, O.: Large deviations techniques and applications, second edn. Springer, New York (1998)
8. Dupuis, P., Wang, H.: Importance sampling, large deviations, and differential games. Stoch. and Stoch. Reports **76**, 481–508 (2004)
9. Iltis, M.: Sharp asymptotics of large deviations in $\mathbb{R}^d$. Journal of Theoretical Probability **8**, 501–522 (1995)
10. Juneja, S., Shahabuddin, P.: Rare event simulation techniques: An introduction and recent advances. In: S.G. Henderson, B.L. Nelson (eds.) Simulation, Handbooks in Operations Research and Management Science. Elsevier, Amsterdam, The Netherlands (2006)
11. L'Ecuyer, P., Blanchet, J., Tuffin, B., Glynn, P.: Asymptotic robustness of estimators in rare-event simulation. ACM TOMACS. To Appear
12. Sadowsky, J.S.: On Monte Carlo estimation of large deviations probabilities. Ann. Appl. Prob. **6**, 399–422 (1996)
13. Siegmund, D.: Importance sampling in the Monte Carlo study of sequential tests. Ann. Stat. **3**, 673–684 (1976)

Distribution of Digital Explicit Inversive Pseudorandom Numbers and Their Binary Threshold Sequence

Zhixiong Chen, Domingo Gomez, and Arne Winterhof

In memory of Edmund Hlawka

Abstract We study the distribution of s-dimensional points of digital explicit inversive pseudorandom numbers with arbitrary lags. We prove a discrepancy bound and derive results on the pseudorandomness of the binary threshold sequence derived from digital explicit inversive pseudorandom numbers in terms of bounds on the correlation measure of order k and the linear complexity profile. The proofs are based on bounds on exponential sums and earlier relations of Mauduit, Niederreiter and Sárközy between discrepancy and correlation measure of order k and of Brandstätter and the third author between correlation measure of order k and linear complexity profile, respectively.

1 Introduction

Inversive methods are attractive alternatives to the linear method for generating pseudorandom numbers, see the recent surveys [11, 12, 17]. In this paper we analyze the distribution of *digital explicit inversive pseudorandom numbers* introduced in [13] and further analyzed in [6, 13, 14, 15, 16].

Zhixiong Chen
1. Key Laboratory of Applied Mathematics, Putian University, Putian, Fujian 351100, P.R. China;
2. Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, Fujian 350007, P.R. China. e-mail: ptczx@126.com.

Domingo Gomez
Faculty of Sciences, University of Cantabria, 39071 Santander, Spain, http://personales.unican.es/gomezd/. e-mail: domingo.gomez@unican.es.

Arne Winterhof
Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Altenbergerstr. 69, 4040 Linz, Austria, http://www.ricam.oeaw.ac.at/people/page.cgi?firstn=Arne;lastn=Winterhof. e-mail: arne.winterhof@oeaw.ac.at.

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_14,

Let $q = p^r$ be a prime power and $\mathbb{F}_q$ the finite field of order q. Let

$$\overline{\gamma} = \begin{cases} \gamma^{-1}, & \text{if } \gamma \in \mathbb{F}_q^*, \\ 0, & \text{if } \gamma = 0. \end{cases}$$

We order the elements of $\mathbb{F}_q = \{\xi_0, \xi_1, \ldots, \xi_{q-1}\}$ using an ordered basis $\{\gamma_1, \ldots, \gamma_r\}$ of $\mathbb{F}_q$ over $\mathbb{F}_p$ for $0 \le n < q$,

$$\xi_n = n_1\gamma_1 + n_2\gamma_2 + \cdots + n_r\gamma_r,$$

if

$$n = n_1 + n_2 p + \cdots + n_r p^{r-1}, \ 0 \le n_i < p, \quad i = 1, \ldots, r.$$

For $n \ge 0$ we define $\xi_{n+q} = \xi_n$. Then the *digital explicit inversive pseudorandom number generator* of period q is defined by

$$\rho_n = \overline{\alpha\xi_n + \beta}, \ n = 0, 1, \ldots$$

for some $\alpha, \beta \in \mathbb{F}_q$ with $\alpha \ne 0$.

If

$$\rho_n = c_{n,1}\gamma_1 + c_{n,2}\gamma_2 + \cdots + c_{n,r}\gamma_r$$

with all $c_{n,i} \in \mathbb{F}_p$, we derive *digital explicit inversive pseudorandom numbers of period* q in the interval $[0, 1)$ by defining

$$y_n = \sum_{j=1}^{r} c_{n,j}\, p^{-j}, \quad n = 0, 1, \ldots. \tag{1}$$

For $s \ge 1$ the distribution of points $(y_n, y_{n\oplus 1}, \ldots, y_{n\oplus(s-1)})$, where $n \oplus k = d$ if $\xi_n + \xi_k = \xi_d$, $0 \le n, k, d < q$, was studied in [13]. Here we study the distribution of the points $(y_{n+d_1}, \ldots, y_{n+d_s})$ for any integers $0 \le d_1 < \cdots < d_s < q$ and the integer addition $+$. We prove a discrepancy bound which is based on estimates for exponential sums generalizing the earlier result of the first author [3] for $s = 2$ using some additional ideas.

As applications we use some results of [4] and [1] to derive bounds on the *correlation measure of order k* and *linear complexity profile* of the binary sequences $\mathcal{R}_q = (r_0, r_1, \ldots, r_{q-1})$ defined by

$$r_n = \begin{cases} 0, & \text{if } 0 \le y_n < \frac{1}{2}, \\ 1, & \text{if } \frac{1}{2} \le y_n < 1, \end{cases} \quad 0 \le n < q. \tag{2}$$

Note that for such applications a discrepancy bound with arbitrary lags $0 \le d_1 < \cdots < d_s < q$ is needed. Most known discrepancy bounds on nonlinear pseudorandom numbers found in the literature consider only the special lags $d_i = i - 1$ for $i = 1, \ldots, s$. In many cases the analysis of the discrepancy becomes much more intricate for arbitrary lags, see for example [10].

We recall that the correlation measure of order k, introduced by Mauduit and Sárközy in [5], is an important measure of pseudorandomness for finite binary sequences. For a finite binary sequence

$$\mathcal{S}_N = \{s_0, s_1, \ldots, s_{N-1}\} \in \{0,1\}^N,$$

the *correlation measure of order k* of $\mathcal{S}_N$ is defined as

$$C_k(\mathcal{S}_N) = \max_{M,D} \left| \sum_{n=1}^{M} (-1)^{s_{n+d_1}+s_{n+d_2}+\cdots+s_{n+d_k}} \right|,$$

where the maximum is taken over all $D = (d_1, \ldots, d_k)$ with non-negative integers $0 \le d_1 < \cdots < d_k$ and M such that $M + d_k \le N - 1$. For a "good" pseudorandom sequence $\mathcal{S}_N$, $C_k(\mathcal{S}_N)$ (for "small" k) is small and is ideally greater than $N^{1/2}$ only by at most a power of $\log N$, see [2].

The linear complexity profile is an important cryptographic characteristic of pseudorandom sequences. A low linear complexity profile has turned out to be undesirable for cryptographical applications.

For a T-periodic binary sequence $\mathcal{S}_T = (s_0, s_1, \ldots, s_{T-1})$ over $\mathbb{F}_2$, the *linear complexity profile* $L(\mathcal{S}_T, N)$ is the function which is defined as the shortest length L of a linear recurrence relation over $\mathbb{F}_2$ for $N > 1$

$$s_{n+L} = c_{L-1}s_{n+L-1} + \cdots + c_0 s_n, \ \ 0 \le n \le N - L - 1,$$

which is satisfied by this sequence.

The discrepancy bound is proved in Section 2, and the bounds on the correlation measure of order k and the linear complexity profile are given in Sections 3 and 4.

2 Discrepancy Bound

In this section we estimate the discrepancy of the points

$$\mathbf{Y}_n = (y_{n+d_1}, \ldots, y_{n+d_s}) \in [0,1)^s, \ n = 0, 1, \ldots, N-1,$$

for any non-negative integers $d_1, \ldots, d_s$ with $0 \le d_1 < \cdots < d_s < q$ and $1 \le N \le q$. We recall that the *discrepancy* of the points $\mathbf{Y}_0, \ldots, \mathbf{Y}_{N-1}$, denoted by $\mathcal{D}_N(d_1, \ldots, d_s)$, is defined by

$$\mathcal{D}_N(d_1, \ldots, d_s) = \sup_{J \subseteq [0,1)^s} \left| \frac{A(J,N)}{N} - |J| \right|,$$

where $A(J,N)$ is the number of points $\mathbf{Y}_0, \ldots, \mathbf{Y}_{N-1}$ which hit the box $J = [\alpha_1, \beta_1) \times \cdots \times [\alpha_s, \beta_s) \subseteq [0,1)^s$, the volume $|J|$ of an interval J is given by $\prod_{i=1}^{s} (\beta_i - \alpha_i)$ and the supremum is taken over all such boxes, see e.g. [9].

Theorem 1. *Let $y_0, y_1, \ldots$ be the sequence defined by (1). For any non-negative integers $d_1, \ldots, d_s$ with $d_1 < \cdots < d_s < q$ and $1 \le N \le q$, the discrepancy $\mathcal{D}_N(d_1, \ldots, d_s)$ of the points*

$$\mathbf{Y}_n = (y_{n+d_1}, \ldots, y_{n+d_s}) \in [0,1)^s,\ n = 0, 1, \ldots, N-1,$$

satisfies

$$\mathcal{D}_N(d_1, \ldots, d_s) = O(N^{-1} 2^{r+rs} rs q^{1/2} (\log q)^s (1 + \log p)^r),$$

where the implied constant is absolute.

Proof. Let $\lambda_{ij} \in \mathbb{F}_p$ $(1 \le i \le s, 1 \le j \le r)$ be not all zero and put $e_p(x) = \exp(2\pi\sqrt{-1}x/p)$ and

$$S_N = S_N(\lambda_{11}, \ldots, \lambda_{sr}) = \sum_{n=0}^{N-1} e_p\left(\sum_{i=1}^{s}\sum_{j=1}^{r} \lambda_{ij} c_{n+d_i, j}\right),$$

where the $c_{i,j}$ are defined in (1). According to [9, Proposition 2.4, Theorem 3.12 and Lemma 3.13] we have

$$\mathcal{D}_N(d_1, \ldots, d_s) \ll 2^s (\log q)^s \frac{1}{N} \max_{\lambda_{11}, \ldots, \lambda_{sr}} |S_N(\lambda_{11}, \ldots, \lambda_{sr})|, \tag{3}$$

where the maximum is taken over all nonzero vectors $(\lambda_{11}, \ldots, \lambda_{sr}) \in \mathbb{F}_p^{sr} \setminus \{(0, \ldots, 0)\}$. Hence it suffices to estimate S_N above.

Let $\{\gamma_1', \ldots, \gamma_r'\}$ be the dual basis of the ordered basis $\{\gamma_1, \ldots, \gamma_r\}$ of $\mathbb{F}_q$ over $\mathbb{F}_p$. Then we have

$$\begin{aligned} S_N &= \sum_{n=0}^{N-1} e_p\left(\sum_{i=1}^{s}\sum_{j=1}^{r} \lambda_{ij} \mathrm{Tr}(\gamma_j' \rho_{n+d_i})\right) \\ &= \sum_{n=0}^{N-1} e_p\left(\mathrm{Tr}\left(\sum_{i=1}^{s}\sum_{j=1}^{r} \lambda_{ij} \gamma_j' \rho_{n+d_i}\right)\right) \\ &= \sum_{n=0}^{N-1} \psi\left(\sum_{i=1}^{s} \mu_i \rho_{n+d_i}\right), \end{aligned}$$

where Tr denotes the absolute trace of $\mathbb{F}_q$, ψ is the additive canonical character of $\mathbb{F}_q$ and

$$\mu_i = \sum_{j=1}^{r} \lambda_{ij} \gamma_j', \quad i = 1, \ldots, s.$$

Since $\lambda_{ij} \in \mathbb{F}_p$ $(1 \le i \le s, 1 \le j \le r)$ are not all zero and $\{\gamma_1', \ldots, \gamma_r'\}$ is a basis of $\mathbb{F}_q$ over $\mathbb{F}_p$, it follows that $\mu_1, \ldots, \mu_s$ are not all zero.

First we present three auxiliary steps for the proof.

(*i*). We call a set of the form $\{\delta + n_1\gamma_1 + \cdots + n_r\gamma_r : 0 \le n_i < N_i, i = 1, \ldots, r\}$ for some integers $0 \le N_1, \ldots, N_r \le p$ and $\delta \in \mathbb{F}_q$ a *box*. Note that the empty set is also a box and that the intersection of a family of boxes is the union of at most 2^r

boxes. (For $r = 1$ this is trivial and in general each r-dimensional box is the direct product of r one-dimensional boxes.)

As in the proof of [7, Theorem 2], it can be verified that for $0 \le \tau, m < q$ there are only 2^{r-1} different $\omega \in \mathbb{F}_q$, namely,

$$\omega = w_2\gamma_2 + \cdots + w_r\gamma_r, \quad w_2, \ldots, w_r \in \{0, 1\}, \tag{4}$$

such that

$$\xi_{m+\tau} = \xi_m + \xi_\tau + \omega,$$

where we used the definition $\xi_{m+q} = \xi_m$, $m = 0, \ldots, q-1$. We are going to prove that the sets

$$S_{\tau,\omega} = \{\xi_m : 0 \le m < q,\ \xi_{m+\tau} = \xi_m + \xi_\tau + \omega\}$$

are boxes. For $0 \le \tau, m < q$, let

$$\tau = \tau_1 + \tau_2 p + \cdots + \tau_r p^{r-1},\ \ 0 \le \tau_1, \tau_2, \ldots, \tau_r < p$$

and

$$m = m_1 + m_2 p + \cdots + m_r p^{r-1},\ \ 0 \le m_1, m_2, \ldots, m_r < p.$$

Put

$$w_1 = 0,\ \ w_{i+1} = \begin{cases} 1, \text{ if } m_i + \tau_i + w_i \ge p, \\ 0, \text{ otherwise,} \end{cases}$$

for $i = 1, 2, \ldots, r$. We get

$$m + \tau = z_1 + z_2 p + \cdots + z_r p^{r-1},\ \ 0 \le z_1, z_2, \ldots, z_r < p$$

where

$$z_i = m_i + \tau_i + w_i - w_{i+1} p,\ \ 1 \le i \le r.$$

Then we get

$$\xi_{m+\tau} = \xi_m + \xi_\tau + \omega,$$

where

$$\omega = w_2\gamma_2 + \cdots + w_r\gamma_r.$$

Note that for fixed τ and ω the sets $S_{\tau,\omega}$ define a partition of $\mathbb{F}_q$ and we have

$$S_{\tau,\omega} = \{\delta + u_1\gamma_1 + \cdots + u_r\gamma_r : 0 \le u_j < k_j,\ j = 1, \ldots, r\},$$

where

$$\delta = \sum_{\substack{j=1 \\ w_{j+1}=1}}^{r-1} (p - \tau_j - w_j)\gamma_j$$

and

$$k_j = \begin{cases} p - \tau_j - w_j, & \text{if } w_{j+1} = 0,\ 1 \le j < r, \\ \tau_j + w_j, & \text{if } w_{j+1} = 1,\ 1 \le j < r, \\ p, & \text{if } j = r. \end{cases}$$

So the sets $S_{\tau,\omega}$ are all boxes.

(*ii*). For $0 \le d_1 < d_2 < \cdots < d_s < q$ and $\omega_1, \ldots, \omega_s \in \mathbb{F}_q$ of the form (4) the sets

$$S_{d_1,\omega_1} \cap \cdots \cap S_{d_s,\omega_s} = \{\xi_n : 0 \le n < q, \xi_{n+d_i} = \xi_n + \xi_{d_i} + \omega_i, i = 1, \ldots, s\}$$

are unions of at most 2^r boxes. As in the proof of [7, Theorem 4] for $1 \le N \le q$, below we verify that the intersection of a box B with $\{\xi_0, \ldots, \xi_{N-1}\}$ is a union of r boxes. Write $B' = B \cap \{\xi_0, \ldots, \xi_{N-1}\}$.

Let $l = \left\lfloor \frac{\log N}{\log p} \right\rfloor + 1$, we write

$$N = v_1 + v_2 p + \cdots + v_l p^{l-1}, \quad 0 \le v_1, v_2, \ldots, v_l < p.$$

We give a partition for B' by defining

$$\begin{aligned}
V_{2,\omega} &= \{\xi_m \in B | m_1 \le v_1, m_2 = v_2, \ldots, m_l = v_l\},\\
V_{j,\omega} &= \{\xi_m \in B | 0 \le m_1, \ldots, m_{j-2} < p,\\
&\qquad m_{j-1} \le v_{j-1} - 1, m_j = v_j, \ldots, m_l = v_l\},\\
&\text{where } j = 3, 4, \ldots, l, \text{ and}\\
V_{1,\omega} &= \{\xi_m \in B | 0 \le m_1, \ldots, m_{l-1} < p,\ m_l \le v_l - 1\}.
\end{aligned}$$

It is easy to see that each $V_{j,\omega}$ is a box since on the coefficients of the ξ_m only possibly more constraints are added.

In summary, there are $2^{(r-1)s}$ possible choices for $\omega_1, \ldots, \omega_s \in \mathbb{F}_q$. For fixed $\omega_1, \ldots, \omega_s \in \mathbb{F}_q$, $S_{d_1,\omega_1} \cap \cdots \cap S_{d_s,\omega_s}$ is a union of at most 2^r boxes B, while $B \cap \{\xi_0, \ldots, \xi_{N-1}\}$ is a union of r boxes $V_{j,\omega}$.

(*iii*). Let $B = \{\delta + n_1\gamma_1 + \cdots + n_r\gamma_r : 0 \le n_i < N_i, i = 1, \ldots, r\}$ with $0 \le N_1, \ldots, N_r \le p$ and $\delta \in \mathbb{F}_q$ be a box. By [18, Lemma 6], we have

$$\sum_{\varsigma \in \mathbb{F}_q^*} \left| \sum_{\xi \in B} \psi(\varsigma\xi) \right| < q(1 + \log p)^r.$$

Now we continue the proof. Let

$$\mathbf{I}(\omega_1, \ldots, \omega_s) = S_{d_1,\omega_1} \cap \cdots \cap S_{d_s,\omega_s} \cap \{\xi_0, \ldots, \xi_{N-1}\}.$$

We note that if $\xi_{d_i} + \omega_i = \xi_{d_j} + \omega_j$ for $i < j$, then there is no n with $0 \le n < q$ such that

$$\xi_{n+d_i} = \xi_n + \xi_{d_i} + \omega_i \quad \text{and} \quad \xi_{n+d_j} = \xi_n + \xi_{d_j} + \omega_j.$$

Otherwise, suppose n_0 is such a value then $\xi_{n_0+d_i} = \xi_{n_0+d_j}$, which leads to $d_i \equiv d_j \pmod q$, a contradiction. So for ω_i, ω_j with $\xi_{d_i} + \omega_i = \xi_{d_j} + \omega_j$,

$$S_{d_i,\omega_i} \cap S_{d_j,\omega_j} = \emptyset,$$

which leads to

$$\mathbf{I}(\omega_1, \ldots, \omega_s) = \emptyset.$$

In such case $|\mathbf{I}(\omega_1, \ldots, \omega_s)| = 0$. Hence we obtain

$$\begin{aligned}
S_N &= \sum_{n=0}^{N-1} \psi\left(\sum_{i=1}^{s} \mu_i \rho_{n+d_i}\right) \\
&= \sum_{n=0}^{N-1} \psi\left(\sum_{i=1}^{s} \mu_i \overline{\alpha\xi_{n+d_i} + \beta}\right) \\
&= \sum_{\omega_1,\ldots,\omega_s} \sum_{\xi \in \mathbf{I}(\omega_1,\ldots,\omega_s)} \psi\left(\sum_{i=1}^{s} \mu_i \overline{\alpha(\xi + \xi_{d_i} + \omega_i) + \beta}\right) \\
&= \sum_{\omega_1,\ldots,\omega_s} \sum_{x \in \mathbb{F}_q} \psi\left(\sum_{i=1}^{s} \mu_i \overline{\alpha(x + \xi_{d_i} + \omega_i) + \beta}\right) \\
&\qquad \sum_{\xi \in \mathbf{I}(\omega_1,\ldots,\omega_s)} \frac{1}{q} \sum_{\varsigma \in \mathbb{F}_q} \psi(\varsigma(x - \xi)) \\
&= \frac{1}{q} \sum_{\omega_1,\ldots,\omega_s} \sum_{\varsigma \in \mathbb{F}_q} \sum_{\xi \in \mathbf{I}(\omega_1,\ldots,\omega_s)} \psi(-\varsigma\xi) \\
&\qquad \sum_{x \in \mathbb{F}_q} \psi\left(\sum_{i=1}^{s} \mu_i \overline{\alpha(x + \xi_{d_i} + \omega_i) + \beta} + \varsigma x\right) \\
&= \sum_{\omega_1,\ldots,\omega_s} \frac{|\mathbf{I}(\omega_1,\ldots,\omega_s)|}{q} \sum_{x \in \mathbb{F}_q} \psi\left(\sum_{i=1}^{s} \mu_i \overline{\alpha(x + \xi_{d_i} + \omega_i) + \beta}\right) \\
&\quad + \frac{1}{q} \sum_{\omega_1,\ldots,\omega_s} \sum_{\varsigma \in \mathbb{F}_q^*} \sum_{\xi \in \mathbf{I}(\omega_1,\ldots,\omega_s)} \psi(-\varsigma\xi) \\
&\qquad \sum_{x \in \mathbb{F}_q} \psi\left(\sum_{i=1}^{s} \mu_i \overline{\alpha(x + \xi_{d_i} + \omega_i) + \beta} + \varsigma x\right).
\end{aligned}$$

By [8, Theorem 2] (see also [19, Lemma 1] or [13, Lemma 1]) the sum over x has absolute value $O(sq^{1/2})$ if the rational functions in the argument are not of the form $A^p - A$. This implies

$$S_N \ll 2^{(r-1)s} s q^{1/2} + 2^{(r-1)s} \cdot s q^{1/2} \cdot \frac{1}{q} \sum_{\varsigma \in \mathbb{F}_q^*} \left| \sum_{\xi \in \mathbf{I}(\omega_1,\ldots,\omega_s)} \psi(\varsigma\xi) \right|.$$

In fact in the proof above we only consider the case when $\mathbf{I}(\omega_1, \ldots, \omega_s) \neq \emptyset$, which leads to $\xi_{d_i} + \omega_i \neq \xi_{d_j} + \omega_j$ for all $i \neq j$. So both rational functions

$$\sum_{i=1}^{s} \mu_i (\alpha(X + \xi_{d_i} + \omega_i) + \beta)^{-1}$$

and

$$\sum_{i=1}^{s} \mu_i (\alpha(X + \xi_{d_i} + \omega_i) + \beta)^{-1} + \varsigma X$$

are not of the form $A^p - A$, where A is a rational function over $\overline{\mathbb{F}}_q$, by [19, Lemma 2] or [13, Lemma 2].

Now according to Steps (*ii*) and (*iii*) above, we have

$$\sum_{\varsigma\in\mathbb{F}_q^*}\left|\sum_{\xi\in\mathbf{I}(\omega_1,\ldots,\omega_s)}\psi(\varsigma\xi)\right| \le 2^r \sum_{\varsigma\in\mathbb{F}_q^*}\left|\sum_{j=1}^{l}\sum_{\xi\in V_{j,\omega}}\psi(\varsigma\xi)\right|$$

$$\le 2^r \sum_{j=1}^{l}\sum_{\varsigma\in\mathbb{F}_q^*}\left|\sum_{\xi\in V_{j,\omega}}\psi(\varsigma\xi)\right|$$

$$\ll 2^r lq(1+\log p)^r \le 2^r rq(1+\log p)^r.$$

Putting everything together, we obtain

$$S_N = O\left(2^{(r-1)s}2^r rsq^{1/2}(1+\log p)^r\right).$$

Now (3) yields the theorem. □

Note that the bound converges slowly if s is large.

3 Correlation Measure of Order k

The correlation measure of order $k=2$ of $\mathcal{R}_q$ satisfies

$$C_2(\mathcal{R}_q) = O(q^{1/2}(\log q)^2(1+\log p)^r)$$

with implied constant depending on r, see [3]. In this paper, we now extend this result to the case of $k>2$.

Theorem 2. *The correlation measure of order k of $\mathcal{R}_q$ defined by (2) satisfies*

$$C_k(\mathcal{R}_q) = O(2^r 2^{(r+1)k} rkq^{1/2}(\log q)^k(1+\log p)^r).$$

Proof. By [4, Theorem 1] and Theorem 1, we have

$$\left|\sum_{n=1}^{M}(-1)^{r_{n+d_1}+\cdots+r_{n+d_k}}\right| \le 2^k M\mathcal{D}_{M+d_k}(d_1,\ldots,d_k)$$

$$= O(2^r 2^{(r+1)k} rkq^{1/2}(\log q)^k(1+\log p)^r)$$

and the result follows. □

Note that the result is only nontrivial if p is large enough.

4 Linear Complexity Profile

In [1, Theorem 1], Brandstätter and the third author used the correlation measure of order k to estimate the linear complexity profile for some related binary sequence

$\mathcal{S}_T$:

$$L(\mathcal{S}_T, N) \geq N - \max_{1 \leq k \leq L(\mathcal{S}_T, N)+1} C_k(\mathcal{S}_T) \tag{5}$$

where $2 \leq N \leq T-1$.

Combining (5) and Theorem 2 we get a lower bound on the linear complexity profile of $\mathcal{R}_q$ after simple calculations.

Corollary 1. *The linear complexity profile of $\mathcal{R}_q$ defined by (2) satisfies*

$$L(\mathcal{R}_q, N) = \Omega\left(\frac{\log(Nq^{-1/2}2^{-r}r^{-1}(1+\log p)^{-r})}{r+\log\log q}\right), \quad 2 \leq N < q.$$

Acknowledgements Z.X.C. was partially supported by the Open Funds of Key Lab of Fujian Province University Network Security and Cryptology under grant 07B005, the Funds of the Education Department of Fujian Province under grant JA07164 and the Natural Science Foundation of Fujian Province of China under grant 2007F3086.

D.G. was partially supported by the Spanish Ministry of Education and Science grant MTM2007-67088.

A.W. was partially supported by the Austrian Science Fund (FWF) under research grant P-19004-N18.

The authors thank Harald Niederreiter for useful suggestions.

References

1. Brandstätter, N., Winterhof, A.: Linear complexity profile of binary sequences with small correlation measure. Periodica Mathematica Hungarica 52(2), 1–8 (2006)
2. Cassaigne, J., Mauduit, C., Sárközy, A.: On finite pseudorandom binary sequences, VII: the measures of pseudorandomness. Acta Arithmetica 103(2), 97–118 (2002)
3. Chen, Z.: Finite binary sequences constructed by explicit inversive methods. Finite Fields and Their Applications 14(3), 579–592 (2008)
4. Mauduit, C., Niederreiter, H., Sárközy, A.: On pseudorandom $[0, 1)$ and binary sequences. Publicationes Mathematicae Debrecen 71(3-4), 305–324 (2007)
5. Mauduit, C., Sárközy, A.: On finite pseudorandom binary sequences I: measures of pseudorandomness, the Legendre symbol. Acta Arithmetica 82, 365–377 (1997)
6. Meidl, W., Winterhof, A.: On the linear complexity profile of explicit nonlinear pseudorandom numbers. Information Processing Letters 85(1), 13–18 (2003)
7. Meidl, W., Winterhof, A.: On the autocorrelation of cyclotomic generator. In: Fq7, Lecture Notes in Computer Science, vol. 2948, pp.1–11. Springer, Berlin Heidelberg (2003)
8. Moreno, C.J., Moreno, O.: Exponential sums and Goppa codes: I. Proceedings of the American Mathematical Society 111, 523–531 (1991)
9. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 63. SIAM, Philadelphia, PA (1992)
10. Niederreiter, H., Rivat, J.: On the correlation of pseudorandom numbers generated by inversive methods. Monatshefte für Mathematik 153(3), 251–264 (2008)
11. Niederreiter, H., Shparlinski, I.E.: Recent advances in the theory of nonlinear pseudorandom number generators. In: Monte Carlo and quasi-Monte Carlo methods 2000, pp. 86–102, Springer, Berlin Heidelberg (2002)

12. Niederreiter, H., Shparlinski, I.E.: Dynamical systems generated by rational functions. In: AAECC, Lecture Notes in Computer Science, vol. 2643, pp. 6–17. Springer-Verlag, Berlin Heidelberg (2003)
13. Niederreiter, H., Winterhof, A.: Incomplete exponential sums over finite fields and their applications to new inversive pseudorandom number generators. Acta Arithmetica 93, 387–399 (2000)
14. Niederreiter, H., Winterhof, A.: On a new class of inversive pseudorandom numbers for parallelized simulation methods. Periodica Mathematica Hungarica 42(1), 77–87 (2001)
15. Niederreiter, H., Winterhof, A.: On the lattice structure of pseudorandom numbers generated over arbitrary finite fields. Applicable Algebra in Engineering, Communication and Computing 12(3), 265–272 (2001)
16. Pirsic, G., Winterhof, A.: On the structure of digital explicit nonlinear and inversive pseudorandom number generators. Journal of Complexity, to appear, 2009.
17. Topuzoğlu, A., Winterhof, A.: Pseudorandom sequences. In: Topics in Geometry, Coding Theory and Cryptography. Algebra and Applications, vol. 6, pp. 135–166, Springer, Dordrecht (2007)
18. Winterhof, A.: Some estimates for character sums and applications. Designs, Codes and Cryptography 22(2), 123–131 (2001)
19. Winterhof, A.: On the distribution of some new explicit inversive pseudorandom numbers and vectors. In: Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 487–499, Springer, Berlin Heidelberg (2006)

Extensions of Fibonacci Lattice Rules

Ronald Cools and Dirk Nuyens

Abstract We study the trigonometric degree of pairs of embedded cubature rules for the approximation of two-dimensional integrals, where the basic cubature rule is a Fibonacci lattice rule. The embedded cubature rule is constructed by simply doubling the points which results in adding a shifted version of the basic Fibonacci rule. An explicit expression is derived for the trigonometric degree of this particular extension of the Fibonacci rule based on the index of the Fibonacci number.

Dedicated to Ian Sloan's 70th birthday.

1 Introduction

We consider the approximation of integrals

$$I[f] := \int_{[0,1)^s} f(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

by weighted sums of function values. In practice one wants more than one such approximation to obtain information on the accuracy of the approximation. In order to approximate an integral together with an error estimate, one often uses two approximations $Q_1[f]$ and $Q_2[f]$. Then $|Q_1[f] - Q_2[f]|$ can be used as an approximation of the error of the less precise rule. (In practice it is often used as an estimate of the

Ronald Cools
Department of Computer Science, K.U.Leuven, Leuven, Belgium
e-mail: ronald.cools@cs.kuleuven.be
url: http://www.cs.kuleuven.be/~ronald

Dirk Nuyens
Department of Computer Science, K.U.Leuven, Leuven, Belgium, and School of Mathematics & Statistics, University of NSW, Sydney, Australia
e-mail: dirk.nuyens@cs.kuleuven.be
url: http://www.cs.kuleuven.be/~dirkn

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_15,

error of the most expensive rule, while hoping that this is the most precise. Practical robust error estimates are based on more than one such combinations.)

Given a cubature formula

$$Q_1[f] = \sum_{j=1}^{N} w_j \, f(\mathbf{x}_j)$$

for the approximation of an integral $I[f]$, we are interested in a cubature formula

$$Q_2[f] = \sum_{j=1}^{N} \bar{w}_j \, f(\mathbf{x}_j) + \sum_{j=N+1}^{N+M} \bar{w}_j \, f(\mathbf{x}_j)$$

that reuses the function evaluations of Q_1 and is "better". The quality criterion used in this paper is the *trigonometric degree*, and the cubature formulas will be lattice rules. For other criteria, see [4].

A cubature formula of trigonometric degree d integrates correctly trigonometric polynomials of degree d. Specifically in s dimensions, it integrates $\exp(2\pi \mathrm{i}\, \mathbf{r} \cdot \mathbf{x})$ correctly for all $\mathbf{r} = (r_1, r_2, \ldots, r_s) \in \mathbf{Z}^s$ that satisfy $|\mathbf{r}| := \sum_{k=1}^{s} |r_k| \leq d$.

Embedded pairs of quadrature and cubature formulas of algebraic degree were already studied a long time ago (see, e.g., [6, 2, 3]). We are unaware of an attempt to do this for the trigonometric case. This paper describes a first attempt, limited to the 2-dimensional case. More examples of this type might eventually lead to similar theoretical insights as in the algebraic case.

In the following section we present the necessary background and notation in s dimensions. In §3 we will present the well known class of 2-dimensional Fibonacci lattice rules and known results on their trigonometric degree. In §4 we will investigate a particular extension of these rules to obtain an embedded pair and in §5 we compare their quality with what is theoretically the best possible result.

2 A Short Course on Lattice Rules

For a thorough introduction on lattice rules, we refer tot [9]. Results on the trigonometric degree of lattice rules up to that date were mainly published in the Russian literature and summarized in [1].

Definition 1. A multiple integration lattice Λ in $\mathbf{R}^s$ is a subset of $\mathbf{R}^s$ which is discrete and closed under addition and subtraction and which contains $\mathbf{Z}^s$ as a subset.

A lattice rule for approximating an integral over $[0,1)^s$ is a cubature formula where the N points are the points of a multiple integration lattice Λ that lie in $[0,1)^s$ and all points have the same weight $1/N$.

Definition 2. The dual of the lattice Λ is $\Lambda^\perp := \{\mathbf{r} \in \mathbf{R}^s : \mathbf{r} \cdot \mathbf{x} \in \mathbf{Z}, \forall \mathbf{x} \in \Lambda\}$.

A lattice Λ can be specified by an $s \times s$ matrix M known as a *generator matrix*, whose rows generate the lattice. This means that all elements of Λ are of the form

$\mathbf{x} = \lambda M$, where $\lambda \in \mathbf{Z}^s$. The dual lattice $\Lambda^\perp$ then has generator matrix $B = (M^{-1})^T$. Since Λ is an integration lattice, its dual $\Lambda^\perp$ is an integer lattice and is generated by an integer-valued matrix B.

The dual of a multiple integration lattice plays an important role in the error representation and is the main tool to prove our results. Assume that f can be expanded into an absolutely convergent multiple Fourier series

$$f(\mathbf{x}) = \sum_{\mathbf{r}\in\mathbf{Z}^s} a(\mathbf{r})\, \mathrm{e}^{2\pi \mathrm{i}\, \mathbf{r}\cdot\mathbf{x}} \text{ with } a(\mathbf{r}) = \int_{[0,1)^s} \mathrm{e}^{-2\pi \mathrm{i}\, \mathbf{r}\cdot\mathbf{x}} f(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

then the error is given by the next theorem. Note that this assumption limits the functions to be 1-periodic in each dimension.

Theorem 1. *[10] Let Λ be a multiple integration lattice. Then the corresponding lattice rule Q has an error*

$$Q[f] - I[f] = \sum_{\mathbf{r}\in\Lambda^\perp\setminus\mathbf{0}} a(\mathbf{r}).$$

The trigonometric degree of a lattice rule can be determined from the dual lattice:

$$d(Q) := \min_{\mathbf{r}\in\Lambda^\perp\setminus\mathbf{0}} |\mathbf{r}| - 1\,.$$

There is a diamond shaped region (a crosspolytope to be precise) with no points of the dual lattice except the origin inside, and some points on its boundary. The 1-norm of points on the boundary is $d+1$.

Central symmetry (of the integration region and the points in the cubature formulas) plays an important role in the algebraic case. If this symmetry is present, then the cubature formula integrates the odd polynomials exactly automatically. The lower bound for the number of points required is also fundamentally different for the even and the odd degrees. The role that central symmetry plays in the algebraic case is played by *shift symmetry* in the trigonometric case [5].

Definition 3. A cubature formula Q for an integral I on $[0,1)^s$ is shift symmetric if, whenever $(x_1^{(j)}, \ldots, x_s^{(j)})$ is a point of the formula, then so is $(x_1^{(j)} + \frac{1}{2}, \ldots, x_s^{(j)} + \frac{1}{2})$, with both points having the same weight.

The point $(x_1^{(j)} + \frac{1}{2}, \ldots, x_s^{(j)} + \frac{1}{2})$ in the above theorem should actually be interpreted modulo 1, i.e., wrapped around the edges of the unit cube. However, since we are assuming periodic functions we loosen the notation and do not write the traditional fractional braces around the points to denote the modulo 1.

From Definition 3 it follows that N is even for a shift symmetric cubature formula. Furthermore, because a shift symmetric cubature formulas is automatically exact for all trigonometric monomials of odd degree, such a cubature formula has an odd trigonometric degree, see [5]. Finally note that a lattice rule is shift symmetric if and only if $(\frac{1}{2}, \ldots, \frac{1}{2})$ is a point of the lattice.

3 Fibonacci Lattice Rules

We will restrict our investigations in this paper to two dimensions, starting from a well known family of lattice rules. Let F_k be the kth Fibonacci number, defined by $F_0 := 0$, $F_1 := 1$ and $F_k := F_{k-1} + F_{k-2}$ for $1 < k \in \mathbf{N}$. Consider the following lattice rules:

$$\mathcal{L}_k[f] = \frac{1}{F_k} \sum_{j=0}^{F_k-1} f\left(\frac{j}{F_k}, \frac{j F_{k-1}}{F_k}\right) \tag{1}$$

and

$$\mathcal{L}'_k[f] = \frac{1}{F_k} \sum_{j=0}^{F_k-1} f\left(\frac{j}{F_k}, \frac{j F_{k-2}}{F_k}\right).$$

Lattice rules of this form are called Fibonacci lattice rules. The two rules given above are *geometrically equivalent*, meaning that one point set can be changed into the other one by symmetry operations of the unit cube [9]. In this case a reflection on the second coordinate axis maps $\mathcal{L}_k$ to $\mathcal{L}'_k$ since $j F_{k-1} \equiv -j F_{k-2} \pmod{F_k}$. Geometrically equivalent lattice rules have the same trigonometric degree, as is obvious from their dual lattices.

The kth Fibonacci number F_k is even if and only if k is a multiple of 3. This can be observed by looking at the Fibonacci sequence modulo 2: $(F_k \bmod 2)_{k\geq 1} = (1, 1, 0, 1, 1, 0, 1, 1, 0, \ldots)$, i.e., the 2nd Pisano period is 3. Only in these cases are the Fibonacci lattice rules shift symmetric. Indeed, the point of $\mathcal{L}_k$ generated by $j = F_k/2$ is $\left(\frac{1}{2}, \frac{F_{k-1}}{2}\right)$ and this then maps to $\left(\frac{1}{2}, \frac{1}{2}\right)$.

The trigonometric degree of Fibonacci lattice rules is known explicitly.

Theorem 2. *[1] If $k = 2m+1$ and $m \geq 2$ then the Fibonacci lattice rule has trigonometric degree $F_{m+2} - 1$. If $k = 2m$ then the Fibonacci lattice rule has trigonometric degree $2F_m - 1$.*

Shift symmetric lattice rules have an odd trigonometric degree [5]. The converse is not always true. A rule of odd trigonometric degree is not necessarily shift symmetric. The family of Fibonacci lattice rules has many examples of this.

The proof in [1] of the above theorem is based on the dual lattice. We sketch it here because we will use the same technique in §4 and because it reveals a structure in the lattice. Starting from the evident generator matrix, e.g., for $k = 2m+1$

$$M = \begin{pmatrix} \frac{1}{F_{2m+1}} & \frac{F_{2m}}{F_{2m+1}} \\ 0 & 1 \end{pmatrix}$$

it follows that $B = (M^{-1})^T = \begin{pmatrix} F_{2m+1} & 0 \\ -F_{2m} & 1 \end{pmatrix}$ is a generator matrix for the dual lattice. When U is any unimodular matrix then UB is also a generator matrix. (A unimodular matrix is a square integer matrix with determinant $+1$ or -1.) Using the unimodular matrix $U = \begin{pmatrix} F_m & F_{m+1} \\ -F_{m-1} & -F_m \end{pmatrix}$, it is shown that

$$B = UA \text{ with } A = \begin{pmatrix} F_m & (-1)^{m+1} F_{m+1} \\ F_{m+1} & (-1)^m F_m \end{pmatrix}. \tag{2}$$

Hence, A is also a generator matrix of this dual lattice. Then, it is proven that no nonzero combination of the rows of this matrix leads to a point with 1-norm smaller than the claimed degree.

Observe that the generator matrix A (2) of the dual lattice is an orthogonal matrix (modulo $F_k = F_{2m+1} = F_m^2 + F_{m+1}^2$). In other words, both generating vectors are orthogonal and have the same length (in 2-norm). So the Fibonacci lattice for odd k has a square unit cell, i.e., this lattice corresponds to a rotated regular grid. This fact was obtained in a different way, and explicitly recognized by Niederreiter and Sloan [8].

4 An Extension of Fibonacci Lattice Rules

We are interested in lattice rules that extend Fibonacci lattice rules so that we obtain an embedded pair of lattice rules. The aim is to obtain a pair that requires less function evaluations than two rules of the corresponding degrees. We are partially successful in that respect.

Consider the lattice rule

$$Q_k[f] = \frac{1}{2F_k} \sum_{j=0}^{2F_k-1} f\left(\frac{j}{2F_k}, \frac{jF_{k-1}}{2F_k}\right). \tag{3}$$

This lattice is shift symmetric if F_{k-1} is odd. Indeed for $j = F_k$ the point $\left(\frac{1}{2}, \frac{F_{k-1}}{2}\right)$ is generated. Whenever F_{k-1} is odd, this maps to $\left(\frac{1}{2}, \frac{1}{2}\right)$.

Next we will show that (3) can be seen as a Fibonacci lattice $\mathcal{L}_k$ plus a shifted version of the same Fibonacci lattice. The above rule can be split in two sums, separating even and odd values of j:

$$\begin{aligned} Q_k[f] &= \frac{1}{2F_k} \sum_{j=0}^{F_k-1} f\left(\frac{j}{F_k}, \frac{jF_{k-1}}{F_k}\right) + \frac{1}{2F_k} \sum_{j=0}^{F_k-1} f\left(\frac{2j+1}{2F_k}, \frac{(2j+1)F_{k-1}}{2F_k}\right) \\ &= \frac{1}{2F_k} \sum_{j=0}^{F_k-1} f\left(\frac{j}{F_k}, \frac{jF_{k-1}}{F_k}\right) + \frac{1}{2F_k} \sum_{j=0}^{F_k-1} f\left(\frac{j}{F_k} + \frac{1}{2F_k}, \frac{jF_{k-1}}{F_k} + \frac{F_{k-1}}{2F_k}\right). \end{aligned}$$

Thus the lattice rule (3) is given by the original rule (1) plus the original rule shifted by $(1, F_{k-1})/(2F_k)$. (The weights are adjusted in the obvious way.) We will now show a formula for its trigonometric degree.

Theorem 3. *The lattice rule given by* (3) *for* $k = 6m + \alpha$, *with* $\alpha = 0, 1, \ldots, 5$, *has trigonometric degree* $F_{3m+\alpha-1} + F_{3m+1} - 1$.

Proof. The proof technique we use here was sketched at the end of §3.

The cubature formula (3) with $k = 6m + \alpha$ is a lattice rule with generator matrix of the corresponding lattice:

$$M = \begin{pmatrix} \frac{1}{2F_{6m+\alpha}} & \frac{F_{6m+\alpha-1}}{2F_{6m+\alpha}} \\ 0 & 1 \end{pmatrix}.$$

We will instead investigate the minor modification of this matrix

$$M = \begin{pmatrix} \frac{1}{2F_{6m+\alpha}} & \frac{(-1)^m F_{6m+\alpha-1}}{2F_{6m+\alpha}} \\ 0 & (-1)^m \end{pmatrix}.$$

If m is even, this is the same. If m is odd, we investigate a geometrically equivalent lattice. As mentioned before, geometrically equivalent lattice rules have the same trigonometric degree.

A generator matrix of the corresponding dual lattice is

$$B = (M^{-1})^T = \begin{pmatrix} 2F_{6m+\alpha} & 0 \\ -F_{6m+\alpha-1} & (-1)^m \end{pmatrix}.$$

Observe that the matrix

$$C = \begin{pmatrix} F_{3m+1} & -2F_{3m} \\ -F_{3m}/2 & F_{3m-1} \end{pmatrix}$$

is an integer unimodular matrix. Indeed, we observed in §3 that F_{3m} is an even number and, making use of Cassini's identity, see (5) below, it follows that

$$\det(C) = F_{3m+1}F_{3m-1} - F_{3m}^2 = (-1)^{3m} = (-1)^m.$$

We will first show that

$$A = \begin{pmatrix} 2F_{3m+\alpha} & 2F_{3m} \\ -F_{3m+\alpha-1} & F_{3m+1} \end{pmatrix} \tag{4}$$

is another generator matrix for the dual lattice. This follows from $CA = B$. It requires some Fibonacci magic to show this. The relevant relations for the sequel are [7]

$$\begin{aligned} F_k F_{n+1} + F_{k-1} F_n &= F_{n+k}, \\ F_{n+1} F_{n-1} - F_n^2 &= (-1)^n. \end{aligned} \tag{5}$$

It follows that

$$\begin{aligned} CA &= \begin{pmatrix} 2F_{3m+1}F_{3m+\alpha} + 2F_{3m}F_{3m+\alpha-1} & 2F_{3m+1}F_{3m} - 2F_{3m}F_{3m+1} \\ -F_{3m}F_{3m+\alpha} - F_{3m-1}F_{3m+\alpha-1} & -F_{3m}^2 + F_{3m-1}F_{3m+1} \end{pmatrix} \\ &= \begin{pmatrix} 2F_{3m+1}F_{3m+\alpha} + 2F_{3m}F_{3m+\alpha-1} & 0 \\ -F_{3m}F_{3m+\alpha} - F_{3m-1}F_{3m+\alpha-1} & (-1)^{3m} \end{pmatrix} = B. \end{aligned}$$

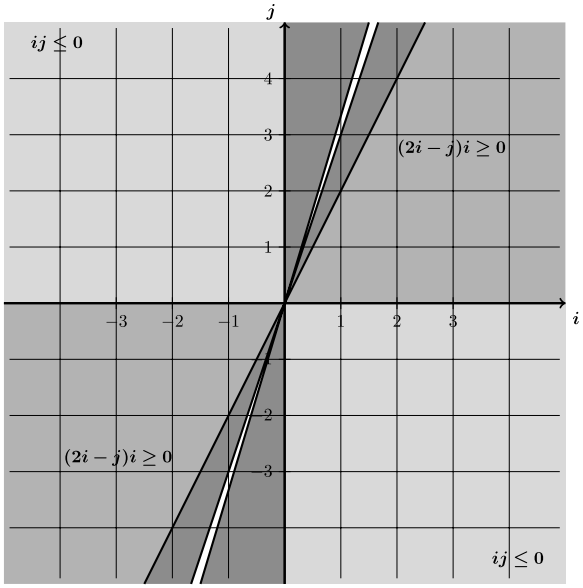


Fig. 1 The different regions to consider for (i, j) when taking linear combinations of the rows of the dual matrix A given by (4).

The points on the dual lattice are generated as

$$(i,\ j)\,A = (i2F_{3m+\alpha} - jF_{3m+\alpha-1},\ i2F_{3m} + jF_{3m+1})$$

for all $i, j \in \mathbf{Z}$. The corresponding lattice rule has trigonometric degree $F_{3m+\alpha-1} + F_{3m+1} - 1$ if all points of the dual lattice, except the point $(0, 0)$, lie outside or on the boundary of the diamond shaped region:

$$|i2F_{3m+\alpha} - jF_{3m+\alpha-1}| + |i2F_{3m} + jF_{3m+1}| \geq F_{3m+\alpha-1} + F_{3m+1}. \tag{6}$$

(Some points fall exactly on the boundary, e.g., when $i = 0$ and $j = 1$.) We prove this inequality for different cases of (i, j). For brevity we name parts of the inequality as follows

$$\begin{aligned}\beta(\alpha) &:= |i2F_{3m+\alpha} - jF_{3m+\alpha-1}|,\\ \delta &:= |i2F_{3m} + jF_{3m+1}|,\\ \gamma(\alpha) &:= F_{3m+\alpha-1} + F_{3m+1},\end{aligned}$$

so that we have to prove for all $\alpha = 0, 1, \ldots, 5$: $\beta(\alpha) + \delta \geq \gamma(\alpha)$. This is a tedious exercise in proving the bound for all different cases. For reference the reader is referred to Fig. 1 on different occasions during the proof, which depicts the possible integer values that the (i, j) can take in generating the dual lattice points.

1. If $i = 0$ then $\beta(\alpha) + \delta = |j|\,(F_{3m+\alpha-1} + F_{3m+1}) \geq F_{3m+\alpha-1} + F_{3m+1}$ since we must have $|j| \geq 1$.
2. If $j = 0$ then $\beta(\alpha) + \delta = |i|\,(2F_{3m+\alpha} + 2F_{3m}) \geq F_{3m+\alpha-1} + F_{3m+1}$ since we must have $|i| \geq 1$ and $2F_{3m} \geq F_{3m} + F_{3m-1} = F_{3m+1}$.
3. If $ij < 0$ (i.e., opposite signs) then $\beta(\alpha) = |i|\,2F_{3m+\alpha} + |j|\,F_{3m+\alpha-1} \geq F_{3m+\alpha-1} + F_{3m+1}$ since $2F_{3m+\alpha} \geq F_{3m+\alpha} + F_{3m+\alpha-1} = F_{3m+\alpha+1} \geq F_{3m+1}$ for all $\alpha \geq 0$.

   We have now checked (and crossed out) the 2nd and 4th quadrant, marked with $ij \leq 0$ on Fig. 1, to fulfill (6).

4. For $ij > 0$ (i.e., same signs) we first look at

$$\begin{aligned}\beta(\alpha) &= |i2F_{3m+\alpha} - jF_{3m+\alpha-1}| \\ &= |(2i - j)F_{3m+\alpha-1} + 2iF_{3m+\alpha-2}|.\end{aligned}$$

   Again here we consider separate cases based on the signs of the two terms. But note first that $\delta = |i|\,2F_{3m} + |j|\,F_{3m+1} \geq 2F_{3m} + F_{3m+1}$, so we always trivially have the F_{3m+1} term from $\gamma(\alpha)$ and we can close a case quickly if $\beta(\alpha) \geq F_{3m+\alpha-1}$. Also note that we can get extra terms of F_{3m} and F_{3m+1} by having larger bounds on respectively $|i|$ and $|j|$.

   a. If $2i - j = 0$ then $\beta(\alpha) = 2|i|\,F_{3m+\alpha-2} \geq 2F_{3m+\alpha-2} \geq F_{3m+\alpha-1}$.
   b. If $(2i - j)\,i > 0$ (i.e., same signs) then $\beta(\alpha) = |2i - j|\,F_{3m+\alpha-1} + 2|i|\,F_{3m+\alpha-2} \geq F_{3m+\alpha-1}$.

      At this time we have checked the largest parts of the 1st and 3rd quadrant, marked with $(2i - j)i \geq 0$ on Fig. 1, to fulfill (6) as well.

   c. If $(2i - j)\,i < 0$ (i.e., opposite signs) then we have more work. First observe that from $(2i - j)\,i < 0$ we can conclude that $|j| > 2$, i.e., $|j| \geq 3$. (This is also visible on the figure.) We can thus refine our estimate for δ in this case to

$$\begin{aligned}\delta &= |i2F_{3m} + jF_{3m+1}| \\ &= |i|\,2F_{3m} + |j|F_{3m+1} \\ &\geq 2F_{3m} + 3F_{3m+1} \\ &= 2F_{3m+2} + F_{3m+1} \\ &= F_{3m+2} + F_{3m+3} \\ &= F_{3m+4}.\end{aligned}$$

      (The different expressions become useful in the following.)
      Now we consider the cases for the different values of α since we can easily obtain the required result by filling in the values, except for the case $\alpha = 5$ where more work is needed.
      i. If $\alpha = 0, 1, 2, 3, 4$ then we get the following results:

α	$\gamma(\alpha) = F_{3m+\alpha-1} + F_{3m+1}$
0	$F_{3m-1} + F_{3m+1}$
1	$F_{3m} + F_{3m+1} = F_{3m+2}$
2	$F_{3m+1} + F_{3m+1} = 2F_{3m+1}$
3	$F_{3m+2} + F_{3m+1} = F_{3m+3}$
4	$F_{3m+3} + F_{3m+1} = F_{3m+2} + 2F_{3m+1}$

We note that for all of these values of α we have that $\delta \geq \gamma(\alpha)$ (using the refined estimate for δ for all conditions we have set).

ii. If $\alpha = 5$ then

$$\begin{aligned} \beta(5) &= |i2F_{3m+5} - jF_{3m+4}|, \\ \delta &= F_{3m+4}, \\ \gamma(5) &= F_{3m+4} + F_{3m+1}, \end{aligned}$$

where we thankfully use the refined value of δ to only have to show $\beta(5) \geq F_{3m+1}$. We can rewrite $\beta(5)$ as follows

$$\begin{aligned} \beta(5) &= |i2F_{3m+5} - jF_{3m+4}| \\ &= |(10i - 3j)F_{3m+1} + 2(3i - j)F_{3m}|. \end{aligned}$$

Set $a = 10i - 3j$ and $b = 3i - j$ and consider the different sign settings.

A. If $a = 0$ then we need integer solutions of $i = \frac{3}{10}j$. This means $|j| \geq 10$ (and $|i| \geq 3$) and we can make a new refinement for δ, $\delta \geq 6F_{3m} + 10F_{3m+1}$, which solves this case since we have $(2F_{3m} + 3F_{3m+1}) + F_{3m+1} = F_{3m+4} + F_{3m+1}$.

B. If $b = 0$ then we obtain $j = 3i$ and as such $a = 10i - 9i = i$ from which it follows that $\beta(5) = F_{3m+1}$. I.e., for $i = 1$ and $j = 3$ we get a point on the boundary.

C. If $ab > 0$ then the result is trivial as clearly then $\beta(5) \geq F_{3m+1} + 2F_{3m}$.

We now have checked the darkest marked part on Fig. 1 to fulfill (6) and are left with the narrow white wedge in between $j = \frac{10}{3}i$ and $j = 3i$.

D. If $ab < 0$ then we either have

$$\begin{cases} 10i - 3j < 0 \\ 6i - 2j > 0 \end{cases} \quad \text{or} \quad \begin{cases} 10i - 3j > 0 \\ 6i - 2j < 0 \end{cases}$$
$$\Leftrightarrow \tfrac{10}{3}i < j < 3i \qquad \Leftrightarrow 3i < j < \tfrac{10}{3}i.$$

From the inequality on the left side follows that both i and j should be negative, while the inequality on the right side makes it that both i and j are positive. Combined we can write:

$$3|i| < |j| < \tfrac{10}{3}|i|.$$

This means $|j| \geq 4$ (and is easily checked on the figure). Refining the estimate of δ we get

$$\begin{aligned}\delta &= |i|2F_{3m} + |j|F_{3m+1} \\ &\geq 2F_{3m} + 4F_{3m+1} \\ &= (2F_{3m} + 3F_{3m+1}) + F_{3m+1} \\ &= F_{3m+4} + F_{3m+1},\end{aligned}$$

and thus $\delta \geq \gamma(5)$.

This completes our proof. □

The result of the theorem is put together in Table 1. It is clear from the table that the cases for $\alpha = 1, 2, 3$ are not interesting. The extended rules have the same degree as the original rules, and the number of additional points is the same as for the original rule. In other words, using the embedded rules costs the same as using two rules. In the next section we will show that the remaining cases are however interesting.

Table 1 Trigonometric degrees of Q_k for the different cases of Theorem 3 compared to the degree of the basic rule $\mathcal{L}_k$.

k mod 6	k mod 2	k mod 3	$d(\mathcal{L}_k)$	$d(Q_k)$	comparison	
0	0	0	$2F_{3m} - 1$	$2F_{3m} + F_{3m-3} - 1$	higher	*
1	1	1	$F_{3m+2} - 1$	$F_{3m+2} - 1$	equal	
2	0	2	$2F_{3m+1} - 1$	$2F_{3m+1} - 1$	equal	
3	1	0	$F_{3m+3} - 1$	$F_{3m+3} - 1$	equal	
4	0	1	$2F_{3m+2} - 1$	$2F_{3m+2} + F_{3m-1} - 1$	higher	**
5	1	2	$F_{3m+4} - 1$	$F_{3m+4} + F_{3m+1} - 1$	higher	

Note that the line marked with a star (*) includes $k = 6$, i.e., $m = 1$ and $\alpha = 0$. In this exceptional case $d(\mathcal{L}_k) = d(Q_k)$ since then $F_{3m-3} = F_0 = 0$. For all other values of $k \equiv 0 \pmod 6$ the rule Q_k does have a higher degree than the basic rule $\mathcal{L}_k$. For the double starred line (**) there is no such problem if one allows negative indices for the Fibonacci numbers (see, e.g., [7]). For $k = 4$, i.e., $m = 0$ and $\alpha = 4$, we then have $d(Q_4) = F_3 + F_1 - 1 = 2$, while $d(\mathcal{L}_4) = 2F_2 - 1 = 1$.

Initially, we recognised the five different cases, and we attempted proofs for each α using a generator matrix that suited each case best. There are indeed other useful generator matrices of the dual lattice. We can, e.g., rewrite the matrix A used in the previous proof:

$$A = \begin{pmatrix} 2F_{3m+\alpha} & 2F_{3m} \\ -F_{3m+\alpha-1} & F_{3m+1} \end{pmatrix} = \begin{pmatrix} F_{3m+2+\alpha} - F_{3m-1+\alpha} & F_{3m+2} - F_{3m-1} \\ -F_{3m+\alpha-1} & F_{3m+1} \end{pmatrix}.$$

This can be transformed as follows

$$\begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} A = \begin{pmatrix} F_{3m+2+\alpha} & F_{3m-2} \\ -F_{3m+\alpha-1} & F_{3m+1} \end{pmatrix}.$$

For specific values of α this generator matrix is more convenient.

Of special interest is the case $\alpha = 5$. It causes extra difficulties in the general proof given above. If treated separately, that part of the proof becomes easier. The difficult part is to show that the generator matrix B can be transformed to

$$A = \begin{pmatrix} F_{3m+1} & F_{3m+\alpha-1} \\ -F_{3m+\alpha-1} & F_{3m+1} \end{pmatrix}$$

by a unimodular matrix if $\alpha = 5$. With hindsight, this matrix was guessed after the general proof was established. Once this is established, proving the degree of the lattice rule is straightforward. This case is special because now there are four points (instead of two) of the dual lattice, one in each quadrant, lying on the boundary. This lattice has a square unit cell. This special structure is immediately evident from the matrix given above, but not from the matrix used in the general proof.

5 Final Remarks

Not all pairs of embedded cubature rules are interesting from a practical point of view. Let N_d^{opt} denote the number of points used by an optimal rule of degree d, i.e., one with the lowest number of points. If we have two embedded rules of respectively degree d_1 and d_2 with N the total number of points, and $d_1 < d_2$, then a measure for its quality is

$$\gamma := \frac{N}{N_{d_1}^{\text{opt}} + N_{d_2}^{\text{opt}}}.$$

Obviously, we prefer rules with $\gamma < 1$.

Lower bounds on the number of points needed to achieve a given trigonometric degree are known (see, e.g., [1, 5, 4]). In two dimensions the lower bound for a degree d rule is given by

$$N \geq N_d^{\text{opt}} := \begin{cases} 2m^2 + 2m + 1, & \text{for } d = 2m, \\ 2(m+1)^2, & \text{for } d = 2m+1. \end{cases}$$

Furthermore, lattice rules are known that attain this lower bound (see, e.g., [1, 5]).

In Table 2 it can be seen that for the cases $k = 6m$, $6m+4$ and $6m+5$ investigated above, we have that $\gamma < 1$. The data in Table 2 was computationally verified completely up to $k = 54$.

Acknowledgements The authors would like to thank Stephen Joe for commenting on a draft version of the manuscript. The second author acknowledges support of the Australian Research Council and the Research Foundation Flanders. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

Table 2 Comparison for the trigonometric degrees of the embedded rule Q_k based on the Fibonacci rule $\mathcal{L}_k$ with the lower bounds for the same degrees. Here $\alpha = k \bmod 6$, $d_1 = d(\mathcal{L}_k)$, $N = 2F_k$ and $d_2 = d(Q_k)$. For $k = 6$ the value of γ should be ignored since there $d_1 = d_2$.

k	α	F_k	d_1	$N_{d_1}^{\text{opt}}$	N	d_2	$N_{d_2}^{\text{opt}}$	$N_{d_1}^{\text{opt}} + N_{d_2}^{\text{opt}}$	γ
4	4	3	1	2	6	2	5	7	0.857
5	5	5	2	5	10	3	8	13	0.769
6	0	8	3	8	16	3	8	16	1.000
10	4	55	9	50	110	10	61	111	0.991
11	5	89	12	85	178	15	128	213	0.836
12	0	144	15	128	288	17	162	290	0.993
16	4	987	41	882	1974	46	1105	1987	0.993
17	5	1597	54	1513	3194	67	2312	3825	0.835
18	0	2584	67	2312	5168	75	2888	5200	0.994
22	4	17711	177	15842	35422	198	19801	35643	0.994
23	5	28657	232	27145	57314	287	41472	68617	0.835
24	0	46368	287	41472	92736	321	51842	93314	0.994
28	4	317811	753	284258	635622	842	355325	639583	0.994
29	5	514229	986	487085	1028458	1219	744200	1231285	0.835
30	0	832040	1219	744200	1664080	1363	930248	1674448	0.994
34	4	5702887	3193	5100818	11405774	3570	6376021	11476839	0.994
35	5	9227465	4180	8740381	18454930	5167	13354112	22094493	0.835
36	0	14930352	5167	13354112	29860704	5777	16692642	30046754	0.994
40	4	102334155	13529	91530450	204668310	15126	114413065	205943515	0.994
41	5	165580141	17710	156839761	331160282	21891	239629832	396469593	0.835
42	0	267914296	21891	239629832	535828592	24475	299537288	539167120	0.994
46	4	1836311903	57313	1642447298	3672623806	64078	2053059121	3695506419	0.994
47	5	2971215073	75024	2814375313	5942430146	92735	4299982848	7114358161	0.835
48	0	4807526976	92735	4299982848	9615053952	103681	5374978562	9674961410	0.994
52	4	32951280099	242785	29472520898	65902560198	271442	36840651125	66313172023	0.994
53	5	53316291173	317810	50501915861	106632582346	392835	77160061448	127661977309	0.835
54	0	86267571272	392835	77160061448	172535142544	439203	96450076808	173610138256	0.994

References

1. Beckers, M., Cools, R.: A relation between cubature formulae of trigonometric degree and lattice rules. In: H. Brass, G. Hämmerlin (eds.) Numerical Integration IV, pp. 13–24. Birkhäuser Verlag, Basel (1993)
2. Cools, R., Haegemans, A.: Optimal addition of knots to cubature formulae for planar regions. Numer. Math. **49**, 269–274 (1986)
3. Cools, R., Haegemans, A.: A lower bound for the number of function evaluations in an error estimate for numerical integration. Constr. Approx. **6**, 353–361 (1990)
4. Cools, R., Nuyens, D.: A Belgian view on lattice rules. In: A. Keller, et al. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 3–21. Springer (2008)
5. Cools, R., Sloan, I.: Minimal cubature formulae of trigonometric degree. Math. Comp. **65**(216), 1583–1600 (1996)
6. Davis, P., Rabinowitz, P.: Methods of Numerical Integration. Academic Press, London (1984)
7. Graham, R.L., Knuth, D.E., Patashnik, O.: Concrete Mathematics, 2nd edn. Addison-Wesley (1994)
8. Niederreiter, H., Sloan, I.: Integration of nonperiodic functions of two variables by Fibonacci lattice rules. J. Comput. Appl. Math. **51**, 57–70 (1994)
9. Sloan, I., Joe, S.: Lattice Methods for Multiple Integration. Oxford University Press (1994)
10. Sloan, I., Kachoyan, P.: Lattice methods for multiple integration: theory, error analysis and examples. SIAM J. Numer. Anal. **24**, 116–128 (1987)

Efficient Search for Two-Dimensional Rank-1 Lattices with Applications in Graphics

Sabrina Dammertz, Holger Dammertz, and Alexander Keller

Abstract Selecting rank-1 lattices with respect to maximized mutual minimum distance has been shown to be very useful for image representation and synthesis in computer graphics. While algorithms using rank-1 lattices are very simple and efficient, the selection of their generator vectors often has to resort to exhaustive computer searches, which is prohibitively slow. For the two-dimensional setting, we introduce an efficient approximate search algorithm and transfer the principle to the search for maximum minimum distance rank-1 lattice sequences. We then extend the search for rank-1 lattices to approximate a given spectrum and present new algorithms for anti-aliasing and texture representation in computer graphics.

1 Introduction

Due to their algorithmic efficiency, rank-1 lattices [17, 20] and rank-1 lattice sequences [9, 10] are very interesting objects for computer graphics [3, 4]: The n points $\mathbf{x}_i$ of an s-dimensional rank-1 lattice

$$L_{n,\mathbf{g}} := \left\{ \mathbf{x}_i := \frac{i}{n}\mathbf{g} \bmod 1 : i = 0, \ldots, n-1 \right\} \subset [0,1)^s \tag{1}$$

are generated by a suitable vector $\mathbf{g} \in \mathbb{N}^s$. Rank-1 lattices $L_{n,a}$ in Korobov form [20] use generator vectors of the restricted form $\mathbf{g} = (1, a, a^2, \ldots, a^{s-1})$.

Sabrina Dammertz
Institute of Media Informatics, Ulm University, Germany
e-mail: sabrina.dammertz@uni-ulm.de

Holger Dammertz
Institute of Media Informatics, Ulm University, Germany
e-mail: holger.dammertz@uni-ulm.de

Alexander Keller
mental images GmbH, Berlin, Germany
e-mail: alex@mental.com

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_16,

Using a van der Corput sequence (radical inverse) Φ_b in base b [17] instead of the fraction $\frac{i}{n}$ extends rank-1 lattices to rank-1 lattice sequences

$$L_{\mathbf{g}}^{\Phi_b} := \{\mathbf{x}_i := \Phi_b(i)\cdot\mathbf{g} \bmod 1 : i \in \mathbb{N}_0\} \subset [0,1)^s \tag{2}$$

in the sense that for any $m \in \mathbb{N}_0$ the first b^m points $\mathbf{x}_0, \ldots, \mathbf{x}_{b^m-1}$ are a rank-1 lattice $L_{b^m,\mathbf{g}}$ [10]. Thereby the van der Corput sequence Φ_b mirrors the b-ary representation of an integer i at the decimal point

$$\begin{aligned} \Phi_b(i) : \mathbb{N}_0 &\longrightarrow \mathbb{Q}\cap[0,1) \\ i = \sum_{j=0}^{\infty} a_j(i) b^l &\longmapsto \sum_{j=0}^{\infty} a_j(i) b^{-j-1}, \end{aligned} \tag{3}$$

where $a_j(i)$ denotes the j-th digit of the integer i represented in base b.

In [4] we investigated the concept of maximized minimum distance (MMD) rank-1 lattices with applications to image synthesis and representation. Since lattices are closed under addition and subtraction, the minimum distance

$$d_{\min}(L_{n,\mathbf{g}}) := \min_{0<i<n} \|\mathbf{x}_i\| \tag{4}$$

of a rank-1 lattice $L_{n,\mathbf{g}}$ is determined by the minimum norm of the lattice points themselves. In this paper we use the L_2-norm on the unit torus unless noted otherwise.

Algorithms for computing the shortest vector in a general lattice have been developed in [5, 7, 12] and efficient implementations exist even for higher dimensions [14]. Specializing the setting to rank-1 lattices in two dimensions allows one to take simpler approaches, as for example the Gaussian basis reduction [11, 18]. This basis reduction is a simple algorithm to efficiently determine a lattice basis where the first basis vector is the shortest vector in the lattice and thus yields its minimum distance. In two dimensions, this algorithm computes a Minkowski-reduced basis and has a computational complexity of $\mathcal{O}(\log n)$ which is sufficient for our application [11].

The problem of constructing lattices with longest possible shortest nonzero vectors for a given lattice density is connected to the problem of finding the densest packing of spheres which has been studied for a long time [1, 16, 19]. Computer searches for good lattices based on the lengths of shortest nonzero vectors have been reported in [13, 15] for example. They focus on the dual lattice, though, and use either exhaustive or random searches, the latter of which poses the problem of deciding how much time to spend on the search process. Due to the low-dimensional structure of many graphics applications, we will consider only $s = 2$ dimensions henceforth. However, the number of potential generator vectors for the number n of points required in graphics applications is so large that a naïve search algorithm for MMD rank-1 lattices as well as tables become prohibitive in time and space. For image storage or sampling it is not uncommon have $n > 4000^2$.

We present efficient approximate search algorithms for MMD rank-1 lattices and sequences, and introduce a method that searches rank-1 lattices to better represent

and integrate functions with an anisotropic Fourier spectrum. The findings result in new algorithms for anti-aliasing and texture representation [3], i.e. numerical integration and function approximation.

2 Efficient Search by Restricting the Search Space

There exists a construction for MMD rank-1 lattices [2], where the generator vector $\mathbf{g}$ and the number of lattice points n are described by the sequence of convergents of the continued fraction equal to $\sqrt{3}$. However, the number n of points of this construction increases very fast, reducing their applicability in practical applications. For other n the generator vector has to be determined by computer search. The naïve algorithm enumerates all possible generator vectors in order to find MMD lattices. Already for only $s = 2$ dimensions, scanning $\mathcal{O}((n-1)^2)$ candidates becomes prohibitive for large n as used in our applications. Restricting the search space to lattices in Korobov form (i.e. $\mathbf{g} = (1, a)$), the minimum distance can be determined efficiently, as described in [12]. However, not all MMD rank-1 lattices can be represented in Korobov form [4]. For example the MMD rank-1 lattice for $n = 56$ points has the generator vector $\mathbf{g} = (4, 7)$. In the following we examine a restriction of the search space for which the efficient search algorithm resembles rasterization algorithms as used in computer graphics. This search is not restricted to Korobov lattices and we show that it can find MMD rank-1 lattices that cannot be represented in Korobov form. This allows a much more flexible use of rank-1 lattices.

2.1 Approximate Search for MMD Rank-1 Lattices

In order to enable an accelerated search for MMD rank-1 lattices we restrict the search space to a small subset of all possible lattice generator vectors. We base our restriction on two observations: First, for any lattice there is more than one generator vector for the identical lattice. For example if the number n of lattice points is prime, all lattice points scaled by n are generator vectors and thus the shortest vector is generator vector, too. For arbitrary n we noticed that it is still often the case that the shortest vector is also a generator vector. The second observation is, that the largest possible minimum distance l would result from a point set, whose triangulation consists of only equilateral triangles [1] (analogue to hexagonal lattices). This distance is an upper bound on the maximized minimum distance that can only be approximated by rank-1 lattices. Equating the area $A = \frac{1}{n}$ of the basis cell of a rank-1 lattice and twice the area of such an equilateral triangle of side length l yields

$$A = \frac{1}{n} = 2\left(\frac{1}{2} \cdot l \cdot h\right), \quad h = l \cdot \frac{\sqrt{3}}{2} \quad \Longleftrightarrow \quad l = \sqrt{\frac{2}{n \cdot \sqrt{3}}}. \tag{5}$$

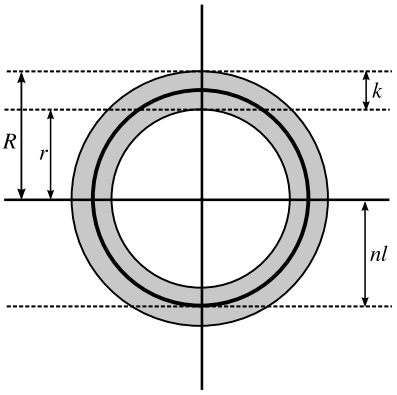


Fig. 1 Idea of the restricted search space.

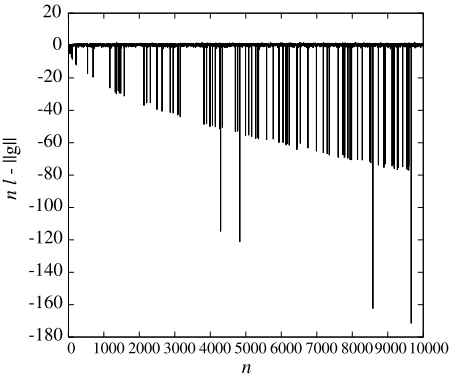


Fig. 2 Difference $n \cdot l - |\mathbf{g}|$ of the maximally possible length l scaled by n and the shortest generator vector of the exhaustive search for $n = 4, \ldots, 10000$.

With the assumption that the generator vector is also the shortest vector it would suffice to search the integer generator vector only within a circle of the radius $n \cdot l$. However as noted above this is not always the case. Experiments showed that using a slightly larger upper bound allows one to find better lattices. To perform the approximate search we restrict the search space for the generator vectors $\mathbf{g}$ to a ring around the origin with inner radius r and outer radius R, where $r = n \cdot l - \frac{k}{2} < n \cdot l < n \cdot l + \frac{k}{2} = R$ and k is a selected positive integer (see Figure 1).

By rasterizing this ring on the integer lattice $\mathbb{Z}^2$ using efficient algorithms from computer graphics [8], all potential generator vectors are enumerated. However, due to symmetry only one eighth of the ring needs to be rasterized (see Figure 3). Fixing the ring width k independent of n, the rasterization runs in $\mathcal{O}(n \cdot l) = \mathcal{O}(\sqrt{n})$ time. The approximate search then runs in $\mathcal{O}(\sqrt{n} \log n)$ time, where the minimum distances are computed using the Gaussian basis reduction.

2.1.1 Restriction of the Search Space

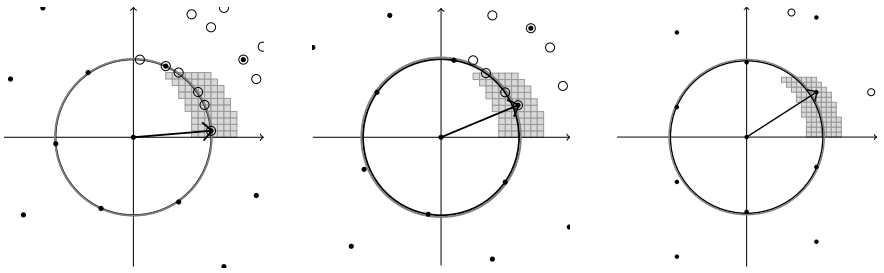
We computed the difference $n \cdot l - \|\mathbf{g}\|$ for $n = 4, \ldots, 10000$, where $\|\mathbf{g}\|$ is the length of the shortest generator vectors found by the exhaustive search. Generator vectors are integer vectors and therefore l has to be scaled by n. Note that when the generator vector of the MMD lattice is not the shortest vector the difference can be negative. The graph in Figure 2 justifies the approach to restrict the search space to a ring of a fixed width. Due to the complexity of the exhaustive search, the range of $n > 10000$ has been investigated for random samples only. An empirically chosen value of $k = 6$ has proven to be a reasonable ring width as described now.

2.1.2 Numerical Evidence

For $n = 4, \ldots, 10000$ and $k = 6$ we now compare the approximate rasterization search to the exact exhaustive search. In 99.1% (i.e. 9908 out of 9997 cases), the approximate algorithm finds the optimal generator vector with respect to maximized minimum distance. The percentage of lattices for which a generator vector coincides with a shortest vector equals 71% (7098 cases), whereas in 28.1% (2810 cases) a generator vector producing a lattice with maximum possible minimum distance is determined inside the ring with width k even if the generator vector is not the shortest vector. Otherwise the new search algorithm yields a maximized minimum distance that is never worse than 90% of the optimum.

The restricted search always yields the correct results for n being prime. We showed above that it is likely to also find generator vectors for MMD rank-1 lattices for arbitrary n. Additionally if the best lattice is not found, at least an acceptable one is found (i.e. one with a minimum distance not worse than 90% of the optimum). Examples for the different cases are visualized in Figure 3. The search space is depicted by the light gray squares, which represent the rasterized region of a ring with radius $n \cdot l$ and width $k = 6$. Due to a very simple rasterization algorithm the rasterized region is slightly larger than required. The light gray circle is of radius $n \cdot l$, while the black circle's radius is the maximized minimum distance $\mathrm{MMD_e}$ determined by the exhaustive search. The set of generator vectors which result from this exhaustive search algorithm and lie in the displayed range are plotted using hollow dots. The solid dots belong to the lattice generated by the displayed vector as one element of the generator vectors resulting from the approximate search with maximized minimum distance $\mathrm{MMD_r}$.

In order to show the improvements of our new algorithm we compare the best lattice found in Korobov form with minimum distance ($\mathrm{MMD_k}$) and the resulting maximized minimum distance using the approximate search ($\mathrm{MMD_r}$). In Figure 4 the ratio $\mathrm{MMD_k}/\mathrm{MMD_r}$ for $n = 4, \ldots, 10000$ is plotted. As is apparent from the graph the new search yields nearly optimal results with respect to the search



(a) $L_{127,(12,1)}$ (b) $L_{134,(12,5)}$ (c) $L_{210,(14,9)}$

Fig. 3 Illustration of the rasterization search. (a) $n = 127$, $\mathrm{MMD_r} = \mathrm{MMD_e}$. The generator vector $\mathbf{g} = (12, 1)$ is a shortest vector in the lattice. (b) $n = 134$, $\mathrm{MMD_r} = \mathrm{MMD_e}$. $\mathbf{g} = (12, 5)$ does not correspond to a shortest vector. (c) $n = 210$, $\mathrm{MMD_r} < \mathrm{MMD_e}$.

criterion and delivers better results than the Korobov form in most cases. More precisely $\mathrm{MMD_r} \geq \mathrm{MMD_k}$ in 99.1% of the cases, of which for 6.2% we have $\mathrm{MMD_r} > \mathrm{MMD_k}$, if the MMD rank-1 lattice cannot be represented in Korobov form, and $\mathrm{MMD_r} = \mathrm{MMD_k}$ in 92.9% of the cases.

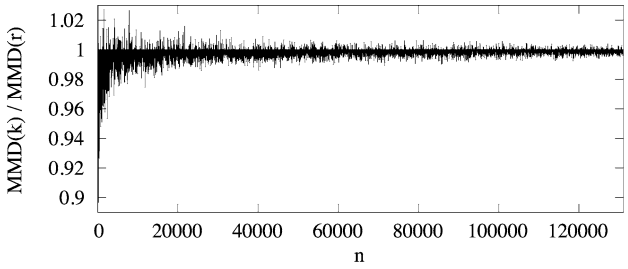


Fig. 4 Ratio $\mathrm{MMD_k}/\mathrm{MMD_r}$ for $n \in [4, 131072)$.

2.2 Search for MMD Rank-1 Lattice Sequences

Using a fixed n for the number of lattice points is often insufficient for graphics applications. For example hierarchical representations of images or progressive sampling need a varying number of sample points. Lattice sequences can provide this functionality and we examine two approaches in this section how to construct rank-1 lattice sequences with MMD property.

As defined in Section 1, a rank-1 lattice sequence $L_{\mathbf{g}}^{\Phi_b}$ contains a sequence of rank-1 lattices $L_{b^m,\mathbf{g}}$ for $m \in \mathbb{N}_0$. We search for rank-1 lattice sequences with maximized minimum distance in the sense that the weighted sum

$$\sum_{m=m_{\min}}^{m_{\max}} (d_{\min}(L_{b^m,\mathbf{g}}))^2 b^m \tag{6}$$

is maximized. Scaling the squared minimum distance by b^m assigns equal importance to all lattices of the sequence since the area of a basis cell is $\frac{1}{b^m}$.

2.2.1 Lattice Sequences based on an Initial MMD Rank-1 Lattice

One way of constructing a MMD rank-1 lattice sequence is by taking a generator vector $\mathbf{g}$ of a MMD rank-1 lattice $L_{b^m,\mathbf{g}}$ and using it in Equation (2). For $q \in \mathbb{N}_0$ and a fixed m, each set of points $\{x_{q\cdot b^m}, \ldots, x_{(q+1)b^m-1}\} \subset L_{\mathbf{g}}^{\Phi_b}$ is a copy of $L_{b^m,\mathbf{g}}$ shifted by $\Delta(q) := \Phi_b(q)b^{-m}\mathbf{g}$ [10]. The minimum distance of all copies is iden-

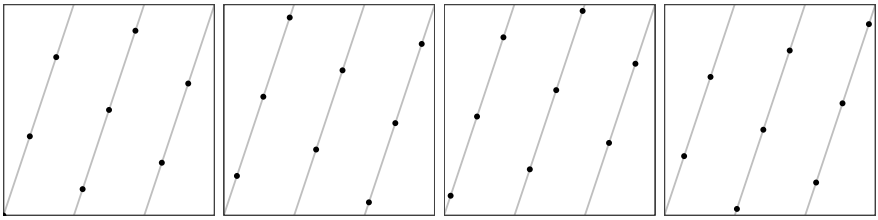


Fig. 5 The shifted lattices $L_{8,(1,3)} + \Delta(q) = L_{8,(1,3)} + \Phi_2(q)2^{-3}(1,3)$ for $q = 0, 1, 2, 3$ from the lattice sequence $L^{\Phi_2}_{(1,3)}$.

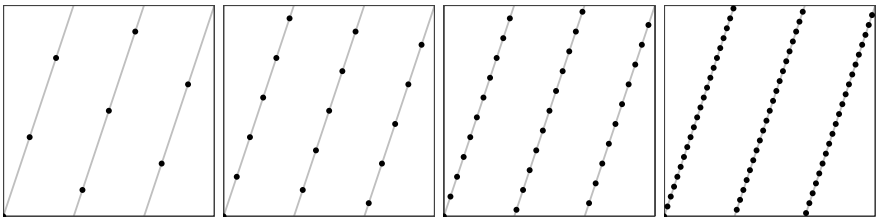


Fig. 6 The lattices $L_{2^m,(1,3)}$ of the lattice sequence $L^{\Phi_2}_{(1,3)}$ started with the initial MMD rank-1 lattice $L_{8,(1,3)}$.

tical, as $d_{\min}$ is shift invariant. For the example of $L^{\Phi_2}_{(1,3)}$ this structural property [9] is depicted in Figure 5. We now consider a two-dimensional generator vector $\mathbf{g} = (g_1, g_2)$ with $\gcd(n, g_1, g_2) = 1$. Then all points of the rank-1 lattice sequence $L^{\Phi_b}_{\mathbf{g}}$ lie on at most $n_h = g_1 + g_2 - 1$ hyperplanes, independent on the number of points. As a consequence, all points of the previous example $L^{\Phi_2}_{(1,3)}$ reside on three hyperplanes (induced by the generator vector), as illustrated in Figure 6. This means that the generator vector has to be modified such that the undesirable uniform bound on the minimum distance induced by the number of hyperplanes is improved.

Considering generator vectors of the form

$$\mathbf{g}_{i,j} := (g_1 + i \cdot b^m, g_2 + j \cdot b^m) \text{ for } i, j \in \mathbb{N}_0,$$

we have $\mathbf{g}_{i,j} \equiv \mathbf{g} \bmod b^m$. As a consequence $L_{b^m,\mathbf{g}} = L_{b^m,\mathbf{g}_{i,j}}$, i.e. the minimum distance remains unchanged for b^m points. However, the upper bound on the number of hyperplanes is increased to $n_h = g_1 + i \cdot b^m + g_2 + j \cdot b^m - 1$, as desired. For example $L^{\Phi_2}_{(41,11)}$ with $(41, 11) = (1 + 5 \cdot 8, 3 + 1 \cdot 8)$ does not restrict points to only three hyperplanes, but for $n = 8$ points generates the same rank-1 lattice as $L^{\Phi_2}_{(1,3)}$, i.e. $L_{8,(1,3)} = L_{8,(41,11)}$ (compare Figures 6 and 7).

The search procedure is started by selecting both a minimum number of points $b^{m_{\min}}$ and maximum $b^{m_{\max}}$. First a search of the previous section is run to find an

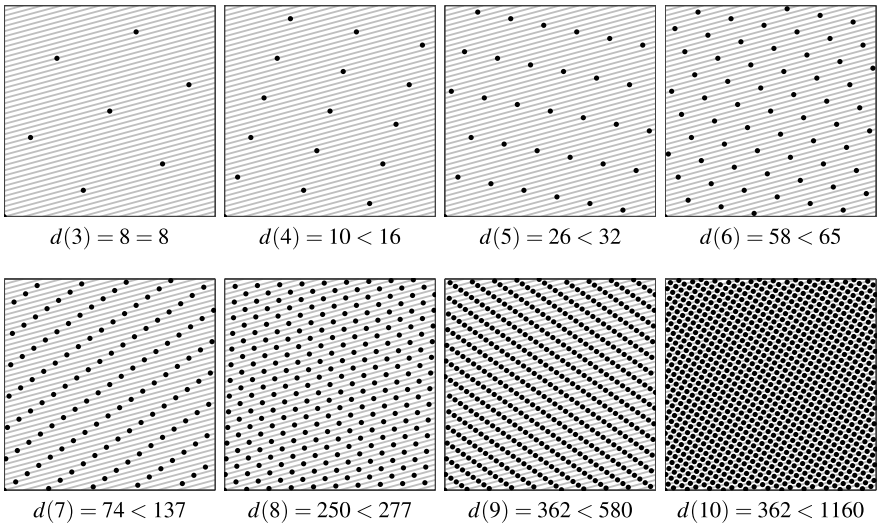


Fig. 7 Searching an MMD rank-1 lattice sequence for the initial lattice $L_{2^3,(1,3)}$ (see Figure 6) and $m_{\max} = 7$, yields $L^{\Phi_2}_{(41,11)}$ with $\mathbf{g}_{5,1} = (1 + 5 \cdot 2^3, 3 + 1 \cdot 2^3) = (41, 11)$. The gray lines show all possible hyperplanes. For each lattice of the rank-1 lattice sequence we compare its minimum distance $d(m) := d_{\min}(L_{b^m,\mathbf{g}})^2 b^m$ to the maximum minimum distance that can be obtained by a single MMD rank-1 lattice.

initial MMD rank-1 lattice generator vector $\mathbf{g}$ for $b^{m_{\min}}$ points. Then the sum of minimum distances (6) is evaluated for each potential generator vector $\mathbf{g}_{i,j}$ in order to find the maximum, where the search range is determined by

$$g_1 + i \cdot b^{m_{\min}} \le b^{m_{\max}} \Rightarrow i \le \frac{b^{m_{\max}} - g_1}{b^{m_{\min}}} < b^{m_{\max}-m_{\min}} \text{ and}$$
$$g_2 + j \cdot b^{m_{\min}} \le b^{m_{\max}} \Rightarrow j \le \frac{b^{m_{\max}} - g_2}{b^{m_{\min}}} < b^{m_{\max}-m_{\min}}.$$

Due to symmetry, an obvious optimization is to bound the range of j by $b^{m_{\max}-m_{\min}} - i$. Again, minimum distances are computed using the Gaussian basis reduction. An example result of the search is illustrated in Figure 7, where minimum distances obtained by the rank-1 lattice sequence are compared to the distances that can be obtained by rank-1 lattices alone.

2.2.2 Approximate Search by Restricting the Search Space

In the second approach the search is not based on an initial MMD rank-1 lattice. Instead we choose $m_{\min} = 1$ and fix a value for $m_{\max}$, looking for a generator vector that maximizes Equation (6).

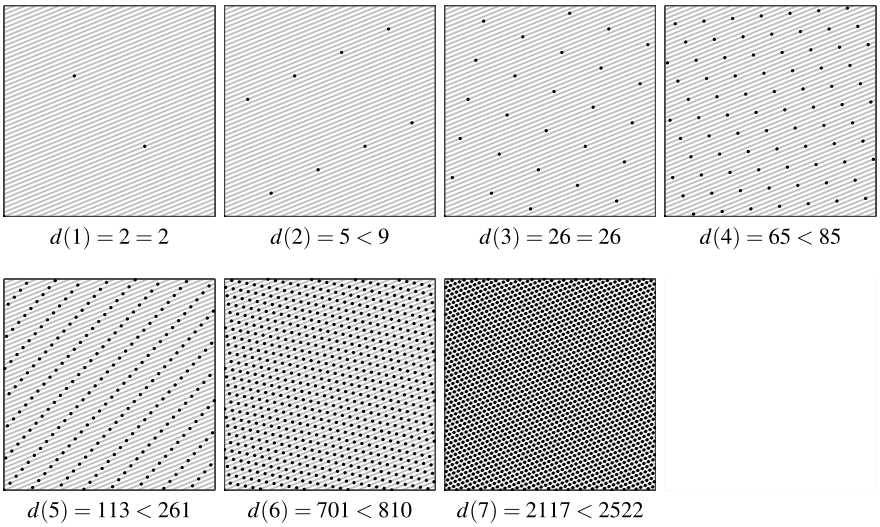


Fig. 8 $L^{\Phi_3}_{(47,19)}$ in base $b = 3$. For each lattice of the rank-1 lattice sequence we compare its minimum distance $d(m) := d_{\min}(L_{b^m,\mathbf{g}})^2 b^m$ to the maximum minimum distance that can be obtained by a single MMD rank-1 lattice.

In order to accelerate the search process, the search space can be restricted using the same strategy as in the rasterization search algorithm for rank-1 lattices (see Section 2.1). Then the search space is the union of the restricted search spaces for $L_{b^m,\mathbf{g}}$, $1 < m \leq m_{\max}$. In experiments, the restricted search achieved the same results as the exhaustive computer search for $n_{\max} := b^{m_{\max}} \leq 256$ and $b = 2, 3, 4$, simultaneously reducing the run-time from $\mathcal{O}(n_{\max}^2 \log n_{\max})$ to $\mathcal{O}(\sqrt{n_{\max}} \log n_{\max})$.

We compare the two search approaches presented in this section by summing the minimum distances of the first b^m points of each sequence

$$\sum_{m=2}^{7} d(m) = \sum_{m=2}^{7} d_{\min}(L_{b^m,\mathbf{g}})^2 b^m.$$

Although the second approach is more general than the first one, the lattices produced by the sequence might not necessarily have the maximal possible minimum distance for the corresponding $n = b^m$ points, which is assured at least for the initial lattice in the first approach. Figure 8 shows the resulting lattice sequence for $b = 3$ and $m_{\max} = 7$, while Table 1 shows the results of the numerical comparison. By definition for $m = 2$ the lattice of the sequence $L^{\Phi_3}_{(82,129)}$ represents an MMD rank-1 lattice, whereas for $m = 3$ the rank-1 lattice of the sequence $L^{\Phi_3}_{(47,19)}$ achieves the largest possible minimum distance as well.

Table 1 Comparing the lattice sequences $L^{\Phi_3}_{(82,129)}$ and $L^{\Phi_3}_{(47,19)}$ with respect to the minimum distance of the first b^m points of the lattice sequence for $b = 3$ and $2 \le m \le 7$. The initial MMD rank-1 lattice for $L^{\Phi_3}_{(82,129)}$ is given by $L_{3^2,(1,3)}$.

m	$d(m)$ first approach	$d(m)$ second approach
2	9	5
3	25	26
4	34	65
5	229	113
6	745	701
7	1033	2117
Σ	2075	3027

3 Search of Anisotropic Rank-1 Lattices to Approximate Spectra

In many graphics applications the image functions exhibit a strong anisotropic behavior in their Fourier spectrum. By constructing rank-1 lattices with knowledge of these functions the image quality can be improved. The Fourier transform of the Shah function

$$\text{Ш}_{L_{n,\mathbf{g}}}(\mathbf{x}) := \sum_{\mathbf{p}\in\mathbb{Z}^s} \delta(\mathbf{x} - B\cdot\mathbf{p})$$

over the lattice $L_{n,\mathbf{g}}$ with basis B, where $\delta(\mathbf{x})$ is Dirac's delta function, yields another Shah function over its dual lattice $L^{\perp}_{n,\mathbf{g}}$ [6]. This means that we can describe the spectrum $\mathcal{S}_{n,\mathbf{g}}$ of $L_{n,\mathbf{g}}$ by the fundamental Voronoi cell of the dual lattice $L^{\perp}_{n,\mathbf{g}}$.We characterize the shape of this cell by two parameters, namely by its orientation $\overrightarrow{\omega}_L$ and by its width w_L, which are computed by means of the basis $B^{\perp}$ of $L^{\perp}_{n,\mathbf{g}}$. Given a lattice basis $B = (\mathbf{b}_1\mathbf{b}_2)^{\mathrm{t}}$, where t means transposed, the dual basis can be easily determined by $B^{\perp} = (B^{-1})^{\mathrm{t}}$. In order to assure that $B^{\perp}$ spans the Delaunay triangulation and thus the Voronoi diagram, the dual basis has to be reduced, for example using the Gaussian basis reduction. Let

$$\mathbf{v} := \begin{cases} \mathbf{b}^{\perp}_1 + \mathbf{b}^{\perp}_2 & \text{if } \mathbf{b}^{\perp}_1 \cdot \mathbf{b}^{\perp}_2 < 0 \\ \mathbf{b}^{\perp}_2 - \mathbf{b}^{\perp}_1 & \text{otherwise} \end{cases}$$

be the diagonal of the basis cell spanned by $\mathbf{b}^{\perp}_1$ and $\mathbf{b}^{\perp}_2$, such that $\mathbf{v}$ and $\mathbf{b}^{\perp}_1$ or $\mathbf{v}$ and $\mathbf{b}^{\perp}_2$ form a valid basis of the dual lattice as well. Then we approximate the orientation of the fundamental Voronoi cell by

$$\overrightarrow{\omega}_L := \mathbf{b}^{\perp}_2 + \mathbf{v} = \begin{cases} 2\cdot\mathbf{b}^{\perp}_2 + \mathbf{b}^{\perp}_1 & \text{if } \mathbf{b}^{\perp}_1 \cdot \mathbf{b}^{\perp}_2 < 0 \\ 2\cdot\mathbf{b}^{\perp}_2 - \mathbf{b}^{\perp}_1 & \text{otherwise.} \end{cases} \tag{7}$$

The width w_L of $\mathcal{S}_{n,\mathbf{g}}$ is defined as the length of the shortest basis vector normalized by the hexagonal bound l, i.e.

$$w_L = \frac{||\mathbf{b}_1^{\perp}||}{l \cdot n}. \tag{8}$$

Note that $l \cdot n$ also represents an upper bound on the maximized minimum distance of the dual lattice, as the length of shortest vector in $L_{n,\mathbf{g}}^{\perp}$ corresponds to the length of the shortest vector in $L_{n,\mathbf{g}}$ scaled by n [4].

The spectrum $\mathcal{T}_{\mathbf{d},w}$, according to which we want to search the rank-1 lattice, is specified by its main direction, i.e. orientation, $\mathbf{d} \in \mathbb{R}^2$ and its width w. The two-dimensional vector $\mathbf{d}$ and the scalar w are passed as an input parameter to the lattice search by an application. The width w takes values in the range of $[0, 1]$ and represents the measure of desired anisotropy. The most anisotropic spectrum is denoted by $w = 0$, whereas $w = 1$ represents the isotropic one. Note that we have to allow $g_i = 0$, $i = 1, 2$ for the generator vector in order to be able to approximate spectra aligned to axes of the Cartesian coordinate system.

For $n \in \mathbb{N}$ the search algorithm steps through all distinct lattices. This can be realized for example by using an $n \times n$ array, where the generator vectors of identical lattices are marked. Given any $\mathbf{g} \in [0, n)^2$, the set of vectors yielding identical lattices is $\{k \cdot \mathbf{g} \bmod n : \gcd(n, k) = 1, k = 1, \ldots, n-1\}$. After computing a Minkowski-reduced basis of the dual lattice, the orientation and width of the fundamental Voronoi cell are determined according to Equations (7) and (8). Then the lattices are sorted with respect to $|w_L - w|$. For the smallest difference we choose the lattice, whose orientation $\overrightarrow{\omega}_L$ best approximates the main direction $\mathbf{d}$ of $\mathcal{T}_{\mathbf{d},w}$. Thereby the similarity

$$\text{sim} = \frac{\mathbf{d} \cdot \overrightarrow{\omega}_L}{\|\mathbf{d}\| \cdot \|\overrightarrow{\omega}_L\|}$$

between those two vectors is measured by calculating the cosine of the angle between $\overrightarrow{\omega}_L$ and $\mathbf{d}$. Figure 9 shows an example for anisotropic rank-1 lattices having $n = 56$ points, where the spectrum is specified by $\mathbf{d} = (\cos\alpha, \sin\alpha)$ with $\alpha = 303°$ and the width varies from 0.1 to 1.0 in steps of 0.1. Using the Gaussian basis reduction for the lattice basis search, the algorithm runs in $\mathcal{O}(n^2 \log n)$ time.

4 Weighted Norms

So far we considered rank-1 lattices only on the unit square. However, graphics applications often require arbitrary rectangular regions. Just selecting a corresponding region of the lattice defined in the entire real space and scaling it to the unit square is not an option as this would destroy for example the needed periodicity and complicate address computations in image applications. We now show how to extend our approximate search for isotropic and anisotropic rank-1 lattices to such regions.

All that needs to be done is considering the weighted norm $||B^{\mathrm{r}}\mathbf{x}_i||$ in the definition of the minimum distance in Equation (4) instead of the Euclidean norm where B^{r} describes the transformation of the unit square to the desired region. Note that as before the distance to the origin has to be computed with respect to the unit torus.

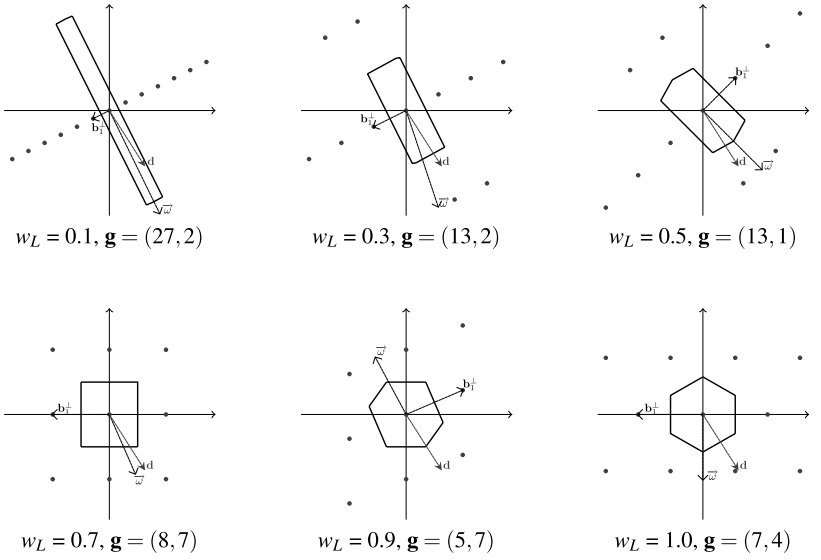


Fig. 9 Resulting spectra for a fixed direction $\mathbf{d} = (\cos 303°, \sin 303°)$ and width varying from 0.1 to 1.0.

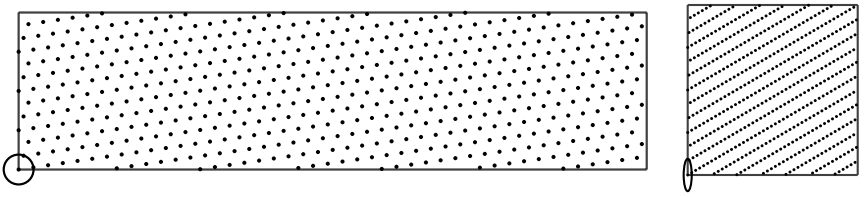


Fig. 10 Searching on a rectangular domain. Left: MMD rank-1 lattice $L_{512,(4,45)}$ in a domain of width-to-height ratio $x : y = 4 : 1$ in world coordinates. Right: The same lattice in the scaled basis with $x : y = 1 : 1$. The search region becomes an ellipse.

Approximate Search for MMD rank-1 lattices

For the special case of scaled rectangular domains, i.e. $B^{\mathrm{r}} = (\mathbf{b}_1^{\mathrm{r}} \mathbf{b}_2^{\mathrm{r}}) = \big((x,0)^{\mathrm{t}}(0,y)^{\mathrm{t}}\big)$, the rasterization search can be adapted easily. Therefore the lattice basis B has to be transformed into world coordinates before computing its determinant, i.e. area A. For the "weighted" lattice basis $B^{\mathrm{w}} = B^{\mathrm{r}} \cdot B$ the area of the basis cell is

$$A = |\det B^{\mathrm{w}}| = |\det B^{\mathrm{r}}| \cdot |\det B| = \frac{x \cdot y}{n} \quad \Rightarrow \quad l = \sqrt{\frac{2 \cdot x \cdot y}{n \cdot \sqrt{3}}}$$

in analogy to Equation (5).

Since we perform the rasterization directly in the sheared basis, the shortest vectors lie within an ellipse (see Figure 10). Its axes $\mathbf{a}_x = ((n \cdot l)/x, 0)^{\mathrm{t}}$ and $\mathbf{a}_y = (0, (n \cdot l)/y)^{\mathrm{t}}$ result from transforming the circle axes $((n \cdot l), 0)^{\mathrm{t}}$ and $(0, (n \cdot l))^{\mathrm{t}}$ into the sheared basis B^{r} of the actual region.

As the rasterization runs in less than $\mathcal{O}(||\mathbf{a}_x|| + ||\mathbf{a}_y||)$, with $||\mathbf{a}_x||, ||\mathbf{a}_y|| \in \mathcal{O}(\sqrt{n})$, we still have a run-time complexity of $\mathcal{O}(\sqrt{n})$. Finally the Gaussian basis reduction needs to be adapted to weighted norms in order to compute the minimum distance. For that purpose the only modification consists in weighting the initial basis before performing the reduction steps. Therefore the search algorithm maintains a run-time complexity of $\mathcal{O}(\sqrt{n} \log n)$.

Anisotropic Rank-1 Lattices

Using the algorithm from Section 3 with weighted norms only requires to transform the desired main direction $\mathbf{d} \in \mathbb{R}^2$ into the sheared basis B^{r} of the desired domain.

5 Applications in Computer Graphics

The search algorithms from Section 2.1 allow one to find suitable generator vectors for the graphics applications introduced in [3, 4] much faster. Here, we introduce two new applications of anisotropic rank-1 lattices.

5.1 Anti-Aliasing by Anisotropic Rank-1 Lattices

In graphics applications rank-1 lattices can be used to integrate the image function over the pixels. By adapting the quadrature rule to the Fourier spectrum of the image function in a way that more of the important frequencies are captured, aliasing artifacts can be reduced. The improved anti-aliasing is illustrated by a comparison in Figure 11.

Given the algorithm from Section 3, an anisotropic MMD rank-1 lattice is specified by the main direction $\mathbf{d}$ and the width w. We globally assume maximum anisotropy by fixing $w = 0$. The main direction $\mathbf{d}$ is determined by projecting the normal of the first object intersected by a ray through the center of a pixel onto the image plane and normalizing the resulting vector. This way the samples from the anisotropic rank-1 lattice in the pixel become isotropic and more uniform, when projected onto the surface seen in the scene (see Figure 12 on the left). As a consequence the texture is averaged more efficiently, resulting in reduced aliasing. Note that for this argument, we assumed only one plane perpendicular to the normal seen through a pixel, which is a useful approximation in many cases.

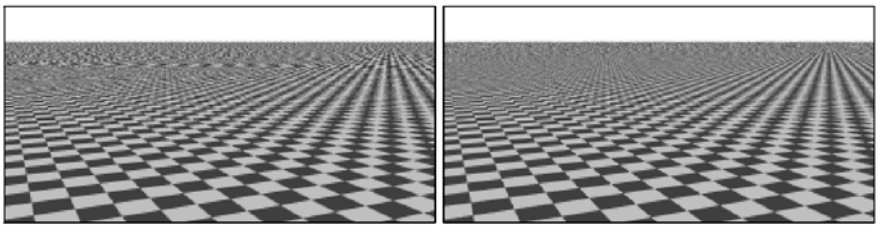


Fig. 11 An infinite checker board rendered with 16 samples for each pixel. The left image uses the same MMD rank-1 lattice $L_{16,(1,4)}$ for all pixels, while in the right image an anisotropic MMD rank-1 lattice adapted to the spectrum of each pixel is used. Clearly some aliases under the horizon become much more attenuated.

As the perspective projection does not have an impact on the variance of the checker board until a certain distance from the camera, anisotropic rank-1 lattices are used only for those pixels for which the hit point of a ray through a pixel midpoint and the checker board exceeds a certain distance to the camera (which is determined experimentally for this special setting). Otherwise MMD rank-1 lattices are used per pixel.

In Figure 12 on the right, we compared the anisotropic rank-1 lattices to MMD rank-1 lattices and jittered grid by computing the L_2-norm of a converged reference image to the corresponding test images for an increasing number of sampling points per pixel. Note that both axes in the error graph are scaled logarithmically and that the reference image was computed by applying a jittered grid sampling pattern with 1024×1024 samples at each pixel. We observe that using the anisotropic rank-1 lattice outperforms the other sampling patterns especially for lower sampling rates. In contrast to the MMD rank-1 lattices, the error curve of the anisotropic lattices does not expose a strong oscillation any more.

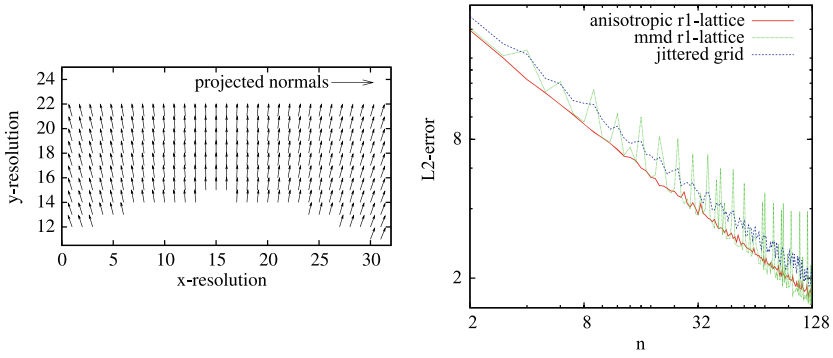


Fig. 12 Left: The arrows indicate the pixels and directions for which anisotropic rank-1 lattices are used. Right: Comparison of the anisotropic rank-1 lattices, to MMD rank-1 lattices and jittered grid.

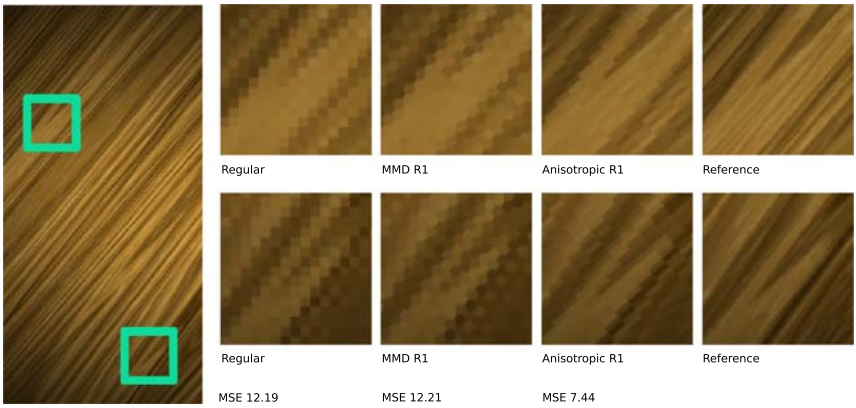


Fig. 13 Magnifications of the highlighted squares in the texture on the left represented on the regular grid, MMD rank-1 lattice, and anisotropic rank-1 lattice by 16384 pixels each. Note that for the anisotropic rank-1 lattice the mean square error (MSE) to the high resolution reference on the right is about half of the regular and MMD rank-1 lattice.

5.2 Rank-1 Lattice Images and Textures

In [4] the Voronoi diagram of MMD rank-1 lattices was used as an approximation to hexagonal pixel layout. While the visual quality at the same number of pixels was superior to classic rectangular layouts, the algorithms were simpler than for hexagonal layouts.

If now an image, or more specifically a texture, exposes an anisotropy, anisotropic MMD rank-1 lattices can be used to further improve the visual appearance, i.e. the approximation power. This is illustrated in Figure 13 for a wood grain texture, which exposes one main direction with large variance.

The parameters for determining the anisotropic MMD rank-1 lattice are computed from the structure tensor of each pixel. Without loss of generality let $\lambda_{1,i} > \lambda_{2,i}$ be the eigenvalues of the structure tensor and $\mathbf{v}_{1,i}$ and $\mathbf{v}_{2,i}$ the corresponding eigenvectors for each pixel $i \in [0, xRes \cdot yRes)$. Then the main direction $\mathbf{d}$ is computed by averaging the eigenvector of the largest eigenvalue over all pixels. The width

$$w = 1.0 - \frac{1}{A_{max}} \cdot \sum_{i=0}^{xRes \cdot yRes} \frac{\lambda_{1,i}}{\lambda_{2,i}}$$

subtracts the normalized texture anisotropy from 1, since 0 means maximum anisotropy for the search algorithm from Section 3. The normalization constant A_{max} must be determined experimentally for a set of textures.

In Figure 14 isotropic rank-1 lattice textures are compared to anisotropic ones by means of the L_2-error of the test images to a reference solution for an increasing

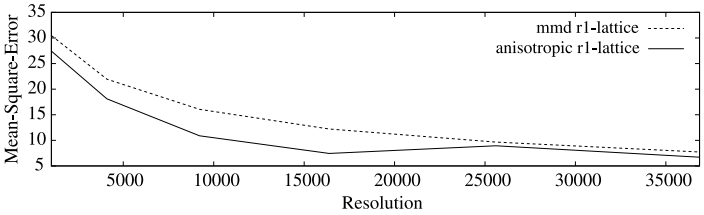


Fig. 14 Error graph showing the different approximation qualities measured with respect to a reference image.

number of lattice points for the source image of Figure 13. As can be seen from the error graph, the anisotropic rank-1 lattice textures are superior, as they are able to capture even small details, which are lost in the isotropic case.

6 Conclusions

We introduced algorithms that efficiently search for generator vectors of rank-1 lattices and sequences with important new applications in computer graphics. Useful results were obtained for both image synthesis and representation. Future research will concentrate on applications of rank-1 lattice sequences and the fast search of generator vectors for the anisotropic case.

Acknowledgements The authors would like to thank mental images GmbH for support and funding of this research.

References

1. Conway, J., Sloane, N., Bannai, E.: Sphere-packings, Lattices, and Groups. Springer-Verlag, New York, Inc. (1987)
2. Cools, R., Reztsov, A.: Different quality indexes for lattice rules. Journal of Complexity **13**(2), 235–258 (1997)
3. Dammertz, H., Keller, A., Dammertz, S.: Simulation on rank-1 lattices. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 205–216. Springer (2008)
4. Dammertz, S., Keller, A.: Image synthesis by rank-1 lattices. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 217–236. Springer (2008)
5. Dieter, U.: How to Calculate Shortest Vectors in a Lattice. Math. Comp. **29**(131), 827–833 (1975)
6. Entezari, A., Dyer, R., Möller, T.: From sphere packing to the theory of optimal lattice sampling. In: T. Möller, B. Hamann, R. Russell (eds.) Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration. Springer (2009)

7. Fincke, U., Pohst, M.: Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis. Math. Comp. **44**, 463–471 (1985)
8. Foley, J., van Dam, A., Feiner, S., Hughes, J.: Computer Graphics, Principles and Practice, 2nd Edition in C. Addison-Wesley (1996)
9. Hickernell, F., Hong, H.: Computing multivariate normal probabilities using rank-1 lattice sequences. In: G. Golub, S. Lui, F. Luk, R. Plemmons (eds.) Proceedings of the Workshop on Scientific Computing (Hong Kong), pp. 209–215. Springer (1997)
10. Hickernell, F., Hong, H., L'Ecuyer, P., Lemieux, C.: Extensible lattice sequences for quasi-Monte Carlo quadrature. SIAM J. Sci. Comput. **22**, 1117–1138 (2001)
11. Kannan, R.: Algorithmic Geometry of Numbers. Annual Reviews in Computer Science **2**, 231–267 (1987)
12. Knuth, D.: The Art of Computer Programming Vol. 2: Seminumerical Algorithms. Addison Wesley (1981)
13. L'Ecuyer, P.: Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure. Math. Comput. **68**(225), 249–260 (1999)
14. L'Ecuyer, P., Couture, R.: An Implementation of the Lattice and Spectral Tests for Multiple Recursive Linear Random Number Generators. INFORMS Journal on Computing **9**(2), 206–217 (1997)
15. L'Ecuyer, P., Lemieux, C.: Variance Reduction via Lattice Rules. Manage. Sci. **46**(9), 1214–1235 (2000)
16. Martinet, J.: Perfect Lattices in Euclidean Spaces. Springer-Verlag (2003)
17. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
18. Rote, G.: Finding a Shortest Vector in a Two-Dimensional Lattice Modulo m. Theoretical Computer Science **172**(1–2), 303–308 (1997)
19. Siegel, C.: Lectures on Geometry of Numbers. Springer-Verlag (1989)
20. Sloan, I., Joe, S.: Lattice Methods for Multiple Integration. Clarendon Press, Oxford (1994)

Parallel Random Number Generators Based on Large Order Multiple Recursive Generators

Lih-Yuan Deng, Jyh-Jen Horng Shiau, and Gwei-Hung Tsai

Abstract Classical random number generators like Linear Congruential Generators (LCG) and Multiple Recursive Generators (MRG) are popular for large-scale simulation studies. To speed up the simulation process, a systematic method is needed to construct and parallelize the random number generators so that they can run simultaneously on several computers or processors. LCGs and MRGs have served as baseline generators for some parallel random number generator (PRNG) constructed in the literature. In this paper, we consider the parallelization problem particularly for a general class of efficient and portable large-order MRGs, in which most coefficients of the recurrence equation are nonzero. With many nonzero terms, such MRGs have an advantage over the MRGs with just a few nonzero terms that they can recover more quickly from a bad initialization. With the special structure imposed on the nonzero coefficients of the new class of generators, the proposed PRNGs can be implemented efficiently. A method of automatic generation of the corresponding MRGs for parallel computation is presented.

1 Introduction

Many scientific researches as well as technology developments demand large- or very-large-scale simulation studies. Random number generators (RNG) definitely

Lih-Yuan Deng
Department of Mathematical Sciences, University of Memphis, Memphis, TN 38152, U.S.A.
e-mail: lihdeng@memphis.edu

Jyh-Jen Horng Shiau
Institute of Statistics, National Chiao Tung University, Hsinchu, Taiwan, 30010, R. O. C.
e-mail: jyhjen@stat.nctu.edu.tw

Gwei-Hung Tsai
Department of Applied Statistics and Information Science, Ming Chuan University, Taoyuan, Taiwan, 333, R.O.C.
e-mail: herbtsai@mail.mcu.edu.tw

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_17,

plays a crucial role in these scientific applications. For example, we need good RNGs for Monte Carlo integration, computer modeling, and simulations.

To speed up the simulation process, it is common to run simulations in parallel on several processors. Thus a systematic method is needed to construct and parallelize the random number generators so that they can run simultaneously on multiple computers or processors. A good parallel random number generator (PRNG) is desired to have the following features: (i) the individual generator should possess all the nice properties of a good RNG, such as uniformity, "randomness", equi-distribution over high dimensions, etc.; (ii) the sequences generated from different processors should be "independent"; (iii) a new and distinct RNG can be automatically constructed when needed.

To construct a PRNG, we usually need a baseline generator. For this, two popular RNGs, Linear Congruential Generators (LCG) and Multiple Recursive Generators (MRG), are often used. For various methods of using LCG or MRG to construct PRNG, see [8, 11, 15], and the references cited therein. Among these methods, a fairly common strategy to generate separate sequences for different processors is using a skip-ahead scheme on the same RNG.

Another approach is to use different multipliers for different processors. Deng [2] proposed an automatic generating method to construct many maximum-period MRGs from a single MRG. Based on his method, recently, for the case of Sophie-Germain primes, Deng, Li, and Shiau [4] developed an efficient method to generate primitive polynomials from an already known primitive polynomial, from which maximum-period MRGs can be produced for PRNGs. In this paper, we modify this algorithm for the case of non-Sophie-Germain primes.

The class of the baseline generators used in [4] to develop PRNGs is the DX-k generators proposed by Deng and Xu [6]. It is a special class of MRGs with equal and only up to four nonzero coefficients in the k-th order recurrence equation, hence is very efficient in computation. However, an RNG with very few nonzero coefficients usually has a drawback of "bad initialization effect", namely, when the k-dimensional state vector is close to the zero vector, the subsequent numbers generated may stay within a neighborhood of zero for quite many of them before they can break away from this near-zero land, a property apparently not desirable in the sense of randomness. Consequently, two generated sequences using the same DX generator with nearly identical state vectors may not depart from each other quickly enough. This "bad initialization effect" was observed by Panneton, L'Ecuyer, and Matsumoto [14] for MT19937, a popular generator proposed by Matsumoto and Nishimura [13].

To avoid the above potential problem, Deng, Li, Shiau, and Tsai [5] considered MRGs with very few zero coefficients. To make such generators efficient and portable, they proposed selecting the same nonzero value for all coefficients in the recurrence equation. With this feature, the proposed generators (named the DL generators) can be implemented efficiently via a higher-order recurrence with very few nonzero coefficients.

In this paper, we extend the PRNG construction method given in [4] to the DL generators and a new class of efficient generators called DT generators.

The remaining of the paper is organized as follows. In Section 2, we review some results of the classical MRG, in particular the DL generators, and describe the proposed DT generators. In Section 3, we describe the new automatic generating method for parallel MRGs designed particularly for non-Sophie-Germain primes. We then present efficient implementations of PRNGs when using DL and DT generators as baseline generators.

2 MRGs and Classes of Efficient Generators

An MRG of order k generates pseudo random numbers sequentially based on the following linear recurrence equation:

$$X_i = (\alpha_1 X_{i-1} + \cdots + \alpha_k X_{i-k}) \bmod p, \quad i \geq k, \tag{1}$$

where the multipliers $\alpha_1, \cdots, \alpha_k$ and prime modulus p are all positive integers, and $X_0, \cdots, X_{k-1}$ are k initial seeds that are not zeroes. The corresponding characteristic polynomial has the following form:

$$f(x) = x^k - \alpha_1 x^{k-1} - \cdots - \alpha_k. \tag{2}$$

If $f(x)$ is a primitive polynomial modulo p, then the MRG in (1) can achieve the maximum period of $p^k - 1$. See Knuth [7]. A maximum-period MRG of order k enjoys a nice equi-distribution property up to k dimensions. See [12].

2.1 Searching for Large-Order MRGs

Alanen and Knuth [1] and Knuth [7] described some conditions for a polynomial to be a primitive polynomial. The major difficulty is on factoring a large number. L'Ecuyer, Blouin, and Couture [10] proposed a method that can bypass this difficulty and found some maximum-period MRGs of order up to 7 by finding p such that $(p^k - 1)/(p - 1)$ is a prime. Later, following the same idea of bypassing factorization, Deng [2] developed an efficient algorithm that are able to find maximum-period MRGs of large order by providing an early exit strategy for failed searches.

In addition, one can also require that both p and $Q \equiv (p-1)/2$ are prime numbers. For a given k, requiring that both $R(k, p)$ and Q are primes was the strategy used in L'Ecuyer [9], where a short list of parameters that satisfy these conditions was also given, as well as tables of specific MRGs. If p and Q are both primes, Q is commonly called a Sophie-Germain prime number and p is usually called a "safe prime" in the area of cryptography. However, there is no particular strong advantage to choose a "safe prime" in the area of computer simulation. Denote $SG \equiv \{p \mid \text{both } p \text{ and } (p-1)/2 \text{ are primes}\}$ and $NSG \equiv \{p \mid p \text{ is a prime and } (p -$

$1)/2$ is not a prime}. For convenience, primes in SG and NSG will be referred to as SG-primes and NSG-primes, respectively, hereafter.

Table 1 lists examples of p being NSG-primes as well as SG-primes with respect to some order k, for which $(p^k-1)/(p-1)$ is prime. Because of the limited time and space, we only consider the smallest prime k in each interval of 100 starting from $k=101$ to $k=4001$. From Table 1, we can see that, NSG-primes are greater than SG-primes for the same k. Hence the MRG constructed with a NSG-prime has the advantage of slightly longer period length (*i.e.*, p^k-1) than its SG-prime counterpart.

Table 1 List of w (with $p=2^{31}-w$) of NSG-primes and SG-primes for various k.

k	w (NSG)	w (SG)	k	w (NSG)	w (SG)
101	699	82845	2111	3637	3536385
211	8839	841329	2203	126831	6043089
307	3279	52545	2309	155391	340185
401	41685	57189	2411	127197	9256449
503	3637	174489	2503	50301	13539249
601	5067	1327485	2609	99625	8811681
701	15847	220665	2707	38571	1113585
809	699	2010789	2801	85141	1095609
907	8811	4400889	2903	136035	14055825
1009	107979	2368869	3001	32725	3058401
1103	118075	7316361	3109	28537	6741129
1201	14157	1113705	3203	107589	4718889
1301	11235	1070901	3301	56607	14881185
1409	125947	4320189	3407	54451	6243009
1511	55719	2771205	3511	158155	1412961
1601	40087	368961	3607	27159	1026585
1709	34915	1032441	3701	38857	11576625
1801	65335	5789241	3803	13525	32058129
1901	99777	267321	3907	305925	17381649
2003	44961	44961	4001	30801	4412481

The primes considered in [3], [4], and [5] are mostly SG-primes. In this paper, we also consider NSG-primes as listed in Table 1 to construct PRNGs.

2.2 Efficient Classes of DL and DT Generators

Deng, Li, Shiau, and Tsai [5] considered a class of DL-k generators as

$$X_i = B(X_{i-1}+X_{i-2}+\cdots+X_{i-k}) \bmod p, \quad i \geq k. \tag{3}$$

That is, MRGs with $\alpha_i = B$ for $i=1,2,\cdots,k$. Such generators can be implemented efficiently by

$$X_i = X_{i-1}+B(X_{i-1}-X_{i-(k+1)}) \bmod p, \quad i \geq k+1. \tag{4}$$

While DL generators can escape quickly from a near-zero k-dimensional state vector, it will stay in near-zero states for a long time for large k, when its k-dimensional state vector is of the form $(0, \cdots, 0, -v, v)'$ for any nonzero integer v.

To avoid this problem, we can consider a new class called DT generators, which has many nonzero terms with geometric weights:

$$X_i = (B^k X_{i-1} + B^{k-1} X_{i-2} + \cdots + B X_{i-k}) \bmod p, \quad i \geq k. \tag{5}$$

Similar to DL generators, DT generators can be efficiently implemented by the $(k+1)$-order equation below:

$$X_i = ((B^{-1} + B^k) X_{i-1} - X_{i-k-1}) \bmod p, \quad i \geq k+1, \tag{6}$$

where $D \equiv (B^{-1} + B^k) \bmod p$ can be pre-computed.

A DT generator with multiplier B could also stay in near-zero states for a long time, when k is large and the state vector is of the form $(0, \cdots, 0, -v, Bv)'$. But this most likely would only happen when one purposely chooses an initial state vector of the above form with the pre-specified multiplier B.

For SG-primes and NSG-primes listed in Table 1, we find DL and DT generators via computer search and list them in Tables 2 and 3, respectively. Unlike DL generators, we choose the smallest B for DT generators, because the multipliers (in the form of a geometric sequence) in the recurrence equation of a DT generator are all different. We have conducted an empirical study and found that all the listed DL and DT generators pass the Crush battery of TestU01 test suite.

Table 2 List of w and B of DL-k generators with modulus $p = 2^{31} - w$ corresponding to NSG-primes and SG-primes for various k. where $B = x$*ddddd* with $x = 10737$.

k	w (SG)	B	w (NSG)	B	k	w (SG)	B	w (NSG)	B
101	699	x41777	82845	x41723	2111	3637	x39400	3536385	x25977
211	8839	x41453	841329	x41805	2203	126831	x38735	6043089	x41471
307	3279	x41594	52545	x41682	2309	155391	x41131	340185	x39495
401	41685	x41092	57189	x41021	2411	127197	x38767	9256449	x37213
503	3637	x33579	174489	x40808	2503	50301	x28943	13539249	x40713
601	5067	x41559	1327485	x41084	2609	99625	x34382	8811681	x25313
701	15847	x38803	220665	x40279	2707	38571	x35221	1113585	x41455
809	699	x40699	2010789	x40429	2801	85141	x23130	1095609	x31033
907	8811	x38326	4400889	x38083	2903	136035	x40760	14055825	x28445
1009	107979	x39780	2368869	x37435	3001	32725	x39004	3058401	x41224
1103	118075	x41439	7316361	x38922	3109	28537	x37000	6741129	x40319
1201	14157	x37498	1113705	x41605	3203	107589	x41736	4718889	x40314
1301	11235	x40131	1070901	x39703	3301	56607	x29509	14881185	x40195
1409	125947	x37395	4320189	x35452	3407	54451	x25629	6243009	x41500
1511	55719	x36725	2771205	x38144	3511	158155	x26113	1412961	x38372
1601	40087	x36626	368961	x41683	3607	27159	x14920	1026585	x38483
1709	34915	x40775	1032441	x38218	3701	38857	x37074	11576625	x40341
1801	65335	x26147	5789241	x41319	3803	13525	x08780	32058129	x08122
1901	99777	x39543	267321	x39927	3907	305925	x39218	17381649	x41047
2003	44961	x41810	44961	x41810	4001	30801	x28037	4412481	x35759

Table 3 List of w and B of DT-k generators with modulus $p = 2^{31} - w$ corresponding to NSG-primes and SG-primes for various k.

k	w (SG)	B	w (NSG)	B	k	w (SG)	B	w (NSG)	B
101	699	231	82845	83	211	8839	170	841329	547
307	3279	366	52545	502	401	41685	569	57189	345
503	3637	2198	174489	861	601	5067	1283	1327485	960
701	15847	2405	220665	266	809	699	923	2010789	1810
907	8811	5725	4400889	2361	1009	107979	2558	2368869	1097
1103	118075	1071	7316361	952	1201	14157	1800	1113705	3000
1301	11235	477	1070901	637	1409	125947	9459	4320189	1611
1511	55719	5142	2771205	1206	1601	40087	1540	368961	3411
1709	34915	14914	1032441	1620	1801	65335	898	5789241	2164
1901	99777	5449	267321	2918	2003	44961	667	44961	667
2111	3637	4697	3536385	967	2203	126831	855	6043089	3189
2309	155391	6522	340185	1964	2411	127197	242	9256449	1897
2503	50301	10133	13539249	1967	2609	99625	2468	8811681	1287
2707	38571	1775	1113585	3980	2801	85141	316	1095609	5098
2903	136035	1458	14055825	4217	3001	32725	9633	3058401	8581
3109	28537	6594	6741129	3960	3203	107589	2471	4718889	416
3301	56607	1269	14881185	11703	3407	54451	3154	6243009	4646
3511	158155	3608	1412961	5935	3607	27159	7855	1026585	550
3701	38857	7604	11576625	11923	3803	13525	4874	32058129	2395
3907	305925	1063	17381649	1252	4001	30801	8272	4412481	4790

3 Automatic Generating Method for Parallel MRGs

3.1 Constructing MRGs with Different Multipliers

It is well known that, if $f(x)$ is an irreducible polynomial, then $c^{-k} f(cx)$ and $x^k f(c/x)$ are also irreducible polynomials for any nonzero constant c. Using this fact, Deng, Li, and Shiau [4] developed an automatic generating algorithm that is most suitable for SG-primes. When p is an NSG-prime, this algorithm still works if the condition $\gcd(k, p-1) = 1$ holds. We remark that this condition holds for all of the NSG-primes listed in Table 1. One slight drawback is the number of different maximum-period MRGs can be produced is fewer than that with SG-primes. Next, we provide a simple alternative automatic generating algorithm for both SG-primes and NSG-primes.

Algorithm. *Let $f(x)$ in (2) be a primitive polynomial corresponding to a DL or a DT generator in which the nonzero coefficient is B as in Equation (3) or (5). The steps below will randomly generate a set of maximum-period MRGs.*

1. *Whenever a new processor is initiated, continue generating a new d_n via a simple LCG: $d_n = W d_{n-1} \bmod p$ until $\gcd(kd_n - 1, p-1) = 1$, where W is a primitive element modulus p.*
2. *Compute $c_n = B^{d_n} \bmod p$ and the primitive polynomial $G(x)$ by*

$$G(x) = c_n^{-k} f(c_n x) \equiv x^k - g_1 x^{k-1} - g_2 x^{k-2} - \cdots - g_k \ mod\ p. \tag{7}$$

3. *The new processor can use the newly constructed maximum-period MRG corresponding to the characteristic polynomial $G(x)$ in (7) as follows:*

$$X_i = g_1 X_{i-1} + \cdots + g_k X_{i-k} \ mod\ p. \tag{8}$$

When the baseline generator is a DL or DT generator, the generated MRGs have k nonzero terms. Next, we present efficient implementations for the constructed MRGs in (8).

3.2 Construction of Parallel MRGs from DL-k and DT-k

Given a maximum-period DL generator in (3), say, from Table 2, we can use the corresponding characteristic polynomial $f(x) = x^k - Bx^{k-1} - \cdots - Bx - B$ to create a set of k-th degree primitive polynomials as $G(x) = c^{-k} f(cx) \bmod p$ with c in the set of numbers $\{c_n\}$ obtained in Step 2 of the above algorithm. For a DL generator, $G(x)$ in (7) has $g_i = c^{-i} B \bmod p$ (or $g_i = c^{-1} g_{i-1} \bmod p$).

Then the MRG in (8) can be efficiently implemented by the following recurrence equation of order $(k+1)$ with only two nonzero terms:

$$X_i = (c^{-1} + g_1) X_{i-1} - c^{-1} g_k X_{i-(k+1)} \bmod p, \quad i \geq k+1.$$

Then, since $g_i = c^{-i} B \bmod p$, we have

$$X_i = (c^{-1}(B+1)) X_{i-1} - c^{-(k+1)} B X_{i-(k+1)} \bmod p, \quad i \geq k+1.$$

For the DT generator in (5), the characteristic polynomial is $f(x) = x^k - B^k x^{k-1} - \cdots - B^2 x - B$. Given a maximum-period DT generator in Table 3, we can similarly produce MRGs in (8), where

$$g_i = c^{-i} B^{k-i+1} \bmod p \quad (\text{or} \quad g_i = c^{-1} B^{-1} g_{i-1} \bmod p).$$

This MRG can be efficiently implemented by the recurrence of order $(k+1)$ below:

$$X_i = (c^{-1} B^{-1} + c^{-1} B^k) X_{i-1} - c^{-(k+1)} X_{i-(k+1)} \bmod p, \quad i \geq k+1,$$

which has only two nonzero terms left.

Acknowledgements The authors are grateful to Professor Pierre L'Ecuyer and two referees for their helpful suggestions. This work was done while the first author was visiting Institute of Statistics, National Chiao Tung University, Hsinchu, Taiwan, which was partially supported by the National Science Council in Taiwan, R.O.C., Grant No. NSC96-2118-M-009-004-MY2 and NSC97-2118-M-009-002-MY2.

References

1. Alanen, J.D., Knuth, D.E.: Tables of finite fields. Sankhyā **26**, 305–328 (1964)
2. Deng, L.Y.: Generalized mersenne prime number and its application to random number generation. In: H. Niederreiter (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 167–180. Springer-Verlag (2004)
3. Deng, L.Y.: Issues on computer search for large order multiple recursive generators. In: S. Heinrich, A. Keller, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 251–261. Springer-Verlag (2008)
4. Deng, L.Y., Li, H., Shiau, J.J.H.: Scalable parallel multiple recursive generators of large order. Parallel Computing **35**, 29–37 (2009)
5. Deng, L.Y., Li, H., Shiau, J.J.H., Tsai, G.H.: Design and implementation of efficient and portable multiple recursive generators with few zero coefficients. In: S. Heinrich, A. Keller, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 263–273. S. Heinrich and A. Keller and H. Niederreiter (2008)
6. Deng, L.Y., Xu, H.: A system of high-dimensional, efficient, long-cycle and portable uniform random number generators. ACM Transactions on Modeling and Computer Simulation **13**(4), 299–309 (2003)
7. Knuth, D.E.: The Art of Computer Programming, vol. 2: Seminumerical Algorithms, third edn. Addison-Wesley, Reading, MA. 1998)
8. L'Ecuyer, P.: Random numbers for simulation. Communications of the ACM **33**(10), 85–97 (1990)
9. L'Ecuyer, P.: Good parameter sets for combined multiple recursive random number generators. Operations Research **47**, 159–164 (1999)
10. L'Ecuyer, P., Blouin, F., Couture, R.: A search for good multiple recursive linear random number generators. ACM Transactions on Modeling and Computer Simulation **3**, 87–98 (1993)
11. L'Ecuyer, P., Simard, R., Chen, E.J., Kelton, W.D.: An objected-oriented random-number package with many long streams and substreams. Operations Research **50**(6), 1073–1075 (2002)
12. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications, revised edn. Cambridge University Press, Cambridge, MA. (1994)
13. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation **8**(1), 3–30 (1998)
14. Panneton, F., L'Ecuyer, P., Matsumoto, M.: Improved long-period generators based on linear recurrences modulo 2. ACM Transactions on Mathematical Software **32**(1), 1–16 (2006)
15. Srinivasan, A., Mascagni, M., Ceperley, D.: Tesitng parallel random number generators. Parallel Computing **29**, 69–94 (2003)

Efficient Numerical Inversion for Financial Simulations

Gerhard Derflinger, Wolfgang Hörmann, Josef Leydold, and Halis Sak

Abstract Generating samples from generalized hyperbolic distributions and noncentral chi-square distributions by inversion has become an important task for the simulation of recent models in finance in the framework of (quasi-) Monte Carlo. However, their distribution functions are quite expensive to evaluate and thus numerical methods like root finding algorithms are extremely slow. In this paper we demonstrate how our new method based on Newton interpolation and Gauss-Lobatto quadrature can be utilized for financial applications. Its fast marginal generation times make it competitive, even for situations where the parameters are not always constant.

1 Introduction

The evaluation of the quantile function of a given distribution is an inevitable task in the framework of Quasi-Monte Carlo methods (QMC) or for copula methods. Unfortunately, fast and accurate implementations of such functions are available only for a few of the important distributions, e.g., for the Gaussian distribution. Otherwise, one has to invert one's cumulative distribution function (CDF), for which it is usually easier to find a ready-to-use implementation (or at least a published algorithm). This procedure is called the *inversion method* in the random variate generation literature. Thus one has to apply some numerical root finding algorithms,

Gerhard Derflinger · Josef Leydold · Halis Sak
Department of Statistics and Mathematics, WU (Vienna University of Economics and Business), Augasse 2-6, A-1090 Vienna, Austria
e-mail: Gerhard.Derflinger@wu.ac.at
e-mail: Josef.Leydold@wu.ac.at
e-mail: Halis.Sak@wu.ac.at

Wolfgang Hörmann
Department of Industrial Engineering, Boğaziçi University, 34342 Bebek-Istanbul, Turkey
e-mail: hormannw@boun.edu.tr

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_18,

usually Newton's method, variants of the secant method or interval bisection. Such an approach is even necessary for standard distributions having shape parameters. Then numerical inversion is combined with rough approximations to get starting points for recursive algorithms, see, e.g., the implementation of quantile functions for Gamma and t distributions in R [12].

For the *fixed parameter* case, i.e., when we have to draw (large) samples from the same distribution, a table based approach combined with interpolation is much faster. Possible implementations are proposed in [2, 6, 14]. Although the computation of the necessary coefficients during the setup can be quite expensive these methods are applicable to all distributions that fulfill some regularity conditions (like smooth and bounded density) and they have fast marginal generation times that hardly depend on the target distribution.

Recent developments of more realistic models for the dynamics of asset prices, interest or exchange rates lead to an increased application of less frequently used distributions. During the MCQMC'08 conference in Montreal we were impressed to see how many talks dealt with models that require the simulation of the variance gamma distribution or more generally of the generalized hyperbolic distribution. Also several authors report the good fit of the generalized hyperbolic distribution to daily stock-returns (see, e.g., [1] and [3]). Both at the conference and in the literature we observed that most researchers seemingly considered generating generalized hyperbolic random variates by inversion as too slow or even impossible. An exception was the talk by Tan [7] who used the algorithm described in [6]. He also mentioned that the setup took a long time. The slow setup is due to the fact that this algorithm requires the CDF which is extremely expensive for the generalized hyperbolic distribution. This talk motivated us to demonstrate the practical application of our recently proposed algorithm [5] to such distributions. It only requires the probability density function (PDF) of the target distribution and computes the CDF during the setup. The synergy of using interpolation together with numerical integration speeds up the setup for the generalized hyperbolic distribution by about hundred times compared to methods that are based on the direct evaluation of the CDF. Moreover, the algorithm allows to control the accuracy of the approximate inverse CDF.

A second numerically difficult distribution required in financial simulations is the non-central chi-square distribution. It is, e.g., required for simulating the increments of the well known Cox-Ingersoll-Ross model for the dynamics of short-term interest rates. Here the application of inversion algorithms is more difficult as the non-centrality parameters λ varies. Nevertheless, as the value of λ is close to 0 for all practically relevant choices of the parameters of that process, we were able to apply our algorithm for this simulation problem.

In this paper we explain the main idea of our newly proposed algorithm and discuss how an inaccurate evaluation of the PDF influences the error of the algorithm. We then develop the details necessary for its application to two simulation problems of quantitative finance requiring the generalized hyperbolic distribution with fixed parameters and the non-central chi-square distribution. A ready-to-use implementation of our new black-box algorithm can be found as method PINV in our library UNU.RAN [9, 10].

All our experiments were performed in R as it provides a convenient platform for interactive computing. Densities for our target distributions are already available and the package *Runuran* [9] provides an interface to our C library. Of course our experiments can be conducted using C or any appropriate computing environment that provides an API to use a C library.

2 The Automatic Algorithm

Our algorithm has been designed as a black-box algorithm, i.e., the user has to provide a function that evaluates the PDF together with a "typical" point of the target distribution, and a maximal tolerated approximation error. As it is not tailored for a particular distribution family it works for distributions with smooth and bounded densities but requires some setup where the corresponding tables have to be computed. We only sketch the basic idea and refer the reader to [5] for all details (including the pseudo-code) and for arguments for the particular choice of the interpolation method and quadrature rule.

Measuring the Accuracy of Approximate Inversion

A main concern of any numerical inversion algorithm must be the control of the approximation error, i.e., the deviation of the approximate inverse CDF F_a^{-1} from the exact function F^{-1}. We are convinced that the *u-error* defined by

$$\varepsilon_u(u) = |u - F(F_a^{-1}(u))| \tag{1}$$

is well-suited for this task. In particular it can easily be computed during the setup and it can be interpreted with respect to the resolution of the underlying uniform pseudo-random number generator or low discrepancy set (see [5] for details). We therefore call the maximal tolerated u-error the *u-resolution* of the algorithm and denote it by ε_u in the sequel. We should mention here that the x-error, $|F^{-1}(u) - F_a^{-1}(u)|$, may be large in the tails of the target distribution. Hence our algorithms are not designed for calculating exact quantiles in the far tails of the distribution.

Newton's Interpolation Formula and Gauss-Lobatto Quadrature

For an interval $[b_l, b_r]$ we select a fixed number of points $b_l = x_0 < x_1 < \cdots < x_n = b_r$ and compute $u_i = F(x_i) = F(x_{i-1}) + \int_{x_{i-1}}^{x_i} f(x)\,dx$ recursively using $u_0 = 0$. The numeric integration is performed by means of Gauss-Lobatto quadrature with 5 points. The integration error is typically much smaller than the interpolation error and can be controlled using adaptive integration. We then construct a polynomial $P_n(x)$ of order n through the $n+1$ pairs (u_i, x_i), thus avoiding the evaluation

of the inverse CDF F^{-1}. The coefficients of the polynomial are calculated using inverse Newton interpolation. Note that using numeric integration is often more stable than the direct use of an accurate implementation of the CDF due to loss of significant digits in the right tail. The interpolation error can be computed during the setup. It is possible to search for the maximal error over $[F(b_l), F(b_r)]$, but we suggest to use a much cheaper heuristic, that estimates the location of the maximal error using the roots of Chebyshev polynomials. The intervals $[b_l, b_r]$ are constructed sequentially from left to right in the setup. The length of every interval is shortened till the estimated u-error is slightly smaller than the required u-resolution.

Cut-off Points for the Domain

The interpolation does not work for densities where the inverse CDF becomes too steep. In particular this happens in the tails of distributions with unbounded domains. Thus we have to find the computational relevant part of the domain, i.e., we have to cut off the tails such that the probability of either tail is negligible, say about 5% of the given u-resolution ε_u. Thus it does not increase the u-error significantly. We approximate the tail of the distribution by a function of the form x^{-c} fitted to the tail in a starting point x_0 and take the cut-off value of that approximation.

3 Considerations for Approximate Densities

The error control of our algorithm assumes that the density can be evaluated precisely. However, in practice we only have a (albeit accurate) approximate PDF $f_e(x)$ available. The pointwise error of the corresponding approximate CDF F_e is bounded by the L_1-error of f_e which is defined by

$$|F(x) - F_e(x)| \le L_1\text{-error} = \varepsilon_1 = \int_{-\infty}^{\infty} |f(x) - f_e(x)|\,dx\ .$$

Hence, the total resulting maximal u-error of the approximation F_a^{-1} is bounded by $\varepsilon_u + \varepsilon_1$. Thus, if we can obtain an upper bound $\tilde{\varepsilon}_1$ for the L_1-error, we can reduce the parameter ε_u of our algorithm by $\tilde{\varepsilon}_1$ to get an algorithm that is guaranteed to have the required u-resolution. (Of course this requires that ε_1 is sufficiently small.)

The L_1-error can be estimated by means of high precision arithmetic (we used *Mathematica* [15] for this task). We compared our implementation of a particular PDF with one that computes 30 significant decimal digits at 500,000 equidistributed points and calculated the L_1-error.

4 The Generalized Hyperbolic Distribution

The generalized hyperbolic distribution is considered to be a more realistic albeit numerically difficult model for the increment of financial processes or the marginal distributions of portfolio risk. However, for both, the generation of a variance gamma process using QMC as well as the generation of the marginal distribution from (e.g.) a t-copula the inversion method is inevitable. We therefore discuss here the application of our numerical inversion algorithm to that important distribution family.

Our tests were performed with the parameter values estimated for four different German stocks in [11]. We first estimated the L_1-errors of our implementation of the PDF (using the R library function for the modified Bessel function of third kind) which was always below 10^{-15}, i.e., close to machine precision of the double format of the IEEE floating point standard.

Our numerical inversion algorithm works for all parameter sets. A u-resolution of 10^{-12} and an order $n = 5$ for the polynomial interpolation requires 140 intervals and the setup was executed in less than 0.2 seconds. This is about 100 times faster than using the inversion algorithm of [6] that requires the CDF instead of the PDF. The observed u-error remained always below the required u-resolution (we used R package *ghyp* [4] as an independent implementation of the CDF of the generalized hyperbolic distribution). The marginal execution time is very fast, less than 0.2 seconds for generating one million variates. So including the setup we are able to sample 10^7 variates of a generalized hyperbolic distribution in less than 2 seconds which is quite fast. Compared to inversion using the quantile function of the *ghyp* package, that requires 35 seconds to generate 1000 variates, the speed up factor is 10^5 which is impressive.

This means that for the realistic return model using the t-copula with generalized hyperbolic marginals our new numeric inversion allows to obtain precise estimates of value at risk or expected shortfall in acceptable time. In that model the return distribution of the d assets of the portfolio has different parameters. As the generation of a return vector requires the evaluation of the inverse CDF of each of these d parameter sets it is therefore necessary to start the simulation with calculating d sets of constants that are stored in d different "generator objects". Thus our numeric inversion algorithm can be used in this semi-varying parameter situation, also values of d around 50 or 100 are no problem.

5 The Non-Central Chi-Square Distribution

The density of the non-central chi-square distribution is difficult to evaluate. Its R implementation uses the representation of the non-central chi-square distribution as a Poisson mixture of central chi-square distributions [8, p. 436]. The density is bounded for $\nu \geq 2$ degrees of freedom. Our numerical estimation shows that the L_1-error of the R implementation is never larger than 10^{-14}. So our algorithm can be applied to get a fast and sufficiently accurate inversion algorithm for the non-

central chi-square distribution. An u-resolution of 10^{-12} and an order $n = 5$ for the polynomial interpolation requires a bit more than 200 intervals for all parameter values of our experiments. The setup together with the generation of 10^6 variates by inversion took about 0.27 seconds and was thus faster than generating 100 variates by inversion using the built in R function `qchisq`. So here the algorithm reached a speed-up factor above 10^4. Note that our inversion algorithm is also about 30 percent faster than using the random variate generation function `rchisq` that is using the mixture representation of the non-central chi-square distribution and is therefore not an inversion algorithm.

Random Variate Generation for the CIR Model

The Cox-Ingersoll-Ross or CIR model is a well known single factor interest rate model defined by the stochastic differential equation

$$dr_t = \kappa(\theta - r_t)dt + \sigma\sqrt{r_t}d\omega_t \ .$$

The increments of the process r_t for a time jump of size T follow a multiple of the non-central chi-square distribution with $\nu = 4\kappa\theta/\sigma^2$ degrees of freedom and non-centrality parameter $\lambda = (1 - \exp(-\kappa T))\sigma^2 r_t \exp(-\kappa T)/\kappa$.

So path simulation requires only the generation of non-central chi-square variates. For fixed process parameters, ν is fixed, whereas λ depends on the current value r_t and thus changes for every call. The changing parameter situation seems to be an obstacle for using a table based inversion algorithm. But a closer look at the formula for λ shows that for sensible process parameters, λ is always close to 0. To check this observation we investigated a total of 24 parameter sets estimated in [13] for the overnight rates of European interbank, of London interbank for Euro, and the overnight rates for Poland, Slovakia, Hungary and the Czech Republic. The parameters were estimated for the four quarters of 2003 separately. Assuming that r_t is not larger than 10θ and trying values of T between 0.001 and 1 it turns out that we always had $\lambda < 0.011$. As ν is fixed and the range of possible λ values is so small, we can use our inversion algorithm for this varying parameter situation. The simplest approach is to run the setup of the algorithm for the parameters $(\nu, \lambda = 0)$ and for $(\nu, \lambda = 0.011)$. To make inversion for an arbitrary value of λ in $(0, 0.011)$ we calculate both x-values, that for $(\nu, \lambda = 0)$ and $(\nu, \lambda = 0.011)$, and then use linear interpolation in λ. It is enough to use $\varepsilon_u = 10^{-10}$ such that the result of the simple linear interpolation in λ has an u-error smaller than 10^{-7} for $\nu \geq 6$ and smaller than $3.5 \cdot 10^{-7}$ for $\nu \geq 3$.

Remark 1. As λ is so close to zero we first thought that it could be enough to use $\lambda = 0$ (i.e., inversion with the ordinary chi-square distribution) as approximation. But that simple approximation leads, e.g., for $\nu = 4$ and $\lambda = 0.005$ to a u-error larger than 10^{-3} which is certainly not acceptable.

If smaller u-errors or larger λ values are required one may use quadratic interpolation in λ. For example for $\lambda \leq 0.1$ we run the setup and store the respective

```
library(Runuran)   ## load library

qnccsi <- function(u, nu, lambda) {
   ## approx. inverse CDF of non-central chi-square distribution.
   ##   u      ... probabilities (vector of length n)
   ##   nu     ... degrees of freedoms (numeric)
   ##   lambda ... non-central values (vector of lenght n)
   ## quadratic interpolation for lambda.
   ## (default) uresolution=1e-10 for generator PINV

   # maximum of non-centrality parameter
   maxlambda <- max(lambda)
   # "typical" point of distributions
   xc <- 0.5*nu
   # generators for distributions
   myf0 <- function(x) dchisq(x,df=nu,ncp=0)
   gen0 <- pinv.new(pdf=myf0,lb=0,ub=Inf,center=xc)
   myf1 <- function(x) dchisq(x,df=nu,ncp=maxlambda*0.5)
   gen1 <- pinv.new(pdf=myf1,lb=0,ub=Inf,center=xc)
   myf2 <- function(x) dchisq(x,df=nu,ncp=maxlambda)
   gen2 <- pinv.new(pdf=myf2,lb=0,ub=Inf,center=xc)
   # generate points from these distributions
   x0<-uq(gen0,u); x1<-uq(gen1,u); x2<-uq(gen2,u)
   # interpolate for particular non-centrality parameters
   lam <- lambda*2/maxlambda-1
   x <- 0.5*(((x0+x2)*lam+(x2-x0))*lam)+x1*(1-lam*lam)
   # return random sample
   x
}

## draw a random sample with randomly selected lambda values
x <- qnccsi(u=runif(1e6),nu=20,lambda=runif(1e6)*0.1)
```

Fig. 1 R code for approximate inverse CDF of a non-central chi-square distribution with `nu` degrees of freedom (fixed) and varying non-centrality parameter `lambda`. (This code requires R version 2.8.1 or later since otherwise `dchisq` hangs.)

constants for $\lambda_0 = 0$, $\lambda_1 = 0.05$, and $\lambda_2 = 0.1$. For an arbitrary $\lambda \le 0.1$ and u we evaluate the inverse CDF and calculate $x_i = F^{-1}(u, \lambda_i)$ for $i = 0, 1, 2$ using the three stored tables. The final result is then obtained using quadratic interpolation of the three pairs (λ_i, x_i). Figure 1 lists an R function that implements such an approach. Using quadratic interpolation in λ we observed an u-error smaller than $1.5 \cdot 10^{-7}$ for $\nu \ge 3$, $\lambda \le 0.1$ and an u-error smaller than $2.3 \cdot 10^{-8}$ for $\nu \ge 6$, $\lambda \le 0.1$. The u-error is more than 100 times smaller than when using linear interpolation for those parameter values. Of course the quadratic interpolation and the evaluation of three quantiles takes some time. Generating 10^6 variates for varying λ-values with the R code in Fig. 1 takes about 0.7 seconds. Compared to the 2.5 seconds it takes to generate 1000 of the same variates with the built-in quantile function of R we can still observe a speed-up factor above 3000.

References

1. Aas, K., Haff, I.H.: The generalized hyperbolic skew Student's t-distribution. Journal of Financial Econometrics **4**(2), 275–309 (2006)
2. Ahrens, J.H., Kohrt, K.D.: Computer methods for efficient sampling from largely arbitrary statistical distributions. Computing **26**, 19–31 (1981)
3. Behr, A., Pötter, U.: Alternatives to the normal model of stock returns: Gaussian mixture, generalised logF and generalised hyperbolic models. Annals of Finance **5**(1), 49–68 (2009). DOI 10.1007/s10436-007-0089-8
4. Breymann, W., Lüthi, D.: ghyp: A package on generalized hyperbolic distributions. Tech. rep., Institute of Data Analysis and Process Design (2008). http://cran.r-project.org/
5. Derflinger, G., Hörmann, W., Leydold, J.: Random variate generation by numerical inversion when only the density is known. ACM Trans. Model. Comput. Simul. (2009), to appear. Preprint available at http://epub.wu-wien.ac.at/english/
6. Hörmann, W., Leydold, J.: Continuous random variate generation by fast numerical inversion. ACM Transactions on Modeling and Computer Simulation **13**(4), 347–362 (2003)
7. Imai, J., Tan, K.S.: An enhanced quasi-Monte Carlo method for simulating generalized hyperbolic Levy process (2008). Talk at the MCQMC08 in Montreal
8. Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous Univariate Distributions, vol. 2, 2nd edn. Wiley, New York (1995)
9. Leydold, J., Hörmann, W.: Runuran – R interface to the UNU. RAN random variate generators, Version 0.8. Department of Statistics and Mathematics, WU Wien, A-1090 Wien, Austria (2008). http://cran.r-project.org/
10. Leydold, J., Hörmann, W.: UNU.RAN – A Library for Non-Uniform Universal Random Variate Generation, Version 1.3. Department of Statistics and Mathematics, WU Wien, A-1090 Wien, Austria (2008). http://statmath.wu-wien.ac.at/unuran/
11. Prause, K.: Modelling financial data using generalized hyperbolic distributions. Tech. rep., University of Freiburg (1997)
12. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008). http://www.R-project.org. ISBN 3-900051-07-0
13. Ševčovič, D., Urbánová Csajková, A.: On a two-pahse minmax method for parameter estimation of the cox, ingersoll, and ross interest rate model. Central European Journal of Operations Research **13**, 169–188 (2005)
14. Ulrich, G., Watson, L.: A method for computer generation of variates from arbitrary continuous distributions. SIAM J. Sci. Statist. Comput. **8**, 185–197 (1987)
15. Wolfram Research, Inc.: Mathematica, Version 6.0. Champaign, IL (2007)

Equidistribution Properties of Generalized Nets and Sequences

Josef Dick and Jan Baldeaux

Abstract Generalized digital nets and sequences have been introduced for the numerical integration of smooth functions using quasi-Monte Carlo rules. In this paper we study geometrical properties of such nets and sequences. The definition of these nets and sequences does not depend on linear algebra over finite fields, it only requires that the point set or sequence satisfies certain distributional properties. Generalized digital nets and sequences appear as special cases. We prove some propagation rules and give bounds on the quality parameter t.

1 Introduction

In this paper we study the equidistribution properties of generalized digital nets and sequences as introduced in [2], see also [1, 3]. Such nets and sequences have been introduced since they can achieve arbitrarily high convergence rates of the integration error when used in a quasi-Monte Carlo rule as quadrature points. To be more precise, if the function $f : [0,1]^s \to \mathbb{R}$, $s \geq 1$, under consideration has mixed partial derivatives up to order $\alpha \geq 1$ in each variable which are square-integrable, then the integration error is of $\mathcal{O}(q^{-(\beta n - t)}(\beta n - t)^{s\alpha})$, for a digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$. Explicit constructions of digital $(t, \alpha, \beta, n \times m, s)$-nets over $\mathbb{F}_q$ with $\beta n = \alpha m$ and t bounded independently of m are also given in [1, 2]. Note that a digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$ has q^m points.

In the next section we define digital $(t, \alpha, \beta, n \times m, s)$-nets and digital $(t, \alpha, \beta, \sigma, s)$-sequences and recall some of their properties as well as explicit con-

Josef Dick
School of Mathematics and Statistics, The University of New South Wales, Sydney, NSW, Australia
e-mail: josef.dick@unsw.edu.au

Jan Baldeaux
School of Mathematics and Statistics, The University of New South Wales, Sydney, NSW, Australia
e-mail: Jan.Baldeaux@student.unsw.edu.au

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_19,

structions from [2]. In Section 3, generalized nets and sequences are introduced. In order to do so, we introduce the concept of a generalized elementary interval in Subsection 3.1. We prove some properties of such sets and then give the definition of $(t, \alpha, \beta, n, m, s)$-nets and $(t, \alpha, \beta, \sigma, s)$-sequences. In Subsection 3.2, propagation rules for these types of point sets and sequences are shown and we also prove some lower and upper bounds on the quality parameter t. In particular, we show that the quality parameter t of a $(t, \alpha, \beta, \sigma, s)$-sequence with smallest possible value of t satisfies $t \asymp \alpha^2 s$, which also holds for digital sequences. For the remainder of the paper we use the following nomenclature: $(t, \alpha, \beta, n, m, s)$-nets and $(t, \alpha, \beta, \sigma, s)$-sequences as introduced in Section 3 of this paper will be referred to as *generalized nets* and *generalized sequences*, digital $(t, \alpha, \beta, n \times m, s)$-nets and digital $(t, \alpha, \beta, \sigma, s)$-sequences, as introduced in [2], will be referred to as *generalized digital nets* and *generalized digital sequences*, (t, m, s)-nets and (t, s)-sequences, [9, 10] will be referred to as *classical nets* and *classical sequences*, digital (t, m, s)-nets and digital (t, s)-sequences, [9, 10] as *classical digital nets* and *classical digital sequences*.

2 Definition of Digital $(t, \alpha, \beta, n \times m, s)$-Nets and Digital $(t, \alpha, \beta, \sigma, s)$-Sequences

Before providing a geometric approach to digital $(t, \alpha, \beta, n \times m, s)$-nets, we need to recall the following concepts: We start with the digital construction scheme, which digital $(t, \alpha, \beta, n \times m, s)$-nets are based upon. This digital construction scheme stems from the construction of digital (t, m, s)-nets, see [10].

Throughout the paper $\mathbb{N}$ denotes the set of natural numbers and $\mathbb{N}_0$ the set of nonnegative integers. Having defined digital $(t, \alpha, \beta, n \times m, s)$-nets and digital $(t, \alpha, \beta, \sigma, s)$-sequences, we will explain the meaning of the parameters in Remark 1.

Definition 1. Let q be a prime power and let $n, m, s \geq 1$ be integers. Let $C_1, \ldots, C_s$ be $n \times m$ matrices over the finite field $\mathbb{F}_q$ of order q. Now we construct q^m points in $[0, 1)^s$: For $0 \leq h < q^m$ let $h = h_0 + h_1 q + \cdots + h_{m-1} q^{m-1}$ be the q-adic expansion of h. Consider an arbitrary but fixed bijection $\varphi : \{0, 1, \ldots, q-1\} \to \mathbb{F}_q$. Identify h with the vector $\mathbf{h} = (\varphi(h_0), \ldots, \varphi(h_{m-1}))^\top \in \mathbb{F}_q^m$, where $\top$ denotes the transpose of the vector. For $1 \leq j \leq s$, multiply the matrix C_j by $\mathbf{h}$, i.e.,

$$C_j \mathbf{h} := (y_{j,1}(h), \ldots, y_{j,n}(h))^\top \in \mathbb{F}_q^n,$$

and set

$$x_{h,j} := \frac{\varphi^{-1}(y_{j,1}(h))}{q} + \cdots + \frac{\varphi^{-1}(y_{j,n}(h))}{q^n}.$$

The point set $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{q^m-1}\}$ is called a digital net (over $\mathbb{F}_q$) (with generating matrices $C_1, \ldots, C_s$). For $n, m = \infty$ we obtain a sequence $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\}$, which is called a digital sequence (over $\mathbb{F}_q$) (with generating matrices $C_1, \ldots, C_s$).

It is clear from the definition, that all the information about the properties of the point set is contained in the generating matrices $C_1, \ldots, C_s$. Hence in order to be able to deal with the properties of these point sets, it is enough to introduce a criterion on the generating matrices. To define such a criterion we first define the dual space [4, 5, 11] of the generating matrices $C_1, \ldots, C_s \in \mathbb{F}_q^{n\times m}$ for a digital net, given by

$$\mathcal{D} = \{\boldsymbol{k} \in \mathbb{N}_0^s : C_1^\top \mathbf{k}_1 + \cdots + C_s^\top \mathbf{k}_s = \mathbf{0} \in \mathbb{F}_q^m\},$$

where for $\boldsymbol{k} = (k_1, \ldots, k_s)$ with $k_j = k_{j,0} + k_{j,1}q + \cdots$ we define the vector $\mathbf{k}_j = (k_{j,0}, \ldots, k_{j,n-1})^\top \in \mathbb{F}_q^n$.

The following criterion was first introduced in the context of applying digital nets to the numerical integration of smooth functions, see [2]: For $k \in \mathbb{N}$ and $\alpha \geq 1$ let $\mu_\alpha(k) = a_1 + \cdots + a_{\min(\nu,\alpha)}$, where $k = \kappa_1 q^{a_1-1} + \cdots + \kappa_\nu q^{a_\nu - 1}$ with $0 < \kappa_1, \ldots, \kappa_\nu < q$ and $1 \leq a_\nu < \cdots < a_1$. Further we set $\mu_\alpha(0) = 0$. For a vector $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s$ we define $\mu_\alpha(\boldsymbol{k}) = \mu_\alpha(k_1) + \cdots + \mu_\alpha(k_s)$. The following definition was motivated in [3].

Definition 2. Let $n, m, \alpha \in \mathbb{N}$, let $0 < \beta \leq \min(1, \alpha m/n)$ be a real number, and let $0 \leq t \leq \beta n$ be a nonnegative integer. Let $\mathbb{F}_q$ be the finite field of prime power order q and let $C_1, \ldots, C_s \in \mathbb{F}_q^{n\times m}$ with $C_j = (\mathbf{c}_{j,1}, \ldots, \mathbf{c}_{j,n})^T$. If for all $1 \leq i_{j,\nu_j} < \cdots < i_{j,1}$, where $0 \leq \nu_j$ for all $j = 1, \ldots, s$, with

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j,\alpha)} i_{j,l} \leq \beta n - t$$

the vectors

$$\mathbf{c}_{1,i_{1,\nu_1}}, \ldots, \mathbf{c}_{1,i_{1,1}}, \ldots, \mathbf{c}_{s,i_{s,\nu_s}}, \ldots, \mathbf{c}_{s,i_{s,1}}$$

are linearly independent over $\mathbb{F}_q$, then the digital net with generating matrices $C_1, \ldots, C_s$ is called a digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$.

If t is the smallest nonnegative integer such that the digital net generated by $C_1, \ldots, C_s$ is a digital $(t, \alpha, \beta, n \times m, s)$-net, then we call the digital net a strict digital $(t, \alpha, \beta, n \times m, s)$-net.

Note that the condition $\sum_{j=1}^{s} \sum_{l=1}^{\nu_j} i_{j,l} \leq \beta n - t$ implies that $i_{j,1} \leq n$, as $\beta \leq 1$ and $t \geq 0$.

Similarly, we can recall the definition of digital $(t, \alpha, \beta, \sigma, s)$-sequences over $\mathbb{F}_q$ from [2].

Definition 3. Let $\alpha, \sigma \geq 1$ and $t \geq 0$ be integers and let $0 < \beta \leq \alpha/\sigma$ be a real number. Let $\mathbb{F}_q$ be the finite field of prime power order q and let $C_1, \ldots, C_s \in \mathbb{F}_q^{\infty\times\infty}$ with $C_j = (\mathbf{c}_{j,1}, \mathbf{c}_{j,2}, \ldots)^\top$. Further let $C_{j,\sigma m\times m}$ denote the left upper $\sigma m \times m$ submatrix of C_j. If for all $m > t/(\beta\sigma)$ the matrices $C_{1,\sigma m\times m}, \ldots, C_{s,\sigma m\times m}$ generate a

digital $(t, \alpha, \beta, \sigma m \times m, s)$-net, then the digital sequence with generating matrices $C_1, \ldots, C_s$ is called a digital $(t, \alpha, \beta, \sigma, s)$-sequence over $\mathbb{F}_q$.

If t is the smallest nonnegative integer such that the digital sequence generated by $C_1, \ldots, C_s$ is a digital $(t, \alpha, \beta, \sigma, s)$-sequence, then we call the digital sequence a strict digital $(t, \alpha, \beta, \sigma, s)$-sequence.

Remark 1. In the following we explain the meaning of the parameters t, α, β, n, m and s used in the context of generalized digital $(t, \alpha, \beta, n \times m, s)$-nets; see also [2, Remark 4.5]:

- s denotes the dimensionality of the point set;
- the logarithm in base q of the number of points is m, i.e., a digital $(t, \alpha, \beta, n \times m, s)$-net has q^m points;
- n denotes the number of rows of the generating matrices and therefore corresponds to the maximum number of non-zero digits in the base q expansion of each coordinate of each point; hence n determines how precise each point is placed in the unit cube, which has a direct influence on the convergence of the integration error as can be seen from the next point;
- $\beta n - t$ denotes the quality of the point set, which can be referred to as the strength of the net; in particular, the integration error is $\mathcal{O}(q^{-\beta n + t}(\beta n - t)^{\alpha s})$;
- digital $(t, \alpha, \beta, n \times m, s)$-nets were introduced in the context of numerical integration, where α is a variable parameter, which denotes the smoothness of the integrand. We assume that the smoothness α is not known explicitly.

Finally, following [2], we now recall a method of explicitly constructing digital $(t, \alpha, \beta, n \times m, s)$-nets, which was first presented in [2, Section 4.4]. This way we obtain digital $(t, \alpha, \min(1, \alpha/d), dm \times m, s)$-nets for all $\alpha \geq 1$, where $d \in \mathbb{N}$ is a parameter which can be chosen freely.

Let $d \geq 1$ and let $C_1, \ldots, C_{sd}$ be the generating matrices of a digital (t', m, sd)-net; we recall that many explicit examples of such generating matrices are known, see e.g., [6, 7, 8, 10, 12, 18] and the references therein. As we will see later, the choice of the underlying digital (t', m, sd)-net has a direct impact on the bound on the t-value of the digital $(t, \alpha, \min(1, \frac{\alpha}{d}), dm \times m, s)$-net, which was proven in [2]. Let $C_j = (\mathbf{c}_{j,1}, \ldots, \mathbf{c}_{j,m})^\top$ for $j = 1, \ldots, sd$; i.e., $\mathbf{c}_{j,l}$ are the row vectors of C_j. Now let the matrix $C_j^{(d)}$ consist of the first rows of the matrices $C_{(j-1)d+1}, \ldots, C_{jd}$, then the second rows of $C_{(j-1)d+1}, \ldots, C_{jd}$, and so on, in the order described in the following: The matrix $C_j^{(d)}$ is a $dm \times m$ matrix; i.e., $C_j^{(d)} = (\mathbf{c}_{j,1}^{(d)}, \ldots, \mathbf{c}_{j,dm}^{(d)})^\top$, where $\mathbf{c}_{j,l}^{(d)} = \mathbf{c}_{u,v}$ with $l = (v-j)d + u$, $1 \leq v \leq m$, and $(j-1)d < u \leq jd$ for $l = 1, \ldots, dm$ and $j = 1, \ldots, s$. We remark that this construction can be extended to digital $(t, \alpha, \beta, \sigma, s)$-sequences by letting $\widetilde{C}_j = (\widetilde{\mathbf{c}}_{j,1}, \widetilde{\mathbf{c}}_{j,2}, \ldots)^\top$, for $j = 1, \ldots, sd$, denote the generating matrices of a digital (t', sd)-sequence; the resulting matrices $\widetilde{C}_j^{(d)}$, $j = 1, \ldots, s$, are now $\infty \times \infty$ matrices, where again we have $\widetilde{C}_j^{(d)} = (\widetilde{\mathbf{c}}_{j,1}^{(d)}, \widetilde{\mathbf{c}}_{j,2}^{(d)}, \ldots)^\top$, where $\widetilde{\mathbf{c}}_{j,l}^{(d)} = \widetilde{\mathbf{c}}_{u,v}$ with $l = (v-j)d + u$, $v \geq 1$, and $(j-1)d < u \leq jd$ for $l = 1, 2, \ldots$ and $j = 1, \ldots, s$.

The following result improves [2, Theorem 4.11] for some cases. For a proof see [4].

Theorem 1. *Let $d \geq 1$ be a natural number and let $C_1, \ldots, C_{sd}$ be the generating matrices of a digital (t', m, sd)-net over the finite field $\mathbb{F}_q$ of prime power order q. Let $C_1^{(d)}, \ldots, C_s^{(d)}$ be defined as above. Then for any $\alpha \in \mathbb{N}$, the matrices $C_1^{(d)}, \ldots, C_s^{(d)}$ are the generating matrices of a digital $(t, \alpha, \min(1, \alpha/d), dm \times m, s)$-net over $\mathbb{F}_q$ with*

$$t = \min(\alpha, d) \min\left(m, t' + \left\lfloor \frac{s(d-1)}{2} \right\rfloor\right). \tag{1}$$

Furthermore, the matrices $\widetilde{C}_1^{(d)}, \ldots, \widetilde{C}_s^{(d)}$ obtained from the generating matrices $\widetilde{C}_1, \ldots, \widetilde{C}_{sd}$ of a digital (t', sd)-sequence over $\mathbb{F}_q$ are the generating matrices of a digital $(t, \alpha, \min(1, \alpha/d), d, s)$-sequence over $\mathbb{F}_q$ with

$$t = \min(\alpha, d) \left(t' + \left\lfloor \frac{s(d-1)}{2} \right\rfloor\right).$$

In the following example we show that the above result cannot be improved on in general.

Example 1. Let $d = 2$ and $s = 1$ and generate a digital $(t, \alpha, \min(1, \alpha/2), 2m \times m, 1)$-net over $\mathbb{F}_q$ from a digital $(0, m, 2)$-net over $\mathbb{F}_q$ (such nets exist, for example one can take the Hammersley net). Then Theorem 1 implies that we can choose $t = \min(\alpha, 2)0 + \min(\alpha, 2)\lfloor 1 \cdot 1/2 \rfloor = 0$, which is already best possible.

On the other hand it can be checked that the bound on the t-value in Theorem 1 for particular digital nets is not necessarily best possible. That is, if we use a strict digital (t', m, sd)-net over $\mathbb{F}_q$ for the construction of the generating matrices $C_1^{(d)}, \ldots, C_s^{(d)}$, then these generating matrices do not necessarily generate a strict digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$, where t is given by (1). This is illustrated in the next example.

Example 2. The following matrices generate a strict digital $(1, 3, 4)$-net over $\mathbb{F}_2$ and stem from a Niederreiter-Xing sequence as implemented by Pirsic [16]:

$$C_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, C_3 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, C_4 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

Using the method described in [2, Section 4.4] with $d = 2$, we construct the generating matrices $C_1^{(2)}$ and $C_2^{(2)}$, which are given by:

$$C_1^{(2)} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, C_2^{(2)} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

For any $\alpha \geq 2$, Theorem 1 yields a digital $(4, \alpha, 1, 6 \times 3, 2)$-net and for $\alpha = 1$ a digital $(2, 1, 1/2, 6 \times 3, 2)$-net.

We now show that the exact t-value of this digital net is smaller than the one obtained from Theorem 1. It can be confirmed by inspection that the matrices $C_1^{(2)}$ and $C_2^{(2)}$ generate a digital $(2, \alpha, 1, 6 \times 3, 2)$-net for all $\alpha \geq 2$, by checking that for all $1 \leq i_{j,\nu_j} < \cdots < i_{j,1}$, where $0 \leq \nu_j$, $j = 1, 2$, with

$$\sum_{j=1}^{2} \sum_{l=1}^{\min(\nu_j, \alpha)} i_{j,l} \leq 6 - 2 = 4$$

the vectors $\mathbf{c}_{1,i_{1,\nu_1}}^{(2)}, \ldots, \mathbf{c}_{1,i_{1,1}}^{(2)}, \mathbf{c}_{2,i_{2,\nu_2}}^{(2)}, \ldots, \mathbf{c}_{2,i_{2,1}}^{(2)}$ are linearly independent over $\mathbb{F}_2$. Furthermore, it can be confirmed that the two matrices $C_1^{(2)}$ and $C_2^{(2)}$ do not generate a digital $(1, \alpha, 1, 6 \times 3, 2)$-net for any $\alpha \geq 2$, as for $\nu_1 = 0$, $\nu_2 = 2$, $i_{2,2} = 1$, $i_{2,1} = 4$, $\mathbf{c}_{2,i_{2,1}}^{(2)}$ and $\mathbf{c}_{2,i_{2,2}}^{(2)}$ are linearly dependent. Hence, for any $\alpha \geq 2$, the matrices $C_1^{(2)}$ and $C_2^{(2)}$ generate a strict digital $(2, \alpha, 1, 6 \times 3, 2)$-net.

For $\alpha = 1$ on the other hand, it can be checked that the matrices $C_1^{(2)}$ and $C_2^{(2)}$ generate a strict digital $(0, 1, 1/2, 6 \times 3, 2)$-net.

Thus, for this example, Theorem 1 does not yield the best possible result for any $\alpha \geq 1$.

Next we present an example which might be counterintuitive at first: We present a strict digital $(2, 3, 4)$-net, which generates a strict digital $(1, \alpha, 1, 6 \times 3, 2)$-net for any $\alpha \geq 2$, and, for $\alpha = 1$, a strict digital $(0, 1, 1/2, 6 \times 3, 2)$-net.

Example 3. The following matrices generate a strict digital $(2, 3, 4)$-net over $\mathbb{F}_2$:

$$K_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, K_2 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, K_3 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, K_4 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Using the method described in [2, Section 4.4] with $d = 2$, we construct the generating matrices $K_1^{(2)}$ and $K_2^{(2)}$, which are given by:

$$K_1^{(2)} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, K_2^{(2)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

For any $\alpha \geq 2$, Theorem 1 yields a digital $(6, \alpha, 1, 6 \times 3, 2)$-net, and for $\alpha = 1$ a digital $(3, 1, 1/2, 6 \times 3, 2)$-net.

As in Example 2, it can be confirmed by inspection that the matrices $K_1^{(2)}$ and $K_2^{(2)}$ generate a digital $(1, \alpha, 1, 6 \times 3, 2)$-net for all $\alpha \geq 2$. Furthermore, it can be

confirmed that the two matrices $K_1^{(2)}$ and $K_2^{(2)}$ do not generate a digital $(0, \alpha, 1, 6 \times 3, 2)$-net for $\alpha \geq 2$, as for $\nu_1 = 2$, $\nu_2 = 2$, $i_{1,2} = 1$, $i_{1,1} = 2$, $i_{2,2} = 1$ and $i_{2,1} = 2$, $\mathbf{k}_{1,i_{1,2}}^{(2)}$, $\mathbf{k}_{1,i_{1,1}}^{(2)}$, $\mathbf{k}_{2,i_{2,2}}^{(2)}$ and $\mathbf{k}_{2,i_{2,1}}^{(2)}$ are linearly dependent, where $\mathbf{k}_{j,i}^{(2)}$ denotes the ith row of the matrix $K_j^{(2)}$.

For $\alpha = 1$ on the other hand, it can be checked that the matrices $K_1^{(2)}$ and $K_2^{(2)}$ generate a strict digital $(0, 1, 1/2, 6 \times 3, 2)$-net.

The last two examples show that Theorem 1 does not always yield the best possible bounds on the t-value for digital $(t, \alpha, \beta, n \times m, s)$-nets constructed from particular classical digital nets. (This could mean that it might be possible to improve the bound on the t-value for generalized digital nets constructed from particular classical nets (or sequences).) On the other hand, at least for digital $(t, \alpha, \beta, \sigma, s)$-sequences, we will see below that Theorem 1 does yield the asymptotically optimal dependence of the t-value on α and s, see Theorem 7 below.

Remark 2. Note that even though the strict digital $(1, 3, 4)$-net used in Example 2 has a better t-value (in the classical sense) than the strict digital $(2, 3, 4)$-net in Example 3, the latter generates the better digital $(t, \alpha, 1, 6 \times 3, 2)$-net for any $\alpha \geq 2$, as measured by the generalized t-value. However, it is possible to find a strict digital $(1, 3, 4)$-net which generates a strict digital $(1, \alpha, 1, 6 \times 3, 2)$-net for any $\alpha \geq 2$. Consider for example

$$\tilde{K}_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \tilde{K}_2 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \tilde{K}_3 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tilde{K}_4 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Remark 3. It can be checked that the matrices $C_1^{(2)}$ and $C_2^{(2)}$ from Example 2 can also be interpreted as generating matrices of a digital $(0, 3, 2)$-net over $\mathbb{F}_2$. However, if we set $\tilde{C}_2 = C_2^{(2)}$, but

$$\tilde{C}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

we have an example of a strict digital $(2, \alpha, 1, 6 \times 3, 2)$-net over $\mathbb{F}_2$, $\alpha \geq 2$, which is a strict digital $(1, 3, 2)$-net.

3 Equidistribution Properties of Generalized Nets and Sequences

Generalized digital nets and sequences, as introduced in [2], rely on linear algebra over finite fields. The quality of such point sets is determined by linear in-

dependence properties of the generating matrices. In this section we remove this restriction by introducing the essential geometrical properties satisfied by digital $(t, \alpha, \beta, n \times m, s)$-nets and digital $(t, \alpha, \beta, \sigma, s)$-sequences. This is analogous to the link between (t, m, s)-nets and digital (t, m, s)-nets in the classical theory (or (t, s)-sequences and digital (t, s)-sequences), where the former includes the latter as a special case and (t, m, s)-nets (and (t, s)-sequences) are defined using only geometrical features of the point set.

3.1 Definition of $(t, \alpha, \beta, n, m, s)$-Nets and $(t, \alpha, \beta, \sigma, s)$-Sequences

We recall that the definition of (t, m, s)-nets is based on the concept of an elementary interval, see e.g. [10]. In the following we introduce a concept analogous to that of an elementary interval, namely that of a generalized elementary interval. Before we do so we need some notation: let $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_s)$, let $|\boldsymbol{\nu}|_1 = \sum_{j=1}^{s} \nu_j$, let $\mathbf{i}_{\boldsymbol{\nu}} = (i_{1,1}, \ldots, i_{1,\nu_1}, \ldots, i_{s,1}, \ldots, i_{s,\nu_s})$, let $\mathbf{a}_{\boldsymbol{\nu}} \in \{0, \ldots, q-1\}^{|\boldsymbol{\nu}|_1}$, and let $\mathbf{a}_{\boldsymbol{\nu}} = (a_{1,i_{1,1}}, \ldots, a_{1,i_{1,\nu_1}}, \ldots, a_{s,i_{s,1}}, \ldots, a_{s,i_{s,\nu_s}})$, where the components $i_{j,l}$ and $a_{j,l}$, $l = 1, \ldots, \nu_j$, do not appear in the vectors $\mathbf{i}_{\boldsymbol{\nu}}$ and $\mathbf{a}_{\boldsymbol{\nu}}$ in case $\nu_j = 0$.

By a *generalized elementary interval* we mean a subset of $[0, 1)^s$ of the form

$$J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}}) = \prod_{j=1}^{s} \bigcup_{\substack{a_{j,l}=0 \\ l \in \{1,\ldots,n\} \setminus \{i_{j,1},\ldots,i_{j,\nu_j}\}}}^{q-1} \left[\frac{a_{j,1}}{q} + \cdots + \frac{a_{j,n}}{q^n}, \frac{a_{j,1}}{q} + \cdots + \frac{a_{j,n}}{q^n} + \frac{1}{q^n} \right),$$

where $q \geq 2$ is an integer and where for $j = 1, \ldots, s$ we have $1 \leq i_{j,\nu_j} < \cdots < i_{j,1} \leq n$ in case $\nu_j > 0$ and $\{i_{j,1}, \ldots, i_{j,\nu_j}\} = \emptyset$ in case $\nu_j = 0$.

We note that a generalized elementary interval is not always an elementary interval, but can be a union of several elementary intervals, see for example Figure 1.

Generalized elementary intervals posses properties similar to those of classical elementary intervals as we show in the following.

Lemma 1. *Let $\boldsymbol{\nu} \in \{0, \ldots, n\}^s$ and $\mathbf{i}_{\boldsymbol{\nu}}$ be defined as above and fixed. Then the generalized elementary intervals $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ for $\mathbf{a}_{\boldsymbol{\nu}} \in \{0, \ldots, q-1\}^{|\boldsymbol{\nu}|_1}$, form a partition of $[0, 1)^s$, i.e. $\bigcup_{\mathbf{a}_{\boldsymbol{\nu}} \in \{0,\ldots,q-1\}^{|\boldsymbol{\nu}|_1}} J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}}) = [0, 1)^s$ and $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}}) \cap J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}'_{\boldsymbol{\nu}}) = \emptyset$, for all $\mathbf{a}_{\boldsymbol{\nu}} \neq \mathbf{a}'_{\boldsymbol{\nu}} \in \{0, \ldots, q-1\}^{|\boldsymbol{\nu}|_1}$.*

Proof. First we have

$$\bigcup_{\mathbf{a}_{\boldsymbol{\nu}} \in \{0,\ldots,q-1\}^{|\boldsymbol{\nu}|_1}} J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$$

$$= \prod_{j=1}^{s} \bigcup_{\substack{a_{j,l}=0 \\ l\in\{1,\dots,n\}}}^{q-1} \left[\frac{a_{j,1}}{q} + \cdots + \frac{a_{j,n}}{q^n}, \frac{a_{j,1}}{q} + \cdots + \frac{a_{j,n}}{q^n} + \frac{1}{q^n} \right)$$
$$= [0,1)^s.$$

To show the second part, we note that, for $\mathbf{i}_{\boldsymbol{\nu}}$ fixed and $\mathbf{a}_{\boldsymbol{\nu}} \neq \mathbf{a}'_{\boldsymbol{\nu}}$, there exists a $j \in \{1,\dots,s\}$, and a $k \in \left\{i_{j,1},\dots,i_{j,\nu_j}\right\}$, such that $a_{j,k} \neq a'_{j,k}$. Let $\mathbf{x} = (x_1,\dots,x_s)$ where each coordinate x_j, $j = 1,\dots,s$, has base q expansion $x_j = x_{j,1}q^{-1} + x_{j,2}q^{-2} + \dots$ (we assume that for each $j \in \{1,\dots,s\}$ infinitely many $x_{j,k} \neq q-1$). Then $\mathbf{x} \in J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ if and only if for all $j = 1,\dots,s$ and all $k \in \{i_{j,1},\dots,i_{j,\nu_j}\}$ we have $x_{j,k} = a_{j,k}$. But as there exists a j and k such that $a_{j,k} \neq a'_{j,k}$, $\mathbf{x}$ cannot be in $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ and $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}'_{\boldsymbol{\nu}})$ simultaneously. Hence $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}}) \cap J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}'_{\boldsymbol{\nu}}) = \emptyset$ and the result follows.

□

In the following lemma, we compute the volume of a generalized elementary interval.

Lemma 2. *Let $\boldsymbol{\nu}$, $\mathbf{i}_{\boldsymbol{\nu}}$ and $\mathbf{a}_{\boldsymbol{\nu}}$ be as above. Then the volume of $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ is $q^{-|\boldsymbol{\nu}|_1}$.*

Proof. Let $\boldsymbol{\nu}$ and $\mathbf{i}_{\boldsymbol{\nu}}$ be fixed. Then we have seen in Lemma 1 that the $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$, $\mathbf{a}_{\boldsymbol{\nu}} \in \{0,\dots,q-1\}^{|\boldsymbol{\nu}|_1}$ form a partition of $[0,1)^s$. From the definition of generalized elementary intervals one can see that $\mathrm{Vol}(J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})) = \mathrm{Vol}(J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}'_{\boldsymbol{\nu}}))$ for all $\mathbf{a}_{\boldsymbol{\nu}}, \mathbf{a}'_{\boldsymbol{\nu}} \in \{0,\dots,q-1\}^{|\boldsymbol{\nu}|_1}$, where $\mathrm{Vol}(J)$ denotes the volume of an interval J, as the intervals $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ and $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}'_{\boldsymbol{\nu}})$ are only shifted versions of each other. Hence

$$\mathrm{Vol}(J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})) = \frac{1}{|\{\mathbf{a}_{\boldsymbol{\nu}} \in \{0,\dots,q-1\}^{|\boldsymbol{\nu}|_1}\}|} = \frac{1}{q^{|\boldsymbol{\nu}|_1}}.$$

□

We are now in a position to define a (t,α,β,n,m,s)-net, which is based on the concept of a generalized elementary interval and Lemma 2.

Definition 4. Let $n, m, \alpha \geq 1$ be natural numbers, let $0 < \beta \leq 1$ be a real number, and let $0 \leq t \leq \beta n$ be an integer. Let $q \geq 2$ be an integer and $P = \{\boldsymbol{x}_0,\dots,\boldsymbol{x}_{q^m-1}\} \subseteq [0,1)^s$ be a point set in the s-dimensional unit cube, $s \geq 1$. We say that P is a (t,α,β,n,m,s)-net (in base q), if for all integers $1 \leq i_{j,\nu_j} < \cdots < i_{j,1}$, where $\nu_j \geq 0$, with

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j,\alpha)} i_{j,l} \leq \beta n - t,$$

where for $\nu_j = 0$ we set the empty sum $\sum_{l=1}^{0} i_{j,l} = 0$, the generalized elementary interval $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ contains exactly $q^{m-|\boldsymbol{\nu}|_1}$ points of P for each $\mathbf{a}_{\boldsymbol{\nu}} \in \{0,\dots,q-1\}^{|\boldsymbol{\nu}|_1}$.

Remark 4. Note that $q^{m-|\boldsymbol{\nu}|_1} = q^m \mathrm{Vol}(J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}}))$. For an interval $J \subseteq [0,1)^s$ and a point set $P \subset [0,1)^s$, let $|P(J)|$ denote the number of points of P in J. Then Definition 4 says that the proportion of points of P in $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$, which is given by $|P(J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}}))|/|P([0,1)^s)|$, equals the volume of $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$.

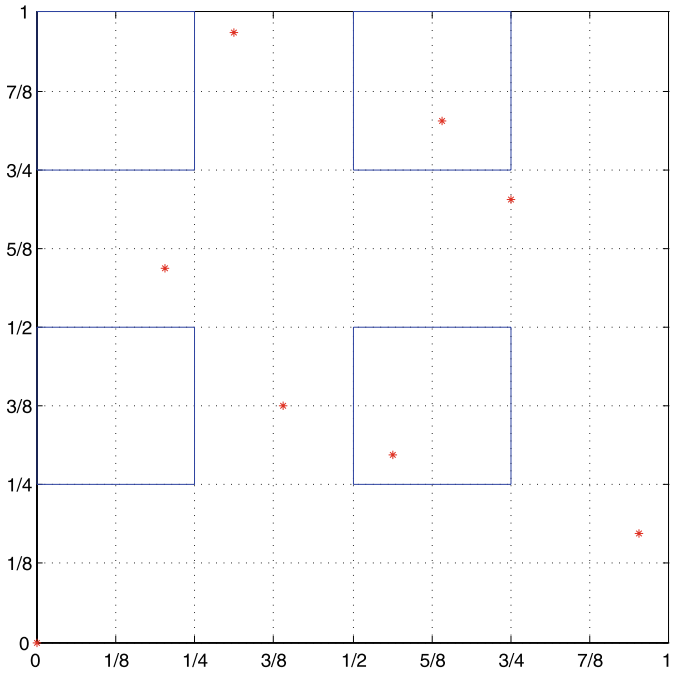


Fig. 1 The picture shows a $(2, \alpha, 1, 6, 3, 2)$-net in base 2 for any $\alpha \geq 2$ and a generalized elementary interval $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$, where $\nu_1 = \nu_2 = 1$, $i_{1,1} = i_{2,1} = 2$, and $a_{i_{1,1}} = 0$ and $a_{i_{2,1}} = 1$.

Remark 5. Note that $(t, \alpha, \beta, n, m, s)$-nets can only exist for parameters t, α, β, n, m, s where the definition implies that $\nu_1 + \cdots + \nu_s \leq m$.

Consider for example the choice of parameters $\beta = 1$, $t = \alpha = s = 2$, $m = 3$ and $n = 6$; such a $(2, 2, 1, 6, 3, 2)$-net can exist, since if $\nu_1 + \nu_2 > 3$ we have for all choices of $1 \leq i_{j,\nu_j} < \cdots < i_{j,1} \leq 6$, for $j = 1, 2$, that $\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha)} i_{j,l} > 4 = \beta n - t$. (On the other hand, that does not imply that such a net really does exist, it only allows for the possibility to exist.)

But a $(0, 2, 1, 6, 3, 2)$-net, i.e. we set $t = 0$ and leave the remaining parameters unchanged, cannot exist, since we could choose $\nu_1 = \nu_2 = 2$, $i_{1,1} = i_{2,1} = 2$ and $i_{1,2} = i_{2,2} = 1$, in which case we have $i_{1,1} + i_{1,2} + i_{2,1} + i_{2,2} = 6 = \beta n - t$, and thereby obtain a generalized elementary interval which has to contain exactly $q^{m-\nu_1-\nu_2} = q^{-1}$ points, which is of course absurd. Hence $t = 0$ is not possible for this choice of parameters. (Regarding $t = 1$, we have explicitly constructed digital $(1, 2, 1, 6 \times 3, 2)$-nets in Example 3 and Remark 2, which by Theorem 6 below also form a $(1, 2, 1, 6, 3, 2)$-net.)

Remark 6. We obtain the definition of a classical (t,m,s)-net from Definition 4 by setting $\alpha=\beta=1, n=m$, and considering all $\nu_1,\dots,\nu_s\geq 0$ so that $\sum_{j=1}^{s}\nu_j\leq m-t$, where we set $i_{j,k}=\nu_j-k+1$ for $k=1,\dots,\nu_j$. Hence a $(t,1,1,m,m,s)$-net is a (t,m,s)-net.

We shall now discuss the additional parameters α,β, and n, which do not appear in the definition of classical (t,m,s)-nets. The case $\alpha=1$ is strongly related to classical (t,m,s)-nets. We can, w.l.o.g., choose ν_j, $j=1,\dots,s$ so that $\sum_{j=1}^{s}\nu_j=\lfloor\beta n\rfloor-t$ and set $i_{j,l}=\nu_j+1-l$ for $l=1,\dots,\nu_j$, as in this case we obtain the most stringent condition on the points, i.e., all other conditions are automatically included in this choice of the $i_{j,l}$. Then a $(t,1,\beta,n,m,s)$-net is a classical (t',m,s)-net with $t'=m-\lfloor\beta n\rfloor+t$.

We have the following theorem.

Theorem 2. *Assume that $n,m,\alpha\in\mathbb{N}$, $0<\beta\leq 1$ a real number, and $0\leq t\leq\beta n$ an integer, such that there exists a (t,α,β,n,m,s)-net in base q. For $1\leq j_0\leq s$ let $0\leq\ell_{j_0}<j_0$ be given by $\ell_{j_0}\equiv m \pmod{j_0}$. Then for $j_0=1,\dots,s$ we have*

$$\beta n-t<\alpha m-j_0\frac{\alpha(\alpha-1)}{2}+\alpha,\qquad \text{for } m\geq\alpha j_0,$$

and

$$\beta n-t<\frac{1}{2}\alpha j_0\left\lfloor\frac{m}{j_0}\right\rfloor+(\ell_{j_0}+1)\left(\left\lfloor\frac{m}{j_0}\right\rfloor+1\right),\qquad \text{for } m<\alpha j_0.$$

Proof. As elaborated in Remark 5, for every choice of $1\leq i_{j,\nu_j}<\cdots<i_{j,1}$, $\nu_j\geq 0$, for $j=1,\dots,s$, with $\sum_{j=1}^{s}\sum_{l=1}^{\min(\nu_j,\alpha)}i_{j,l}\leq\beta n-t$, we must have that $|\boldsymbol{\nu}|_1\leq m$.

Let $1\leq j_0\leq s$ and let

$$\nu_j=\begin{cases}\lfloor m/j_0\rfloor+1 & \text{for } 1\leq j\leq\ell_{j_0}+1,\\ \lfloor m/j_0\rfloor & \text{for } \ell_{j_0}+2\leq j\leq j_0,\\ 0 & \text{for } j_0+1\leq j\leq s.\end{cases}$$

Further set $i_{j,l}=\nu_j+1-l$ for $l=1,\dots,\nu_j$ for $j=1,\dots,j_0$. Note that for this choice of $\nu_1,\dots,\nu_s$ we have

$$|\boldsymbol{\nu}|_1=j_0\left\lfloor\frac{m}{j_0}\right\rfloor+\ell_{j_0}+1=j_0\frac{m-\ell_{j_0}}{j_0}+\ell_{j_0}+1=m+1.$$

Consider the case where $\alpha\leq\lfloor m/j_0\rfloor$. Then

$$\begin{aligned}\sum_{j=1}^{s}\sum_{l=1}^{\min(\nu_j,\alpha)}i_{j,l}&=j_0\left(\left\lfloor\frac{m}{j_0}\right\rfloor+\left\lfloor\frac{m}{j_0}\right\rfloor-1+\cdots+\left\lfloor\frac{m}{j_0}\right\rfloor-(\alpha-1)\right)+\alpha(\ell_{j_0}+1)\\&=\alpha j_0\left\lfloor\frac{m}{j_0}\right\rfloor-j_0\frac{\alpha(\alpha-1)}{2}+\alpha(\ell_{j_0}+1)\\&=\alpha j_0\frac{m-\ell_{j_0}}{j_0}-j_0\frac{\alpha(\alpha-1)}{2}+\alpha\ell_{j_0}+\alpha\end{aligned}$$

$$= \alpha m - j_0 \frac{\alpha(\alpha-1)}{2} + \alpha.$$

Thus we get a contradiction if the last term is smaller or equal to $\beta n - t$ and hence the first result follows.

Now we consider the case where $\alpha \geq \lfloor m/j_0 \rfloor + 1$. Then

$$\begin{aligned}\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j,\alpha)} i_{j,l} &= j_0 \left(\left\lfloor \frac{m}{j_0} \right\rfloor + \left\lfloor \frac{m}{j_0} \right\rfloor - 1 + \cdots + 1 \right) + (\ell_{j_0}+1) \left(\left\lfloor \frac{m}{j_0} \right\rfloor + 1 \right) \\ &= j_0 \frac{\lfloor m/j_0 \rfloor (\lfloor m/j_0 \rfloor + 1)}{2} + (\ell_{j_0}+1) \left(\left\lfloor \frac{m}{j_0} \right\rfloor + 1 \right) \\ &\leq \frac{1}{2} \alpha j_0 \left\lfloor \frac{m}{j_0} \right\rfloor + (\ell_{j_0}+1) \left(\left\lfloor \frac{m}{j_0} \right\rfloor + 1 \right).\end{aligned}$$

Again we get a contradiction if the last term is smaller or equal to $\beta n - t$ and hence also the second result follows.

□

Note, Theorem 2 implies for $\alpha = 1, 2$ that $\beta n - t < \alpha m + 1$ (choose $j_0 = 1$) and, based on the proof of Theorem 2, one can show that $\beta n - t < \alpha m$ for $\alpha \geq 3$ (choose $j_0 = 1$). Thus, as $\beta n - t < \alpha m + 1$, we can w.l.o.g. choose β and n such that $\beta n < \alpha m + 1$ (for $\beta n \geq \alpha m + 1$ we must have $t > 0$, hence we do not exclude any cases by choosing $\beta n < \alpha m + 1$), or if β is such that βn is an integer, we have $\beta \leq \alpha m / n$.

Choosing $j_0 = s$ in Theorem 2 and estimating $\ell_{j_0} + 1 \leq j_0$, we obtain the following corollary.

Corollary 1. *Assume that $n, m, \alpha \in \mathbb{N}$, $0 < \beta \leq 1$ a real number, and $0 \leq t \leq \beta n$ an integer, such that there exists a $(t, \alpha, \beta, n, m, s)$-net in base q. Then we have*

$$\beta n - t < \alpha m - s \frac{\alpha(\alpha-1)}{2} + \alpha, \qquad \text{for } m \geq \alpha s,$$

and

$$\beta n - t < \frac{1}{2} \alpha m + m + s, \qquad \text{for } m < \alpha s.$$

As in the classical case, we can also define sequences.

Definition 5. Let $\alpha, \sigma \geq 1, t \geq 0$ be integers, and $0 < \beta \leq 1$ be a real number. Let $S = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\}$ be a sequence of points in $[0,1)^s$. Then S is a $(t, \alpha, \beta, \sigma, s)$-sequence in base q if for all $k \geq 0$ and $m > t/(\beta\sigma)$ we have that $\boldsymbol{x}_{kq^m}, \boldsymbol{x}_{kq^m+1}, \ldots, \boldsymbol{x}_{(k+1)q^m-1}$ is a $(t, \alpha, \beta, \sigma m, m, s)$-net in base q.

Remark 7. We obtain the definition of a classical (t, s)-sequence from Definition 5 and Remark 6 by setting $\alpha = \beta = \sigma = 1$. Hence a $(t, 1, 1, 1, s)$-sequence is a (t, s)-sequence.

3.2 Some Properties of $(t, \alpha, \beta, n, m, s)$-Nets and $(t, \alpha, \beta, \sigma, s)$-Sequences

In this subsection we establish a few propagation rules for $(t, \alpha, \beta, n, m, s)$-nets and $(t, \alpha, \beta, \sigma, s)$-sequences in base q. Furthermore, we establish that every digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$ is a $(t, \alpha, \beta, n, m, s)$-net in base q and that every digital $(t, \alpha, \beta, \sigma, s)$-sequence over $\mathbb{F}_q$ is also a $(t, \alpha, \beta, \sigma, s)$-sequence in base q. Finally, we produce lower and upper bounds on the quality parameter t for $(t, \alpha, \beta, \sigma, s)$-sequences.

The following theorem is in analogy to [2, Theorem 4.10].

Theorem 3. *Let P be a $(t, \alpha, \beta, n, m, s)$-net in base q and let S be a $(t, \alpha, \beta, \sigma, s)$-sequence in base q. Then we have the following:*

(i) *P is a $(t', \alpha, \beta', n, m, s)$-net for all $0 < \beta' \le \beta$ and all $t \le t' \le \beta' n$, and S is a $(t', \alpha, \beta', \sigma, s)$-sequence for all $0 < \beta' \le \beta$ and all $t \le t'$.*

(ii) *P is a $(t', \alpha', \beta', n, m, s)$-net for all $\alpha' \ge 1$ where $\beta' = \beta \min(\alpha, \alpha')/\alpha$ and $t' = \lceil t \min(\alpha, \alpha')/\alpha \rceil$, and S is a $(t', \alpha', \beta', \sigma, s)$-sequence for all $\alpha' \ge 1$ where $\beta' = \beta \min(\alpha, \alpha')/\alpha$ and where $t' = \lceil t \min(\alpha, \alpha')/\alpha \rceil$.*

(iii) *Any $(t, \alpha, \beta, \sigma, s)$-sequence is a $(t, \alpha, \beta, \sigma', s)$-sequence for all $1 \le \sigma' \le \sigma$.*

(iv) *Any $(t, \alpha, \beta, n, m, s)$-net is a classical $(m - \lfloor \beta n/\alpha \rfloor + \lceil t/\alpha \rceil, m, s)$-net and any $(t, \alpha, \beta, \sigma, s)$-sequence with $\alpha = \beta\sigma$ is a classical $(\lceil t/\alpha \rceil, s)$-sequence.*

Proof. For the first part note that $\beta' n - t' \le \beta n - t$ and hence the condition on P in Definition 4 is either the same or weaker. The same holds for S, hence the first part follows.

To prove the second part we consider firstly the case $\alpha' \ge \alpha$. Let $1 \le i_{j,\nu_j} < \cdots < i_{j,1}$, $\nu_j \ge 0$, for $j = 1, \ldots, s$ with

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha')} i_{j,l} \le \beta n - t.$$

As

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha)} i_{j,l} \le \sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha')} i_{j,l}$$

and P is a $(t, \alpha, \beta, n, m, s)$-net, it follows that $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ contains $q^{m - |\boldsymbol{\nu}|_1}$ points for all admissible $\mathbf{a}_{\boldsymbol{\nu}}$ and hence this case follows for nets.

Let now $\alpha' < \alpha$ and assume

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha')} i_{j,l} \le \beta' n - t' = \frac{\alpha'}{\alpha} \beta n - \left\lceil t \frac{\alpha'}{\alpha} \right\rceil.$$

As

$$\frac{1}{\alpha} \sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha)} i_{j,l} \le \frac{1}{\alpha'} \sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha')} i_{j,l},$$

it follows that

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j,\alpha)} i_{j,l} \le \frac{\alpha}{\alpha'}(\beta' n - t') \le \beta n - t.$$

As P is a (t,α,β,n,m,s)-net, it follows that $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ contains exactly $q^{m-|\boldsymbol{\nu}|_1}$ points for all admissible $\mathbf{a}_{\boldsymbol{\nu}}$, completing the proof for nets. For sequences the result follows from the result for nets and Definition 5.

For the third part we have to show that every point set $\boldsymbol{x}_{kq^m}, \dots, \boldsymbol{x}_{(k+1)q^m-1}$ is a $(t,\alpha,\beta,\sigma' m, m, s)$-net. We know that this point set is a $(t,\alpha,\beta,\sigma m, m, s)$-net from Definition 5. As $\sigma' m - t \le \sigma m - t$ this follows as the condition on the points $\boldsymbol{x}_{kq^m}, \dots, \boldsymbol{x}_{(k+1)q^m-1}$ can only become weaker, which implies the result.

For the last part we use (ii), which shows that every (t,α,β,n,m,s)-net P is also a $(\lceil t/\alpha \rceil, 1, \beta/\alpha, n, m, s)$-net. After Remark 6, it was shown that Definition 4 implies that a $(t,1,\beta,n,m,s)$-net is a (t',m,s)-net, with $t' = m - \lfloor \beta n \rfloor + t$, hence P is also a classical (t',m,s)-net, where

$$t' = m - \left\lfloor \frac{\beta}{\alpha} n \right\rfloor + \left\lceil \frac{t}{\alpha} \right\rceil.$$

Now consider a $(t,\alpha,\beta,\sigma,s)$-sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \dots$. For any $k \ge 0$, the set of points $\boldsymbol{x}_{kq^m}, \dots, \boldsymbol{x}_{(k+1)q^m-1}$ forms a $(t,\alpha,\beta,\sigma m, m, s)$-net. Hence the above result implies that this is a classical (t',m,s)-net where

$$t' = m - \left\lfloor \frac{\beta}{\alpha} \sigma m \right\rfloor + \left\lceil \frac{t}{\alpha} \right\rceil = \left\lceil \frac{t}{\alpha} \right\rceil.$$

As $\boldsymbol{x}_{kq^m}, \dots, \boldsymbol{x}_{(k+1)q^m-1}$ is a classical (t',m,s)-net for all $k \ge 0$, the result follows.

□

Remark 8. By Theorem 3, a $(2,\alpha,1,6,3,2)$-net, $\alpha \ge 2$, is a classical $(4 - \lfloor \frac{6}{\alpha} \rfloor, 3, 2)$-net. By the forthcoming Theorem 6, the digital $(2,\alpha,1,6\times 3,2)$-net from Remark 3 is a $(2,\alpha,1,6,3,2)$-net, hence we have an example of a $(2,\alpha,1,6,3,2)$-net which is a strict $(1,3,2)$-net. See also Figure 1 for an example of a $(2,\alpha,1,6,3,2)$-net, which is a $(0,3,2)$-net.

In part (iv) of the above theorem we had the restriction that $\alpha = \beta\sigma$. If S is a $(t,\alpha,\beta,\sigma,s)$-sequence with $\alpha > \beta\sigma$, then we cannot use (iv) of the above theorem to imply that S is a classical (t',s)-sequence, as then we would obtain a t'-value of the subnets $\boldsymbol{x}_{kq^m}, \dots, \boldsymbol{x}_{(k+1)q^m-1}$ which grows with m. Hence we do not obtain a classical sequence this way. On the other hand, we always have $\alpha \ge \beta\sigma$, as we show in the following theorem.

Theorem 4. *Assume that $t, \alpha, \sigma, s \in \mathbb{N}$, and $\beta \in \mathbb{R}$, $0 < \beta \le 1$ are such that there exists a $(t,\alpha,\beta,\sigma,s)$-sequence. Then $\beta\sigma \le \alpha$.*

Proof. Let $\boldsymbol{x}_0, \boldsymbol{x}_1, \dots$ be a $(t,\alpha,\beta,\sigma,s)$-sequence. Then the set of points $\boldsymbol{x}_0, \dots, \boldsymbol{x}_{q^m-1}$ forms a $(t,\alpha,\beta,\sigma m, m, s)$-net for all $m > t/(\beta\sigma)$.

Assume to the contrary that $\alpha < \beta\sigma$. As $\beta\sigma m - t < \alpha m + 1$, which was shown after the proof of Theorem 2, we can choose an m large enough to obtain a contradiction. Hence $\beta\sigma \leq \alpha$.

□

Digital sequences for which $\alpha = \beta\sigma$ are of interest, as in this case we get the optimal rate of convergence of the integration error for functions with square integrable partial mixed derivatives of order α in each variable, whereas for $\alpha > \beta\sigma$ we do not get the optimal rate, see [2]. But for the case $\alpha = \beta\sigma$ we get the following bound on the value of t from Theorem 2.

Theorem 5. *Assume that $t, \alpha, \sigma, s \in \mathbb{N}$, and $\beta \in \mathbb{R}$, $0 < \beta \leq 1$, are such that $\alpha = \beta\sigma$ and such that there exists a $(t, \alpha, \beta, \sigma, s)$-sequence. Then for all $\alpha \geq 2$ we have*

$$t > s\frac{\alpha(\alpha-1)}{2} - \alpha.$$

Proof. Let $m_0 = \alpha s$. Then the first q^{m_0} points of a $(t, \alpha, \beta, \sigma, s)$-sequence form a $(t, \alpha, \beta, \sigma m_0, m_0, s)$-net. By Corollary 1 we obtain that

$$\beta\sigma m_0 - t < \alpha m_0 - s\frac{\alpha(\alpha-1)}{2} + \alpha.$$

By substituting α for $\beta\sigma$ in the last equation we obtain the result.

□

The next theorem establishes that a digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$ is a $(t, \alpha, \beta, n, m, s)$-net in base q and analogously for sequences. This also yields explicit constructions of $(t, \alpha, \beta, n, m, s)$-nets and $(t, \alpha, \beta, \sigma, s)$-sequences as digital constructions are known from [2].

Theorem 6. *Every digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$ is a $(t, \alpha, \beta, n, m, s)$-net in base q and every digital $(t, \alpha, \beta, \sigma, s)$-sequence over $\mathbb{F}_q$ is a $(t, \alpha, \beta, \sigma, s)$-sequence in base q.*

Proof. Assume we are given an arbitrary generalized elementary interval

$$J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}}) = \prod_{j=1}^{s} \bigcup_{\substack{a_{j,l}=0 \\ l \in \{1,\ldots,n\} \setminus \{i_{j,1},\ldots,i_{j,\nu_j}\}}}^{q-1} \left[\frac{a_{j,1}}{q} + \cdots + \frac{a_{j,n}}{q^n}, \frac{a_{j,1}}{q} + \cdots + \frac{a_{j,n}}{q^n} + \frac{1}{q^n}\right),$$

for some given values of $\boldsymbol{\nu}$, $\mathbf{i}_{\boldsymbol{\nu}}$, and $\mathbf{a}_{\boldsymbol{\nu}}$ such that $1 \leq i_{j,\nu_j} < \cdots < i_{j,1}$, $j = 1, \ldots, s$, $\nu_j \geq 0$, and

$$\sum_{j=1}^{s} \sum_{l=1}^{\min(\nu_j, \alpha)} i_{j,l} \leq \beta n - t. \tag{2}$$

We have to show that $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ contains exactly $q^{m-|\boldsymbol{\nu}|_1}$ points of the digital $(t, \alpha, \beta, n \times m, s)$-net, which we denote by $\boldsymbol{x}_0, \dots, \boldsymbol{x}_{q^m-1}$. Let $\boldsymbol{x}_h = (x_{h,1}, \dots, x_{h,s})$ and $x_{h,j} = x_{h,j,1}q^{-1} + x_{h,j,2}q^{-2} + \dots$ be the q-adic representation of $x_{h,j}$.

Then for each $0 \le h < q^m$ it follows that $\boldsymbol{x}_h \in J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ if and only if $x_{h,j,k} = a_{j,k}$ for all $k \in \{i_{j,1}, \dots, i_{j,\nu_j}\}$ and all $j = 1, \dots, s$. The value of $x_{h,j,k}$ is obtained from the digital construction scheme in the following way: Let $C_1, \dots, C_s$ denote the generator matrices of the digital $(t, \alpha, \beta, n \times m, s)$-net over $\mathbb{F}_q$. Then $x_{h,j,k} = \varphi^{-1}(\mathbf{c}_{j,k}\mathbf{h})$, where $\mathbf{c}_{j,k}$ denotes the kth row of C_j. Thus $\mathbf{c}_{j,k}\mathbf{h} = \varphi(x_{h,j,k})$.

Let $C = (\mathbf{c}_{1,i_{1,1}}^\top, \dots, \mathbf{c}_{1,i_{1,\nu_1}}^\top, \dots, \mathbf{c}_{s,i_{s,1}}^\top, \dots, \mathbf{c}_{s,i_{s,\nu_s}}^\top)^\top$ and further we define the vector $\mathbf{b} = (\varphi(a_{1,i_{1,1}}), \dots, \varphi(a_{1,i_{1,\nu_1}}), \dots, \varphi(a_{s,i_{s,1}}), \dots, \varphi(a_{s,i_{s,\nu_s}}))^\top$. Then, by the above, it follows that $\boldsymbol{x}_h \in J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ if and only if $C\mathbf{h} = \mathbf{b}$.

We now investigate how many solutions $\mathbf{h}$ the system of equations $C\mathbf{h} = \mathbf{b}$ has. As (2) is satisfied, Definition 2 implies that the rows of the matrix C are linearly independent. As C has $|\boldsymbol{\nu}|_1$ ($|\boldsymbol{\nu}|_1 \le m$) rows, there are exactly $q^{m-|\boldsymbol{\nu}|_1}$ solutions to this system, and hence $q^{m-|\boldsymbol{\nu}|_1}$ of the $\boldsymbol{x}_0, \dots, \boldsymbol{x}_{q^m-1}$ fall into $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$, which shows that every digital $(t, \alpha, \beta, n \times m, s)$-net is also a $(t, \alpha, \beta, n, m, s)$-net.

Now we turn to sequences. Let $\boldsymbol{x}_0, \boldsymbol{x}_1, \dots$ be a digital $(t, \alpha, \beta, \sigma, s)$-sequence over the finite field $\mathbb{F}_q$. Let $k \ge 0$ and $m > t/(\beta\sigma)$. Then the point set $\boldsymbol{x}_{\ell q^m}, \dots, \boldsymbol{x}_{(\ell+1)q^m-1}$ can be obtained from the digital construction scheme with an added digital shift, i.e., there are matrices $C_1, \dots, C_s \in \mathbb{F}_q^{n \times m}$ and vectors $\mathbf{d}_{j,\ell} = (d_{j,1,\ell}, \dots, d_{j,n,\ell})^\top \in \mathbb{F}_q^n$, $1 \le j \le s$, which depend on ℓ, such that $x_{h,j,k} = \varphi^{-1}(\mathbf{c}_{j,k}\mathbf{h} + d_{j,k,\ell})$. Thus we have $\mathbf{c}_{j,k}\mathbf{h} = \varphi(x_{h,j,k}) - d_{j,k,\ell} \in \mathbb{F}_q$. For some given generalized elementary interval $J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ we have $\boldsymbol{x}_h \in J(\mathbf{i}_{\boldsymbol{\nu}}, \mathbf{a}_{\boldsymbol{\nu}})$ if and only if $\mathbf{c}_{j,k}\mathbf{h} = \varphi(a_{j,k}) - d_{j,k,\ell}$ for all $k \in \{i_{1,1}, \dots, i_{1,\nu_1}, \dots, i_{s,1}, \dots, i_{s,\nu_s}\}$ and $j = 1, \dots, s$. Thus the same argument as for nets applies and the result follows.

□

Definition 6. Let q be a prime power. Then let $d_q(\alpha, s)$ denote the smallest value of t such that there exists a digital $(t, \alpha, \beta, \sigma, s)$-sequence over the finite field $\mathbb{F}_q$ with $\alpha = \beta\sigma$.

The analogy of Definition 6 for classical digital sequences, i.e. the case $\alpha = 1$, has already appeared in [13], see also [14, Definition 8]. For $\alpha = \beta = \sigma = 1$, i.e. digital (t, s)-sequences, it is true that

$$\frac{s}{q-1} - \mathcal{O}(\log s) < d_q(1, s) \le \frac{c}{\log q} s + 1,$$

for all $s \ge 1$, where $c > 0$ is an absolute constant. The lower bound was shown in [17] and also holds for (t, s)-sequences, whereas the upper bound can be found in [13, Theorem 4] and [14, Corollary 1]. Improved results for several special values of q can also be found in [15].

The following theorem now considers the case $\alpha \ge 2$.

Theorem 7. *Let q be a prime power. Then for all $s \ge 1$ and $\alpha \ge 2$ we have*

$$s\frac{\alpha(\alpha-1)}{2}-\alpha < d_q(\alpha,s) \le s\alpha^2\frac{c}{\log q}+\alpha+\alpha\left\lfloor\frac{s(\alpha-1)}{2}\right\rfloor,$$

where $c > 0$ is an absolute constant.

Proof. The lower bound is taken from Theorem 5. To prove the upper bound we use Theorem 1 with $d = \alpha$ to obtain a digital $(t, \alpha, 1, \alpha, s)$-sequence over $\mathbb{F}_q$ with

$$t = \alpha t' + \alpha\left\lfloor\frac{s(\alpha-1)}{2}\right\rfloor,$$

where t' is the quality parameter of the classical digital $(t', s\alpha)$-sequence upon which the construction is based. From [13, Theorem 4], [14, Corollary 1] we know that there exist digital (t', s)-sequences for which $t' \le \frac{c}{\log q}s + 1$. Upon combining the last two formulae, where we replace s with αs in the last formula as we consider $(t', s\alpha)$-sequences, the result follows.

□

Note that the bounds in Theorem 7 also apply to (non-digital) $(t, \alpha, \beta, \sigma, s)$-sequences with $\alpha = \beta\sigma$ and t value as small as possible.

Acknowledgements The support of the ARC under its Centre of Excellence program is gratefully acknowledged. The authors thank Peter Kritzer for providing them with the strict digital $(1, 3, 4)$-net and the strict digital $(2, 3, 4)$-net used in Example 3 and Remark 2 respectively. Further we gratefully acknowledge the helpful comments of Harald Niederreiter, Art Owen, and the anonymous referees.

References

1. J. Dick, Explicit Constructions of Quasi-Monte Carlo Rules for the Numerical Integration of High-Dimensional Periodic Functions. SIAM J. Numer. Anal., 45, 2141–2176, 2007.
2. J. Dick, Walsh spaces containing smooth functions and Quasi-Monte Carlo Rules of Arbitrary high order. SIAM J. Numer. Anal., 46, 1519–1553, 2008.
3. J. Dick, On quasi-Monte Carlo rules achieving higher order convergence. To appear in: P. L'Ecuyer and A.B. Owen (eds.), Monte Carlo and quasi-Monte Carlo methods 2008, Springer Verlag, to appear 2010.
4. J. Dick and P. Kritzer, Duality theory and propagation rules for generalized digital nets. To appear in Math. Comp., 2010.
5. J. Dick and F. Pillichshammer, Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces. J. Complexity, 21, 149–195, 2005.
6. H. Faure, Discrépance de suites associées à un système de numération (en dimension s). Acta Arith., 41, 337–351, 1982.
7. H. Niederreiter, Constructions of (t, m, s)-nets and (t, s)-sequences. Finite Fields Appl., 11, 578–600, 2005.
8. H. Niederreiter, Nets, (t, s)-sequences and codes. In: A. Keller, S. Heinrich, and H. Niederreiter (eds.), Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 83–100, Springer, Berlin, 2008.

9. H. Niederreiter, Point sets and sequences with small discrepancy. Monatsh. Math., 104, 273–337, 1987.
10. H. Niederreiter, Random number generation and quasi-Monte Carlo methods, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 63, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
11. H. Niederreiter and G. Pirsic, Duality for digital nets and its applications. Acta Arith., 97, 173–182, 2001.
12. H. Niederreiter and C.P. Xing, Global function fields with many rational places and their applications. In: G.L. Mullen and P.J.-S. Shiue (eds.), Finite fields: theory, applications, and algorithms (Waterloo, ON, 1997), Contemp. Math., Vol. 225, pp. 87–111, Amer. Math. Soc., Providence, RI, 1999.
13. H. Niederreiter and C. Xing, Low-discrepancy sequences and Global Function Fields with Many Rational Places. Finite Fields Appl., 2, 241–273, 1996.
14. H. Niederreiter and C.P. Xing, Quasirandom points and global function fields. In: Finite Fields and Applications, S. Cohen and H. Niederreiter, volume 233 of London Math. Soc. Lecture Note Ser., pages 269–296. Cambridge University Press, Cambridge, 1996.
15. H. Niederreiter and C.P. Xing, Rational points on curves over finite fields: theory and applications, London Mathematical Society Lecture Note Series, Vol. 285, Cambridge University Press, Cambridge, 2001.
16. G. Pirsic, A software implementation of Niederreiter-Xing sequences. In: K.T. Fang, F.J. Hickernell, and H. Niederreiter (eds.), Monte Carlo and quasi-Monte Carlo methods 2000, pp. 434–445, Springer Verlag, Berlin, 2002.
17. R. Schürer, A New Lower Bound on the t-Parameter of (t,s)-Sequences. In: A. Keller, S. Heinrich, and H. Niederreiter (eds.), Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 623–632, Springer Verlag, Berlin, 2008.
18. I.M. Sobol', Distribution of points in a cube and approximate evaluation of integrals, Z. Vyčisl. Mat. i Mat. Fiz., 7, 784–802, 1967.

Implementation of a Component-By-Component Algorithm to Generate Small Low-Discrepancy Samples

Benjamin Doerr, Michael Gnewuch, and Magnus Wahlström

Abstract In [B. Doerr, M. Gnewuch, P. Kritzer, F. Pillichshammer. Monte Carlo Methods Appl., 14:129–149, 2008], a component-by-component (CBC) approach to generate small low-discrepancy samples was proposed and analyzed. The method is based on randomized rounding satisfying hard constraints and its derandomization. In this paper we discuss how to implement the algorithm and present first numerical experiments. We observe that the generated points in many cases have a significantly better star discrepancy than what is guaranteed by the theoretical upper bound. Moreover, we exhibit that the actual discrepancy is mainly caused by the underlying grid structure, whereas the rounding errors have a negligible contribution. Hence to improve the algorithm, we propose and analyze a randomized point placement. We also study a hybrid approach which combines classical low-discrepancy sequences and the CBC algorithm.

1 Introduction

The *star discrepancy* of an N-point set $\mathcal{P}_s = \{p_0, \ldots, p_{N-1}\}$ in the s-dimensional unit cube $[0,1]^s$ is defined by

$$D_N^*(\mathcal{P}_s) := \sup_{x \in [0,1]^s} |\Delta_s(x, \mathcal{P}_s)|.$$

Here the discrepancy function Δ_s of the set $\mathcal{P}_s$ is given, for $x = (x_1, \ldots, x_s)$, by

Benjamin Doerr and Magnus Wahlström
Max-Planck-Institut für Informatik, Saarbrücken, Germany
url: http://www.mpi-inf.mpg.de/~{doerr|wahl}

Michael Gnewuch
Department of Computer Science, Columbia University, New York, USA, and Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Kiel, Germany
url: http://www.numerik.uni-kiel.de/~mig/

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_20,

$$\Delta_s(x, \mathcal{P}_s) = \lambda_s([0,x)) - \frac{1}{N}\sum_{j=0}^{N-1} 1_{[0,x)}(p_j),$$

where λ_s is the s-dimensional Lebesgue measure and $1_{[0,x)}$ is the characteristic function of the s-dimensional half-open box $[0,x) = [0,x_1) \times \cdots \times [0,x_s)$.

It is well known that the star discrepancy is intimately related to multivariate integration. If we have a function f defined on $[0,1]^s$ and a point set $\mathcal{P}_s = \{p_0, \ldots, p_{N-1}\}$, then the Koksma-Hlawka inequality states

$$\left| \int_{[0,1]^s} f(x)\,\mathrm{d}x - \frac{1}{N}\sum_{k=0}^{N-1} f(p_k) \right| \le D_N^*(\mathcal{P}_s) V(f),$$

where $V(f)$ denotes the variation of f in the sense of Hardy and Krause, see, e.g., [11, 12]. Thus for multivariate integration it is important to find small point sets with low discrepancy.

For fixed dimension s various N-point sets $\mathcal{P}_s$ have been constructed satisfying

$$D_N^*(\mathcal{P}_s) \le C_{\mathcal{P}_s} \log(N)^{s-1} N^{-1}, \quad N \ge 2, \tag{1}$$

with $C_{\mathcal{P}_s}$ a suitable constant depending on $\mathcal{P}_s$, see, e.g., [12]. These constructions typically suffer from two difficulties. One is that often the constant $C_{\mathcal{P}_s}$ is not known sufficiently well or depends unfavorably on the dimension s. The other is that these bounds for large s and moderate N unfortunately give no useful information, since $\log(N)^{s-1}N^{-1}$ is an increasing function in N for $N \le e^{s-1}$.

A bound more helpful for high-dimensional integration was established by Heinrich et al. [10]. They proved in a non-constructive way that there is a constant $c > 0$ such that for all $s, N \in \mathbb{N}$ an N-point set $\mathcal{P}_s \subseteq [0,1)^s$ exists satisfying

$$D_N^*(\mathcal{P}_s) \le c s^{1/2} N^{-1/2}. \tag{2}$$

Which point sets do satisfy bounds like (2)? Discrepancy calculations performed by Thiémard [17, 18] indicate that known constructions which exhibit the asymptotic behavior (1) may not satisfy (2). As pointed out in [9], actually the seemingly easier question if any of the known constructions of low-discrepancy point sets $\mathcal{P}_{N,s}$ satisfies estimates of the form $D_N^*(\mathcal{P}_{N,s}) \le c s^\kappa N^{-\alpha}$ for all $s, N \in \mathbb{N}$, where c, κ, α are positive constants not depending on N and s, still remains open.

In [1, 3] the first non-trivial deterministic algorithmic point constructions were proposed whose star discrepancy grows for a fixed number of points with respect to the dimension s at most like $\sqrt{s}$. A drawback of these algorithms is their large runtime. A considerable speed-up was achieved by Kritzer, Pillichshammer and the first two authors [2]. They perform the derandomization in a component-by-component (CBC) fashion. An additional advantage of this new algorithm is that it allows a simple exact computation of the star discrepancy of the output point set.

The disadvantage is that the theoretical upper bound on the star discrepancy of the generated points grows like $s^{3/2}$ (if one generates the set component by com-

ponent starting in dimension 1) instead of like $\sqrt{s}$. The paper gives no indication of whether this is likely to happen in practice or not. A second question left open in [2] is how useful it is in practice to extend given low-discrepancy point sets in the dimension via the CBC algorithm. This approach is related to padding quasi-Monte Carlo (QMC) by Monte Carlo (MC), see also Sect. 3.5.

For these reasons, we implemented the CBC algorithm in the programming language C99. In this paper, we describe some details of the implementation and report the results of several numerical experiments. Among other outcomes, they show that the discrepancy of the point sets constructed by our algorithm for reasonable settings grows only linearly in the dimension s. Also, we do observe a significant advantage of extending existing good point sets in small dimensions via our approach to higher dimensions.

Another motivation for this paper from a different subarea of discrepancy theory is the following. There is a vast literature on the theory of randomized rounding and its derandomization, starting with the monograph of Spencer [16] and the celebrated paper of Raghavan [14]. But although derandomization is an algorithmic tool, seemingly none of these publications cares about practical aspects as, e.g., explicit descriptions of the resulting algorithms, the run-times of algorithms, or the rounding errors observed in practice. It seems that this work, together with [4] are the first to fight this short-coming.

2 Description of the Algorithm

2.1 General Method

The CBC approach presented in [2] aims at using an existing $(s-1)$-dimensional low-discrepancy point set to construct an s-dimensional one. Let a point set $\mathcal{P}_{s-1} = \{y_0, \ldots, y_{N-1}\} \subset [0,1)^{s-1}$ be given. For an integer $m_s \geq 2$ the algorithm determines numbers $X_0, \ldots, X_{N-1}$, each chosen from the set

$$G_s = \left\{ \frac{1}{2m_s}, \frac{3}{2m_s}, \ldots, \frac{2m_s-1}{2m_s} \right\} \tag{3}$$

and returns a point set $\mathcal{P}_s = \mathcal{P}_s(X_0, \ldots, X_{N-1}) := \{(y_0, X_0), \ldots, (y_{N-1}, X_{N-1})\} \subset [0,1)^s$. If we choose

$$m_s = m_s(N) := \left\lceil \frac{\sqrt{N}}{\sqrt{2}} \left(s \log\left(\rho'(N,s)\right) + \log(4) \right)^{-1/2} \right\rceil, \tag{4}$$

where

$$\rho'(N,s) = 2\sqrt{\mathrm{e}} \left(\max\{1, N/((1+2\log(2))s)\} \right)^{1/2},$$

then the output set $\mathcal{P}_s$ satisfies the discrepancy bound

$$D_N^*(\mathcal{P}_s) \le \left(\sqrt{3}+\frac{1}{\sqrt{2}}\right)\frac{\sqrt{s}}{\sqrt{N}}\left(\log\left(\rho'(N,s)\right)+\frac{1}{s}\log(4)\right)^{1/2}+D_N^*(\mathcal{P}_{s-1}). \quad (5)$$

Here we confine ourselves to sets $\mathcal{P}_s$ that are subsets of the anisotropic grid

$$\mathcal{G}_s := G_1 \times \cdots \times G_s,$$

where G_d is defined as in (3) and (4), but with s replaced by d for $d = 1, \ldots, s$. Let

$$G_d^* := \left\{\frac{1}{m_d}, \frac{2}{m_d}, \ldots, \frac{m_d-1}{m_d}, 1\right\}$$

for $d = 1, \ldots, s$, and let

$$\mathcal{G}_s^* := G_1^* \times \cdots \times G_s^*.$$

Let $t_1, \ldots, t_n, n := \Pi_{d=1}^s m_d$, be an enumeration of $\mathcal{G}_s^*$ and

$$\mathcal{T}_s^* := \{[0, t_i) \,|\, i = 1, \ldots, n\}.$$

When deciding the values for $X_0, \ldots, X_{N-1} \in G_s$, our algorithm tries to obtain a low discrepancy in all boxes of $\mathcal{T}_s^*$. As described in [2], this together with the fact that $\mathcal{P}_s \subset \mathcal{G}_s$ allows to calculate the exact star discrepancy of the output set within the course of the algorithm without essentially increasing the effort.

Formulation as Rounding Problem

The CBC algorithm derandomizes randomized rounding satisfying hard constraints. To describe it more precisely, let us formulate our problem as a rounding problem.

As discussed before, if $\mathcal{P}_{s-1} = \{y_0, \ldots, y_{N-1}\} \subset \mathcal{G}_{s-1}$ is given, we aim at finding $X_0, \ldots, X_{N-1} \in G_s$ such that $\mathcal{P}_s = \{(y_j, X_j) : 0 \le j < N\}$ has small discrepancy in the sense of (5).

Let $\mathcal{X}$ be the set of all $x \in [0,1]^{\{0,\ldots,N-1\}\times\{1,\ldots m_s\}}$ such that $x\mathbf{1}_{m_s} = \mathbf{1}_N$. Let $\overline{x} \in \mathcal{X}$ be defined by $\overline{x}_{jk} = \frac{1}{m_s}$ for all j, k. A *rounding* of $\overline{x}$ is an $x \in \mathcal{X} \cap \{0,1\}^{\{0,\ldots,N-1\}\times\{1,\ldots m_s\}}$. That is, each x_{jk} is a rounding of $\overline{x}_{jk}$ and the *hard constraints* $\sum_{k=1}^{m_s} x_{jk} = 1$ for all j are satisfied.

Let us define the linear function $A : \mathbb{R}^{\{0,\ldots,N-1\}\times\{1,\ldots m_s\}} \to \mathbb{R}^n$ by

$$A(x)_i = \sum_{j=0}^{N-1}\sum_{k=1}^{m_s} x_{jk} 1_{[0,t_i)}(y_j, \hat{k}) \quad \text{for } i = 1, \ldots, n;$$

here we used the shorthand $\hat{k} = \frac{2k-1}{2m_s}$ for the kth point of G_s. If $x \in \mathcal{X}$ is binary, put $X_j = \hat{k}_j$ for $j = 0, 1, \ldots, N-1$, where k_j is the index for which $x_{jk_j} = 1$. Then

$$\mathcal{P}(x) := \mathcal{P}(X_0, \ldots, X_{N-1}) = \{(y_j, X_j) : 0 \le j < N\}$$

is an N–point set in $\mathcal{G}_s \subset [0,1]^s$ and $A(x)_i$ equals $|\mathcal{P}(x) \cap [0,t_i)|$. In [2, Sect. 4] an elementary calculation showed that

$$D_N^*(\mathcal{P}(x)) \leq \tfrac{1}{N}\|A(x) - A(\overline{x})\|_\infty + D_N^*(\mathcal{P}_{s-1}) + \tfrac{1}{2m_s}. \tag{6}$$

This implies that point sets $\mathcal{P}(x)$ with small discrepancy correspond to roundings $x \in \mathcal{X}$ of $\overline{x}$ with small rounding error $\|A(x) - A(\overline{x})\|_\infty$.

Solving the Rounding Problem

How does our algorithm generate roundings x that satisfy the hard constraints $\sum_{k=1}^{m_s} x_{jk} = 1$ for all j and exhibit a small rounding error? The randomized construction is to choose for each j independently a $k_j \in \{1,\ldots,m_s\}$ at random such that $\mathbb{P}[k_j = k] = \overline{x}_{jk}$ for all j,k. Then for all j we define binary random variables X_{jk} by $X_{jk} = 1$ if and only if $k = k_j$, and $X_{jk} = 0$ otherwise. Note that any outcome of X lies in $\mathcal{X}$. Put

$$\sigma := \frac{\sqrt{N}}{\sqrt{2}}\left(s\log(\rho'(N,s)) + \log(4)\right)^{1/2}. \tag{7}$$

As shown in [2], we get $P := \sum_i \mathbb{P}[|(A(X-\overline{x}))_i| \geq \sigma] \leq 1/2$ ("small initial failure probability"). In particular, there is an $x \in \mathcal{X}$ such that $|(A(x-\overline{x}))_i| \leq \sigma$ for all i.

We can compute such roundings x by derandomizing the probabilistic construction above. For $k = 1,\ldots,m_s$, let us consider the conditional probability

$$P_k := \sum_i \mathbb{P}\big[|(A(X-\overline{x}))_i| \geq \sigma \,\big|\, k_0 = k\big].$$

Since $P = \sum_{k=1}^{m_s} \frac{1}{m_s} P_k$, there is a $1 \leq k_0^* \leq m_s$ such that $P_{k_0^*} \leq P \leq 1/2$ ("decreasing failure probability"). Next, let

$$P_{k_0^* k} := \sum_i \mathbb{P}\big[|(A(X-\overline{x})_i| \geq \sigma \,\big|\, k_0 = k_0^*, k_1 = k\big].$$

Again, $P_{k_0^*} = \sum_{k=1}^{m_s} \frac{1}{m_s} P_{k_0^* k}$, and there is a $1 \leq k_1^* \leq m_s$ such that $P_{k_0^* k_1^*} \leq P_{k_0^*} \leq 1/2$. Proceeding like this we end up with $k_0^*,\ldots,k_{N-1}^*$ such that

$$P_{k_0^*\ldots k_{N-1}^*} := \mathbb{P}\big[|(A(X-\overline{x}))_i| \geq \sigma \,\big|\, \forall\, 0 \leq j < N : k_j = k_j^*\big] \leq 1/2.$$

Since $P_{k_0^*\ldots k_{N-1}^*}$ involves no randomness, we actually have $P_{k_0^*\ldots k_{N-1}^*} = 0$. Then we define x as follows: For each $0 \leq j < N$, we set $x_{jk_j^*} := 1$ and $x_{jk} := 0$ for all other k. This yields a binary $x \in \mathcal{X}$ such that $|(A(x-\overline{x}))_i| \leq \sigma$ for all i.

In practice we usually cannot compute the conditional probabilities $P_{k_0^* k_1^* \ldots}$ in time polynomially bounded in N, m_s and n. However, we can compute (in polynomial time) upper bounds $U_{k_0^* k_1^* \ldots}$ for the exact conditional probabilities $P_{k_0^* k_1^* \ldots}$ such that the following key properties are maintained:

- Small initial (estimated) failure probability: $U < 1$.

- Decreasing (estimated) failure probability: For all $0 \le \ell < N$ and $k_0^*, \dots, k_{\ell-1}^* \in \{1, \dots, m_s\}$ there is a $1 \le k \le m_s$ such that $U_{k_0^* k_1^* \dots k_{\ell-1}^* k} \le U_{k_0^* k_1^* \dots k_{\ell-1}^*}$.

The quantities $U_{k_0^* k_1^* \dots}$ are called *pessimistic estimators* for the conditional probabilities $P_{k_0^* k_1^* \dots}$. This notion was introduced by Raghavan [14], who also showed that such pessimistic estimators exist for the conditional probabilities that occur in our derandomization.

To achieve that the initial estimated failure probability U is less than one, we have to choose σ in equation (7) to be $\sqrt{6}$ times larger than there. This choice implies that the output set $\mathcal{P}_s$ of the CBC algorithm satisfies the discrepancy bound (5).

With suitable implementation, the run-time of the derandomized algorithm is $O(nNm_s)$. Under the assumption $s \le N/3$, this is

$$O(nNm_s) = O\left(\frac{c^s N^{\frac{s+3}{2}}}{s^{\frac{s}{2}+\frac{3}{4}} \log\left(\frac{N}{s}\right)^{\frac{s+1}{2}}}\right),$$

where c is some constant independent of N and s.

We can use the derandomized algorithm iteratively. That is, we first use it to generate a point set $\mathcal{P}_1 \subset [0,1)$ and then to repeatedly add dimensions until we obtain the desired point set $\mathcal{P}_s$. When doing so, the final point set $\mathcal{P}_s$ satisfies the discrepancy estimate

$$D_N^*(\mathcal{P}_s) \le \left(\sqrt{3} + \frac{1}{\sqrt{2}}\right) \frac{s^{3/2}}{\sqrt{N}} \left(\log\left(\rho'(N,s)\right) + \frac{1}{s}\log(4)\right)^{1/2}.$$

2.2 Implementation Details

As seen in the previous subsection, we estimate the probability that some box $[0, t_i)$ receives too few or too many points by the sum (taken over the boxes) of the probabilities that the box has too few or too many points. We shall estimate such a failure probability by the sum of the two probabilities that (i) the box receives too few and (ii) too many points. For the case that the box receives too many points, [14, Sect. 3] gives an upper bound on the failure probability of

$$\exp(-c_i^+ W_i^+) \prod_{j=0}^{N-1} \left(\sum_{k=1}^{m} \widetilde{x}_{jk} \exp(1_{[0,t_i)}(y_j, \hat{k}) c_i^+) \right), \tag{8}$$

where c_i^+, W_i^+ and $\widetilde{x}_{jk}$ are as follows.

By $\widetilde{x}_{jk}$ we denote the expected value of the random variable X_{jk} if it has not been rounded, or the outcome of the rounding thereafter. Here and in the following, probabilities, expectations etc. refer to the randomized construction which we aim to derandomize. At no occasion, our algorithm will use randomness itself. Let W_i denote the expected number of points to lie in the box $[0, t_i)$. By construction, it

is equal to number of points $N\lambda_s([0,t_i))$ we aim at having in the box $[0,t_i)$. Let $W_i^+ \geq W_i$ be the maximum number of points we want to tolerate in this box. Define δ_i^+ via $(1+\delta_i^+)W_i = W_i^+$ and $c_i^+ := \log(1+\delta_i^+)$.

With these constants, the sum in equation (8) simplifies as follows. Let $y_j \in \mathcal{P}_{s-1}$ be such that y_j lies in the projection of $[0,t_i)$ to the first $s-1$ coordinates. If $(X_{jk})_{k=1}^m$ is unrounded, then $\sum_{k=1}^m \widetilde{x}_{jk} \exp(1_{[0,t_i)}(y_j,\hat{k})c_i^+) = 1+\delta_i^+ t_i(s)$, where $t_i(s)$ is the last coordinate of t_i. If $(X_{jk})_{k=1}^m$ is rounded, this sum evaluates to either $1+\delta_i^+$ or 1, depending on whether the jth point of $\mathcal{P}_s$ was placed inside box $[0,t_i)$ or not.

The case that the box receives too few points is not covered by Raghavan, but can be treated similarly by regarding the probability that the complement of the box receives too many points. Ignoring points that already miss box $[0,t_i)$ due to an earlier coordinate, the formulas work out the same, with $t_i(s)$ replaced by $1-t_i(s)$, and with a different error tolerance δ_i^-.

Thus we can compute the pessimistic estimator efficiently: when determining the rounded value for $(\widetilde{x}_{jk})_{k=1}^m$ for some j, we only need to replace the terms involving these variables with the m_s possible choices for $(\widetilde{x}_{jk})_{k=1}^m$. This can be done quite efficiently in time $O(nm_s)$, and both computing the initial value of the estimator and computing *all* subsequent values takes time $O(nNm_s)$.

The discussion so far works for all values of δ_i^+ and δ_i^-, provided the initial estimator evaluates to less than one. Potentially, since these represent guarantees on the star discrepancy of $\mathcal{P}_s$, they seem to provide many opportunities for minimizing the resulting discrepancy by setting them according to a desired total discrepancy bound. In practice, however, we observed best results by choosing each δ_i^+ and δ_i^- in such a way that the corresponding failure probability was just less than $1/(2n)$. The values for δ_i^+ and δ_i^- leading to these failure probabilities can be approximated conveniently via binary search.

To compute the discrepancy of the resulting point set, we could track the updates of the pessimistic estimators (since, as noted above, they contain a factor $1+\delta_i^+$ for each point placed inside the corresponding box). However, since the points are placed along a grid of reasonable size, it is just as practical to calculate the discrepancy explicitly in a naive way, that is, looking at all boxes constructible from the coordinates used by the point set, as described in [2, equation (4.1)] or [7, equation (1)].

3 Numerical Experiments

In our numerical experiments we use the star discrepancy of an output set as measure of quality. Calculating the actual star discrepancy of an arbitrary given set is a difficult problem, which was proved to be NP-hard in [7]. Also all known algorithms that approximate the star discrepancy up to some user-specified error have a run-time exponential in s and only a very limited range of application, see [17, 18, 6] and the references therein. If we construct an s-dimensional point set component by component via the CBC algorithm starting in dimension 1, then the output set is

a subset of the relatively small grid $\mathcal{G}_s$ and its exact discrepancy can be calculated without essentially increasing the effort. But if we start in some dimension $1 \leq s' < s$ with a given point set and extend it via the CBC algorithm to an s-dimensional point set (cf. Sect. 3.5) or if we randomize the output set (cf. Sect. 3.4), then the exact calculation of the star discrepancies will in general be infeasible. That is why we have to use in such cases different estimators of quality.

As for the memory and time requirements of the algorithm, and more generally the instance sizes for which our algorithm is feasible, the answer depends on whether memory or time is the critical issue. Since the algorithm very frequently needs to ask the question "does point number i lie inside box number j?", it speeds up the execution time significantly to store the answers to these questions in a data structure. If this is done, then the key issue becomes one of memory usage—at $s = 10$ and 500 points, the execution time on our computation server is roughly twenty minutes, but the memory requirements exceed 2 GB. On the other hand, without such caching, the running time for the same instance setting rises to need several hours. Going up from 500 to 1000 points at $s = 10$ scales the number of grid boxes by a factor of fifteen, and so would be infeasible with both our variants. In the rest of this section, when we talk about infeasible instances, we mean that the memory requirements for the caching method significantly exceed 2 GB. Our computation server is equipped with AMD Opteron 2220 SE 2.8 GHz processors and 16 GB of memory.

3.1 Dependence on the Dimension

An obvious disadvantage of the CBC approach of [2] compared to the multidimensional approach of [1] is how the discrepancy guarantee depends on the dimension s. While the latter has a discrepancy guarantee roughly proportional to $s^{1/2}$, the discrepancy guarantee of the CBC approach contains a factor of $s^{3/2}$.

To see to which extent the point sets actually display this disadvantage, we computed point sets of 1000 points each in dimension $s = 2, \ldots, 8$ via the CBC algorithm. Their discrepancies together with the theoretical guarantees proven in [2] are depicted in Figure 1. We should be cautious here, since we do not know how the discrepancies behave in higher dimensions. However, our data gives the impression that the actual discrepancies are not of order $s^{3/2}$, but rather depend linearly on the dimension.

3.2 Analysing the Discrepancy: Rounding Error vs. Placement Error

The description of the CBC approach in the previous section reveals that the increase of the discrepancy in a single iteration stems from two causes. One is the rounding

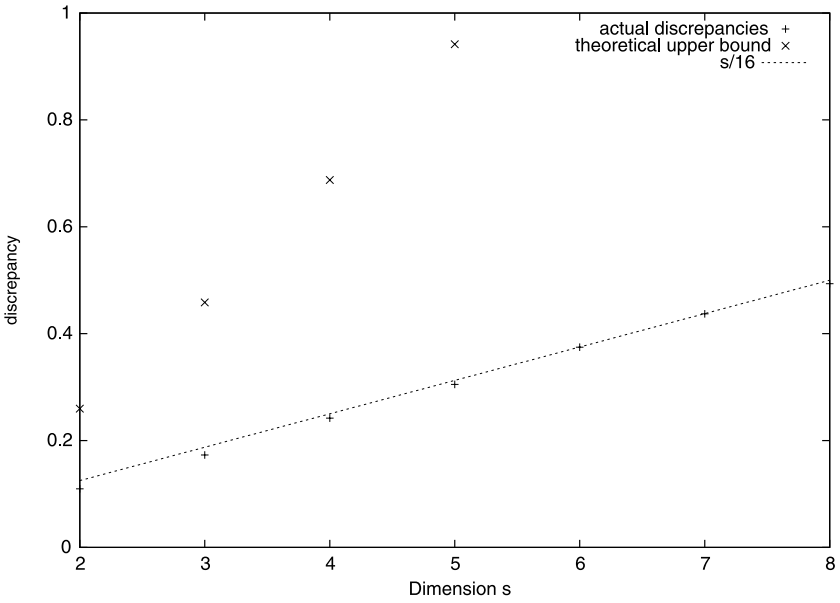


Fig. 1 Predicted and actual discrepancies of 1000 points constructed via the CBC approach.

error inflicted by the derandomized rounding procedure. This is the term $\frac{1}{N}||A(x) - A(\overline{x})||_\infty$ in (6), which is of order $O(\sqrt{\log(|\mathcal{G}_s^*|)/N})$, where we recall that $|\mathcal{G}_s^*| = \prod_{d=1}^{s} m_d$. The other is the additional error stemming from placing the points on the grid G_s in the sth coordinate, that is, by putting each point in the middle of an interval defined by G_s. This error was bounded by $1/(2m_s)$ in (6).

If we regard the outcome of all iterations, then the notion of *rounding error* naturally becomes the maximum discrepancy in a box aligned with the grid $\mathcal{G}_s^*$. We define

$$R(\mathcal{P}_s) := \max_{x \in \mathcal{G}_s^*} |\Delta_s(x, \mathcal{P}_s)|\,. \tag{9}$$

The additional error stemming from placing the points in the center of the grid cells of $\mathcal{G}_s^*$ (*placement error*) can be bounded by what we shall call the *grid gap*

$$\Delta(m_1, \ldots, m_s) := 1 - \prod_{d=1}^{s} (1 - 1/(2m_d)).$$

Indeed, if $\mathcal{P}_s^*$ is an arbitrary N-point set in $[0,1)^s$, and if $\mathcal{P}_s$ is the set we get after translating each point of $\mathcal{P}_s^*$ into the center of the grid cell of $\mathcal{G}_s^*$ it is contained in, then we get

$$D_N^*(\mathcal{P}_s) - \Delta(m_1, \ldots, m_s) \le D_N^*(\mathcal{P}_s^*) \le D_N^*(\mathcal{P}_s) + \Delta(m_1, \ldots, m_s).$$

In [2], the grid widths m_d were chosen in a way that the estimates for the rounding error and the placement error in a single iteration were roughly equal, and hence their sum was nearly minimized. Note that a finer grid naturally reduces the placement error, but at the same time increases the rounding error due to the $\sqrt{\log|\mathcal{G}_s^*|}$ term in the error bound (which is known to actually occur in many rounding problems). Since the actual discrepancies are much smaller than the predicted ones, it makes sense to analyze the contributions that each cause makes, and possibly adjust the trade-off between rounding error and placement error.

To this purpose, let us first note that the estimate for the placement error cannot be improved, and in consequence, that it always is a lower bound for the resulting discrepancy. Indeed, let $\mathcal{P}$ be any subset of the grid $\mathcal{G}_s$ (as, e.g., an output set of the CBC algorithm). Let $z_\varepsilon := (1-1/(2m_1)+\varepsilon, \ldots, 1-1/(2m_s)+\varepsilon)$ for a small $\varepsilon > 0$. Then $|\Delta_s(z_\varepsilon, \mathcal{P})| = 1-\lambda_s([0,z_\varepsilon)) = 1-\prod_{d=1}^{s}(1-1/(2m_d)+\varepsilon)$. Hence $\lim_{\varepsilon\to 0}|\Delta_s(z_\varepsilon,\mathcal{P})| = \Delta(m_1,\ldots,m_s)$. Thus $\Delta(m_1,\ldots,m_s)$ is a lower bound of $D_N^*(\mathcal{P})$.

A comparison of the actual discrepancy of point sets $\mathcal{P}_s$ in different dimensions s with the corresponding grid gap $\Delta(m_1,\ldots,m_s)$ and the occurring rounding error can be found in Figure 2; the data points relevant to this discussion are those labeled "standard grid".

The surprising results visible in the figure is that for these data points, the discrepancy $D_N^*(\mathcal{P}_s)$ is in every case very close to the trivial lower bound of $\Delta(m_1,\ldots,m_s)$. This is good news in the sense that the rounding procedure contributes almost nothing to the discrepancy. Furthermore, it means that an output set $\mathcal{P}_s$ of the CBC algorithm of size N has more or less the same discrepancy as the full grid $\mathcal{G}_s$, whose cardinality is roughly of order $O(N^{s/2}/\sqrt{s!})$ – note that for the interesting case of s not too small and $s \ll N$ the output set $\mathcal{P}_s$ is a sparse subset of the grid $\mathcal{G}_s$.

3.3 Finetuning the Algorithm

In the light of the previous insight, it makes sense to run the algorithm with finer grids than what was proposed in [2]. This would increase the currently almost nonexisting rounding error and reduce the currently dominant grid gap. Unfortunately, since the run-time of our algorithm is linear in the grid size, it also increases the run-time.

Figure 2 presents some data of this type. Besides showing what constructions are possible in reasonable time, we see the following. Clearly, making the grid finer does reduce the grid gap significantly. When taking twice as many grid subdivisions in each dimension, however, the rounding error remains insignificant compared to the grid gap. (We may appreciate this from the practical point of view, since it seems that we take full advantage of the finer grid.)

For the grid with four times the number of subdivisions, we do observe a visible rounding error, even if it is still small compared to the grid gap. This means that here we are getting closer to the optimal balance of rounding and placement error.

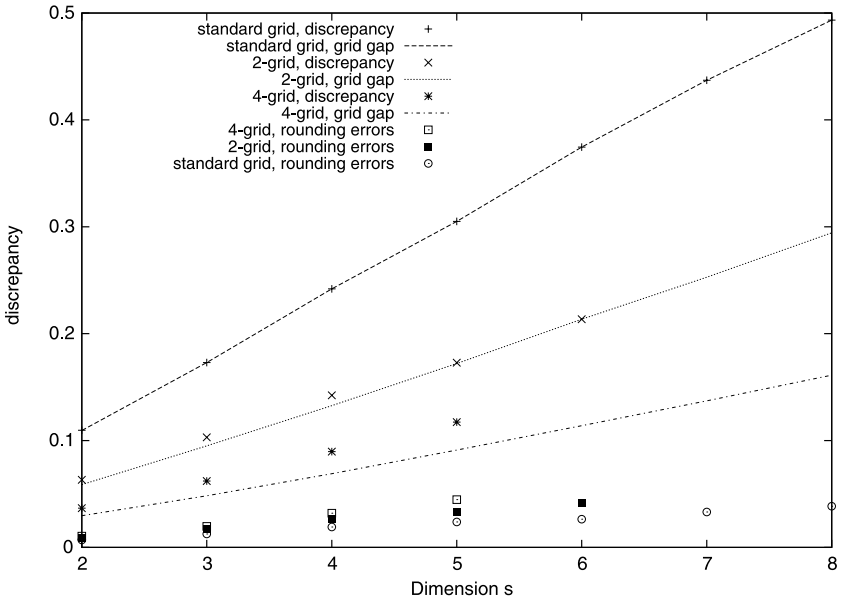


Fig. 2 Discrepancies, grid gaps and rounding errors observed for $N = 1000$ when using the grid proposed in [2] ("standard grid", m_s grid lines in dimension s), and grids with two ("2–grid") and four ("4–grid") times the number of grid subdivisions in each dimension.

Unfortunately, taking an even finer grid was not feasable. Already conducting the $s = 5$ experiment in the last grid took 4.5 hours of computation time (using the caching method, requiring 3 GB of memory; see beginning of Section 3).

3.4 Randomization of the Output Set

In the previous subsection, we saw that taking a finer grid does reduce the grid gap, but at the cost of computation times that quickly make this approach infeasible. Hence taking finer grids generally did not suffice to reduce the grid gap to an extent that the discrepancy guarantee was not dominated by the grid gap.

To overcome this issue, we now propose an additional idea to reduce the placement error and prove superior bounds on it. The idea is simple: Instead of placing the points on the centers of the grid cells, we place them on randomly chosen locations in the grid cell they belong to. Let us make this more precise: We consider the partition $\mathcal{B}^*$ of $[0, 1)^s$ into n axis-parallel half-open boxes of equal size with centers in $\mathcal{G}_s$ and right upper corners in $\mathcal{G}_s^*$. We transform the output set $\mathcal{P}_s$ of the CBC algorithm into a set $\mathcal{P}_s^*$ by substituting in each box $B \in \mathcal{B}^*$ the points $p \in \mathcal{P}_s \cap B$ by random points p^* that are independently and uniformly distributed within B. This

randomization may enhance the quality of the output set and has the advantage that $\Delta(m_1,\dots,m_s)$ is not necessarily any longer a lower bound for $D_N^*(\mathcal{P}_s^*)$.

The practical problem that occurs now is that the actual star discrepancy of $\mathcal{P}_s^*$ would be much harder to calculate than the one of the subset $\mathcal{P}_s$ of $\mathcal{G}_s$. Therefore we use an estimator for the discrepancy of $\mathcal{P}_s^*$, which we describe here shortly. To this purpose let us restate a definition and a lemma from [3].

Definition 1. A finite set $\Gamma \subset [0,1]^s$ is called a δ-cover of $[0,1]^s$ if for every $y=(y_1,\dots,y_s)\in[0,1]^s$ there are $x=(x_1,\dots,x_s), z=(z_1,\dots,z_s)\in\Gamma\cup\{0\}$ with $\lambda_s([0,z))-\lambda_s([0,x))\le\delta$ and $x_i\le y_i\le z_i$ for all $1\le i\le s$.

Lemma 1. *Let Γ be a δ-cover of $[0,1]^s$. Then for any N-point set $\mathcal{P}_s\subset[0,1)^s$ we have $D_N^*(\mathcal{P}_s)\le D_N^\Gamma(\mathcal{P}_s)+\delta$, where $D_N^\Gamma(\mathcal{P}_s)=\max_{x\in\Gamma}|\Delta_s(x,\mathcal{P}_s)|$.*

Let $R(\mathcal{P}_s)$ be the rounding error defined in (9). It is not hard to see that $\mathcal{G}_s^*$ is a δ-cover for

$$\delta = 1-\prod_{d=1}^{s}\left(1-\frac{1}{m_d}\right).$$

Due to Lemma 1 we get thus $D_N^*(\mathcal{P}_s^*)\le R(\mathcal{P}_s^*)+\delta = R(\mathcal{P}_s)+\delta$. But we can give a much better estimate: For $0<\delta'\le\delta$ and $p\in(0,1)$ define now

$$\theta=\theta(p,\delta,\delta')=\left(\frac{\delta+2R(\mathcal{P}_s)}{2N}\right)^{1/2}\left(\log\left(\frac{2}{p}\right)+Q(\delta')\right)^{1/2},$$

where

$$Q(\delta')=\min\left\{s\log(k),\log\left(\frac{2^s s^s}{s!}\right)+s\log((\delta')^{-1}+1)\right\}$$

and

$$k=k(\delta',s)=\left\lceil\frac{s}{s-1}\,\frac{\log(1-(1-\delta')^{1/s})-\log(\delta')}{\log(1-\delta')}\right\rceil+1.$$

Theorem 1. *$D_N^*(\mathcal{P}_s^*)\le R(\mathcal{P}_s)+\theta(p,\delta,\delta')+\delta'$ with probability at least $1-p$.*

The proof employs Hoeffding's large deviation bound for all test boxes $[0,x)$, with x from a minimal δ'-cover G'. To perform a union bound, the results [3, Thm.2.3] and [6, Thm.1.15] are used to upper-bound the cardinality of G'.

In our tests we chose $p=0.05$ and used the estimator

$$R(\mathcal{P}_s)+\min_{0<\delta'\le\delta}\big(\theta(0.05,\delta,\delta')+\delta'\big),$$

which is an upper bound for $D_N^*(\mathcal{P}_s^*)$ with probability at least 95%.

Figure 3 shows the resulting estimates for the discrepancies of randomized output sets $\mathcal{P}_s^*$ in dimension $s=10$ for values of N between 50 and 500, together with the corresponding rounding and placement errors of the related (deterministic) output sets $\mathcal{P}_s$. These experiments strongly indicate that the final (local) randomization procedure enhances the quality of the output sets significantly, since the estimated

discrepancies are all clearly smaller than the corresponding grid gaps, which are lower bounds for the discrepancies of the (non-randomized) output sets. Therefore the randomized CBC method seems to be a promising alternative to the conventional CBC method which overcomes the lower discrepancy bound in form of the grid gap.

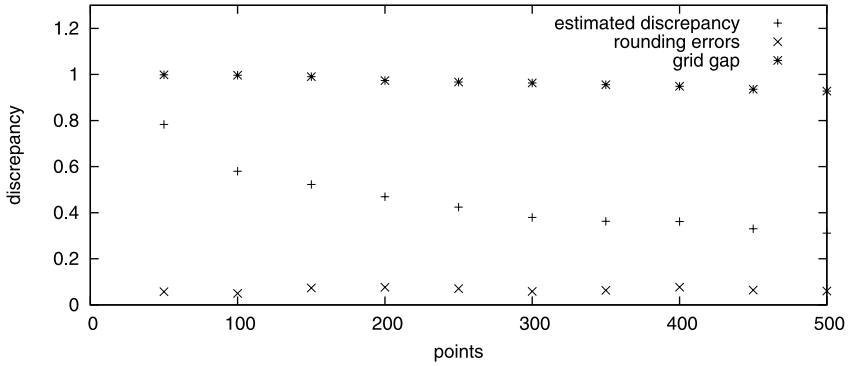


Fig. 3 Comparison of the estimated discrepancy of 10-dimensional randomized output sets with the corresponding rounding errors and grid gaps.

In Table 1 we listed the run-times of the randomized CBC algorithm for some representative values of N. Note that the CBC algorithm used is the version with higher memory requirements (see beginning of Section 3), and that the time needed for the final randomization of the point set is negligible.

Table 1 Times for creating point sets using the randomized CBC method.

N	Running time, $s = 5$	Running time, $s = 10$
100	0.037s	0.598s
300	0.644s	1m45.967s
500	2.196s	20m49.5s

3.5 A Hybrid Approach: Extending Halton-Hammersley Point Sets

One of the strengths of the CBC method is that we may also use it to extend an existing point set to one in higher dimension. This could be a promising idea, because the classical constructions lead to very good point sets in small dimensions. This advantage is used by a related idea called padding quasi-Monte Carlo (QMC) by Monte Carlo (MC) – there a low-discrepancy sequence is extended in the dimension by choosing the additional coordinates randomly, see, e.g., [15, 13, 5]. The resulting point sets show in many applications a better performance than pure QMC

or MC points. Here we want to study whether the same is true for extending low-discrepancy point sets by the CBC algorithm. Therefore, we pursue the following.

Fix N, the number of points to be constructed, and s, the dimension the final point set shall have. For some $s' \leq s$, let $\mathcal{P}^*_{s'}$ be a Halton-Hammersley point set of N points in dimension s' (see, e.g., [8] or [12]). Then we perform $s - s'$ CBC iterations to obtain an N-point set $\mathcal{P}_s$ in $[0,1)^s$. We want to stress that in our experiments the decisions of the CBC algorithm which additional components to choose to extend $\mathcal{P}_{s'}$ are only based on the discrepancies induced by all boxes of the form $[0,t)$, $t \in \mathcal{G}^*_d$, $d = s'+1,\ldots,s$. (In general it would be possible to substitute $\mathcal{G}^*_d$ by a finer grid whose projection onto the first s' components is the grid generated by the coordinate values of the points of $\mathcal{P}_{s'}$. But such an s'-dimensional grid is already of size $\Theta(N^{s'})$ and will therefore increase the computation time crucially.) If $s' \geq 1$, the resulting point set $\mathcal{P}_s$ is not any longer a subset of the (relatively small) grid $\mathcal{G}_s$. For $s = 10$ and N in the hundreds this means practically that calculating the exact discrepancy of $\mathcal{P}_s$ is in general infeasible. Therefore we restrict ourselves to computing the rounding errors of the resulting point sets, as defined in (9).

We performed our experiments for $s = 10$. The results for $s' = 0$ (the pure CBC approach), 3, 5, 7 and 10 (the pure 10–dimensional Halton-Hammersley set) and $N = 50, 100, 150, \ldots, 500$ are depicted in Figure 4.

The results support our theoretical considerations. For small numbers of points, the relatively large discrepancy of high-dimensional Halton-Hammersley sets leads

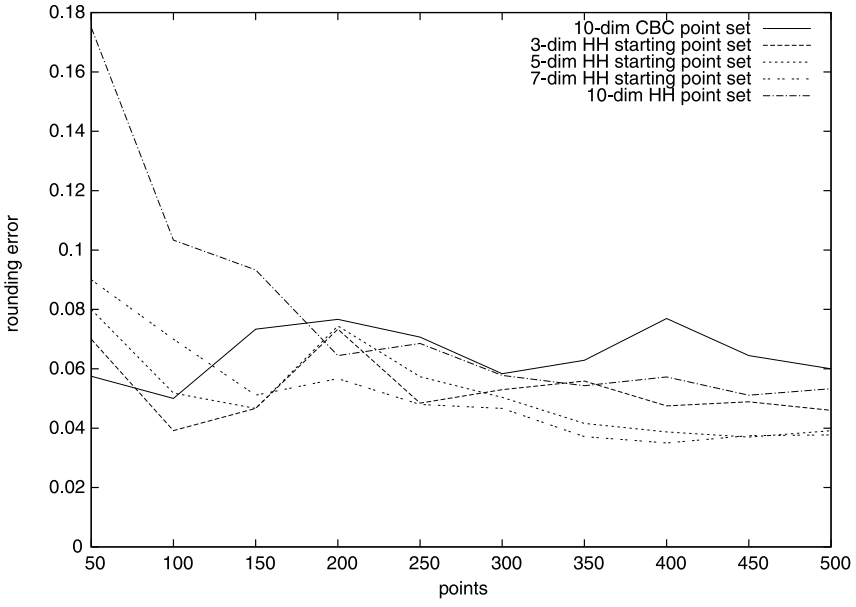


Fig. 4 Rounding errors of 10-dimensional point sets stemming from different versions of the hybrid approach (starting with a 3, 5 and 7 dimensional Halton-Hammersley set and then adding dimensions via the CBC approach), together with the two pure approaches.

to inferior results for large values of s'. For larger numbers of points, the pure Halton-Hammersley set is not very good, but becomes better than the pure CBC construction. Better results are obtained for the intermediate values $s' = 3, 5, 7$.

Acknowledgements The authors thank the editor Art B. Owen and two anonymous referees for their comments, which greatly helped to improve the presentation. Benjamin Doerr and Magnus Wahlström gratefully acknowledge support from the German Science Foundation (DFG) via its priority programme "SPP 1307: Algorithm Engineering", grant DO 479/4-1. Michael Gnewuch gratefully acknowledges support from the DFG under grant GN 91/3-1.

References

1. Doerr, B., Gnewuch, M.: Construction of low-discrepancy point sets of small size by bracketing covers and dependent randomized rounding. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 299–312. Springer-Verlag, Berlin Heidelberg (2008)
2. Doerr, B., Gnewuch, M., Kritzer, P., Pillichshammer, F.: Component-by-component construction of low-discrepancy point sets of small size. Monte Carlo Methods Appl. **14**, 129–149 (2008)
3. Doerr, B., Gnewuch, M., Srivastav, A.: Bounds and constructions for the star discrepancy via δ-covers. J. Complexity **21**, 691–709 (2005)
4. Doerr, B., Wahlström, M.: Randomized rounding in the presence of a cardinality constraint. In: Proceedings of ALENEX, pp. 162–174. SIAM (2009)
5. Gnewuch, M.: On probabilistic results for the discrepancy of a hybrid-Monte Carlo sequence. J. Complexity **25**, 312–317 (2009)
6. Gnewuch, M.: Bracketing numbers for axis-parallel boxes and applications to geometric discrepancy. J. Complexity **24**, 154–172 (2008)
7. Gnewuch, M., Srivastav, A., Winzen, C.: Finding optimal volume subintervals with k points and calculating the star discrepancy are NP-hard problems. J. Complexity **25**, 115–127 (2009)
8. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multidimensional integrals. Numer. Math. **2**, 84–90 (1960)
9. Heinrich, S.: Some open problems concerning the star-discrepancy. J. Complexity **19**, 416–419 (2003)
10. Heinrich, S., Novak, E., Wasilkowski, G.W., Woźniakowski, H.: The inverse of the star-discrepancy depends linearly on the dimension. Acta Arith. **96**, 279–302 (2001)
11. Hickernell, F.J., Sloan, I.H., Wasilkowski, G.W.: On tractability of weighted integration over bounded and unbounded regions in $\mathbb{R}^s$. Math. Comp. **73**, 1885–1901 (2004)
12. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia (1992)
13. Ökten, G., Tuffin, B., Burago, V.: A central limit theorem and improved error bounds for a hybrid-Monte Carlo sequence with applications in computational finance. J. Complexity **22**, 435–458 (2006)
14. Raghavan, P.: Probabilistic construction of deterministic algorithms: Approximating packing integer programs. J. Comput. Syst. Sci. **37**, 130–143 (1988)
15. Spanier, J.: Quasi-Monte Carlo methods for particle transport problems. In: H. Niederreiter, P.J.S. Shiue (eds.) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, pp. 121–148. Springer-Verlag, Berlin (1995)
16. Spencer, J.: Ten Lectures on the Probabilistic Method. SIAM, Philadelphia (1987)
17. Thiémard, E.: An algorithm to compute bounds for the star discrepancy. J. Complexity **17**, 850–880 (2001)

18. Thiémard, E.: Optimal volume subintervals with k points and star discrepancy via integer programming. Math. Meth. Oper. Res. **54**, 21–45 (2001). Extended version available at http://ina2.eivd.ch/Collaborateurs/etr/

Quasi-Monte Carlo Simulation of Diffusion in a Spatially Nonhomogeneous Medium

Rami El Haddad, Christian Lécot, and Gopalakrishnan Venkiteswaran

Abstract We propose and test a quasi-Monte Carlo (QMC) method for solving the diffusion equation in the spatially nonhomogeneous case. For a constant diffusion coefficient, the Monte Carlo (MC) method is a valuable tool for simulating the equation: the solution is approximated by using particles and in every time step the displacement of each particle is drawn from a Gaussian distribution with constant variance. But for a spatially dependent diffusion coefficient, the straightforward extension using a spatially variable variance leads to biased results. A correction to the Gaussian steplength was recently proposed and provides satisfactory results. In the present work, we devise a QMC variant of this corrected MC scheme. We present the results of some numerical experiments showing that our QMC algorithm converges better than the corresponding MC method for the same number of particles.

1 Introduction

The random walk technique is commonly used for investigating processes involving the diffusion of substances. In the case of a constant diffusion coefficient D_0, the substance is modelled by an assemblage of discrete particles. Time is discretized and in a short time interval Δt, each particle is moved with an increment chosen from a

Rami El Haddad
Département de Mathématiques, Faculté des Sciences, Université Saint-Joseph, BP 11-514 Riad El Solh, Beyrouth 1107 2050, Liban
e-mail: rami.haddad@fs.usj.edu.lb

Christian Lécot
Laboratoire de Mathématiques, UMR 5127 CNRS & Université de Savoie, Campus scientifique, 73376 Le Bourget-du-Lac Cedex, France
e-mail: Christian.Lecot@univ-savoie.fr

Gopalakrishnan Venkiteswaran
Department of Mathematics, Birla Institute of Technology and Science, Vidya Vihar Campus, Pilani 333 031 Rajasthan, India
e-mail: gvenki@bits-pilani.ac.in

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_21,

Gaussian distribution with variance $2D_0\Delta t$. Advantages of the MC method include the simplicity of the algorithm and the ability to deal with complicated geometries. A disadvantage is that many runs are often needed to obtain reliable mean values.

In some cases, one may have a diffusion coefficient that varies with position. The naive extension of the constant diffusion coefficient case is to use a spatially variable variance $2D(x)\Delta t$. It is found in practice that this leads to an apparent advection in directions of decreasing diffusivity and a concentration of particles in regions of low diffusivity [4, 1, 2]. This can be explained as follows (see [4]). Let c denote the concentration, i.e., the solution of the diffusion equation. If everything is smooth, one can define (at least in the one-dimensional case) a new coordinate system and a transformed concentration γ such that the diffusion coefficient becomes constant. But the transformed equation includes advection in directions of increasing diffusivity. The naive approach of MC simulation neglects this advection and leads to biased results. A simple correction to the Gaussian steplength has been recently proposed [1]. It was shown that the procedure gives satisfactory results.

One possibility to improve the accuracy of the MC method is to replace pseudo-random numbers with quasi-random numbers; in addition, sorting the ensemble of particles at each time step can improve convergence [8].

In this paper we propose a quasi-Monte Carlo version of the corrected MC algorithm of [1]. It aims to solve problems involving the diffusion of substances in a spatially nonhomogeneous medium. The quasi-random numbers used are (t, m, s)-nets: we refer to [9] for a comprehensive and detailed exposition of the theory of (t, m, s)-nets and (t, s)-sequences. The number of simulation particles can vary with time: to make a proper use of the better uniformity of nets, we split the ensemble of particles into subsets after each time step.

The paper is organized as follows. In Section 2, we recall the classical MC scheme for the simulation of diffusion in a one-dimensional homogeneous medium and we introduce a QMC version of the algorithm. In Section 3, we consider the nonhomogeneous case and we present the correction to the Gaussian steplength proposed in [1]. In Section 4, we describe the QMC strategy for the corrected scheme. In Section 5, we report the results of numerical experiments that compare our method with standard MC through the simulation of diffusion of calcium ions in both cases of constant and variable diffusivity. Conclusions are stated in the final section.

2 Simulation of Diffusion Using a Random Walk Method

The diffusion equation describes the transport of molecules from regions of high concentration to those of low concentration, attributed to Brownian motion. For the simplest case of an infinite 1-dimensional medium, the equation is as follows.

$$\frac{\partial c}{\partial t}(x,t) = \frac{\partial}{\partial x}\Big(D\frac{\partial c}{\partial x}\Big)(x,t), \quad x \in \mathbf{R}, \quad t > 0, \tag{1}$$

$$c(x,0) = c_0(x), \quad x \in \mathbf{R}, \tag{2}$$

where $c(x,t)$ is the concentration of particles at location x and time t and $D = D(x,t)$ is the diffusion coefficient.

Mass conservation is expressed as:

$$\forall t > 0 \quad \int_{\mathbf{R}} c(x,t)dx = \int_{\mathbf{R}} c_0(x)dx, \tag{3}$$

if we suppose that c_0 is a nonnegative integrable function.

Diffusion processes arise in many engineering applications, and a large body of numerical methods exists for the solution of the diffusion equation. When other mechanisms (convection or reaction) besides diffusion are involved, standard computational algorithms (finite differences or finite elements methods) may suffer from problems associated with the discretization techniques. In such situations, grid-free methods are developed. Here we explore the simulation of diffusion using stochastic particle methods [3].

In the case of a constant diffusion coefficient D_0, equation (1) can be written

$$\frac{\partial c}{\partial t}(x,t) = D_0 \frac{\partial^2 c}{\partial x^2}(x,t). \tag{4}$$

The *fundamental solution* of equation (4) is

$$E(x,t) := \frac{\mathrm{e}^{-x^2/4D_0 t}}{\sqrt{4\pi D_0 t}}, \quad x \in \mathbf{R},\ t > 0.$$

Hence, if a particle is released from the origin, the probability of finding it in an infinitesimal range dx around x after a time step Δt is $f_X(x)dx$, where

$$f_X(x) = \frac{\mathrm{e}^{-x^2/4D_0\Delta t}}{\sqrt{4\pi D_0 \Delta t}}. \tag{5}$$

This is the density of a Gaussian distribution with zero mean and variance $2D_0\Delta t$.

This gives the basis of the random walk scheme for solving the diffusion equation. The simulation is conducted by first sampling N particles from the initial distribution $c_0(x)$. Time is discretized using a time step Δt. At every time step, each particle is moved by a random displacement drawn from a Gaussian distribution with mean zero and variance $2D_0\Delta t$: the random distance Δx that the particle moves can be computed as

$$\Delta x = \sqrt{2D_0\Delta t}\, Z, \tag{6}$$

where Z is a standard Gaussian random variable.

A deterministic version of the previous algorithm was proposed in [8] and generalized to the multidimensional case in [6]. In the MC scheme, the standard Gaussian random variable can be generated from (uniform) pseudo-random numbers using inversion technique. A simple replacement of these numbers by quasi-random ones may destroy the method, because the quasi-random points are highly correlated. It was found that if the particles are relabeled according to their position at each time

step, the difficulty disappears: an error bound is given in arbitrary dimension; the results of numerical experiments in one and two dimensions show that the QMC version outperforms the original MC method (see [6]). The reordering technique was first used in [5] for the QMC simulation of the Boltzmann equation. We sketch the quasi-random walk method described in [6].

We consider equation (4) and we normalize the initial data so that

$$\int_{\mathbf{R}} c_0(x)dx = 1. \tag{7}$$

Let $\mathcal{M}_+(\mathbf{R})$ denote the set of all measurable nonnegative functions on $\mathbf{R}$. Using the fundamental solution $E(x,t)$, we can write for $t \geq 0$ and $\Delta t \geq 0$:

$$c(x, t+\Delta t) = \int_{\mathbf{R}} E(x-y, \Delta t)c(y,t)dy, \quad x \in \mathbf{R}. \tag{8}$$

We choose integers $b \geq 2$, $m \geq 0$ and we put $N := b^m$. We generate N samples x_j^0, $0 \leq j < N$ (particles) from the initial probability distribution. This can be done by the inversion method:

$$x_j^0 := C_0^{-1}(\xi_j), \tag{9}$$

where $C_0(x)$ is the initial cumulative distribution function

$$C_0(x) := \int_{-\infty}^{x} c_0(y)dy,$$

and $\{\xi_0, \ldots, \xi_{N-1}\}$ is a one-dimensional low discrepancy point set. A possible choice is

$$\xi_j := \frac{2j+1}{2N} \quad \text{for } 0 \leq j < N,$$

since the minimum of the discrepancy is attained for these points (see [9]). For the quasi-random walks, we need a low discrepancy sequence: $Y = \{\mathbf{y}_0, \mathbf{y}_1, \ldots\} \subset [0,1)^2$. For $n \in \mathbf{N}$, denote

$$Y^n := \{\mathbf{y}_\ell : nN \leq \ell < (n+1)N\}.$$

If Π' and Π'' are the maps defined by

$$\Pi'(\mathbf{u}) := u_1, \quad \Pi''(\mathbf{u}) := u_2, \quad \mathbf{u} = (u_1, u_2) \in [0,1)^2,$$

we assume that for all $n \in \mathbf{N}$,

$$\Pi' Y^n \quad \text{is a } (0, m, 1)\text{-net in base } b, \tag{10}$$

$$\Pi'' Y^n \subset (0,1). \tag{11}$$

We discretize time into intervals of length Δt; we set $t_n := n\Delta t$ and $c_n(x) := c(x, t_n)$. Assuming that we have computed an approximation c^n of c_n by a sum of Dirac masses (particles) located at positions $x_0^n, \ldots, x_{N-1}^n$:

$$c^n := \frac{1}{N} \sum_{0 \le j < N} \delta_{x_j^n} \approx c_n,$$

we compute an approximation c^{n+1} at time t_{n+1} in two steps.

(1) *Sorting the particles.* The particles are labeled such that

$$x_0^n \le x_1^n \le \cdots \le x_{N-1}^n. \tag{12}$$

This reordering was first introduced in [5]. It guarantees convergence of the scheme (see [6]).

(2) *Updating the positions of particles through QMC integration.* Replacing c by c^n on the right hand side of equation (8), we define an approximation of the solution at time t_{n+1} of the form:

$$\widetilde{c}^{n+1}(x) := \frac{1}{N} \sum_{0 \le j < N} E(x - x_j^n, \Delta t).$$

Consequently, for all $\chi \in \mathcal{M}_+(\mathbf{R})$ we have

$$\int_{\mathbf{R}} \chi(x) \widetilde{c}^{n+1}(x) dx = \frac{1}{N} \sum_{0 \le j < N} \int_0^1 \chi\left(x_j^n + \sqrt{2D_0 \Delta t}\, \Phi^{-1}(u)\right) du, \tag{13}$$

where Φ is the standard normal cumulative distribution function. Let 1_j denote the indicator function of the interval

$$I_j := \left[\frac{j}{N}, \frac{j+1}{N}\right).$$

To $\chi \in \mathcal{M}_+(\mathbf{R})$, we associate $C^n \chi$ the function defined on $[0,1) \times (0,1)$ by

$$C^n \chi(\mathbf{u}) := \sum_{0 \le j < N} 1_j(u_1) \chi\left(x_j^n + \sqrt{2D_0 \Delta t}\, \Phi^{-1}(u_2)\right), \quad \mathbf{u} = (u_1, u_2).$$

Then

$$\forall \chi \in \mathcal{M}_+(\mathbf{R}) \quad \int_{\mathbf{R}} \chi(x) \widetilde{c}^{n+1}(x) dx = \int_{(0,1)^2} C^n \chi(\mathbf{u}) d\mathbf{u}. \tag{14}$$

Using QMC integration, we determine the approximation c^{n+1} by:

$$\forall \chi \in \mathcal{M}_+(\mathbf{R}) \quad \int_{\mathbf{R}} \chi(x) c^{n+1}(x) = \frac{1}{N} \sum_{0 \le k < N} C^n \chi(\mathbf{y}_{nN+k}). \tag{15}$$

This can be reworded as follows. For $u \in [0,1)$, put $j(u) := \lfloor Nu \rfloor$, where $\lfloor x \rfloor$ denotes the greatest integer $\le x$. It follows from (10) that the mapping

$$\ell \in \{nN, nN+1, \ldots, (n+1)N-1\} \to j(y_{\ell,1})$$

is one-to-one. Then, the positions at time t_{n+1} are defined by:

$$x^{n+1}_{j(y_{nN+k,1})} = x^n_{j(y_{nN+k,1})} + \sqrt{2D_0\Delta t}\,\Phi^{-1}(y_{nN+k,2}), \quad 0 \le k < N \tag{16}$$

(note that condition (11) ensures that it is possible to compute $\Phi^{-1}(y_{nN+k,2})$). The convergence of the algorithm was established and the results of numerical experiments were reported in [6]. The following kind of convergence is guaranteed (here we restrict ourselves to the one-dimensional case for simplicity). Let $X^n := \{x^n_j : 0 \le j < N\}$ be the set of approximating particles at time t_n, and $D^*_N(X^n; c_n)$ be the star c_n-discrepancy of X^n, defined as

$$D^*_N(X^n; c_n) := \sup_{y\in\mathbf{R}} \left| \frac{1}{N} \sum_{0\le j<N} 1_y(x^n_j) - \int_{\mathbf{R}} 1_y(x) c_n(x)\,dx \right|,$$

where 1_y denotes the indicator function of the interval $(-\infty, y)$. This is the Kolmogorov distance of the distributions defined by c^n and c_n. Then it is proved that $D^*_N(X^n; c_n) = \mathcal{O}(N^{-1/2})$. This is a worst-case error estimate, so the result is stronger than the probabilistic rate of $\mathcal{O}(N^{-1/2})$ for MC. The computational experiments indicate that a significant improvement is achieved over standard MC simulation. The convergence rate of QMC in the one-dimensional test cases chosen in [6] is $\mathcal{O}(N^{-0.71})$.

For MC, the positions of the particles are updated as follows:

$$x^{n+1}_j = x^n_j + \sqrt{2D_0\Delta t}\,z_{nN+j}, \quad 0 \le j < N, \tag{17}$$

where $\{z_\ell : \ell \ge 0\}$ is a sequence of normally distributed random numbers, with mean 0, and variance 1.

In some applications, one may have a diffusion coefficient that varies with distance. In this case, straightforward modification of the previous method, involving replacing the constant diffusion coefficient by the spatially dependent one in the steplength formula, leads to a systematic error. A correction to the steplength was proposed by Farnell and Gibson [1]. We recall their algorithm in the next section.

3 Correction to the Steplength

We suppose that the diffusion coefficient $D(x) > 0$ depends on position. Let $x_0 \in \mathbf{R}$ and f_X be the probability density function:

$$f_X(x) := \frac{\mathrm{e}^{-x^2/4D(x_0)\Delta t}}{\sqrt{4\pi\, D(x_0)\Delta t}}.$$

If $\Delta x := \sqrt{2D(x_0)\Delta t}\,Z$ is the uncorrected steplength, it is related to a uniform random deviate U by:

$$U = \int_{-\infty}^{\Delta x} f_X(x)\,dx. \tag{18}$$

If f_W is the density function of the exact steplength Δw, then

$$U = \int_{-\infty}^{\Delta w} f_W(w)dw. \tag{19}$$

The correction ϵ satisfies: $\Delta w = \Delta x + \epsilon$. Equations (18)–(19) lead to:

$$\begin{aligned}\int_{-\infty}^{\Delta x} f_X(x)dx &= \int_{-\infty}^{\Delta x} f_W(w)dw + \int_{\Delta x}^{\Delta x+\epsilon} f_W(w)dw \\ &= \int_{-\infty}^{\Delta x} f_W(w)dw + \epsilon f_W(\Delta x) + \mathcal{O}(\epsilon^2),\end{aligned}$$

if f_W is differentiable. The first order correction term is then:

$$\epsilon = \frac{\int_{-\infty}^{\Delta x} f_X(x)dx - \int_{-\infty}^{\Delta x} f_W(w)dw}{f_W(\Delta x)}. \tag{20}$$

This formula is not computationally convenient, since it requires the evaluation of integrals; we are looking for an easily computable correction. We first consider the linear case.

Linearly varying diffusion coefficient. We suppose that

$$D(x) := D_0(1 + \alpha x),$$

where $D_0 > 0$ and α are constants. In this case, if $c_0 = \delta_0$, the solution of the diffusion equation (1)–(2) is, for $1 + \alpha x > 0$:

$$c(x,t) = \frac{1}{D_0|\alpha|t} \mathrm{e}^{-(2+\alpha x)/D_0\alpha^2 t} I_0\left(\frac{2\sqrt{1+\alpha x}}{D_0\alpha^2 t}\right), \tag{21}$$

where I_0 is the modified Bessel function. Hence the density function of the exact steplength is given by:

$$f_W(w) = \frac{1}{D_0|\alpha|\Delta t} \mathrm{e}^{-(2+\alpha w)/D_0\alpha^2\Delta t} I_0\left(\frac{2\sqrt{1+\alpha w}}{D_0\alpha^2\Delta t}\right). \tag{22}$$

Using an asymptotic expansion of I_0, we can write, for small values of Δt and $w = \mathcal{O}(\sqrt{\Delta t})$:

$$f_W(w) = \frac{\mathrm{e}^{-w^2/4D_0\Delta t}}{\sqrt{4\pi D_0\Delta t}}\left(1 - \frac{\alpha}{4}w + \frac{\alpha}{8D_0\Delta t}w^3 + \mathcal{O}(w^2)\right). \tag{23}$$

Putting this into equation (20), we get:

$$\epsilon \approx \epsilon_1 := \frac{\frac{\alpha D_0\Delta t}{2} + \frac{\alpha(\Delta x)^2}{4}}{1 - \frac{\alpha\Delta x}{4} + \frac{\alpha(\Delta x)^3}{8D_0\Delta t}}. \tag{24}$$

If a particle starts from 0, the corrected steplength is $\Delta x + \epsilon_1$, with Δx given by (6).

General variable diffusion coefficient. In the general case, we use a local linear approximation of $D(x)$. If x_0 denotes the location of a particle, we write:

$$D(x_0+\Delta x) = D(x_0)\left(1+\frac{D'(x_0)}{D(x_0)}\Delta x\right)+\mathcal{O}((\Delta x)^2). \tag{25}$$

The random distance that the particle located at x_0 moves in a time interval Δt is $\Delta x+\epsilon$, with

$$\Delta x := \sqrt{2D(x_0)\Delta t}\,Z \qquad \text{and}$$
$$\epsilon := \frac{\frac{D'(x_0)\Delta t}{2}+\frac{D'(x_0)(\Delta x)^2}{4D(x_0)}}{1-\frac{D'(x_0)\Delta x}{4D(x_0)}+\frac{D'(x_0)(\Delta x)^3}{8(D(x_0))^2\Delta t}},$$

where Z is a standard Gaussian random variable.

In the next section, we propose a QMC version of the corrected random walk scheme described above.

4 QMC Scheme in a Nonhomogeneous Medium

The scheme is based on the QMC method described in Section 2. To make a better use of the outstanding uniformity of nets, the number of particles of this method was kept constant. In some cases, the number of particles used may vary in time. We generalize the method so that it can be applied to this case.

4.1 Approximation with a Fixed Number of Particles

Let $b \geq 2$, $m \geq 0$ and $N := b^m$. We use a low discrepancy sequence $Y = \{\mathbf{y}_0, \mathbf{y}_1, \ldots\} \subset [0,1)^2$. If Y^n denotes the n-th segment of length N of the sequence, we suppose that conditions (10) and (11) are fulfilled. Let Δt be a time step, $t_n := n\Delta t$, and $c_n(x) := c(x,t_n)$ for $n \in \mathbf{N}$. We are looking for an approximation of c_n of the form

$$c^n := \frac{1}{N}\sum_{0\leq j<N}\delta_{x_j^n},$$

where $x_0^n, \ldots, x_{N-1}^n$ are the positions of the particles at time t_n. The algorithm consists of several steps.

Initialization. We introduce a set of N particles sampled from the initial distribution c_0. The locations $\{x_0^0, \ldots, x_{N-1}^0\}$ may be defined by equation (9). In our numerical experiments, the particles are released from a point source at origin; in this case we take $x_0^0 = \cdots = x_{N-1}^0 := 0$.

Displacement of particles. Suppose that at time t_n, the particles are in positions $x_0^n, x_1^n, \ldots, x_{N-1}^n$. We determine the positions at time t_{n+1} in two steps.

- *Reordering of particles.* This means relabeling the particles so that

$$x_0^n \le x_1^n \le \cdots \le x_{N-1}^n. \tag{26}$$

- *Quasi-random walk.* For $u \in [0,1)$, put $j(u) := \lfloor Nu \rfloor$. For $x \in \mathbf{R}$ and $u \in (0,1)$, let

$$f(x,u) := \sqrt{2D(x)\Delta t}\,\Phi^{-1}(u),$$

and

$$\epsilon(x,u) := \frac{\frac{D'(x)\Delta t}{2} + \frac{D'(x)(f(x,u))^2}{4D(x)}}{1 - \frac{D'(x)f(x,u)}{4D(x)} + \frac{D'(x)(f(x,u))^3}{8(D(x))^2\Delta t}}.$$

The new positions (at time t_{n+1}) are computed as follows:

$$x_{j(y_{nN+k,1})}^{n+1} = x_{j(y_{nN+k,1})}^n + f(x_{j(y_{nN+k,1})}^n, y_{nN+k,2}) + \epsilon(x_{j(y_{nN+k,1})}^n, y_{nN+k,2}),$$

for $0 \le k < N$.

4.2 Approximation with a Varying Number of Particles

In the QMC algorithm of Section 2, the number N of numerical particles is kept constant and equal to a power of some prime base b, i.e. $N = b^m$, for some integer $m > 0$. This condition ensures that the mapping

$$\ell \in \{nN, nN+1, \ldots, (n+1)N-1\} \to j(y_{\ell,1})$$

is one-to-one: each particle corresponds to a point of a $(0, m, 1)$-net.

In the particular application of interest to us, physical particles (calcium ions) are released from a point source over time intervals. We choose some base b. As the number of particles is changing, it is not always equal to a power of b. If N is the number of particles at time t_n, we write the digit expansion of N in base b:

$$N = \sum_{i=0}^{\infty} a_i(N) b^i,$$

where $0 \le a_i(N) < b$ and $a_i(N) = 0$ for all sufficiently large i. The set of N particles is split into $a_0(N)$ subsets of size b^0, $a_1(N)$ subsets of size b^1, and so forth. Each subset is treated apart (for relabeling and displacement). At the end of the time step all the subsets are merged in one set of N particles. Then we can add to the system the new particles that are released from the source during the time interval.

5 Numerical Examples

In this section, we assess the accuracy of the QMC algorithm described above and we compare it to a classical MC scheme. We are interested in biological applications and the context is Ca^{2+} diffusion inside the neuromuscular junction of the crayfish [7]. Ions are released from a point source as the result of a train of action potentials arriving at that location, with each impulse resulting in release. These ions then diffuse independently through the surrounding medium and act on receptors at certain distances from the release point, chosen as the origin of the x-axis.

We have performed three series of numerical experiments:

1. constant diffusion coefficient with a periodic instantaneous emission of particles;
2. variable diffusion coefficient with a periodic instantaneous emission of particles;
3. variable diffusion coefficient with a non-instantaneous periodic emission of particles.

In the first case, the exact solution is available, so that we can verify that the new method gives the correct answer and is more accurate than the MC scheme, using the same number of simulation particles. In the other cases, where no analytical solution is known, the results are compared with those given by a MC simulation using a very large number of simulation particles.

The quasi-random sequence used in the simulation is the $(0, 2)$-sequence of Faure in base $b = 2$ (see [9] for details). We consider three successive emissions of particles. The period is equal to 10; the final time is then $T = 30$. The number of time steps is $P = 187\,500$ (62 500 time steps per period). Hence $\Delta t := T/P = 1.6\ 10^{-4}$. The number of particles released in every period is $N = 2^{13}$.

The concentration in a space interval is computed as the number of particles in the interval divided by the total number of particles. To smooth the curves and to make comparison clearer, we average the results over a time interval of amplitude 0.16. We compare the MC and QMC results by calculating the time evolution of the concentration of ions in the following space intervals:

$$[40, 60], \quad [100, 120], \quad [180, 200].$$

5.1 Constant Diffusion Coefficient and Instantaneous Emission

Here $D = 4\,000$. The results of simulations are compared with the exact solution which is available in this case. The concentrations in the varying intervals are shown in Figure 1. The QMC results agree closely with the analytical solution; they clearly outperform the MC outputs. We compute the mean of the absolute error in concentration on the intervals chosen: the results of MC and QMC simulations are compared in Table 1.

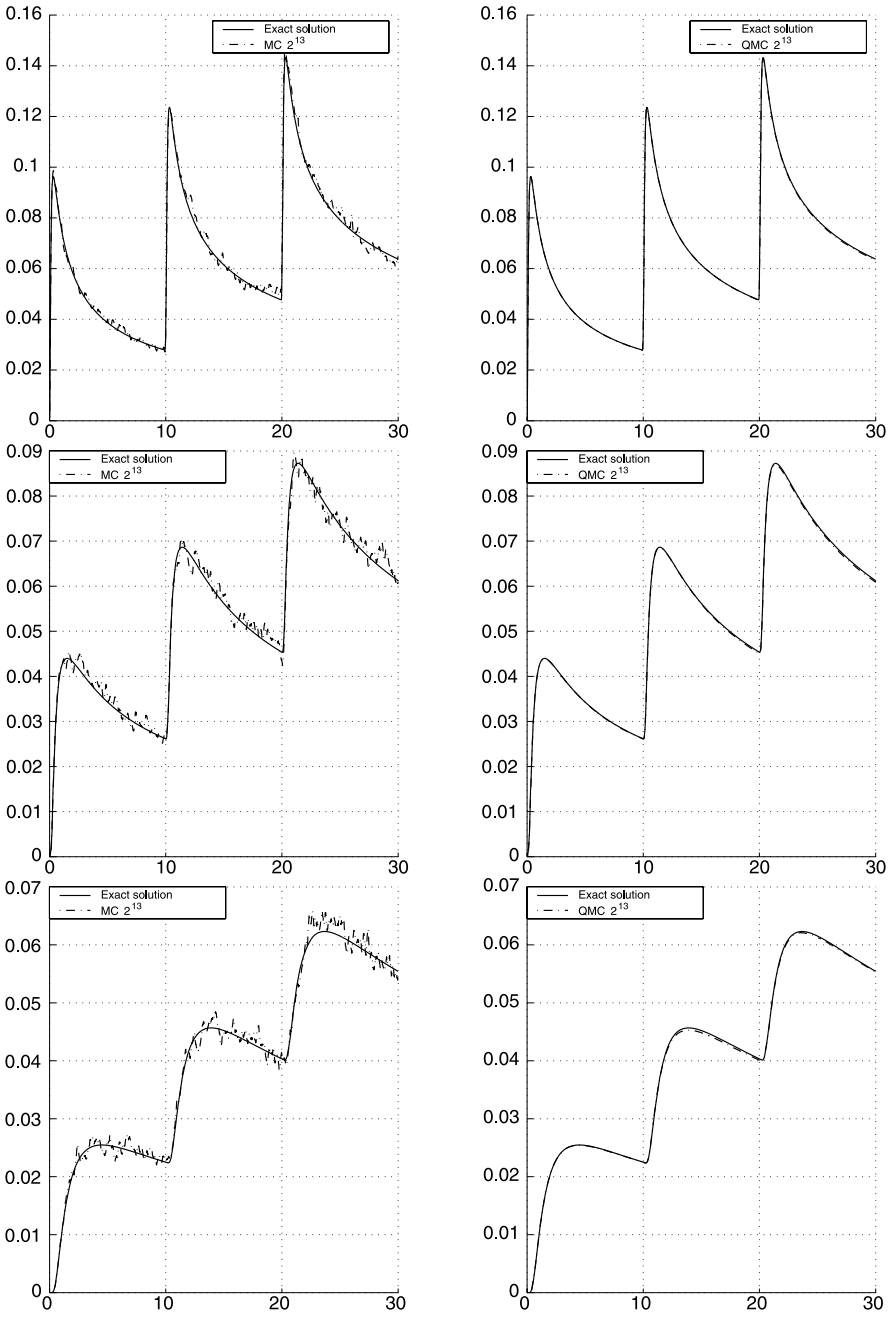


Fig. 1 Constant diffusion coefficient and instantaneous emission. Time evolution of the concentration in the intervals [40, 60] (up), [100, 120] (middle), and [180, 200] (down). Comparison of MC (left) and QMC (right) simulations.

Table 1 Mean error in concentration on three space intervals: MC vs. QMC.

Space interval	$[40, 60]$	$[100, 120]$	$[180, 200]$
MC	0.001590	0.001384	0.001038
QMC	0.000146	0.000181	0.000172

5.2 Variable Diffusion Coefficient and Instantaneous Emission

We consider the spatially varying diffusion coefficient used in [1], after nondimensionalization:

$$D(x) := \widehat{D}(1 - 0.8u(x)), \tag{27}$$

where $\widehat{D} := 4000$ and

$$u(x) := \frac{1}{2}(\tanh(A(b-x)) + 1), \quad A = 3.5\ 10^{-2}, \quad b = 200.$$

In this case, no exact solution is known. We compare the results with the outputs of a MC simulation using 2^{20} particles. The concentrations are displayed in Figure 2. Once again, the quasi-random strategy produces more accurate approximations than the standard random walk method. The mean of the absolute error in concentration, for both MC and QMC, is reported in Table 2 (here the error of a computation is defined as the absolute difference between the result of a given simulation and the result of the MC simulation using 2^{20} particles).

Table 2 Mean error in concentration on three space intervals: MC vs. QMC.

Space interval	$[40, 60]$	$[100, 120]$	$[180, 200]$
MC	0.002445	0.001798	0.000906
QMC	0.000342	0.000247	0.000178

5.3 Variable Diffusion Coefficient and Non-Instantaneous Emission

We use the same diffusion coefficient (27) as in the previous experiment but here the ions are released over a time interval of length 1.31072 following the arrival of each impulse (one particle is released from $x = 0$ at every time step Δt). As before, no exact solution is available and we take as a reference the result of a MC simulation with 2^{20} particles. Figure 3 shows the outputs of the computations. Using quasi-random numbers in place of pseudo-random numbers and reordering

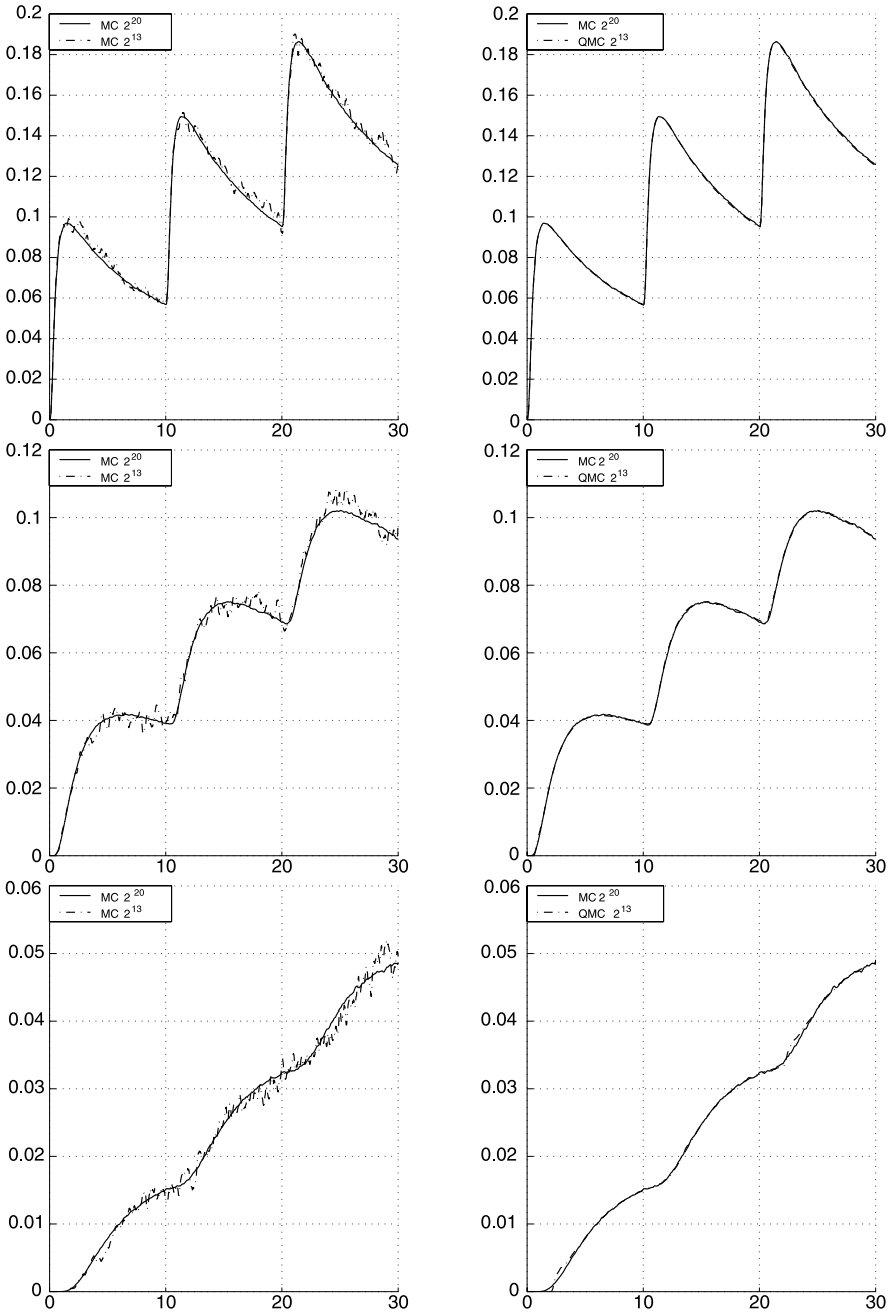


Fig. 2 Variable diffusion coefficient and instantaneous emission. Time evolution of the concentration in the intervals [40, 60] (up), [100, 120] (middle), and [180, 200] (down). Comparison of MC (left) and QMC (right) simulations.

the particles clearly reduce scattering in the results. The mean of the absolute error in concentration is computed and MC and QMC results are compared in Table 3 (the error is defined as before).

Table 3 Mean error in concentration on three space intervals: MC vs. QMC.

Space interval	$[40, 60]$	$[100, 120]$	$[180, 200]$
MC	0.001993	0.001801	0.000876
QMC	0.000516	0.000235	0.000207

6 Conclusion

In this paper, we have proposed a QMC method for the simulation of diffusion equation in a spatially nonhomogeneous medium. The method generalizes the QMC algorithm described in [6] in the case of a constant diffusion coefficient.

Time is discretized and diffusion is simulated by the random walk displacement of a set of particles. The scheme uses a $(0, 2)$-sequence in some base b in place of random numbers and the particles are reordered according to their position at every time step. The usual Gaussian steplength appropriate for a constant diffusion coefficient is modified by using a correction arising from the spatial dependence [1]. If the number of simulation particles varies with time, the set of particles is split into subsets after each time step, in order to make a proper use of the great uniformity of $(0, m, 1)$-nets. The particles of each subset are reordered and move independently of those of the other groups. The results of numerical experiments show a strong improvement over standard random walk.

The calculations presented here have been given in one dimension. Multidimensional versions of the QMC algorithm for a constant diffusion coefficient are covered in [6]. A direction for future research would be to apply the algorithm proposed here for a non-constant diffusion coefficient to higher dimensions. Other directions include the implementation of boundary conditions and the development of a QMC version of an alternative method for simulating diffusion in a spatially nonhomogeneous medium, which is based on a biased random walk on an asymmetrical lattice [2].

Acknowledgements We thank the Editor Art B. Owen and two anonymous reviewers for several suggestions that helped to clarify the paper.

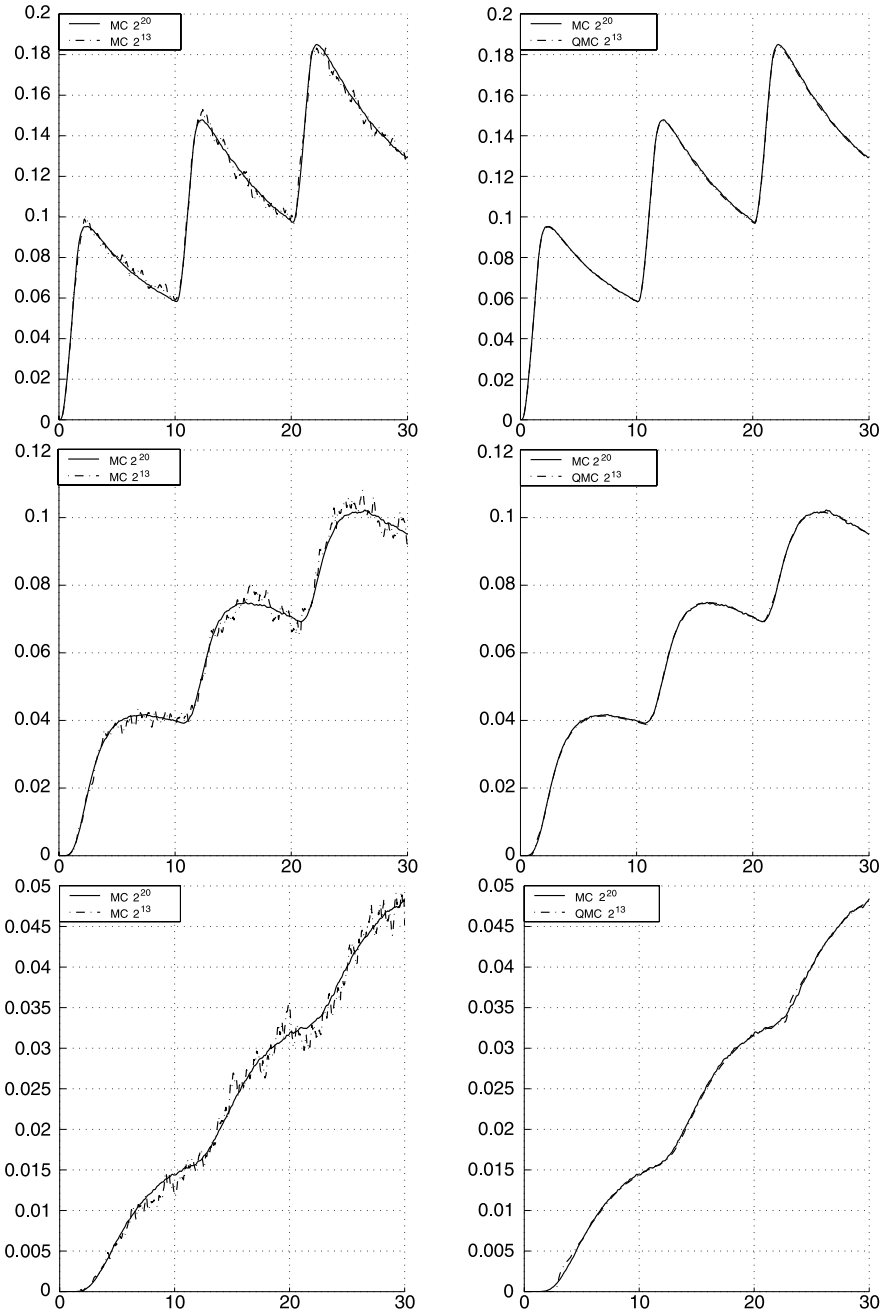


Fig. 3 Variable diffusion coefficient and non-instantaneous emission. Time evolution of the concentration in the intervals $[40, 60]$ (up), $[100, 120]$ (middle), and $[180, 200]$ (down). Comparison of MC (left) and QMC (right) simulations.

References

1. Farnell, L., Gibson, W.G.: Monte Carlo simulation of diffusion in a spatially nonhomogeneous medium: correction to the Gaussian steplength. Journal of Computational Physics **198**, 65–79 (2004)
2. Farnell, L., Gibson, W.G.: Monte Carlo simulation of diffusion in a spatially non-homogeneous medium: A biased random walk on an asymmetrical lattice. Journal of Computational Physics **208**, 253–265 (2005)
3. Ghoniem, A.F., Sherman, F.S.: Grid-free simulation of diffusion using random walk methods. Journal of Computational Physics **61**, 1–37 (1985)
4. Hunter, J.R., Craig, P.D., Phillips, H.E.: On the use of random walk models with spatially variable diffusivity. Journal of Computational Physics **106**, 366–376 (1993)
5. Lécot, C.: A direct simulation Monte Carlo scheme and uniformly distributed sequences for solving the Boltzmann equation. Computing **41**, 41–57 (1989)
6. Lécot, C., El Khettabi, F.: Quasi-Monte Carlo simulation of diffusion. Journal of Complexity **15**, 342–359 (1999)
7. Matveev, V., Sherman, A., Zucker, R.S.: New and corrected simulations of synaptic facilitation. Biophysical Journal **83**, 1368–1373 (2002)
8. Morokoff, W.J., Caflisch, R.E.: A quasi-Monte Carlo approach to particle simulation of the heat equation. SIAM Journal on Numerical Analysis **30**, 1558–1573 (1993)
9. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia (1992)

L_2 Discrepancy of Two-Dimensional Digitally Shifted Hammersley Point Sets in Base b

Henri Faure and Friedrich Pillichshammer

Abstract We give an exact formula for the L_2 discrepancy of two-dimensional digitally shifted Hammersley point sets in base b. This formula shows that for certain bases b and certain shifts the L_2 discrepancy is of best possible order with respect to the general lower bound due to Roth. Hence, for the first time, it is proved that, for a thin, but infinite subsequence of bases b starting with $5, 19, 71, \ldots$, a single permutation only can achieve this best possible order, unlike previous results of White (1975) who needs b permutations and Faure & Pillichshammer (2008) who need 2 permutations.

1 Introduction and Statement of the Results

For a finite point set $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of $N \geq 1$ (not necessarily distinct) points in the unit-square $[0,1)^2$ the L_2 *discrepancy* is defined by

$$L_2(\mathcal{P}) := \left(\int_0^1 \int_0^1 |E(x,y,\mathcal{P})|^2 \, \mathrm{d}x \, \mathrm{d}y \right)^{1/2},$$

where the *discrepancy function* is given as $E(x,y,\mathcal{P}) = A([0,x) \times [0,y), \mathcal{P}) - Nxy$, where $A([0,x) \times [0,y), \mathcal{P})$ denotes the number of indices $1 \leq M \leq N$ for which $\boldsymbol{x}_M \in [0,x) \times [0,y)$. The L_2 discrepancy is a quantitative measure for the irregularity of distribution of $\mathcal{P}$, i.e., the deviation from perfect uniform distribution

Henri Faure
Institut de Mathématiques de Luminy, U.M.R. 6206 CNRS, 163 avenue de Luminy, case 907, 13288 Marseille Cedex 09 and Université Paul Cézanne (Aix-Marseille III), France
e-mail: faure(AT)iml.univ-mrs.fr

Friedrich Pillichshammer
Institut für Finanzmathematik, Universität Linz, Altenbergerstraße 69, A-4040 Linz, Austria
e-mail: friedrich.pillichshammer(AT)jku.at

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_22,

modulo one, which has a close relationship with the worst-case and average-case errors of quasi-Monte Carlo integration of functions from certain function classes. An introduction to the theory of uniform distribution modulo one and the discrepancy of sequences can be found in the books of Kuipers & Niederreiter [11] or of Drmota & Tichy [3]. Concerning the relationship between L_2 discrepancy and quasi-Monte Carlo integration we further refer to [16, 19, 20] for example.

It was first shown by Roth [15] (see also [11, Chapter 2, Section 2]) that there is a constant $c > 0$ with the property that for the L_2 discrepancy of any finite point set $\mathcal{P}$ consisting of N points in $[0,1)^2$ we have

$$L_2(\mathcal{P}) \geq c\sqrt{\log N}. \tag{1}$$

In this paper we will consider the L_2 discrepancy of so-called digitally shifted Hammersley point sets in base b with b^n points. These point sets form a sub-class of generalized Hammersley point sets in base b (the Hammersley point set is also known as Roth net for $b = 2$), which can be considered as finite two-dimensional versions of the generalized van der Corput sequences in base b as introduced by Faure [5].

Throughout the paper let $b \geq 2$ be an integer and let $\mathfrak{S}_b$ be the set of all permutations of $\{0, 1, \ldots, b-1\}$.

Definition 1 (generalized Hammersley point set). Let $b \geq 2$ and $n \geq 0$ be integers and let $\Sigma = (\sigma_0, \ldots, \sigma_{n-1}) \in \mathfrak{S}_b^n$. For an integer $1 \leq N \leq b^n$, write $N - 1 = \sum_{r=0}^{n-1} a_r(N) b^r$ in the b-adic system and define $S_b^{\Sigma}(N) := \sum_{r=0}^{n-1} \frac{\sigma_r(a_r(N))}{b^{r+1}}$. Then the *generalized two-dimensional Hammersley point set in base b* consisting of b^n points associated with Σ is defined by

$$\mathcal{H}_{b,n}^{\Sigma} := \left\{ \left(S_b^{\Sigma}(N), \frac{N-1}{b^n} \right) : 1 \leq N \leq b^n \right\}.$$

In case of $\sigma_i = \sigma$ for all $0 \leq i < n$, we also write $\mathcal{H}_{b,n}^{\sigma}$ instead of $\mathcal{H}_{b,n}^{\Sigma}$. If $\sigma = \mathrm{id}$, the identical permutation, then we obtain the classical two-dimensional Hammersley point set in base b.

Exact formulas for the L_2 discrepancy of the classical two-dimensional Hammersley point set $\mathcal{H}_{b,n}^{\mathrm{id}}$ in base b have been proved by Vilenkin [17], Halton & Zaremba [9] and Pillichshammer [13] in base $b = 2$ and by White [18] and Faure & Pillichshammer [8] for arbitrary bases. These results show that the classical Hammersley point set cannot achieve the best possible order of L_2 discrepancy with respect to Roth's general lower bound (1).

The first who obtained the best possible order of L_2 discrepancy for finite two-dimensional point sets was Davenport [2], with a modification of so-called $(N\alpha)$-sequences (α having a continued fraction expansion with bounded partial quotients), more precisely with the set consisting of the $2M$ points $\left(\{\pm N\alpha\}, \frac{N}{M}\right)$ for $1 \leq N \leq M$ where M is a positive integer and $\{x\}$ denotes the fractional part of x.

Next, observing that $\{-N\alpha\} = 1 - \{N\alpha\}$, Proinov [14] obtained the same result with the same set where generalized van der Corput sequences take the place of

$(N\alpha)$-sequences and he named this process *symmetrization of a sequence*. Later on, the same process was used by Chaix & Faure [1] for infinite van der Corput sequences (improving at the same time the constants of Proinov) and by Larcher & Pillichshammer [12] for $(0, m, 2)$-nets and $(0, 1)$-sequences in base 2. It is important to note that all these results using the symmetrization process give the exact order with bounds only for the implied constants whereas in the following, with various cleverly generalized Hammersley point sets, different authors obtain exact formulas and hence exact values for the implied constants.

Below we first give a survey of results concerning generalized Hammersley point sets with best possible order of L_2 discrepancy together with some comparisons between the methods, showing the interest in considering only one permutation, i.e., a single sequence $\mathcal{H}^{\sigma}_{b,n}$.

First results were available in base $b = 2$: Let id be the identity and $\mathrm{id}_1(k) := k + 1 \pmod 2$ be the *digital shift* in base 2; then Halton & Zaremba [9] and later, in a much more general form, Kritzer & Pillichshammer [10] gave sequences of permutations $\Sigma \in \{\mathrm{id}, \mathrm{id}_1\}^n$ (although they did not use this terminology), for which the generalized Hammersley point set $\mathcal{H}^{\Sigma}_{2,n}$ in base 2 achieves the best possible order of L_2 discrepancy in the sense of Roth (1). For more detailed results we refer to [10].

Results for arbitrary bases were first given by White [18] who generalized the result from [9] in a certain way. He considered sequences Σ of the form

$$\Sigma = (\mathrm{id}_0, \mathrm{id}_1, \ldots, \mathrm{id}_{b-1}, \mathrm{id}_0, \mathrm{id}_1, \ldots, \mathrm{id}_{b-1}, \ldots) \tag{2}$$

of length n where $\mathrm{id}_l(k) := k + l \pmod b$ for $0 \le l, k < b$ (White did not use this terminology). The permutations id_l are called *digital shifts* in base b; they are natural generalizations of the digital shift in base 2 used by Halton & Zaremba and Kritzer & Pillichshammer. For this specific Σ, White gave an exact formula for the L_2 discrepancy of the corresponding generalized Hammersley point set. Essentially this formula states that

$$\left(L_2(\mathcal{H}^{\Sigma}_{b,n})\right)^2 = n\frac{(b^2-1)(3b^2+13)}{720b^2} + O(1) \tag{3}$$

whenever Σ is of the form (2).

Setting $b = 2$ in this formula gives the same sequence as in [9] and the simplest sequence in [10], that is $\Sigma = (\mathrm{id}_0, \mathrm{id}_1, \mathrm{id}_0, \mathrm{id}_1, \ldots)$, with the same constant 5/192. Note that we need only two permutations and therefore the formula for base 2 starts being valid for integers $n \ge 2$, that is, sets of $2^2 = 4$ points at least, which is very few.

The problem for arbitrary b is that we need $n \ge b$, i.e., sets of b^b points at least. Even for small bases like $b = 10$ the property requires sets consisting of more than 10^{10} points which is far away from usual numbers of points allowed in quasi-Monte Carlo simulation. If we want to use generalized Hammersley point sets in applications (image-processing, optimization of printers for instance), we must find a better way than White (in fact White used a trick due to Halton & Warnock, see [18, p. 221]) to improve the L_2 discrepancy of the original Hammersley point sets.

Another approach consists of using the so-called *swapping permutation* τ defined by $\tau(k) = b-k-1$, for $0 \le k < b$, instead of shifts (the term *swapping* is introduced and justified in [6] and [7, Section 2]). Applied to the L_2 discrepancy of Hammersley point sets, this generalization gives formula (3) with the simplest sequence $\Sigma = (\mathrm{id}_0, \tau, \mathrm{id}_0, \tau, \ldots)$ in arbitrary bases. We refer to [8] for detailed proofs together with extensions to the L_p discrepancy. Once again, we need only two permutations but our results are valid for arbitrary bases whereas Halton & Zaremba and Kritzer & Pillichshammer deal only with base 2. We also remark that in base 2, shift and swap is the same permutation, so that [8] fully generalizes the results of [9] (for L_2 discrepancy) and [10] from base 2 to base b.

Now, after White who needs b permutations and Faure & Pillichshammer who need two, the question arises if only one permutation is enough to get the same property, i.e., the best order of L_2 discrepancy.

In this paper, we consider this question for shifts in base b and we deal with sequences of permutations of the form $\Sigma_l := (\mathrm{id}_l, \ldots, \mathrm{id}_l)$ for arbitrary fixed integer $0 \le l < b$, i.e., with our notation after Definition 1, we study generalized Hammersley point sets $\mathcal{H}_{b,n}^{\mathrm{id}_l}$. We call such sets *digitally shifted Hammersley point sets in base b*. We can prove an exact formula for the L_2 discrepancy of these sets which permits to answer the question above for the sub-class of digitally shifted Hammersley point sets. The proof relies on the approach of [8] and uses the fundamental Lemmas 1 and 2 from this paper. However here, for the first time, we have to manage with true permutations while in [8] we dealt with identity only (τ being simply a mirror of it); on the other hand, we obtained more results in this specific case.

Section 2 contains prerequisites and auxiliary results, and Section 3 contains the proof of the following result:

Theorem 1. *For the L_2 discrepancy of a digitally shifted Hammersley point set $\mathcal{H}_{b,n}^{\mathrm{id}_l}$, with integers $b \ge 2$, $0 \le l < b$ and $n \ge 1$, we have*

$$\begin{aligned}
&\left(L_2\left(\mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\right)^2 \\
&= \left(\frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right)\right)^2 - \frac{1}{2b^n}\frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right) \\
&\quad + \frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2} + \frac{(b^2-1)(3b^2+13)}{720b}\right) + \frac{3}{8} + \frac{1}{4b^n} - \frac{1}{72b^{2n}}.
\end{aligned}$$

If we choose $l = 0$ then $\mathcal{H}_{b,n}^{\mathrm{id}_0}$ is the classical Hammersley point set and our formula recovers [8, Theorem 1] and [18, Eq. (15)].

From Theorem 1 one can see that for certain values of b and l one can obtain the optimal order of L_2 discrepancy in the sense of Roth (1) with a single shift. In this case the implied leading constant is the same as in White's and Faure & Pillichshammer's result (3).

Corollary 1. *For integers $b \ge 2$, $0 \le l < b$ and $n \ge 1$ we have*

$$\left(L_2\left(\mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\right)^2 = n\frac{(b^2-1)(3b^2+13)}{720b^2} + \frac{3}{8} + \frac{1}{4b^n} - \frac{1}{72b^{2n}} \tag{4}$$

if and only if b satisfies the Pell-Fermat equation $b^2 - 3c^2 = -2$ with a suitable integer c and $l = \frac{1}{2}(b \pm c)$.

All solutions of this equation are given by $b + c\sqrt{3} = \pm(1+\sqrt{3})(2+\sqrt{3})^m$ with $m \in \mathbb{N}_0$.

Proof. Of course Eq. (4) holds if and only if $\frac{b^2-1}{12} = \frac{l(b-l)}{2}$ and this is equivalent to $l = \frac{1}{2}\left(b \pm \sqrt{\frac{b^2+2}{3}}\right)$. Since l is an integer this is equivalent to $\frac{b^2+2}{3} = c^2$ for some integer c or equivalently $b^2 - 3c^2 = -2$. Note that all solutions (b, c) have to consist of odd b and c only. This is in accordance with the fact that $l = \frac{1}{2}(b \pm c)$ is an integer.

For $z = x + y\sqrt{d}$ and its conjugate $\overline{z} = x - y\sqrt{d}$ we write $N(z) = z \cdot \overline{z} = x^2 - y^2 d$. It is known (see, for example [4]) that the general solution z (if it exists) of a Pell-Fermat equation $N(z) = a$ can be obtained as the product of the solution of the special Pell-Fermat equation $N(z) = 1$, which is given by $z = \pm(z_0)^m$, $m \in \mathbb{N}$, where $z_0 > 1$ is the minimal solution, with a special solution of $N(z) = a$ with $0 \le z \le z_0$.

In our case we have the minimal solution $z_0 = 2 + \sqrt{3}$ and the special solution $1 + \sqrt{3}$. Hence, all solutions are given by $z = \pm(1+\sqrt{3})(2+\sqrt{3})^m$, $m \in \mathbb{N}_0$. □

The first few of the infinitely many pairs (b, l) for which Eq. (4) holds are $(5,1)$, $(5,4)$, $(19,4)$, $(19,15)$, $(71,15)$, $(71,56)$, $(265,56)$, $(265,209)$, $(989,209)$, $(989,780)$, $(3691,780)$, $(3691,2911)$, ...

Hence, we have proved that for a thin (but infinite) subsequence of bases b a single shift only is sufficient to obtain the optimal order of L_2 discrepancy. Between the necessity of b shifts with White's method and the few bases we have found with a single shift, there are surely many other possibilities. Finding such alternatives will need more investigations and we plan to pursue this work in the near future.

2 Auxiliary Results

In this section we provide the main tools for the proof of Theorem 1. The analysis of the L_2 discrepancy is based on special functions which have been first introduced by Faure in [5] and which are defined as follows.

For $\sigma \in \mathfrak{S}_b$ let $\mathcal{Z}_b^\sigma = (\sigma(0)/b, \sigma(1)/b, \ldots, \sigma(b-1)/b)$. For $h \in \{0, 1, \ldots, b-1\}$ and $x \in [(k-1)/b, k/b)$, where $k \in \{1, \ldots, b\}$, we define

$$\varphi_{b,h}^\sigma(x) = \begin{cases} A([0,h/b); k; \mathcal{Z}_b^\sigma) - hx & \text{if } 0 \le h \le \sigma(k-1), \\ (b-h)x - A([h/b,1); k; \mathcal{Z}_b^\sigma) & \text{if } \sigma(k-1) < h < b, \end{cases}$$

where here for a sequence $X = (x_M)_{M \ge 1}$ we denote by $A(I; k; X)$ the number of indices $1 \le M \le k$ such that $x_M \in I$. Further, the function $\varphi_{b,h}^\sigma$ is extended to the

reals by periodicity. Note that $\varphi_{b,0}^{\sigma} = 0$ and $\varphi_{b,h}^{\sigma}(0) = 0$ for any $\sigma \in \mathfrak{S}_b$ and any $0 \le h < b$.

Furthermore, we define $\varphi_b^{\sigma} := \sum_{h=0}^{b-1} \varphi_{b,h}^{\sigma}$ and $\phi_b^{\sigma} := \sum_{h=0}^{b-1} (\varphi_{b,h}^{\sigma})^2$. Note that φ_b^{σ} is continuous, piecewise linear on the intervals $[k/b, (k+1)/b]$ and $\varphi_b^{\sigma}(0) = \varphi_b^{\sigma}(1)$. For example for $\sigma = \mathrm{id}$ we have

$$\varphi_{b,h}^{\mathrm{id}}(x) = \begin{cases} (b-h)x & \text{if } x \in [0, h/b], \\ h(1-x) & \text{if } x \in [h/b, 1], \end{cases} \tag{5}$$

from which one obtains (see [8, Lemma 3] for details) that for $x \in \left[\frac{k}{b}, \frac{k+1}{b}\right]$, $0 \le k < b$, we have

$$\varphi_b^{\mathrm{id}}(x) = \frac{b(b-2k-1)}{2}\left(x - \frac{k}{b}\right) + \frac{k(b-k)}{2} \tag{6}$$

and

$$\phi_b^{\mathrm{id}}(x) = (1-x)^2 \frac{k(k+1)(2k+1)}{6} + x^2 \frac{(b-k)(b-k-1)(2b-2k-1)}{6}. \tag{7}$$

From (6) we immediately obtain for $y \in \left[0, \frac{1}{b}\right)$ the equation

$$\sum_{k=0}^{b-1} \varphi_b^{\mathrm{id}}\left(\frac{k}{b} + y\right) = \frac{b(b^2-1)}{12}. \tag{8}$$

Sometimes we will use the following property from [1, Propriété 3.4] stating that

$$(\varphi_{b,h}^{\sigma})'(k/b + 0) = (\varphi_{b,h}^{\mathrm{id}})'(\sigma(k)/b + 0). \tag{9}$$

Here and later on by $f'(x+0)$ we mean the right-derivative of the function f at x.

The following lemma gives a relationship between the family of $\varphi_{b,h}^{\sigma}$ functions with respect to the permutations id and id_l.

Lemma 1. *For any $0 \le h, l < b$ and $x \in [0,1]$ we have*

$$\varphi_{b,h}^{\mathrm{id}_l}(x) = \varphi_{b,h}^{\mathrm{id}}\left(x + \frac{l}{b}\right) - \varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right) \tag{10}$$

and in particular,

$$\varphi_b^{\mathrm{id}_l}(x) = \varphi_b^{\mathrm{id}}\left(x + \frac{l}{b}\right) - \varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right).$$

Proof. It is enough to show that the equality holds for $x = k/b$, $k \in \{0, \ldots, b-1\}$. Since the functions $\varphi_{b,h}^{\sigma}$ are continuous and linear on $\left[\frac{j}{b}, \frac{j+1}{b}\right)$, $0 \le j < b$, invoking Eq. (9) we have

$$\varphi_{b,h}^{\mathrm{id}_l}\left(\frac{k}{b}\right) = \frac{1}{b}\sum_{j=0}^{k-1}\left(\varphi_{b,h}^{\mathrm{id}_l}\right)'\left(\frac{j}{b} + 0\right) = \frac{1}{b}\sum_{j=0}^{k-1}\left(\varphi_{b,h}^{\mathrm{id}}\right)'\left(\frac{\mathrm{id}_l(j)}{b} + 0\right)$$

$$= \frac{1}{b} \sum_{j=l}^{k+l-1} \left(\varphi_{b,h}^{\mathrm{id}}\right)' \left(\frac{j}{b} + 0\right) = \varphi_{b,h}^{\mathrm{id}}\left(\frac{k+l}{b}\right) - \varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right)$$

as desired. □

The following lemma provides a formula for the discrepancy function of generalized Hammersley point sets.

Lemma 2. *For integers* $1 \le \lambda, N \le b^n$ *and* $\Sigma = (\sigma_0, \ldots, \sigma_{n-1}) \in \mathfrak{S}_b^n$ *we have*

$$E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\Sigma}\right) = \sum_{j=1}^{n} \varphi_{b,\varepsilon_j}^{\sigma_{j-1}}\left(\frac{N}{b^j}\right),$$

where the $\varepsilon_j = \varepsilon_j(\lambda, n, N)$ *can be given explicitly.*

A proof of this result together with formulas for $\varepsilon_j = \varepsilon_j(\lambda, n, N)$ can be found in [8, Lemma 1].

Remark 1. Let $0 \le x, y \le 1$ be arbitrary. Since all points from $\mathcal{H}_{b,n}^{\Sigma}$ have coordinates of the form α/b^n for some $\alpha \in \{0, 1, \ldots, b^n - 1\}$, we have

$$E(x, y, \mathcal{H}_{b,n}^{\Sigma}) = E(x(n), y(n), \mathcal{H}_{b,n}^{\Sigma}) + b^n(x(n)y(n) - xy), \tag{11}$$

where for $0 \le x \le 1$ we define $x(n) := \min\{\alpha/b^n \ge x : \alpha \in \{0, \ldots, b^n\}\}$.

Now we will give a series of lemmas with further, more involved properties of the functions $\varphi_{b,h}^{\sigma}$, φ_b^{σ} and ϕ_b^{σ}. The first result is a special case of [8, Lemma 2] (see there for a proof).

Lemma 3. *For* $1 \le N \le b^n$ *and* $0 \le j_1 < j_2 < \cdots < j_k < n$ *we have*

$$\sum_{\lambda=1}^{b^n} \prod_{i=1}^{k} \varphi_{b,\varepsilon_{j_i}}^{\sigma_{j_i}}\left(\frac{N}{b^{j_i}}\right) = b^{n-k} \prod_{i=1}^{k} \varphi_b^{\sigma_{j_i}}\left(\frac{N}{b^{j_i}}\right)$$

and

$$\sum_{\lambda=1}^{b^n} \prod_{i=1}^{k} \left(\varphi_{b,\varepsilon_{j_i}}^{\sigma_{j_i}}\left(\frac{N}{b^{j_i}}\right)\right)^2 = b^{n-k} \prod_{i=1}^{k} \phi_b^{\sigma_{j_i}}\left(\frac{N}{b^{j_i}}\right).$$

Lemma 4. *For* $0 \le h < k < n$ *and* $0 \le l < b$ *we have*

$$\sum_{N=1}^{b^n} \varphi_b^{\mathrm{id}_l}\left(\frac{N}{b^h}\right) \varphi_b^{\mathrm{id}_l}\left(\frac{N}{b^k}\right) = b^n \left(\frac{b^2-1}{12} - \varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\right)^2.$$

Proof. Using Lemma 1 we have

$$\sum_{N=1}^{b^n} \varphi_b^{\mathrm{id}_l}\left(\frac{N}{b^h}\right) \varphi_b^{\mathrm{id}_l}\left(\frac{N}{b^k}\right)$$

$$= \sum_{N=1}^{b^n} \varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right)\varphi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right) + b^n\left(\varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\right)^2$$
$$-\varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\sum_{N=1}^{b^n}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right) - \varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\sum_{N=1}^{b^n}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right). \tag{12}$$

Let $N = N_0 + N_1 b + \cdots + N_{n-1}b^{n-1}$ be the b-adic expansion of $N \in \{0,\ldots,b^n-1\}$. From the periodicity of φ_b^{id} and using Eq. (8) we obtain

$$\begin{aligned}\sum_{N=1}^{b^n}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right) &= \sum_{N_0,\ldots,N_{n-1}=0}^{b-1}\varphi_b^{\mathrm{id}}\left(\frac{N_0+\cdots+N_{n-1}b^{n-1}}{b^h}+\frac{l}{b}\right)\\ &= b^{n-h}\sum_{N_0,\ldots,N_{h-1}=0}^{b-1}\varphi_b^{\mathrm{id}}\left(\frac{N_0+\cdots+N_{h-1}b^{h-1}}{b^h}+\frac{l}{b}\right)\\ &= b^{n-h}\sum_{N=0}^{b^{h-1}-1}\sum_{N_{h-1}=0}^{b-1}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{N_{h-1}+l}{b}\right)\\ &= b^{n-h}\sum_{N=0}^{b^{h-1}-1}\sum_{z=0}^{b-1}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{z}{b}\right) = b^n\frac{b^2-1}{12}.\end{aligned} \tag{13}$$

Similar reasoning as above and noting that $h < k$ gives

$$\begin{aligned}&\sum_{N=1}^{b^n}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right)\varphi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)\\ &= b^{n-k}\sum_{N=0}^{b^k-1}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right)\varphi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)\\ &= b^{n-k}\sum_{N=0}^{b^{k-1}-1}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right)\sum_{z=0}^{b-1}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{z}{b}\right)\\ &= \frac{b^2-1}{12}\sum_{N=0}^{b^n-1}\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right) = b^n\left(\frac{b^2-1}{12}\right)^2.\end{aligned} \tag{14}$$

Now the result follows from inserting (13) and (14) into (12). □

Lemma 5. *For $0 \le k < n$ and $0 \le l < b$ we have*

$$\sum_{N=1}^{b^n}\phi_b^{\mathrm{id}_l}\left(\frac{N}{b^k}\right) = b^n\left(\frac{b^4-1}{90b}+\frac{b(b^2-1)}{36b^{2k}}\right)+b^n\phi_b^{\mathrm{id}}\left(\frac{l}{b}\right)$$
$$-\frac{b^{n-1}}{12}l(b-l)(1+b^2+lb-l^2).$$

Proof. We have

$$\phi_b^{\mathrm{id}_l}\left(\frac{N}{b^k}\right)=\sum_{h=0}^{b-1}\left(\varphi_{b,h}^{\mathrm{id}_l}\left(\frac{N}{b^k}\right)\right)^2=\sum_{h=0}^{b-1}\left(\varphi_{b,h}^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)-\varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right)\right)^2$$
$$=\phi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)+\phi_b^{\mathrm{id}}\left(\frac{l}{b}\right)-2\sum_{h=0}^{b-1}\varphi_{b,h}^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)\varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right).$$

By using the periodicity of ϕ_b^{id} we obtain

$$\sum_{N=1}^{b^n}\phi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)=b^{n-k}\sum_{N=1}^{b^k}\phi_b^{\mathrm{id}}\left(\frac{N}{b^k}\right)=b^{n-k}\sum_{j=0}^{b-1}\sum_{N=jb^{k-1}+1}^{(j+1)b^{k-1}}\phi_b^{\mathrm{id}}\left(\frac{N}{b^k}\right).$$

For $jb^{k-1}+1\le N\le(j+1)b^{k-1}$ we have $j/b<N/b^k\le(j+1)/b$ and hence we can use Eq. (7) to obtain

$$\sum_{N=1}^{b^n}\phi_b^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)=b^{n-k}\sum_{j=0}^{b-1}\sum_{N=jb^{k-1}+1}^{(j+1)b^{k-1}}\left[\left(1-\frac{N}{b^k}\right)^2\frac{j(j+1)(2j+1)}{6}\right.$$
$$\left.+\left(\frac{N}{b^k}\right)^2\frac{(b-j)(b-j-1)(2b-2j-1)}{6}\right]$$
$$=b^n\left(\frac{b^4-1}{90b}+\frac{b(b^2-1)}{36b^{2k}}\right).$$

Furthermore we have

$$\sum_{N=1}^{b^n}\sum_{h=0}^{b-1}\varphi_{b,h}^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)\varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right)=\sum_{h=0}^{b-1}\varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right)\sum_{N=1}^{b^n}\varphi_{b,h}^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right).$$

Using the periodicity of $\varphi_{b,h}^{\mathrm{id}}$ and Eq. (5) for the innermost sum we obtain

$$\sum_{N=1}^{b^n}\varphi_{b,h}^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)=b^{n-k}\sum_{N=0}^{b^k-1}\varphi_{b,h}^{\mathrm{id}}\left(\frac{N}{b^k}\right)$$
$$=b^{n-k}\left(\sum_{N=0}^{hb^{k-1}}(b-h)\frac{N}{b^k}+\sum_{N=hb^{k-1}+1}^{b^k-1}h\left(1-\frac{N}{b^k}\right)\right)$$
$$=b^{n-1}\frac{(b-h)h}{2}.$$

Hence, using again Eq. (5),

$$\sum_{N=1}^{b^n}\sum_{h=0}^{b-1}\varphi_{b,h}^{\mathrm{id}}\left(\frac{N}{b^k}+\frac{l}{b}\right)\varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right)=\frac{b^{n-1}}{2}\sum_{h=0}^{b-1}\varphi_{b,h}^{\mathrm{id}}\left(\frac{l}{b}\right)(b-h)h$$
$$=\frac{b^{n-1}}{2}\left(\sum_{h=0}^{l-1}(b-h)h^2\left(1-\frac{l}{b}\right)+\sum_{h=l}^{b-1}(b-h)^2h\frac{l}{b}\right)$$

$$= \frac{b^{n-1}}{24} l(b-l)(1+b^2+lb-l^2).$$

The result follows. □

Lemma 6. *For $0 \le h < n$ and $0 \le l < b$ we have*

$$\sum_{N=1}^{b^n} N\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right) = b^{2n}\frac{b^2-1}{24} + \frac{b^n l(b-l)}{12b}(3b-b^h(b-2l)).$$

Proof. Splitting up the range of summation we have

$$\sum_{N=1}^{b^n} N\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right) = \sum_{k=0}^{b^{n-h+1}-1} \sum_{N=kb^{h-1}+1}^{(k+1)b^{h-1}} N\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right).$$

For $0 \le k < b^{n-h+1}$ let $k = qb+r$ with integers $0 \le r < b$ and $0 \le q < b^{n-h}$. Then for $kb^{h-1}+1 \le N \le (k+1)b^{h-1}$ we have $r/b \le N/b^h - q \le (r+1)/b$. Hence, if $0 \le r < b-l$, then $0 \le (r+l)/b \le N/b^h - q + l/b \le (r+l+1)/b \le 1$ and if $b-l \le r < b$, then $0 \le (r+l-b)/b \le N/b^h - q + l/b - 1 \le (r+l-b+1)/b < 1$. Using the periodicity of φ_b^{id} and Eq. (6) we therefore obtain

$$\begin{aligned}
\sum_{N=1}^{b^n} N\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}+\frac{l}{b}\right) &= \sum_{r=0}^{b-1}\sum_{q=0}^{b^{n-h}-1}\sum_{N=qb^h+rb^{h-1}+1}^{qb^h+(r+1)b^{h-1}} N\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}-q+\frac{l}{b}\right)\\
&= \sum_{r=0}^{b-l-1}\sum_{q=0}^{b^{n-h}-1}\sum_{N=qb^h+rb^{h-1}+1}^{qb^h+(r+1)b^{h-1}} N\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}-q+\frac{l}{b}\right)\\
&\quad + \sum_{r=b-l}^{b-1}\sum_{q=0}^{b^{n-h}-1}\sum_{N=qb^h+rb^{h-1}+1}^{qb^h+(r+1)b^{h-1}} N\varphi_b^{\mathrm{id}}\left(\frac{N}{b^h}-q+\frac{l}{b}-1\right)\\
&= \sum_{r=0}^{b-l-1}\sum_{q=0}^{b^{n-h}-1}\sum_{N=qb^h+rb^{h-1}+1}^{qb^h+(r+1)b^{h-1}} N\left(\frac{b(b-2(r+l)-1)}{2}\left(\frac{N}{b^h}-q-\frac{r}{b}\right)\right.\\
&\qquad\qquad \left. + \frac{(r+l)(b-r-l)}{2}\right)\\
&\quad + \sum_{r=b-l}^{b-1}\sum_{q=0}^{b^{n-h}-1}\sum_{N=qb^h+rb^{h-1}+1}^{qb^h+(r+1)b^{h-1}} N\left(\frac{b(b-2(r+l-b)-1)}{2}\left(\frac{N}{b^h}-q-\frac{r}{b}\right)\right.\\
&\qquad\qquad \left. + \frac{(r+l-b)(2b-r-l)}{2}\right)\\
&= b^{2n}\frac{b^2-1}{24} + \frac{b^n l(b-l)}{12b}(3b-b^h(b-2l)).
\end{aligned}$$

This is the desired result. □

3 The Proof of Theorem 1

First we show a discrete version of Theorem 1. The following result is a generalization of [8, Lemma 6]. The original is obtained when putting $l = 0$ below.

Lemma 7. *For $0 \le l < b$ we have*

$$\frac{1}{b^{2n}} \sum_{\lambda,N=1}^{b^n} E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right) = \frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right) \tag{15}$$

and

$$\begin{aligned} &\frac{1}{b^{2n}} \sum_{\lambda,N=1}^{b^n} \left(E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\right)^2 \\ &= \left(\frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right)\right)^2 + n\frac{(b^2-1)(3b^2+13)}{720b^2} + \frac{1}{36}\left(1 - \frac{1}{b^{2n}}\right). \end{aligned} \tag{16}$$

Proof. We just give the (much more involved) proof of Eq. (16). Using Lemmas 2, 3, 4 and 5 we have

$$\begin{aligned} &\frac{1}{b^{2n}} \sum_{\lambda,N=1}^{b^n} \left(E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\right)^2 = \frac{1}{b^{2n}} \sum_{\lambda,N=1}^{b^n} \sum_{i,j=1}^{n} \varphi_{b,\varepsilon_i}^{\mathrm{id}_l}\left(\frac{N}{b^i}\right) \varphi_{b,\varepsilon_j}^{\mathrm{id}_l}\left(\frac{N}{b^j}\right) \\ &= \frac{1}{b^{2n}} \sum_{i=1}^{n} \sum_{N=1}^{b^n} \sum_{\lambda=1}^{b^n} \left(\varphi_{b,\varepsilon_i}^{\mathrm{id}_l}\left(\frac{N}{b^i}\right)\right)^2 + \frac{1}{b^{2n}} \sum_{\substack{i,j=1 \\ i\neq j}}^{n} \sum_{N=1}^{b^n} \sum_{\lambda=1}^{b^n} \varphi_{b,\varepsilon_i}^{\mathrm{id}_l}\left(\frac{N}{b^i}\right) \varphi_{b,\varepsilon_j}^{\mathrm{id}_l}\left(\frac{N}{b^j}\right) \\ &= \frac{1}{b^{2n}} \sum_{i=1}^{n} \sum_{N=1}^{b^n} b^{n-1} \phi_b^{\mathrm{id}_l}\left(\frac{N}{b^i}\right) + \frac{1}{b^{2n}} \sum_{\substack{i,j=1 \\ i\neq j}}^{n} \sum_{N=1}^{b^n} b^{n-2} \varphi_b^{\mathrm{id}_l}\left(\frac{N}{b^i}\right) \varphi_b^{\mathrm{id}_l}\left(\frac{N}{b^j}\right) \\ &= \frac{1}{b} \sum_{i=1}^{n} \left(\left(\frac{b^4-1}{90b} + \frac{b(b^2-1)}{36b^{2i}}\right) + \phi_b^{\mathrm{id}}\left(\frac{l}{b}\right) - \frac{l(b-l)(1+b^2+lb-l^2)}{12b}\right) \\ &\quad + \frac{n^2-n}{b^2}\left(\frac{b^2-1}{12} - \varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\right)^2 \\ &= \left(\frac{n}{b}\left(\frac{b^2-1}{12} - \varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\right)\right)^2 - \frac{n}{b^2}\left(\frac{b^2-1}{12} - \varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\right)^2 \\ &\quad + n\frac{b^4-1}{90b^2} + \frac{1}{36}\left(1 - \frac{1}{b^{2n}}\right) + \frac{n}{b}\left(\phi_b^{\mathrm{id}}\left(\frac{l}{b}\right) - \frac{l(b-l)(1+b^2+lb-l^2)}{12b}\right) \\ &= \left(\frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right)\right)^2 + n\frac{(b^2-1)(3b^2+13)}{720b^2} + \frac{1}{36}\left(1 - \frac{1}{b^{2n}}\right), \end{aligned}$$

where for the last equality we used that $\varphi_b^{\mathrm{id}}(l/b) = l(b-l)/2$ according to Eq. (6) and $\phi_b^{\mathrm{id}}(l/b) = (1-l/b)^2 l(l+1)(2l+1)/6 + (b-l)(b-l-1)(2b-2l-1)l^2/(6b^2)$ according to Eq. (7). □

Now we give the proof of Theorem 1.

Proof. Using Eq. (11) we obtain

$$\begin{aligned}
\left(L_2\left(\mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\right)^2 &= \int_0^1\int_0^1 \left(E\left(x(n), y(n), \mathcal{H}_{b,n}^{\mathrm{id}_l}\right) + b^n\left(x(n)y(n) - xy\right)\right)^2 \mathrm{d}x\,\mathrm{d}y \\
&= \frac{1}{b^{2n}} \sum_{\lambda,N=1}^{b^n} \left(E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\right)^2 \\
&\quad + 2b^n \sum_{\lambda,N=1}^{b^n} \int_{\frac{\lambda-1}{b^n}}^{\frac{\lambda}{b^n}} \int_{\frac{N-1}{b^n}}^{\frac{N}{b^n}} E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\left(\frac{\lambda}{b^n}\frac{N}{b^n} - xy\right) \mathrm{d}x\,\mathrm{d}y \\
&\quad + b^{2n} \sum_{\lambda,N=1}^{b^n} \int_{\frac{\lambda-1}{b^n}}^{\frac{\lambda}{b^n}} \int_{\frac{N-1}{b^n}}^{\frac{N}{b^n}} \left(\frac{\lambda}{b^n}\frac{N}{b^n} - xy\right)^2 \mathrm{d}x\,\mathrm{d}y \\
&=: S_1 + S_2 + S_3.
\end{aligned}$$

The term S_1 has been evaluated in Lemma 7 and straightforward algebra shows that $S_3 = (1 + 18b^n + 25b^{2n})/(72b^{2n})$. So it remains to deal with S_2.

Evaluating the integral appearing in S_2 we obtain

$$\begin{aligned}
S_2 &= \frac{1}{b^{3n}} \sum_{\lambda,N=1}^{b^n} (\lambda + N) E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right) - \frac{1}{2b^{3n}} \sum_{\lambda,N=1}^{b^n} E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right) \\
&=: S_4 - S_5.
\end{aligned}$$

The term S_5 can be obtained from Lemma 7, Eq. (15). For S_4 we have

$$\begin{aligned}
S_4 &= \frac{1}{b^{3n}} \sum_{\lambda,N=1}^{b^n} \lambda E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right) + \frac{1}{b^{3n}} \sum_{\lambda,N=1}^{b^n} N E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right) \\
&=: \frac{1}{b^{3n}}(S_{4,1} + S_{4,2}).
\end{aligned}$$

With Lemma 2, Lemma 3, Lemma 1 and Lemma 6 we obtain

$$\begin{aligned}
S_{4,2} &= b^{n-1} \sum_{i=1}^{n} \sum_{N=1}^{b^n} N\left(\varphi_b^{\mathrm{id}}\left(\frac{N}{b^i} + \frac{l}{b}\right) - \varphi_b^{\mathrm{id}}\left(\frac{l}{b}\right)\right) \\
&= b^{2n-1} \sum_{i=1}^{n} \left(b^n \frac{b^2-1}{24} + l(b-l)\left(\frac{3b - b^{i+1} + 2lb^i}{12b} - \frac{b^n+1}{4}\right)\right) \\
&= b^{3n} \frac{b^2-1}{24b} n - \frac{b^{2n}}{12b^2} \sum_{i=1}^{n} (b-l)l\left(b^i(b-2l) + 3b^{n+1}\right)
\end{aligned}$$

$$= b^{3n}\frac{b^2-1}{24b}n - \frac{b^{2n}}{12b}(b-l)l\left((b-2l)\frac{b^n-1}{b-1}+3b^n n\right).$$

We turn to $S_{4,1}$. We have

$$\begin{aligned}\mathcal{H}_{b,n}^{\mathrm{id}_l} &= \left\{\left(\frac{\mathrm{id}_l(a_0)}{b}+\cdots+\frac{\mathrm{id}_l(a_{n-1})}{b^n}, \frac{a_{n-1}}{b}+\cdots+\frac{a_0}{b^n}\right) : 0 \le a_i < b\right\}\\ &= \left\{\left(\frac{x_0}{b}+\cdots+\frac{x_{n-1}}{b^{n-1}}, \frac{\mathrm{id}_l^{-1}(x_{n-1})}{b}+\cdots+\frac{\mathrm{id}_l^{-1}(x_0)}{b^n}\right) : 0 \le x_i < b\right\}.\end{aligned}$$

Let $g : [0,1]^2 \to [0,1]^2$ be defined by $g(x,y) = (y,x)$. For $l = 0$ we have $\mathrm{id}_0^{-1} = \mathrm{id}_0$ and for $0 < l < b$ we have $\mathrm{id}_l^{-1} = \mathrm{id}_{b-l}$. Hence we have $\mathcal{H}_{b,n}^{\mathrm{id}_l} = g\left(\mathcal{H}_{b,n}^{\mathrm{id}_{b-l}}\right)$ for $0 < l < b$ and $\mathcal{H}_{b,n}^{\mathrm{id}_0} = g\left(\mathcal{H}_{b,n}^{\mathrm{id}_0}\right)$. Therefore, for $0 < l < b$ we obtain

$$\begin{aligned}S_{4,1} &= \sum_{\lambda,N=1}^{b^n} \lambda E\left(\frac{\lambda}{b^n}, \frac{N}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_l}\right) = \sum_{\lambda,N=1}^{b^n} \lambda E\left(\frac{N}{b^n}, \frac{\lambda}{b^n}, \mathcal{H}_{b,n}^{\mathrm{id}_{b-l}}\right)\\ &= b^{3n}\frac{b^2-1}{24b}n - \frac{b^{2n}}{12b}(b-l)l\left((2l-b)\frac{b^n-1}{b-1}+3b^n n\right)\end{aligned}$$

where we used the formula for $S_{4,2}$ in the last equation. The same formula holds true for $l = 0$.

Hence we have

$$S_4 = \frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right).$$

Now we obtain

$$\begin{aligned}\left(L_2\left(\mathcal{H}_{b,n}^{\mathrm{id}_l}\right)\right)^2 &= \left(\frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right)\right)^2 + n\frac{(b^2-1)(3b^2+13)}{720b^2}\\ &\quad + \frac{1}{36}\left(1-\frac{1}{b^{2n}}\right) + \frac{n}{b}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right)\\ &\quad - \frac{n}{2b^{n+1}}\left(\frac{b^2-1}{12} - \frac{l(b-l)}{2}\right) + \frac{1+18b^n+25b^{2n}}{72b^{2n}}\end{aligned}$$

which yields the desired result. □

Acknowledgements F.P. is supported by the Austrian Science Foundation (FWF), Project S9609, that is part of the Austrian National Research Network "Analytic Combinatorics and Probabilistic Number Theory".

References

1. H. Chaix and H. Faure: Discrépance et diaphonie en dimension un. Acta Arith. **63**: 103–141, 1993.
2. H. Davenport: Note on irregularities of distribution. Mathematika **3**: 131–135, 1956.
3. M. Drmota and R. F. Tichy: *Sequences, Discrepancies and Applications*. In: Lecture Notes in Mathematics, vol. 1651. Springer, Berlin, 1997.
4. D. Djukić: Pell's Equation. http://www.imomath.com/index.php?options=mbb|tekstkut&p=0
5. H. Faure: Discrépance de suites associées à un système de numération (en dimension un). Bull. Soc. Math. France **109**: 143–182, 1981.
6. H. Faure: Improvements on low discrepancy one-dimensional sequences and two-dimensional point sets. In: Keller, A, Heinrich, S and Niederreiter, H (Eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006. Springer, 327–341, 2008.
7. H. Faure: Star extreme discrepancy of generalized two-dimensional Hammersley point sets. Uniform Distribution Theory **3**: 45–65, 2008.
8. H. Faure and F. Pillichshammer: L_p discrepancy of generalized two-dimensional Hammersley point sets. Monatsh. Math. **158**: 31–61, 2009.
9. J. H. Halton and S. K. Zaremba: The extreme and the L^2 discrepancies of some plane sets. Monatsh. Math. **73**: 316–328, 1969.
10. P. Kritzer and F. Pillichshammer: An exact formula for the L_2 discrepancy of the shifted Hammersley point set. Uniform Distribution Theory **1**: 1–13, 2006.
11. L. Kuipers, and H. Niederreiter *Uniform Distribution of Sequences*. John Wiley, New York, 1974.
12. G. Larcher and F. Pillichshammer: Walsh series analysis of the L_2-discrepancy of symmetrisized point sets. Monatsh. Math. **132**: 1–18, 2001.
13. F. Pillichshammer: On the L_p-discrepancy of the Hammersley point set. Monatsh. Math. **136**: 67–79, 2002.
14. P. D. Proinov: Symmetrization of the van der Corput generalized sequences. Proc. Japan Acad. **64**, Ser. A: 159–162, 1988.
15. K. F. Roth: On irregularities of distribution. Mathematika **1**: 73–79, 1954.
16. I. H. Sloan and H. Woźniakowski: When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? J. Complexity **14**: 1–33, 1998.
17. I. V. Vilenkin: Plane nets of Integration. Ž. Vyčisl. Mat. i Mat. Fiz. **7**: 189–196, 1967. (English translation in: U.S.S.R. Computational Math. and Math. Phys. **7**, no. 1: 258–267, 1967.)
18. B. E. White: Mean-square discrepancies of the Hammersley and Zaremba sequences for arbitrary radix. Monatsh. Math. **80**: 219–229, 1975.
19. H. Woźniakowski: Average case complexity of multivariate integration. Bull. Amer. Math. Soc. **24**: 185–194, 1991.
20. S. K. Zaremba: Some applications of multidimensional integration by parts. Ann. Polon. Math. **21**: 85–96, 1968.

Vibrato Monte Carlo Sensitivities

Michael B. Giles

Abstract We show how the benefits of the pathwise sensitivity approach to computing Monte Carlo Greeks can be extended to discontinuous payoff functions through a combination of the pathwise approach and the Likelihood Ratio Method. With a variance reduction modification, this results in an estimator which for timestep h has a variance which is $O(h^{-1/2})$ for discontinuous payoffs and $O(1)$ for continuous payoffs. Numerical results confirm the variance is much lower than the $O(h^{-1})$ variance of the Likelihood Ratio Method, and the approach is also compatible with the use of adjoints to obtain multiple first order sensitivities at a fixed cost.

1 Introduction

Monte Carlo simulation is the most popular approach in computational finance for determining the prices of financial options. This is partly due to its computational efficiency for high-dimensional problems involving multiple assets, interest rates or exchange rates, and partly due to its relative simplicity and the ease with which it can be parallelised across large compute clusters. However, the accurate calculation of prices is only one objective of Monte Carlo simulation. Even more important in some ways is the calculation of the sensitivities of the prices to various input parameters. These sensitivities, known collectively as the "Greeks", are important for risk analysis and mitigation through hedging.

The pathwise sensitivity approach (also known as Infinitesimal Perturbation Analysis) is one of the standard techniques for computing these sensitivities [14]. Giles and Glasserman have recently introduced a particularly efficient implementation of this approach using adjoint techniques [13] which are related to the use of

Oxford-Man Institute of Quantitative Finance
Oxford University Mathematical Institute
24–29 St. Giles, Oxford, U.K. OX1 3LB
url: http://people.maths.ox.ac.uk/~gilesm/

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_23,

reverse mode automatic differentiation [11, 15]. This makes it possible to calculate an unlimited number of first order sensitivities at a total cost which is comparable to the cost of the original pricing calculation.

However, the pathwise approach is not applicable when the financial payoff function is discontinuous, and even when the payoff is continuous and piecewise differentiable, the use of scripting languages in real-world implementations means it can be very difficult in practice to evaluate the derivative of very complex financial products. One solution to these problems is to use the Likelihood Ratio Method (LRM) but its weaknesses are that the variance of the resulting estimator is usually $O(h^{-1})$, where h is the timestep for the path discretisation, and it can not be combined efficiently with the adjoint approach.

Building on the ideas of L'Ecuyer on hybrid pathwise/LRM sensitivity calculations [18, 19], this paper presents an idea which combines the pathwise approach for the stochastic path evolution with LRM for the payoff evaluation. Through the use of antithetic variates for variance reduction, the variance of the resulting estimator is $O(h^{-1/2})$ when the payoff is discontinuous, and $O(1)$ when it is continuous. Numerical examples show it is much more efficient than the standard LRM approach.

2 Pathwise and LRM Sensitivities

Consider the approximate solution of the general SDE driven by Brownian motion,

$$\mathrm{d}S_t = a(S,t)\,\mathrm{d}t + b(S,t)\,\mathrm{d}W_t, \tag{1}$$

using the Euler discretisation with timestep h,

$$\widehat{S}_{n+1} = \widehat{S}_n + a(\widehat{S}_n, t_n)\,h + b(\widehat{S}_n, t_n)\,\Delta W_{n+1}. \tag{2}$$

The Brownian increments ΔW_n can be defined to be a linear transformation of a vector of independent unit Normal random variables Z.

The goal is to efficiently estimate the expected value of some financial payoff function $f(S)$, and numerous first order sensitivities of this value with respect to different input parameters such as the volatility or one component of the initial data $S(0)$. In the simplest cases, $f(S)$ is a function of the value of the underlying solution $S(T)$ at the final time T, but in more general cases it might depend on the values at intermediate times as well.

The pathwise sensitivity approach can be viewed as starting with the expectation expressed as an integral with respect to Z:

$$\widehat{V} \equiv \mathbb{E}\big[f(\widehat{S})\big] = \int f(\widehat{S}(Z,\theta))\; p_Z(Z)\,\mathrm{d}Z. \tag{3}$$

Here θ represents a generic input parameter, and the probability density function for Z is

$$p_Z(Z) = (2\pi)^{-d/2} \exp\left(-\|Z\|_2^2/2\right),$$

where d is the dimension of the vector Z.

If the drift, volatility and payoff functions are all differentiable, (3) may be differentiated to give

$$\frac{\partial \widehat{V}}{\partial \theta} = \int \frac{\partial f}{\partial \widehat{S}} \frac{\partial \widehat{S}}{\partial \theta} \, p_Z(Z) \, \mathrm{d}Z, \tag{4}$$

with $\dfrac{\partial \widehat{S}}{\partial \theta}$ being obtained by differentiating (2) to obtain

$$\frac{\partial \widehat{S}_{n+1}}{\partial \theta} = \frac{\partial \widehat{S}_n}{\partial \theta} + \left(\frac{\partial a_n}{\partial \widehat{S}_n} \frac{\partial \widehat{S}_n}{\partial \theta} + \frac{\partial a_n}{\partial \theta} \right) h + \left(\frac{\partial b_n}{\partial \widehat{S}_n} \frac{\partial \widehat{S}_n}{\partial \theta} + \frac{\partial b_n}{\partial \theta} \right) \Delta W_{n+1}. \tag{5}$$

By considering the limit of a sequence of regularised functions, it can be proved that (4) remains valid when the payoff function is continuous and piecewise differentiable, and the numerical estimate obtained by averaging over M independent path simulations

$$M^{-1} \sum_{m=1}^{M} \frac{\partial f}{\partial \widehat{S}}(\widehat{S}^{(m)}) \frac{\partial \widehat{S}^{(m)}}{\partial \theta}$$

is an unbiased estimate for $\partial \widehat{V}/\partial \theta$ with a variance which is $O(M^{-1})$, independent of h, if $f(S)$ is Lipschitz and the drift and volatility functions satisfy the standard conditions [17].

Performing a change of variables, the expectation can also be expressed as

$$\widehat{V} \equiv \mathbb{E}\Big[f(\widehat{S}) \Big] = \int f(\widehat{S}) \; p_S(\widehat{S}, \theta) \, \mathrm{d}\widehat{S}, \tag{6}$$

where $p_S(\widehat{S}, \theta)$ is the probability density function for $\widehat{S}$ which will depend on all of the inputs parameters. If this is known, (6) can be differentiated to give

$$\frac{\partial \widehat{V}}{\partial \theta} = \int f \frac{\partial p_S}{\partial \theta} \mathrm{d}\widehat{S} = \int f \frac{\partial (\log p_S)}{\partial \theta} \, p_S \, \mathrm{d}\widehat{S} = \mathbb{E}\left[f \frac{\partial (\log p_S)}{\partial \theta} \right].$$

which can be estimated using the unbiased estimator

$$M^{-1} \sum_{m=1}^{M} f(\widehat{S}^{(m)}) \frac{\partial \log p_S(\widehat{S}^{(m)})}{\partial \theta}$$

This is the Likelihood Ratio Method. Its great advantage is that it does not require the differentiation of $f(\widehat{S})$. This makes it applicable to cases in which the payoff is discontinuous, and it also simplifies the practical implementation because banks often have complicated flexible procedures through which traders specify payoffs. However, it does have a number of limitations, one being a requirement of absolute continuity which is not satisfied in a few important applications such as the LIBOR market model [14]. Other drawbacks of LRM are that in most cases it gives an es-

timator with a variance which is $O(M^{-1}h^{-1})$, becoming infinite as $h \to 0$ [14], and there is no way to efficiently incorporate adjoint techniques and hence the computational cost is proportional to the number of first order sensitivities which are needed.

3 Vibrato Monte Carlo

We now introduce a hybrid combination of pathwise and LRM sensitivity calculation, applying the pathwise approach to the differentiable path simulation, and using LRM for the discontinuous payoff evaluation. The idea of combining pathwise and LRM approaches is not new. L'Ecuyer [18, 19] presented a general framework in which the two approaches are just special cases of a more general estimator, and Chen and Glasserman [5] have recently shown that the use of Malliavin calculus [8, 9] can also be viewed as a hybrid pathwise/LRM combination.

The novelty in the present paper lies in the precise form of the hybrid combination and the variance reduction which is achieved, making it a very practical method for finance applications with a discontinuous payoff function.

3.1 Conditional Expectation

The Oxford English Dictionary describes "vibrato" as "a rapid slight variation in pitch in singing or playing some musical instruments". The analogy to Monte Carlo methods is the following: whereas a path simulation in a standard Monte Carlo calculation produces a precise value for the output values from the underlying stochastic process, in the vibrato Monte Carlo approach the output values have a narrow probability distribution.

This is an example of the use of Conditional Monte Carlo simulation [1], and generalises an example discussed by Glasserman in section 7.2.3 of his book [14] as a solution to the problem of computing Greeks for discontinuous payoffs. In his example, a path simulation for a scalar SDE is performed in the usual way for the first $N-1$ timesteps, at each timestep taking a value for the Wiener increment ΔW_n which is a sample from the appropriate Gaussian distribution, and then using (2) to update the solution. On the final timestep, one instead considers the full distribution of possible values for ΔW_N. This gives a Gaussian distribution for $\widehat{S}_N$ at time T, conditional on the value of $\widehat{S}_{N-1}$ at time $T-h$, with probability density function

$$p_S(\widehat{S}_N) = \frac{1}{\sqrt{2\pi}\,\sigma_W} \exp\left(-\frac{(\widehat{S}_N - \mu_W)^2}{2\sigma_W^2}\right) \tag{7}$$

where

$$\mu_W = \widehat{S}_{N-1} + a(\widehat{S}_{N-1}, T-h)\,h, \quad \sigma_W = b(\widehat{S}_{N-1}, T-h)\sqrt{h},$$

with $a(S,t)$ and $b(S,t)$ being the drift and volatility of the SDE described in (1). Hence, the conditional expectation for the value of a digital payoff with strike K,

$$f(S(T)) \;=\; H(S(T)-K) \equiv \begin{cases} 1, & S(T) > K \\ 0, & S(T) \le K \end{cases}$$

is

$$\mathbb{E}\Big[f(\widehat{S}_N) \,|\, \widehat{S}_{N-1} \Big] \;=\; \int_{-\infty}^{\infty} H(\widehat{S}_N - K)\; p_S(\widehat{S}_N)\; \mathrm{d}\widehat{S}_N \;=\; \Phi\left(\frac{\mu_W - K}{\sigma_W} \right)$$

where $\Phi(\cdot)$ is the cumulative Normal distribution function. A Monte Carlo estimator for the option value is therefore

$$M^{-1} \sum_{m=1}^{M} \mathbb{E}[f(\widehat{S}_N) \,|\, \widehat{S}_{N-1}^{(m)}] \;\equiv\; M^{-1} \sum_{m=1}^{M} \Phi\left(\frac{\mu_W^{(m)} - K}{\sigma_W^{(m)}} \right)$$

and because the conditional expectation $\mathbb{E}[f(\widehat{S}_N) \,|\, \widehat{S}_{N-1}]$ is a differentiable function of the input parameters the pathwise sensitivity approach can now be applied.

There are two difficulties in using this form of conditional expectation in practice in financial applications. This first is that the integral arising from the conditional expectation will often become a multi-dimensional integral without an obvious closed-form value, and the second is that it requires a change to the often complex software framework used to specify payoffs.

The solution is to use a Monte Carlo estimate of the conditional expectation, and use LRM to obtain its sensitivity. Thus, the technique combines pathwise sensitivity for the path calculation with LRM sensitivity for the payoff evaluation.

3.2 Vibrato Monte Carlo

The idea is very simple; adopting the conditional expectation approach, each path simulation for a particular set of Wiener increments $W \equiv (\Delta W_1, \Delta W_2, \ldots, \Delta W_{N-1})$ (excluding the increment for the final timestep) computes a conditional Gaussian probability distribution $p_S(\widehat{S}_N | W)$. For a scalar SDE, if μ_W and σ_W are the mean and standard deviation for given W, then

$$\widehat{S}_N(W, Z) = \mu_W + \sigma_W Z,$$

where Z is a unit Normal random variable. The expected payoff can then be expressed as

$$\widehat{V} = \mathbb{E}_W \Big[\mathbb{E}_Z[f(\widehat{S}_N) \,|\, W] \Big] = \int \left\{ \int f(\widehat{S}_N)\; p_S(\widehat{S}_N \,|\, W)\; \mathrm{d}\widehat{S}_N \right\}\; p_W(W)\; \mathrm{d}W.$$

The outer expectation/integral is an average over the discrete Wiener increments, while the inner conditional expectation/integral is averaging over Z.

To compute the sensitivity to the input parameter θ, the first step is to apply the pathwise sensitivity approach for fixed W to obtain $\partial\mu_W/\partial\theta, \partial\sigma_W/\partial\theta$. We then apply LRM to the inner conditional expectation to get

$$\frac{\partial \widehat{V}}{\partial \theta} = \mathbb{E}_W\left[\frac{\partial}{\partial\theta}\mathbb{E}_Z\Big[f(\widehat{S}_N)\,|\,W\Big]\right] = \mathbb{E}_W\left[\mathbb{E}_Z\left[f(\widehat{S}_N)\,\frac{\partial(\log p_S)}{\partial\theta}\;|\;W\right]\right],$$

where p_S is defined in (7) and

$$\frac{\partial(\log p_S)}{\partial\theta} = \frac{\partial(\log p_S)}{\partial\mu_W}\,\frac{\partial\mu_W}{\partial\theta} + \frac{\partial(\log p_S)}{\partial\sigma_W}\,\frac{\partial\sigma_W}{\partial\theta}.$$

The Monte Carlo estimators for $\widehat{V}$ and $\partial\widehat{V}/\partial\theta$ have the form

$$M^{-1}\sum_{m=1}^{M}\widehat{Y}^{(m)}, \qquad M^{-1}\sum_{m=1}^{M}\widehat{Y}_\theta^{(m)},$$

where $\widehat{Y}^{(m)}$ is an unbiased estimator for $\mathbb{E}_Z\left[f(\widehat{S}_N)\,|\,W^{(m)}\right]$ and $\widehat{Y}_\theta^{(m)}$ is an unbiased estimator for $\mathbb{E}_Z\left[f(\widehat{S}_N)\,\frac{\partial(\log p_S)}{\partial\theta}\,|\,W^{(m)}\right]$ for a given set of Brownian increments $W^{(m)}$.

Although the discussion so far has considered an option based on the value of a single underlying value at the terminal time T, it will be shown that the idea extends very naturally to multidimensional cases, producing a conditional multivariate Gaussian distribution, and also to financial payoffs which are dependent on values at intermediate times.

3.3 Efficient Estimators

It is important that $\widehat{Y}$ and $\widehat{Y}_\theta$ have low variance to minimise the number of path simulations which must be performed to achieve a given accuracy. Rather than defining $\widehat{Y}^{(m)}$ to be simply

$$\widehat{Y}^{(m)} = f(\mu_W^{(m)} + \sigma_W^{(m)} Z^{(m)}),$$

using a single independent Z sample for each Brownian path, it is better use antithetic variates to reduce the variance, noting that

$$\mathbb{E}_Z\Big[f(\widehat{S}_N)\,|\,W\Big] = \mathbb{E}_Z\left[\tfrac{1}{2}\Big(f(\mu_W+\sigma_W Z)+f(\mu_W-\sigma_W Z)\Big)\right],$$

and also use multiple independent Z samples for each Brownian path by defining $\widehat{Y}^{(m)}$ to be

$$\widehat{Y}^{(m)} = P^{-1} \sum_{p=1}^{P} \tfrac{1}{2} \Big(f(\mu_W^{(m)} + \sigma_W^{(m)} Z^{(m,p)}) + f(\mu_W^{(m)} - \sigma_W^{(m)} Z^{(m,p)}) \Big).$$

The optimal number of samples will be considered later, but the variance $\mathbb{V}_Z[\widehat{Y}^{(m)} \mid W]$ will be particularly small if $f(S)$ is locally differentiable, and in this case a single Z sample is probably sufficient.

For a scalar SDE and a given W,

$$\log p_S = -\log \sigma_W - \frac{(\widehat{S}_N - \mu_W)^2}{2\sigma_W^2} - \tfrac{1}{2}\log(2\pi)$$

and

$$\mathbb{E}_Z\left[f(\widehat{S}_N) \, \frac{\partial(\log p_S)}{\partial \theta} \mid W \right] = \frac{\partial \mu_W}{\partial \theta} \, \mathbb{E}_Z\left[f(\widehat{S}_N) \, \frac{\partial(\log p_S)}{\partial \mu_W} \mid W \right] + \frac{\partial \sigma_W}{\partial \theta} \, \mathbb{E}_Z\left[f(\widehat{S}_N) \, \frac{\partial(\log p_S)}{\partial \sigma_W} \mid W \right].$$

Looking at the first of the two expectations on the r.h.s., then

$$\begin{aligned} \mathbb{E}_Z\left[f(\widehat{S}_N) \frac{\partial(\log p_S)}{\partial \mu_W} \mid W \right] &= \mathbb{E}_Z\left[\frac{Z}{\sigma_W} \, f(\mu_W + \sigma_W Z) \right] \\ &= \mathbb{E}_Z\left[\frac{Z}{2\sigma_W} \Big(f(\mu_W + \sigma_W Z) - f(\mu_W - \sigma_W Z) \Big) \right]. \end{aligned}$$

If $f(S)$ is locally differentiable, this is the expectation of a quantity which is $O(1)$ in magnitude, and one Z sample is probably sufficient to estimate its value. If $f(S)$ is discontinuous, then for paths near the discontinuity the expectation is of a quantity which is $O(\sigma_W^{-1}) = O(h^{-1/2})$ and it will be more efficient to use multiple samples to estimate the expected value.

Similarly, using the additional result that $\mathbb{E}_Z[Z^2 - 1] = 0$,

$$\begin{aligned} &\mathbb{E}_Z\left[f(\widehat{S}_N) \, \frac{\partial(\log p_S)}{\partial \sigma_W} \mid W \right] \\ &\quad = \mathbb{E}_Z\left[\frac{Z^2 - 1}{\sigma_W} \, f(\mu_W + \sigma_W Z) \right] \\ &\quad = \mathbb{E}_Z\left[\frac{Z^2 - 1}{2\sigma_W} \Big(f(\mu_W + \sigma_W Z) - 2 f(\mu_W) + f(\mu_W - \sigma_W Z) \Big) \right]. \end{aligned}$$

The expression within this expectation is in general no larger than for the previous expectation, and so the same set of samples will suffice.

Combining these two derivations, we finally define $\widehat{Y}_\theta^{(m)}$ to be

$$\widehat{Y}_\theta^{(m)} = \frac{\partial \mu_W}{\partial \theta} \, \widehat{Y}_\mu^{(m)} + \frac{\partial \sigma_W}{\partial \theta} \, \widehat{Y}_\sigma^{(m)}$$

where $\widehat{Y}_\mu^{(m)}$ and $\widehat{Y}_\sigma^{(m)}$ are the following averages based on P independent Z samples,

$$\widehat{Y}_\mu^{(m)} = P^{-1}\sum_{p=1}^{P} \frac{Z^{(m,p)}}{2\sigma_W}\left(f(\mu_W^{(m)}+\sigma_W^{(m)}Z^{(m,p)}) - f(\mu_W^{(m)}-\sigma_W^{(m)}Z^{(m,p)})\right) \quad (8)$$

$$\widehat{Y}_\sigma^{(m)} = P^{-1}\sum_{p=1}^{P} \frac{(Z^{(m,p)})^2-1}{2\sigma_W^{(m)}}\Big(f(\mu_W^{(m)}+\sigma_W^{(m)}Z^{(m,p)}) - 2f(\mu_W^{(m)}) + f(\mu_W^{(m)}-\sigma_W^{(m)}Z^{(m,p)})\Big)$$

3.4 Multivariate Generalisation

These estimators can be generalised to the case of multiple assets with a multivariate Gaussian distribution conditional on the set of Wiener increments which lead to approximation $\widehat{S}_{N-1}$ at time $T-h$. If μ_W is now the column vector of conditional expectations $\mathbb{E}[\widehat{S}_N \mid W]$, and Σ_W is the covariance matrix, then $\widehat{S}_N$ can be written as

$$\widehat{S}_N(W,Z) = \mu_W + C_W Z,$$

where Z is a vector of uncorrelated unit Normal variables and C_W is any matrix such that $\Sigma_W = C_W C_W^T$, with C_W^T denoting the matrix transpose. Provided Σ_W is non-singular, the joint probability density function for S is

$$\log p_S = -\tfrac{1}{2}\log|\Sigma_W| - \tfrac{1}{2}(\widehat{S}_N-\mu_W)^T\Sigma_W^{-1}(\widehat{S}_N-\mu_W) - \tfrac{1}{2}d\log(2\pi),$$

where d is the dimension of Z. Differentiating this (see [7, 20]) gives

$$\frac{\partial \log p_S}{\partial \mu_W} = \Sigma_W^{-1}(\widehat{S}_N-\mu_W) = C_W^{-T}Z,$$

where C_W^{-T} is shorthand for $(C_W^{-1})^T$, and

$$\frac{\partial \log p_S}{\partial \Sigma_W} = -\tfrac{1}{2}\Sigma_W^{-1} + \tfrac{1}{2}\Sigma_W^{-1}(\widehat{S}_N-\mu_W)(\widehat{S}_N-\mu_W)^T\Sigma_W^{-1} = \tfrac{1}{2}C_W^{-T}\left(ZZ^T-I\right)C_W^{-1}.$$

For a given W,

$$\mathbb{E}_Z\left[f(\widehat{S}_N)\,\frac{\partial(\log p_S)}{\partial\theta} \mid W\right] = \left(\frac{\partial\mu_W}{\partial\theta}\right)^T \mathbb{E}_Z\left[f(\widehat{S}_N)\,\frac{\partial(\log p_S)}{\partial\mu_W} \mid W\right] + \mathrm{Trace}\left(\frac{\partial\Sigma_W}{\partial\theta}\,\mathbb{E}_Z\left[f(\widehat{S}_N)\,\frac{\partial(\log p_S)}{\partial\Sigma_W} \mid W\right]\right),$$

where the trace of a matrix is the sum of its diagonal elements. To obtain efficient estimators, we again use antithetic variates to get

$$\mathbb{E}_Z\left[f(\widehat{S}_N)\ \frac{\partial(\log p_S)}{\partial \mu_W} \mid W \right] = \mathbb{E}_Z\left[\tfrac{1}{2}\left(f(\mu_W + C_W Z) - f(\mu_W - C_W Z) \right) C_W^{-T} Z \right],$$

and we use $\mathbb{E}_Z[Z\,Z^T - I\,] = 0$ to give

$$\begin{aligned}
&\mathbb{E}_Z\left[f(\widehat{S}_N)\ \frac{\partial(\log p_S)}{\partial \Sigma_W} \mid W \right] \\
&\quad = \mathbb{E}_Z\left[\tfrac{1}{4}\left(f(\mu_W + C_W Z) - 2f(\mu_W) + f(\mu_W - C_W Z) \right) C_W^{-T} (Z\,Z^T - I)\, C_W^{-1} \right].
\end{aligned}$$

These two results lead to the estimator

$$\widehat{Y}_\theta^{(m)} = \left(\frac{\partial \mu_W}{\partial \theta} \right)^T \widehat{Y}_\mu^{(m)} + \mathrm{Trace}\left(\frac{\partial \Sigma_W}{\partial \theta}\ \widehat{Y}_\Sigma^{(m)} \right)$$

where $\widehat{Y}_\mu^{(m)}$ and $\widehat{Y}_\Sigma^{(m)}$ are defined as

$$\begin{aligned}
\widehat{Y}_\mu^{(m)} &= P^{-1} \sum_{p=1}^{P} \left(f(\mu_W^{(m)} + C_W^{(m)} Z^{(m,p)}) - f(\mu_W^{(m)} - C_W^{(m)} Z^{(m,p)})\ (C_W^{(m)})^{-T} Z^{(m,p)} \right) \\
\widehat{Y}_\Sigma^{(m)} &= P^{-1} \sum_{p=1}^{P} \tfrac{1}{4} \left(f(\mu_W^{(m)} + \sigma_W^{(m)} Z^{(m,p)}) - 2f(\mu_W^{(m)}) + f(\mu_W^{(m)} - \sigma_W^{(m)} Z^{(m,p)}) \right) \\
&\qquad \times (C_W^{(m)})^{-T} \left(Z^{(m,p)} (Z^{(m,p)})^T - I \right) (C_W^{(m)})^{-1}.
\end{aligned} \tag{9}$$

If the payoff also depends on values at intermediate times τ_j, not just at maturity, these can be handled by omitting the simulation time t_n closest to each measurement time τ_j, using a timestep twice as big as usual for the time interval $[t_{n-1}, t_{n+1}]$. Using Brownian interpolation conditional on the values $\widehat{S}_{n\pm1}$, with constant drift and volatility based on $\widehat{S}_{n-1}$, results in a Gaussian distribution for $\widehat{S}(\tau_j)$ of the form

$$\widehat{S}(\tau_j) = \widehat{S}_{n-1} + \frac{\tau_j - t_{n-1}}{2h} \left(\widehat{S}_{n+1} - \widehat{S}_{n-1} \right) + \frac{\sqrt{(t_{n+1} - \tau_j)\,(\tau_j - t_{n-1})}}{2h}\ C_{n-1}\ Z$$

where $C_{n-1} C_{n-1}^T$ is the covariance matrix for $\widehat{S}_{n+1}$ conditional on $\widehat{S}_{n-1}$, and Z is again a vector of uncorrelated unit Normal variables. Collectively, the values $\widehat{S}(\tau_j)$ form a set with a multivariate Normal distribution, conditional on the set of discrete Wiener increments, with the values at different times being independently distributed. One can then apply the theory above to obtain the sensitivities.

The Likelihood Ratio Method is not applicable when the covariance matrix Σ is singular. This situation occurs, for example, in the LIBOR market model driven by a single Brownian motion [3]. A solution is to introduce an additional diffusion in the final timestep, for example by replacing Σ by $\Sigma + \sigma I$, where I is the identity matrix. If the extra diffusion is of a similar magnitude (e.g. σ is approximately equal to the largest eigenvalue of Σ) this will introduce an $O(h)$ bias in the expected payoff and its sensitivity, but this bias is of the same order of magnitude as the weak convergence error associated with the Euler approximation.

3.5 Optimal Number of Samples

The use of multiple samples to estimate the value of the conditional expectations is an example of the splitting technique [1]. If W and Z are independent random variables, then for any function $g(W, Z)$ the estimator

$$\widehat{Y}_{M,P} = M^{-1} \sum_{m=1}^{M} \left(P^{-1} \sum_{p=1}^{P} g(W^{(m)}, Z^{(m,p)}) \right)$$

with independent samples $W^{(m)}$ and $Z^{(m,p)}$ is an unbiased estimator for

$$\mathbb{E}_{W,Z}\left[g(W,Z)\right] \equiv \mathbb{E}_W\Big[\mathbb{E}_Z[g(W,Z)\,|\,W]\Big],$$

and its variance is

$$\mathbb{V}[\widehat{Y}_{M,P}] = M^{-1}\,\mathbb{V}_W\Big[\mathbb{E}_Z[g(W,Z)\,|\,W]\Big] + (MP)^{-1}\,\mathbb{E}_W\Big[\mathbb{V}_Z[g(W,Z)\,|\,W]\Big].$$

Applying this general result to our vibrato estimators with P samples for Z for each simulation path, the variance is of the form

$$v_1\,M^{-1} + v_2\,(MP)^{-1},$$

and the cost of computing $\widehat{Y}_{M,P}$ is proportional to

$$c_1\,M + c_2\,MP,$$

with c_1 corresponding to the path calculation and c_2 corresponding to the payoff evaluation. For a fixed computational cost, the variance can be minimised by minimising the product

$$\left(v_1 + v_2\,P^{-1}\right)(c_1 + c_2\,P) = v_1\,c_2\,P + v_1\,c_1 + v_2\,c_2 + v_2\,c_1\,P^{-1},$$

which gives the optimum value $P_{\text{opt}} = \sqrt{v_2\,c_1/v_1\,c_2}$.

c_1 is $O(h^{-1})$ since the cost is proportional to the number of timesteps, and c_2 is $O(1)$, independent of h. If the payoff is Lipschitz, then $\widehat{Y}_\theta$ is $O(1)$ for all paths, and so v_1 and v_2 are both $O(1)$ and $P_{\text{opt}} = O(h^{-1/2})$. On the other hand, if the payoff is discontinuous with an $O(h^{1/2})$ fraction of paths being within $O(h^{1/2})$ of the discontinuity (which assumes a locally bounded density for the distribution of $S(T)$) then for these paths $\mathbb{E}_Z[\widehat{Y}_\theta|W] = O(h^{-1/2})$ and $\mathbb{V}_Z[\widehat{Y}_\theta|W] = O(h^{-1})$. This leads to v_1 and v_2 both being $O(h^{-1/2})$ and so again $P_{\text{opt}} = O(h^{-1/2})$.

In both cases, as $h \to 0$, the variance is asymptotically equal to $v_1\,M^{-1}$ and the cost is asymptotically equal to $c_1\,M$. Thus the use of the vibrato technique does not, to leading order, increase the variance or the computational cost compared to the use of exact conditional expectation in the few cases for which this exists in a simple closed form.

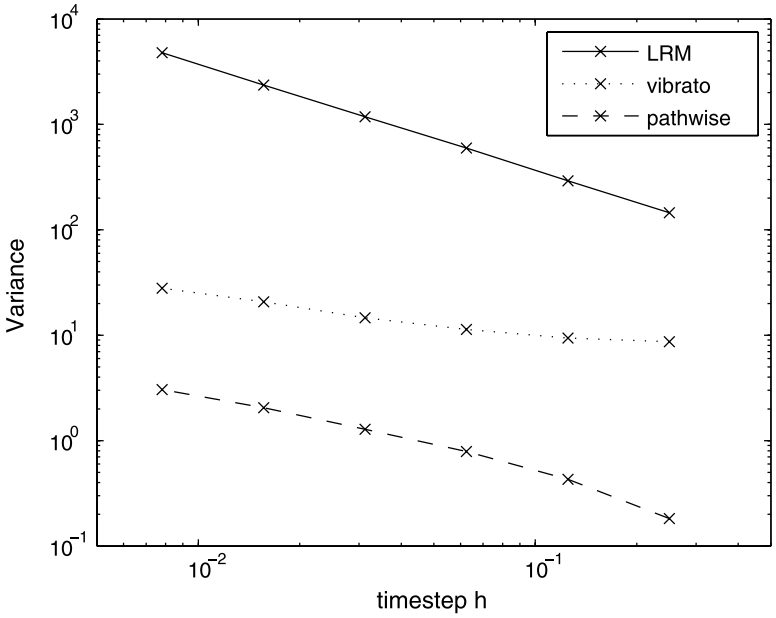


Fig. 1 Comparison of Vega variance for LRM, pathwise and vibrato estimators.

3.6 Numerical Results

We consider a 2-dimensional Geometric Brownian Motion,

$$\begin{aligned} \mathrm{d}S_t^{(1)} &= r\,S_t^{(1)}\,\mathrm{d}t + \sigma^{(1)}\,S_t^{(1)}\,\mathrm{d}W_t^{(1)} \\ \mathrm{d}S_t^{(2)} &= r\,S_t^{(2)}\,\mathrm{d}t + \sigma^{(2)}\,S_t^{(2)}\,\mathrm{d}W_t^{(2)} \end{aligned}$$

with parameters $r = 0.05,\ \sigma^{(1)} = 0.2,\ \sigma^{(2)} = 0.3$ and correlation $\varrho = 0.5$ between the driving Brownian motions. The payoff function is chosen to be a digital call paying a discounted value of $\exp(-rT)$ if and only if the value of $S^{(1)}(T)$ exceeds the strike K. Parameter values $T = 1, K = 100$ are used. This very simple example is chosen so that in Fig. 1 we can compare the variance for the vibrato calculation to the variance of both the LRM method and also the pathwise method in combination with the analytic conditional expectation.

The figure shows the increase in the variance of the estimator for one of the Vegas, $\partial V/\partial\sigma^{(1)}$, as the timestep h is reduced. We see the rapid increase in the variance of the LRM method which is $O(h^{-1})$ asymptotically, and the much slower $O(h^{-1/2})$ growth in the variance of the two sets of results based on the pathwise approach. The difference between the pathwise and vibrato results is due to the number of Z samples used in the vibrato method. Only one sample was used for the results presented here; increasing this number will lead to the vibrato variance

converging to the variance of the pathwise method using the analytic conditional expectation. It is striking how much larger the LRM variance is. With 2 timesteps it is already 10 times larger than the vibrato method with a single Z sample, while for 128 timesteps it is 200 times larger.

4 Adjoint Pathwise Sensitivity Implementation

There is insufficient space in this paper to fully explain the adjoint implementation, but it is important to note that the vibrato approach is completely compatible with an adjoint calculation of the path sensitivity, and thus it is possible to obtain an unlimited number of first order sensitivities at a cost which is similar to the cost of the original calculation.

To give an introduction to the ideas, we follow the terminology used by the Automatic Differentiation community [2, 4, 6, 15]. Forward mode sensitivity calculation (like the standard pathwise sensitivity calculation) starts with a perturbation to an input and derives the corresponding perturbation to all subsequent variables. Doing this within a computer program at the level of individual binary operation (e.g. addition or multiplication) of the form

$$c = g(a, b)$$

leads to the corresponding linear perturbation equation

$$\dot{c} = \frac{\partial c}{\partial a}\,\dot{a} + \frac{\partial c}{\partial b}\,\dot{b}$$

where $\dot{c}$ denotes the derivative of c with respect to the perturbed input parameter.

By contrast, the reverse (or adjoint) mode starts with the fact that the final output of interest has unit sensitivity with respect to itself, and then works backward through the sequence of computer instructions, to determine the sensitivity of the final output to changes in the input parameters of each instruction. Assuming that a and b are only used for the computation of c in the above example (i.e. they are not used as inputs for any other calculation) the corresponding two adjoint equations are

$$\overline{a} = \frac{\partial c}{\partial a}\,\overline{c}, \qquad \overline{b} = \frac{\partial c}{\partial b}\,\overline{c}$$

where $\overline{a}$ represents the sensitivity of the final output to changes in a.

The key point of the adjoint approach is that by working backwards from the payoff calculation through the path evolution back to the start, it can compute the sensitivity of a single output quantity (such as a payoff function) to an unlimited number of input parameters (such as initial price, interest rate, volatility, etc.) at a total cost which is little more than the original calculation. For details on this approach and its use in computational finance, see [11, 13, 16].

In applying these adjoint ideas to the vibrato approach in this paper, for each path in the scalar case one would simulate the path up to $\widehat{S}_{N-1}$ and compute the quantities $\widehat{Y}_\mu$ and $\widehat{Y}_\sigma$ as defined in (8). These values correspond to $\overline{\mu_W}$ and $\overline{\sigma_W}$, the sensitivity of the estimated payoff for that path to changes in μ_W and σ_W. This is the initialisation required for the reverse pass of the adjoint path calculation which will lead to the calculation of $\overline{\theta}$, the sensitivity of the estimated payoff for that path to changes in an input parameter θ. Similarly, in the multivariate case the adjoint initialisation is $\overline{\mu_W} = \widehat{Y}_\mu$ and $\overline{\Sigma_W} = \widehat{Y}_\Sigma$, where $\widehat{Y}_\mu$ and $\widehat{Y}_\Sigma$ are as defined in (10).

5 Conclusions and Future Work

In this paper we have introduced the idea of vibrato Monte Carlo sensitivity calculations. This can be viewed as an application of the Conditional Monte Carlo approach, and is a generalisation of the use of conditional expectation for payoff smoothing. It leads to a hybrid method for calculating sensitivities, applying pathwise sensitivity analysis to the path simulation, and the Likelihood Ratio Method to the payoff evaluation. This offers the computational efficiency of the pathwise method, particularly when combined with an adjoint implementation, together with the greater generality and ease-of-implementation of LRM.

Although the paper discusses only first order sensitivities, the approach extends naturally to higher order derivatives. A similar variance reduction construction for second order derivatives leads to an estimator with a variance which is $O(h^{-1/2})$ for payoffs which are continuous but have a discontinuous derivative, and $O(h^{-3/2})$ for payoffs which are discontinuous.

Another direction for future research is the use of the vibrato idea for multilevel Monte Carlo analysis [12]. Analytic conditional expectation is currently used to treat discontinuous payoffs to obtain improved convergence rates with the Milstein scheme [10]. The vibrato approach will allow this to be generalised to multivariate cases.

Acknowledgements This research has been supported by the Oxford-Man Institute of Quantitative Finance and a fellowship from the UK Engineering and Physical Sciences Research Council. I am very grateful to the reviewers for their helpful comments on the original paper.

References

1. Asmussen, A., Glynn, P.: Stochastic Simulation. Springer, New York (2007)
2. Bischof, C., Bücker, M., Hovland, P., Naumann, U., Utke, J. (eds.): Advances in Automatic Differentiation. Springer-Verlag (2008)
3. Brace, A., Gaterak, D., Musiela, M.: The market model of interest rate dynamics. Mathematical Finance **7**, 127–155 (1997)

4. Bücker, M., Corliss, G., Hovland, P., Naumann, U., Norris, B. (eds.): Automatic Differentiation: Applications, Theory and Implementations. Springer-Verlag (2006)
5. Chen, N., Glasserman, P.: Malliavin Greeks without Malliavin calculus. Stochastic Processes and their Applications **117**, 1689–1723 (2007)
6. Corliss, G., Faure, C., Griewank, A., Hascoët, L., Naumann, U. (eds.): Automatic Differentiation: From Simulation to Optimization. Springer-Verlag (2001)
7. Dwyer, P.: Some applications of matrix derivatives in multivariate analysis. Journal of the American Statistical Association **62**(318), 607–625 (1967)
8. Fournié, E., Lasry, J.M., Lebuchoux, J., Lions, P.L., Touzi, N.: Applications of Malliavin calculus to Monte Carlo methods in finance. Finance and Stochastics **3**, 391–412 (1999)
9. Fournié, E., Lasry, J.M., Lebuchoux, J., Lions, P.L., Touzi, N.: Applications of Malliavin calculus to Monte Carlo methods in finance, II. Finance and Stochastics **5**, 201–236 (2001)
10. Giles, M.: Improved multilevel Monte Carlo convergence using the Milstein scheme. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 343–358. Springer-Verlag (2007)
11. Giles, M.: Monte Carlo evaluation of sensitivities in computational finance. Tech. Rep. NA07/12, Oxford University Computing Laboratory (2007)
12. Giles, M.: Multilevel Monte Carlo path simulation. Operations Research **56**(3), 981–986 (2008)
13. Giles, M., Glasserman, P.: Smoking adjoints: fast Monte Carlo Greeks. RISK (2006)
14. Glasserman, P.: Monte Carlo Methods in Financial Engineering. Springer, New York (2004)
15. Griewank, A.: Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM (2000)
16. Kaebe, C., Maruhn, J., Sachs, E.: Adjoint based Monte Carlo calibration of financial market models. Finance and Stochastics **13**(3), 351–237 (2009)
17. Kloeden, P., Platen, E.: Numerical Solution of Stochastic Differential Equations. Springer, Berlin (1992)
18. L'Ecuyer, P.: A unified view of the IPA, SF and LR gradient estimation techniques. Management Science **36**(11), 1364–1383 (1990)
19. L'Ecuyer, P.: On the interchange of derivative and expectation for likelihood ratio derivative estimators. Management Science **41**(4), 738–748 (1995)
20. Magnus, J., Neudecker, H.: Matrix differential calculus with applications in statistics and econometrics. John Wiley & Sons (1988)

The Weighted Variance Minimization in Jump-Diffusion Stochastic Volatility Models

Anatoly Gormin and Yuri Kashtanov

Abstract The Monte Carlo method is applied to estimation of options in the case of a stochastic volatility model with jumps. An option contract has a number of parameters like a strike, an exercise date, etc. Estimators of option prices with different values of its parameters are constructed on the same trajectories of the underlying asset price process. The problem of minimization of the weighted sum of their variances is considered. Optimal estimators with minimal weighted variance are pointed out. Their approximations are applied to variance reduction.

1 Introduction

The Monte Carlo method provides a general approach for the options prices valuation. It is especially useful in the cases of complicated models, path-dependent options or multidimensional underlyings. Since the main disadvantage of the Monte Carlo method is the low rate of convergence, it is important to reduce variances of estimators.

This problem was investigated by many authors for different models; see for example [10] in the case of diffusion model or [4], [5] in the case of stochastic volatility model. The authors use the methods of importance sampling and control variates to reduce the variance of the particular option estimator and point out the optimal estimators which reduce the variance to zero. The method of control variates was used in [11] and [14] in particular jump-diffusion models.

Faculty of Mathematics and Mechanics
Department of Statistical Simulation
Saint-Petersburg State University
198504 Saint-Petersburg, Russia
e-mail: Anatoliy.Gormin@pobox.spbu.ru
e-mail: Yuri.Kashtanov@paloma.spbu.ru

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_24,

Stochastic volatility models are considered in detail in [12]. Homogeneous jump-diffusion models (Levy processes) are described in [2], the optimal hedging strategies for these models are pointed out. More general jump-diffusion models are considered in [7] and [13], Chapter 8. As mentioned in the last paper, jumps in financial models "provide a better fit to time series data and greater flexibility in matching derivative prices".

Our approach differs from above one. We consider several estimators (on the same paths of the underlying asset price process) corresponding different parameters such as a strike price, a barrier etc. and reduce the weighted sum of variances. The computation of several options is important in portfolio optimization, risk management etc. (see [9]). The method of importance sampling was applied in [3], Chapter 4 to the weighted sum of variances minimization in the case of several integrals calculation with respect to one probability measure. We considered the problem of the weighted variance minimization in [8] in the case of a diffusion model, the methods of importance sampling and control variates were applied to minimization of the weighted variance. Here we extend the results of [8] to the case of the stochastic volatility jump-diffusion model (SVJD) and consider several weighting parameters.

In Section 2 the optimal estimators are pointed out; their specifications for different options are given in Section 3. The optimal estimators can not be applied for computations directly because they refer to the values, the computation of which is of the same complexity as the computation of the option prices themself. But if we have some approximations to these values we can construct unbiased estimators which have less variance than the standard ones. We compute preliminarily these approximations on a grid, and then use their linear interpolations in simulations. Due to the simplicity of this procedure, the time of the trajectory simulation increases insignificantly. To understand in what cases this variance reduction is effective, note that the total computational time T_ε, where ε is an accuracy of estimation, can be expressed in the form

$$T_\varepsilon = t_0 + c_\alpha \frac{D}{\varepsilon^2} t_1, \tag{1}$$

where t_0 is the time of approximations calculation, t_1 is the time of one trajectory simulation, D is the variance of an estimator, c_α is a constant dependent on the confidence level α. Thus, if we are interested in high accuracy, the estimator with smaller variance will be more effective however big t_0 is.

Approximations of optimal functions are constructed in Section 4; they are used in estimators which we apply to price valuation. Results of computations are shown in Section 5.

Below we describe the process of underlying prices S_t and assume that the process of interest rate r_t has the form $r_t = r(t, S_t)$. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, N_t be a Poisson process with constant intensity λ. This means that $N_t = \sum_{n\geq 1} 1_{\{T_n \leq t\}}$, where T_1, $(T_{n+1} - T_n)_{n\geq 1}$ are i.i.d. exponential random variables with parameter λ. Let $(Y_n)_{n\geq 1}$ be a sequence of independent random variables with a distribution $m(dy)$ on a measurable space $(E, \mathcal{E})$, $p(dt, dy)$ be the

counting measure of $(T_n, Y_n)_{n\geq 1}$, and $\nu(dt,dy) = \lambda m(dy)dt$ be the compensator of $p(dt,dy)$. Denote by $\tilde{p}(dt,dy)$ the compensated measure $p(dt,dy) - \nu(dt,dy)$.

Suppose the vector process $(S_t, V_t)_{0\leq t\leq T}$ has the representation

$$S_t = \int_0^t \mu(\tau, S_\tau)S_\tau d\tau + \int_0^t \sigma(\tau, V_\tau)S_\tau dW_\tau^1 + \sum_{n=1}^{N_t} \gamma(T_n, S_{T_n-}, Y_n)S_{T_n-} \tag{2}$$

$$V_t = \int_0^t \eta(\tau, V_\tau)d\tau + \int_0^t \theta(\tau, V_\tau)dW_\tau^2, \tag{3}$$

$$W_t' = \rho W_t^1 + \sqrt{1-\rho^2}W_t^2,$$

where $\mu(t,S_t) = r(t,S_t) - \lambda\int_E \gamma(t,S_{t-},y)m(dy)$, the functions σ, θ are positive, $|\rho| < 1$, $\gamma(t,x,y) > -1$, and $W_t = (W_t^1, W_t^2)^T$ is the standard two-dimensional Brownian motion independent of the sequence $(T_n, Y_n)_{n\geq 1}$. Let a filtration $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$ be the natural filtration generated by $(W_t)_{t\geq 0}$ and $(T_n, Y_n)_{n\geq 1}$.

Note that if we consider (2), (3) as equations for (S_t, V_t) then under well-known conditions on the coefficients r, σ, γ, etc. (see [13], Chapter 3), there exists a unique strong solution. An appropriate choice of the function $\gamma(t,x,y)$ allows one to construct a process with a state-dependent intensity of jumps (see [7]). Usually the coefficients η and θ are chosen so that the process V_t has the mean-reversion property.

We suppose that there exist constants c_1 and c_2 such that for almost every $k \in \Theta$ a payoff function $f_k(S)$ satisfies the inequality $f_k(S) \leq c_1||S|| + c_2$ $\mathbb{P}$-a.s., where S is a random trajectory, $||S|| = \sup_{t\leq T} S_t$.

Let $R_t = e^{-\int_0^t r_s ds}$, under the measure $\mathbb{P}$ the discounted asset price process $(R_t S_t)_{0\leq t\leq T}$ is an $\mathcal{F}_t$-martingale, and we will evaluate the option price given by the formula $C_k = \mathbb{E}(R_T f_k)$. Let the parameter k take values from a set Θ. We shall construct unbiased estimators $\widehat{C}_k$ for C_k on the same trajectories of the process $(S_t)_{0\leq t\leq T}$ with the minimal weighted variance

$$\int_\Theta \mathrm{Var}(\widehat{C}_k)\mathsf{Q}(dk), \tag{4}$$

where $\mathrm{Var}(\xi) = \mathbb{E}(\xi - \mathbb{E}\xi)^2$, the measure Q defines the accuracy of C_k estimation. These weights may be chosen in different ways. The simplest choice is $\mathsf{Q}\{k\} = 1$; if k is the strike price it is more natural to chose $\mathsf{Q}\{k\} = \widetilde{\mathcal{V}}_k$, where $\widetilde{\mathcal{V}}_k$ is an approximation of Vega of the option. For convenience assume that $\mathsf{Q}(\Theta) = 1$.

2 Estimators with the Minimal Weighted Variance

We consider methods of importance sampling and control variates to reduce the weighted variance.

Importance sampling. Let υ_t be an $\mathbb{F}$-adapted two dimensional process, $\kappa_t(y)$ be an $\mathbb{F}$-predictable E-marked process such that $\kappa_{T_n}(Y_n) > -1$ for any $n \geq 1$. Denote by ϑ the pair $\{\upsilon, \kappa\}$. Let $(L_t^\vartheta)_{0\leq t\leq T}$ be the solution of the equation

$$\frac{dL_t^\vartheta}{L_{t-}^\vartheta} = \upsilon_t dW_t + \int_E \kappa_t(y)\tilde{p}(dt,dy), \quad L_0^\vartheta = 1.$$

From Itô's lemma (see [2], Chapter 8) it follows that the solution is given by the formula

$$L_t^\vartheta = \exp\left(\int_0^t \upsilon_s dW_s - \frac{1}{2}\int_0^t |\upsilon_s|^2 ds - \int_0^t\int_E \kappa_s(y)\nu(ds,dy)\right) \times \prod_{n=1}^{N_t}\left(1+\kappa_{T_n}(Y_n)\right). \tag{5}$$

If for each $t \in [0,T]$

$$|\upsilon_t|^2 + \int_E \kappa_t^2(y)m(dy) \leq c(t) \quad \mathbb{P}-\text{a.s.}, \tag{6}$$

where $c(t) \geq 0$ is non-random such that $\int_0^T c(t)dt < \infty$, then $(L_t^\vartheta)_{0\leq t\leq T}$ is a $\mathbb{P}$-martingale and $\mathbb{E}L_T^\vartheta = 1$ (see [13], Chapter 3). Therefore, we can define the measure $\mathbb{P}^\vartheta$ by $d\mathbb{P}^\vartheta = L_T^\vartheta d\mathbb{P}$. Under the measure $\mathbb{P}^\vartheta$ the process $W_t^\upsilon = W_t - \int_0^t \upsilon_s ds$ is a Wiener process and $p(dt,dy)$ has the compensator $(1+\kappa_t(y))\nu(dt,dy)$ (see [13], Chapter 3). Note that if we define $\psi_t = \int_E \kappa_t(y)m(dy) + 1$ and $h_t(y) = (\kappa_t(y) + 1)/\psi_t$, then $p(dt,dy)$ has $(\mathbb{P}^\vartheta, \mathcal{F}_t)$-local characteristics $(\lambda\psi_t, h_t(y)m(dy))$: under the new measure $\mathbb{P}^\vartheta$, the intensity is equal to $\lambda\psi_t$ and the distribution of marks is $h_t(y)m(dy)$. Let us denote by $\mathbb{E}^\vartheta$ the expectation under $\mathbb{P}^\vartheta$. Define $\rho_T^\vartheta = \left(L_T^\vartheta\right)^{-1}$ and consider estimators in the form

$$\widehat{C}_k(\vartheta) = R_T f_k \rho_T^\vartheta. \tag{7}$$

Control variates. Denote by φ the pair $\{z,\zeta\}$, where $(z_t)_{0\leq t\leq T}$ is an $\mathbb{F}$-adapted two dimensional process and $(\zeta_t(y))_{0\leq t\leq T}$ is an $\mathbb{F}$-predictable E-marked process such that

$$\mathbb{E}\int_0^T |z_t|^2 dt < \infty, \quad \mathbb{E}\int_0^T\int_E \zeta_t^2(y)\nu(dt,dy) < \infty.$$

In this case $M_t^\varphi = M^{z,\zeta} = \int_0^t z_s dW_s + \int_0^t\int_E \zeta_s(y)\tilde{p}(ds,dy)$ is a square integrable martingale (see [2], Chapter 8) and we consider estimators of the form

$$\bar{C}_k(\varphi) = R_T f_k + M_T^\varphi. \tag{8}$$

Since $\mathbb{E}^\vartheta \widehat{C}_k(\vartheta) = \mathbb{E}\bar{C}_k(\varphi) = C_k$, the problems of the weighted variance minimization are reduced to the weighted second moment minimization

$$\min_{\vartheta} \int_{\Theta} \mathbb{E}^{\vartheta} \widehat{C}_k^2(\vartheta) \mathsf{Q}(dk), \tag{9}$$

$$\min_{\varphi} \int_{\Theta} \mathbb{E} \bar{C}_k^2(\varphi) \mathsf{Q}(dk). \tag{10}$$

First, consider the case of importance sampling. Denote

$$\widehat{G} = R_T \left(\int_{\Theta} f_k^2 \mathsf{Q}(dk) \right)^{\frac{1}{2}}, \tag{11}$$

then

$$\int_{\Theta} \mathbb{E}^{\vartheta} \widehat{C}_k^2(\vartheta) \mathsf{Q}(dk) = \mathbb{E}^{\vartheta} \left(R_T^2 \int_{\Theta} f_k^2 \mathsf{Q}(dk) (\rho_T^{\vartheta})^2 \right) = \mathbb{E}^{\vartheta} \left(\widehat{G} \rho_T^{\vartheta} \right)^2.$$

Introduce the martingale $\hat{\mu}_t = \mathbb{E}(\widehat{G} | \mathcal{F}_t)$.

Theorem 1. *There exist an $\mathbb{F}$-adapted process $\hat{\alpha}_t$ and an $\mathbb{F}$-predictable E-marked process $\hat{\beta}_t(y)$ such that*

$$d\hat{\mu}_t = \hat{\alpha}_t dW_t + \int_E \hat{\beta}_t(y) \tilde{p}(dt, dy). \tag{12}$$

The minimum of (9) equals $\left(\mathbb{E}\widehat{G}\right)^2$ and is attained when

$$\upsilon_t = \frac{\hat{\alpha}_t}{\hat{\mu}_t}, \quad \kappa_t(y) = \frac{\hat{\beta}_t(y)}{\hat{\mu}_{t-}}, \tag{13}$$

if the condition (6) for υ, κ holds.

Proof. The martingale $\hat{\mu}_t$ is square-integrable and using the martingale representation theorem (see [13], Chapter 2), we get that there exist processes $\hat{\alpha}_t$, $\hat{\beta}_t(y)$ such that the differential for $\hat{\mu}_t$ has the form (12). Define $\hat{\upsilon}_t$, $\hat{\kappa}_t(y)$ as in (13), then from Itô's lemma we have

$$\begin{aligned}
\ln \hat{\mu}_s &= \ln \hat{\mu}_0 + \int_0^s \frac{\hat{\alpha}_t}{\hat{\mu}_t} dW_t - \frac{1}{2} \int_0^s \frac{|\hat{\alpha}_t|^2}{\hat{\mu}_t^2} dt - \int_0^s \int_E \frac{\hat{\beta}_t(y)}{\hat{\mu}_{t-}} \nu(dt, dy) \\
&\quad + \int_0^s \int_E \ln \left(1 + \frac{\hat{\beta}_t(y)}{\hat{\mu}_{t-}} \right) p(dt, dy) \\
&= \int_0^s \hat{\upsilon}_t dW_t - \frac{1}{2} \int_0^s |\hat{\upsilon}_t|^2 dt \\
&\quad - \int_0^s \int_E \hat{\kappa}_t(y) \nu(dt, dy) + \ln \left(\prod_{n=1}^{N_s} (1 + \hat{\kappa}_{T_n}(Y_n)) \right).
\end{aligned}$$

Since $\hat{\beta}_{T_n}(Y_n) = \hat{\mu}_{T_n} - \hat{\mu}_{T_n-}$, we have $\hat{\kappa}_{T_n}(Y_n) = \hat{\mu}_{T_n} / \hat{\mu}_{T_n-} - 1$, and therefore for any $n \geq 1$ $\hat{\kappa}_{T_n}(Y_n) > -1$ $\mathbb{P} - a.s.$ Since $\hat{\upsilon}$, $\hat{\kappa}$ satisfy (6), $\mathbb{E} L_T^{\hat{\upsilon}, \hat{\kappa}} = 1$ and we can

construct the probability measure $\mathbb{P}^{\hat{\upsilon},\hat{\kappa}} = \mathbb{P}^{\hat{\vartheta}}$ with the density

$$\frac{d\mathbb{P}^{\hat{\vartheta}}}{d\mathbb{P}} = L_T^{\hat{\vartheta}} = \exp\left(\ln\hat{\mu}_t - \ln\hat{\mu}_0\right) = \frac{\widehat{G}}{\mathbb{E}\widehat{G}}.$$

Thus, we have $\min_{\vartheta}\mathbb{E}^{\vartheta}\left(\widehat{G}\rho_T^{\vartheta}\right)^2 \le \mathbb{E}^{\hat{\vartheta}}\left(\widehat{G}\rho_T^{\hat{\vartheta}}\right)^2 = \hat{\mu}_0^2$. Since for any ϑ

$$\mathbb{E}^{\vartheta}\left(\widehat{G}\rho_T^{\vartheta}\right)^2 \ge \left(\mathbb{E}^{\vartheta}\widehat{G}\rho_T^{\vartheta}\right)^2 = \hat{\mu}_0^2,$$

minimum (9) is equal to $\left(\mathbb{E}\widehat{G}\right)^2$.

Specifications of this result for different options are given in Section 3; their application to option valuation is described in Sections 4 and 5.

Now consider the case of control variates. Let $\bar{G} = R_T\int_\Theta f_k\mathsf{Q}(dk)$, then

$$\begin{aligned}\int_\Theta \mathbb{E}\bar{C}_k^2(\varphi)\mathsf{Q}(dk) &= \mathbb{E}\widehat{G}^2 + 2\mathbb{E}\bar{G}M_T^{\varphi} + \mathbb{E}\left(M_T^{\varphi}\right)^2\\ &= \mathbb{E}\widehat{G}^2 - \mathrm{Var}(\bar{G}) + \mathrm{Var}\left(\bar{G} + M_T^{\varphi}\right).\end{aligned} \tag{14}$$

Denote by $\bar{\mu}_t$ the martingale $\mathbb{E}(\bar{G}|\mathcal{F}_t)$.

Theorem 2. *The minimum of* (10) *equals* $\mathbb{E}\widehat{G}^2 - \mathrm{Var}(\bar{G})$ *and is attained when*

$$z_t = -\bar{\alpha}_t, \quad \zeta_t(y) = -\bar{\beta}_t(y) \tag{15}$$

where the $\mathbb{F}$*-adapted process* $\bar{\alpha}_t$ *and the* $\mathbb{F}$*-predictable* E*-marked process* $\bar{\beta}_t(y)$ *are such that* $d\bar{\mu}_t = \bar{\alpha}_t dW_t + \int_E \bar{\beta}_t(y)\tilde{p}(dt,dy)$.

Proof. Note that only the last term in (14) depends on $\varphi = \{z,\zeta\}$ and may be minimized. The martingale $\bar{\mu}_t$ is square-integrable, then using the martingale representation theorem (see [13], Chapter 2), we obtain that there exist an $\mathbb{F}$-adapted process $\bar{\alpha}_t$ and $\mathbb{F}$-predictable E-marked process $\bar{\beta}_t(y)$ such that

$$\mathbb{E}\int_0^T |\bar{\alpha}_t|^2 dt < \infty, \quad \mathbb{E}\int_0^T\int_E \bar{\beta}_t^2(y)q(dt,dy) < \infty$$

and

$$\bar{\mu}_t = \bar{\mu}_0 + \int_0^t \bar{\alpha}_t dW_t + \int_0^t\int_E \bar{\beta}_t(y)\tilde{p}(dt,dy).$$

If $z_t = -\bar{\alpha}_t$ and $\zeta_t(y) = -\bar{\beta}_t(y)$, then

$$\mathrm{Var}\left(\bar{G} + M_T^{z,\zeta}\right) = \mathrm{Var}(\bar{G} + \mu_0 - \mu_T) = \mathrm{Var}(\mu_0) = 0.$$

Note that if the measure Q is supported at one point, the minimal weighted variances for importance sampling and control variates methods are equal to zero. In

other cases the difference of the minimal weighted variances of these methods is equal to $\mathrm{Var}(\widehat{G}) - \mathrm{Var}(\bar{G})$ and depends on f_k, Q, Θ.

3 Application to the Options Valuation

Here specifications of general results in the case of a strike as a weighting parameter are given. Optimal functions which minimize the weighted variance are pointed out. Their approximations are given in Section 4 and used for the estimators' construction which reduce the weighted variance in Examples 1, 2.

The case of Asian options. Consider Asian options with the payoff function

$$f_K(S) = \phi_K(Y_T), \quad Y_T = \frac{1}{T}\int_0^T S_t dt,$$

where $\phi_K(x)$ is $(x-K)^+$ for a call option and $(K-x)^+$ for a put option. Let the strikes $K \in \Theta = [K_1, K_2]$.

Importance sampling. The functional $\widehat{G}$ defined in (11) has the form $\widehat{G} = R_T \widehat{H}(Y_T)$, where $\widehat{H}(x) = \left(\int_{K_1}^{K_2} \phi_K^2(x)\mathsf{Q}(dK)\right)^{\frac{1}{2}}$. Considering $X_t = (S_t, V_t, Y_t)$ as a 3-dimensional diffusion process, we obtain

$$\begin{aligned}\hat{\mu}_t &= \mathbb{E}(\widehat{G}|\mathcal{F}_t) = \mathbb{E}\left(R_t e^{-\int_t^T r_s ds} \widehat{H}\left(Y_t + \frac{1}{T}\int_t^T S_\tau d\tau\right)\middle| \mathcal{F}_t\right) \\ &= R_t \mathbb{E}\left(e^{-\int_t^T r_s ds} \widehat{H}\left(Y_t + \frac{1}{T}\int_t^T S_\tau d\tau\right)\middle| S_t, Y_t\right) = R_t u(t, S_t, V_t, Y_t),\end{aligned}$$

where

$$u(t,x,z,\zeta) = \mathbb{E}\left(e^{-\int_t^T r_s ds} \widehat{H}\left(\zeta + \frac{1}{T}\int_t^T S_\tau d\tau\right)\middle| S_t = x, V_t = z\right). \tag{16}$$

Assume that the function $u(t,x,z,\zeta)$ is smooth enough, then using Itô's lemma for $u(t, X_t) = u(t, S_t, V_t, Y_t)$, we get the representation of the $\mathbb{P}$-martingale $\hat{\mu}_t$ in the form

$$\begin{aligned}d\hat{\mu}_t &= R_t\left[\frac{\partial u}{\partial x}(t, X_t)\sigma(V_t)S_t + \frac{\partial u}{\partial z}(t, X_t)\theta_t\rho\right] dW_t^1 + \\ &+ R_t \frac{\partial u}{\partial z}(t, X_t)\theta_t\sqrt{1-\rho^2} dW_t^2 + R_t \int_E (\Gamma u(t, X_{t-}; y) - u(t, X_{t-}))\,\tilde{p}(dt, dy),\end{aligned}$$

where $\Gamma u(t, X_t; y) = u(t, S_t(1+\gamma(t, S_t, y)), V_t, Y_t)$. Thus, we obtain the representations for $\upsilon_t = (\upsilon_t^1, \upsilon_t^2)$, $\kappa(t, y)$ defined in (13):

$$\upsilon_t^1 = \frac{u'_x(t, X_t)\sigma(V_t)S_t + u'_z(t, X_t)\theta_t\rho}{u(t, X_t)},$$

$$\upsilon_t^2 = \frac{u'_z(t, X_t)\theta_t\sqrt{1-\rho^2}}{u(t, X_t)}, \quad \kappa_t(y) = \frac{\Gamma u(t, X_{t-}; y)}{u(t, X_{t-})} - 1. \tag{17}$$

Control variates. In the same way, we can obtain that the optimal processes $z_t = (z_t^1, z_t^2)$ and $\zeta(t, y)$ defined in (15) are given by

$$z_t^1 = -R_t\left(u'_x(t, X_t)\sigma(V_t)S_t + u'_z(t, X_t)\theta_t\rho\right), z_t^2 = -R_t u'_z(t, X_t)\theta_t\sqrt{1-\rho^2},$$
$$\zeta(t, y) = R_t\left(u(t, X_{t-}) - \Gamma u(t, X_{t-}; y)\right), \tag{18}$$

where

$$u(t, x, z, \zeta) = \mathbb{E}\left(e^{-\int_t^T r_s ds}\bar{H}\left(\zeta + \frac{1}{T}\int_t^T S_\tau d\tau\right)\middle| S_t = x, V_t = z\right), \tag{19}$$

the function $\bar{H}(x) = \int_{K_1}^{K_2}\phi_K(x)\mathsf{Q}(dK)$. Approximations of functions (16) and (19) are constructed in Section 4.

The case of Plain Vanilla options. Consider Plain Vanilla options with the pay-off function $\phi_K(S_T)$ and strikes $K \in \Theta = [K_1, K_2]$. In the same way as for Asian options we get that the optimal functions υ_t, $\kappa_t(y)$ for the importance sampling estimator $\widehat{C}_K(\upsilon, \kappa)$ are given in (17), where the process $X_t = (S_t, V_t)$ and

$$u(t, x, z) = \mathbb{E}\left(e^{-\int_t^T r_s ds}\widehat{H}(S_T)\middle| S_t = x, V_t = z\right). \tag{20}$$

For the control variates estimator $\bar{C}(z, \zeta)$ optimal z_t, $\zeta_t(y)$ are given in (18) with $X_t = (S_t, V_t)$ and

$$u(t, x, z) = \mathbb{E}\left(e^{-\int_t^T r_s ds}\bar{H}(S_T)|\, S_t = x, V_t = z\right). \tag{21}$$

The same method as for Asian options is suggested for the function $u(t, x, z)$ approximation in Section 4.

4 Approximations

Approximations of the function u in (16), (19) allow us to construct approximations of optimal functions in (17), (18). Note that the unbiasedness of the estimators (7), (8) doesn't depend on the accuracy of the optimal functions approximations. The function u can be approximated by different methods. We use in simulations the method described below.

Let us define for $\tau > t$, $x > 0$, the process

$$\widetilde{S}_{t,x}(\tau) = x\exp\{\tilde{\mu}(\tau - t) + \tilde{\sigma}(W_\tau - W_t)\}\prod_{n=N_t+1}^{N_\tau}(1 + \tilde{\gamma}(Y_n)), \tag{22}$$

where W_t is the one-dimensional Wiener process, N_t is the Poisson process with constant intensity $\tilde{\lambda}$, and $\tilde{m}(dy)$ is the distribution of marks, $\tilde{\mu} = \tilde{r} - 0.5\tilde{\sigma}^2 - \tilde{\lambda}\int_E \tilde{\gamma}(y)\tilde{m}(dy)$, the function $\tilde{\gamma}(y)$ depends only on y, and $\tilde{r}$, $\tilde{\sigma}$ are positive constants. Let $\tilde{S}_t = \tilde{S}_{0,S_0}(t)$. We approximate the optimal function $u(t,x,z,\zeta)$ in (16), (19) by an approximation of the function

$$\tilde{u}(t,x,z,\zeta) = e^{-\tilde{r}(T-t)}\mathbb{E}\widetilde{H}(\zeta + \frac{1}{T}\tilde{y}_{t,x}), \quad \tilde{y}_{t,x} = \int_t^T \widetilde{S}_{t,x}(\tau)d\tau,$$

where $\widetilde{H}$ is a piecewise linear approximation of $\bar{H}$ or $\widehat{H}$. The first and the second moment of $\tilde{y}_{t,x}$ can be calculated:

$$\mathbb{E}\tilde{y}_{t,x} = \frac{x}{r}\left(e^{\tilde{r}(T-t)} - 1\right), \quad E\tilde{y}_{t,x}^2 = 2x^2\int_0^{T-t}\int_u^{T-t} e^{r(u+v)}e^{Mu}dvdu, \qquad (23)$$

where $M = \tilde{\sigma}^2 + \tilde{\lambda}\int_B \tilde{\gamma}^2(y)\tilde{m}(dy)$ and in the similar way as in [8], the distribution of $\tilde{y}_{t,x}$ can be approximated by the log-normal distribution with the same first two moments. Thus, we can approximate the expectations $\mathbb{E}1_{[\alpha,\beta)}(\tilde{y}_{t,x})$, $\mathbb{E}\tilde{y}_{t,x}1_{[\alpha,\beta)}(\tilde{y}_{t,x})$, and get an analytic expression for the approximation of $\tilde{u}$. Note that the derivative of this approximation with respect to z is equal to null.

In the case of Plain Vanilla options the functions (20), (21) are approximated by $\tilde{u}(t,x,z) = e^{-\tilde{r}(T-t)}\mathbb{E}\widetilde{H}(\widetilde{Z}_{t,x}(T))$, where $\widetilde{H}$ is a piecewise linear function and the random variable $\widetilde{Z}_{t,x}(T)$ has log-normal distribution with the same first and second moments as $\widetilde{S}_{t,x}(T)$.

5 Simulation Results

Here we illustrate the efficiency of above estimators in simulation. The jump-diffusion stochastic volatility model (2), (3) is considered, where

$$\eta(t,V_t) = \xi(\eta - V_t),\ \theta(t,V_t) = \theta\sqrt{V_t},\ \sigma(t,V_t) = \sqrt{V_t},\ \bar{\gamma}(t,S_{t-},z) = e^z - 1.$$

That is, we consider the Heston model with jumps. The parameters of the Cox-Ingersoll-Ross process V_t are as follows: $\xi = 10$, $\eta = 0.04$, $\theta = 0.2$. The measure $m(dz)$ is the discrete distribution on $\{U, D\}$ and $m(U) = p$, $m(D) = 1 - p$, where $U = \ln(1.05)$, $D = \ln(0.9)$, $p = 0.7$. The intensity $\lambda = 5$, the correlation coefficient $\rho = -0.5$, the interest rate $r = 0.05$, and $\mu = r - \lambda(e^D + (e^U - e^D)p - 1)$. The initial stock price $S_0 = 100$ and $V_0 = 0.04$. In all examples the Euler scheme (for example, see [1]) is applied to simulate the process (S_t, V_t) with the discretization step equal to 0.001.

We calculate the relative computational costs as the ratio of computational times $R = T^{(1)}/T^{(0)}$, where $T^{(i)}$ are defined by formula (1); $T^{(0)}$ corresponds to the standard Monte Carlo and $T^{(1)}$ is a computational time of the estimators under consid-

eration. For the standard Monte Carlo, we assume $t_0^{(0)} = 0$, then the ratio is equal to

$$R = \frac{t_0^{(1)}}{T^{(0)}} + \frac{D^{(1)}}{D^{(0)}} \frac{t_1^{(1)}}{t_1^{(0)}}. \tag{24}$$

To present the results of simulation we introduce the following notations: $\mathcal{D} = \frac{D^{(1)}}{D^{(0)}}$, $\mathcal{T} = \frac{t_0^{(1)}}{T^{(0)}}$, $\tau = \frac{t_1^{(1)}}{t_1^{(0)}}$.

Example 1. Asian call options with strikes $K \in \Theta = [80, 120]$ and the maturity $T = 1$ are estimated. The measure Q is the uniform distribution on Θ. Divide the set Θ into four subsets $[80, 90)$, $[90, 100]$, $(100, 110]$, $(110, 120]$. The methods of importance sampling and control variates are applied for each subset separately. The estimators (7), (8) are constructed with the approximations of optimal functions from Section 4. We simulate 10^4 trajectories of the underlying asset price process. According to the formula (24) we calculate the relative computational costs R, the ratio of the weighted variances $\mathcal{D}$, and the value $\mathcal{T}$ for importance sampling estimators (subscript is) and control variates (subscript cv). The ratio of times of one trajectory simulation τ is not greater than 1.1 for importance sampling and 1.06 in the case of control variates. The results are given in Table 1.

Table 1 The comparison of importance sampling and control variates.

Set	$[80, 90)$	$[90, 100]$	$(100, 110]$	$(110, 120]$
R_{is}	0.147	0.152	0.150	0.138
R_{cv}	0.12	0.138	0.155	0.175
$\mathcal{D}_{\text{is}}$	0.035	0.036	0.023	0.012
$\mathcal{D}_{\text{cv}}$	0.007	0.018	0.031	0.047
$\mathcal{T}_{\text{is}}$	0.11	0.114	0.113	0.125
$\mathcal{T}_{\text{cv}}$	0.113	0.119	0.123	0.127

It's interesting to note that the importance sampling works better for the out-of-the-money options pricing and control variates technique - for the in-the-money options pricing. For the sets $[90, 100]$ and $(100, 110]$ the confidence limits of the Asian call option prices with the confidence level 0.99 are shown in Figure 1. The dashed lines "Initial" represent confidence limits of the standard Monte Carlo estimator, the solid lines "CV" represent confidence limits of the control variate estimator for $K \leq 100$, "IS" denotes the confidence limits of the importance sampling estimator for $K > 100$.

When we change the probability measure in the case of importance sampling, we change the drift coefficient of the underlying asset price process, the intensity and the distribution of marks. In this example we have discrete distributions of marks $m(dz)$ and its changing corresponds to the probability of upward jump $p = m(U)$ changing. If we apply importance sampling method for strikes from the set $[90, 100]$

with constant probability of upward jump $p = 0.7$, the weighted variance is reduced 3 times. But if we change p, the weighted variance is reduced 21 times and the graph of the probability-time is shown in Figure 2.

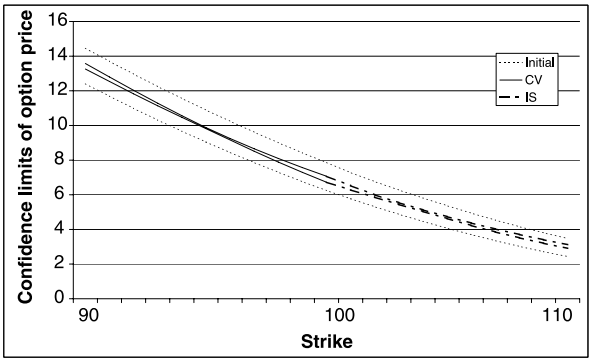


Fig. 1 Asian call option.

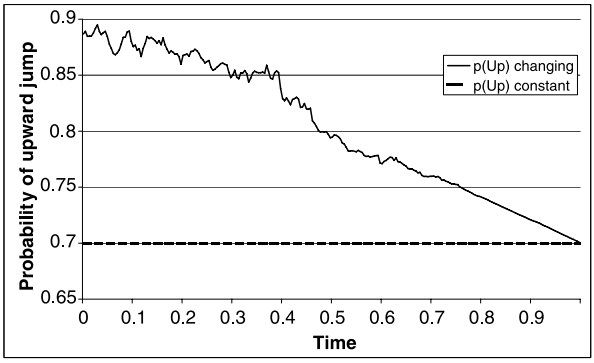


Fig. 2 Probability of upward jump.

Example 2. Consider Plain Vanilla options with the exercise date $T = 1$ and strikes $K \in \Theta = \{98 + i\}_{i=0}^{10}$. Let the measure Q be the measure of the uniform distribution on Θ. We simulate 10^5 trajectories of the underlying asset. The method of importance sampling reduces the weighted variance 88–fold ($\mathcal{D}_{\text{is}} = 0.011$) with the relative computational costs $R_{\text{is}} = 0.014$. The value $\mathcal{T}_{\text{is}} = 0.0015$ and the ratio of times of one trajectories simulation $\tau_{\text{is}} = 1.09$. The control variates estimators reduce the weighted variance 83–fold ($\mathcal{D}_{\text{cv}} = 0.012$) and $R_{\text{cv}} = 0.0145$, $\mathcal{T}_{\text{cv}} = 0.0017$, $\tau_{\text{cv}} = 1.06$.

Since this example is quite simple, we can use it to compare the effectiveness of described methods with a simpler one from [6], Example 4.1.4. Let the estimator

for C_K have the form

$$R_T f_K(S_T) - e^{-\tilde{r}T}\left(f_K\left(\tilde{S}_T\right) - \mathbb{E} f_K\left(\tilde{S}_T\right)\right),$$

where $\tilde{S}_T$ is defined in (22). Trajectories of the process $\tilde{S}_t$ and S_t are simulated using the same realization of the random variables. Note that the variance reduction for this estimator is possible because we can construct the process $\tilde{S}_t$ which is on the one hand well correlated with S_t and on another hand is simple enough to calculate the expectation analytically. This method reduce the weighted variance 78 times ($\mathcal{D} = 0.0128$), the relative computational costs $R = 0.0129$, $\tau = 1.01$.

We don't expect that our methods are more effective for simple models than some simpler methods; but as we see, their efficiencies are similar.

Acknowledgements This research was supported by RFBR under the grant number 08-01-00194.

References

1. N. Bruti-Liberati and E. Platen. Strong approximations of stochastic equations with jumps. *Journal of Computational and Applied Mathematics*, 205(2):982–1001, 2007.
2. R. Cont and P. Tankov. *Financial Modelling with Jump Processes*. Chapman & Hall/CRC, Boca Raton, 2004.
3. S. M. Ermakov. *The Monte Carlo method and related topics*. Nauka, Moscow, 1975.
4. J. P. Fouque and C. H. Han. Variance reduction for Monte Carlo methods to evaluate option prices under multi-factor stochastic volatility models. *Quantitative Finance*, 4(5):597–606, 2004.
5. J. P. Fouque and T. A. Tullie. Variance reduction for Monte Carlo simulation in a stochastic volatiltiy environment. *Quantitative Finance*, 2(1):24–30, 2002.
6. P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2004.
7. P. Glasserman and N. Merener. Numerical solution of jump-diffusion LIBOR market models. *Finance and Stochastic*, 7(1):1–27, 2003.
8. A. A. Gormin and Y. N. Kashtanov. The weighted variance minimization for options pricing. *Monte Carlo Methods and Applications*, 13(5–6):333–351, 2007.
9. S. Lindset and A-C. Lund. A Monte Carlo approach for the American put under stochastic interest rates. *Journal of Economics Dynamics and Control*, 31(4):1081–1105, 2007.
10. Nigel J. Newton. Variance reduction for simulated diffusions. *SIAM Journal on Applied Mathematics*, 54(6):1780–1805, 1994.
11. C. Chiarella, C. Nikitopoulos and E. Schlogl. A control variate method for Monte Carlo simulations of Heath-Jarrow-Morton models with jumps. *Applied Mathematical Finance*, 14(5):365–399, 2007.
12. J.-P. Foque, G. Papanicolaou and K. R. Sircar. *Derivatives in Financial Markets with Stochastic Volatility*. Cambridge University Press, Cambridge, 2000.
13. R. Situ. *Theory of Stochastic Differential Equations with Jumps and Applications*. Springer-Verlag, New York, 2005.
14. Z. Zhu and F. B. Hanson. A Monte-Carlo option-pricing algorithm for log-uniform jump-diffusion model. *Proceedings of Joint 44nd IEEE Conference on Decision and Control and European Control Conference*, pages 5221–5226, 2005.

(t, m, s)-Nets and Maximized Minimum Distance, Part II

Leonhard Grünschloß and Alexander Keller

Abstract The quality parameter t of (t, m, s)-nets controls extensive stratification properties of the generated sample points. However, the definition allows for points that are arbitrarily close across strata boundaries. We continue the investigation of (t, m, s)-nets under the constraint of maximizing the mutual distance of the points on the unit torus and present two new constructions along with algorithms. The first approach is based on the fact that reordering (t, s)-sequences can result in $(t, m, s+1)$-nets with varying toroidal distance, while the second algorithm generates points by permutations instead of matrices.

1 Introduction

Important problems in image synthesis like e.g. anti-aliasing, hemispherical integration, or the illumination by area light sources can be considered as low-dimensional numerical integration problems. Among the most successful approaches to computing this kind of integrals are quasi-Monte Carlo and randomized quasi-Monte Carlo methods [9, 13, 2] based on the two-dimensional Larcher-Pillichshammer points [10], which expose a comparatively large minimum toroidal distance.

These points belong to the class of (t, m, s)-nets in base q (for an extensive overview of the topic we refer to [12, Ch. 4, especially p. 48]), which is given by

Leonhard Grünschloß
mental images GmbH, Fasanenstraße 81, 10623 Berlin, Germany
e-mail: leonhard@gruenschloss.org

Alexander Keller
mental images GmbH, Fasanenstraße 81, 10623 Berlin, Germany
e-mail: alex@mental.com

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_25,

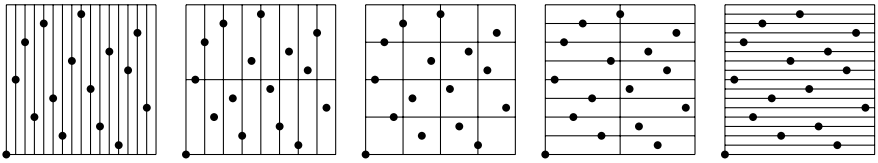


Fig. 1 The Larcher-Pillichshammer points as an example of a $(0, 4, 2)$-net in base 2 superimposed on all possible elementary intervals. There is exactly one point in each elementary interval.

Definition 1. For integers $0 \leq t \leq m$, a (t, m, s)-*net in base* q is a point set of q^m points in $[0, 1)^s$ such that there are exactly q^t points in each elementary interval E with volume q^{t-m}.

Figure 1 shows an instance of a $(0, 4, 2)$-net in base 2 and illustrates the concept of *elementary intervals*

$$E = \prod_{i=1}^{s} \left[a_i q^{-b_i}, (a_i + 1) q^{-b_i} \right) \subseteq [0, 1)^s$$

as used in the definition, where $a_i, b_i \in \mathbb{Z}$, $b_i \geq 0$ and $0 \leq a_i < q^{b_i}$. The concept of (t, m, s)-nets can be generalized for sequences of points as given by

Definition 2. For an integer $t \geq 0$, a sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots$ of points in $[0, 1)^s$ is a (t, s)-*sequence in base* q if, for all integers $k \geq 0$ and $m > t$, the point set $\mathbf{x}_{kq^m}, \ldots, \mathbf{x}_{(k+1)q^m-1}$ is a (t, m, s)-net in base q.

Obviously, the stratification properties of (t, m, s)-nets and (t, s)-sequences are best for the quality parameter $t = 0$, because then every elementary interval contains exactly one point, as illustrated in Figure 1. The conception does not consider the mutual distance of the points, which allows points to lie arbitrarily close together across shared interval boundaries.

In this paper we continue previous work from [6] that used the *minimum toroidal distance*

$$d_{\min}(\{\mathbf{x}_0, \ldots, \mathbf{x}_{N-1}\}) := \min_{0 \leq u < v < N} \|\mathbf{x}_u - \mathbf{x}_v\|_T$$

to classify point sets $\{\mathbf{x}_0, \ldots, \mathbf{x}_{N-1}\}$, where the toroidal distance of two points $\mathbf{x} = (x_1, \ldots, x_s) \in [0, 1)^s$ and $\mathbf{y} = (y_1, \ldots, y_s) \in [0, 1)^s$ is defined as

$$\|\mathbf{x} - \mathbf{y}\|_T := \sqrt{\sum_{i=1}^{s} (\min\{|x_i - y_i|, 1 - |x_i - y_i|\})^2}.$$

Maximizing the shift-invariant measure of toroidal distance further increases uniformity, allows one to tile the resulting point sets, and to consider periodic integrands, which is especially useful in the aforementioned graphics applications.

In Section 2 we therefore extend the construction of Larcher and Pillichshammer to $s = 3$ dimensions resulting in a $(0, m, 3)$-net in base 2, which exhibits a large

minimum toroidal distance. In Section 3 we then present a new permutation-based construction for $(0, m, 2)$-nets in base 2, which often have a larger minimum toroidal distance than can be obtained by any digital net in base 2.

2 A New $(0, m, 3)$-Net in Base 2 with Large Minimum Toroidal Distance

Although both the Hammersley and Larcher-Pillichshammer points are constructions of $(0, m, 2)$-nets in base 2, the latter have a much larger minimum toroidal distance [8]. In fact already in [10, 11] the Hammersley points have been identified to be the worst construction with respect to certain other measures, too.

The extension of the Larcher-Pillichshammer points to $s = 3$ dimensions has been an open problem for long. In the following, we present a construction of a $(0, m, 3)$-net in base 2, where one two-dimensional projection equals the Larcher-Pillichshammer points [9] and therefore benefits from their large minimum toroidal distance.

2.1 Digital Nets and Sequences

We briefly summarize necessary notation and algorithmic facts on digital nets and sequences from [12] in a simplified manner. While in general digital nets and sequences are defined using a finite field $\mathbb{F}_q$, with q being a prime power, we only consider the case $q = 2$ in the following.

Given suitable $m \times m$-matrices $C_1, \ldots, C_s$ over $\mathbb{F}_2$, the i-th component of the n-th point for $1 \le i \le s$ and $0 \le n < 2^m$ can be generated by

$$x_n^{(i)} = \begin{pmatrix} 2^{-1} \\ \vdots \\ 2^{-m} \end{pmatrix}^T \left[C_i \begin{pmatrix} d_0(n) \\ \vdots \\ d_{m-1}(n) \end{pmatrix} \right] \in [0, 1),$$

where the matrix-vector multiplication has to be performed in $\mathbb{F}_2$ and the digits $d_k(n)$ are defined by the binary expansion of

$$n = \sum_{k=0}^{m-1} d_k(n) 2^k.$$

The elements of the i-th generator matrix are denoted by $c_{j,r}^{(i)}$.

The theoretical construction of (t, s)-sequences requires generator matrices of infinite size. However, in practice this does not pose a problem when enumerating the points, since $d_k(n) = 0$ for all sufficiently large k and, in addition, we require the

matrix entries $c_{j,r}^{(i)} = 0$ for all sufficiently large j [12, p. 72, (S6)]. Therefore, only finite upper left submatrices have to be considered when generating points. Finally, due to the finite precision of integer computation, m is finite in any case.

Based on the results in [10, 11, 4], we will consider the $(0, 1)$-sequence in base 2 generated by the non-singular upper triangular matrix

$$C_1' := \left(\begin{cases} 1 & \text{if } j \le r, \\ 0 & \text{otherwise} \end{cases} \right)_{j,r=0}^{\infty} \tag{1}$$

in order to generate the Larcher-Pillichshammer points [9].

2.2 Reordering the Sobol'-Sequence

We now take a look at the first two components of the Sobol'-sequence [15], which are a $(0, 2)$-sequence in base 2, and are defined by the infinite upper triangular matrices

$$C_1 := \left(\delta_{j,r}\right)_{j,r=0}^{\infty} \quad \text{and} \quad C_2 := \left(\binom{r}{j} \bmod 2 \right)_{j,r=0}^{\infty},$$

where $\delta_{j,r} = 1$ for $j = r$ and zero otherwise. In the above definition and the remainder of this work, by $a \bmod m$ we mean the *common residue*. That is the nonnegative value $b < m$, such that $a \equiv b \mod m$.

Suppose that the infinite matrices $C_1, \ldots, C_s$ over $\mathbb{F}_2$ generate a $(0, s)$-sequence in base 2. Furthermore suppose that the infinite matrix C_1' over $\mathbb{F}_2$ generates a $(0, 1)$-sequence in base 2. This implies that C_1' is a nonsingular upper triangular matrix. Then a $(0, s)$-sequence in base 2 is generated by $C_1', C_2 D, \ldots, C_s D$, where $D := C_1^{-1} C_1'$. This new sequence consists of the same points as before, however, in a different order. This result is proven in more general form in [5, Prop. 1].

Note that the first generator matrix C_1' is the one of Larcher-Pillichshammer. Since C_1 is the identity, we have $D = C_1^{-1} C_1' = C_1'$ and obtain

$$\begin{aligned} C_2 D &= \left(\binom{r}{j} \bmod 2 \right)_{j,r=0}^{\infty} C_1' \\ &= \left(\sum_{k=0}^{r} \binom{k}{j} \bmod 2 \right)_{j,r=0}^{\infty} \\ &= \left(\binom{r+1}{j+1} \bmod 2 \right)_{j,r=0}^{\infty} =: C_2', \end{aligned} \tag{2}$$

as the second generator matrix, where the last equality follows from the Christmas Stocking Theorem.

Note that the same reordering can be applied to the Faure-sequence [3], which can be regarded as a generalization of the Sobol'-sequence for any prime base q. An even more general construction for any prime power base can be found in [12].

2.3 Construction

Combining the component $\frac{n}{2^m}$ with the first 2^m points of a $(0, 2)$-sequence in base 2 yields a $(0, m, 3)$-net in base 2 (see [12, Lemma 4.22, p. 62]). The relationship of C_1' and C_2' as expressed in Equation (2) in fact is a property of all $(0, 2)$-sequences in base 2 generated by non-singular upper triangular matrices [7, Prop. 4]. Consequently fixing the first generator matrix uniquely determines the second generator matrix and thus the $(0, m, 3)$-net.

In the previous section we reordered the Sobol'-sequence in base 2 such that one component matches the $(0, 1)$-sequence as defined in [10]. For the construction of the new $(0, m, 3)$-net in base 2, it is sufficient to consider the matrices C_1', C_2' from Section 2.2, because for $s = b = 2$ Sobol's construction is identical to Faure's construction. By construction two dimensions of the resulting three-dimensional net are the Larcher-Pillichshammer points [9].

In Figure 2 we compared the minimum toroidal distance of the new construction to the one that uses the original Sobol' generator matrices C_1 and C_2. Except for $m = 5$ and $m = 6$, the new construction is by far superior for all $m \leq 22$. We did not compute the minimum toroidal distance for $m > 22$, though.

2.3.1 Implementation

The following code in `C99` (the current ANSI standard of the `C` language) returns the n-th point of the $(0, m, 3)$-net generated using C_1' and C_2' (see Equations (1) and (2)) for $m < 32$ in $\mathcal{O}(m)$. The vectorized implementation is based on the fact that addition in $\mathbb{F}_2$ corresponds to the *exclusive or* operation.

```
void x_n(unsigned int n, const unsigned int m, float x[3]) {
  // first component: n / 2^m
  x[0] = (float) n / (1U << m);

  // remaining components by matrix multiplication
  unsigned int r1 = 0, r2 = 0;
  for (unsigned int v1 = 1U << 31, v2 = 3U << 30; n; n >>= 1) {
    if (n & 1) { // vector addition of matrix column by XOR
      r1 ^= v1;
      r2 ^= v2 << 1;
    }
    // update matrix columns
    v1 |= v1 >> 1;
    v2 ^= v2 >> 1;
  }
```

```
  // map to unit cube [0,1)^3
  x[1] = r1 * (1.f / (1ULL << 32));
  x[2] = r2 * (1.f / (1ULL << 32));
}
```

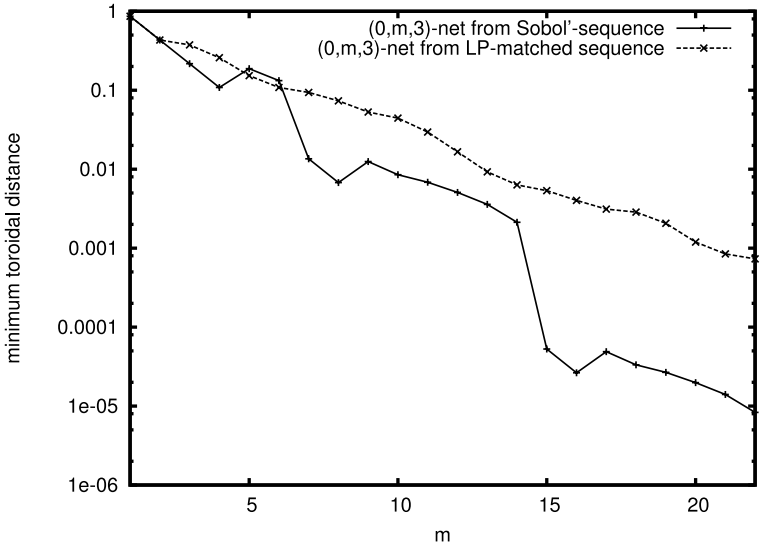


Fig. 2 A plot of the toroidal distance in $[0,1)^3$ for both the Sobol' $(0,m,3)$-net as well as the $(0,m,3)$-net constructed using C_1', C_2', where one of the two-dimensional projections equals the Larcher-Pillichshammer point set. We would like to stress that the minimum distance is scaled logarithmically, so the absolute difference is very significant for increasing m. In contrast to the Sobol'-net, the minimum distance decreases smoothly for the new construction.

3 Permutation-Generated $(0, m, 2)$-Nets in Base 2 with Larger Minimum Toroidal Distance

Taking the integer part of the coordinates of a $(0,m,s)$-net in base q multiplied by the number of points q^m results in each component being a permutation from the symmetric group S_{q^m}. For $s=2$ dimensions and base $q=2$ this is visualized in Figure 1, where each column and each row (the leftmost and rightmost set of elementary intervals in the figure) contain exactly one point.

Using Heap's efficient permutation generation algorithm [14] allows one to enumerate all permutations $\pi : \{0,\dots,2^m-1\} \to \{0,\dots,2^m-1\}$ that represent $(t,m,2)$-nets in base 2 with the points $\frac{1}{2^m}(n,\pi(n)) \in [0,1)^2$. We extended this basic backtracking algorithm by pruning the search tree whenever the elementary-interval property $t=0$ was violated or already a net with larger minimum toroidal distance

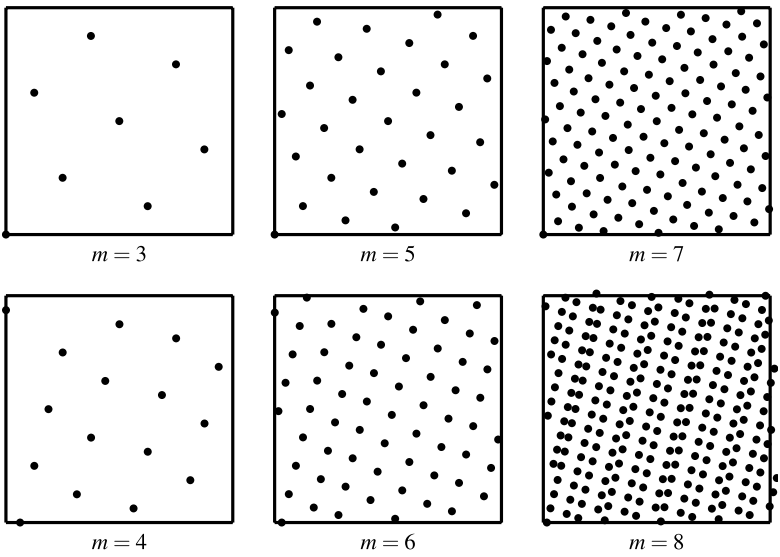


Fig. 3 Examples of the permutation-generated $(0, m, 2)$-nets in base 2 for $m = 3, \ldots, 8$.

than the current one had been found. Verifying the $t = 0$ property is simple using the code published in [6, Section 2.2].

Performing such an exhaustive combinatorial computer search seems hopeless given the number of possible permutations $|S_{2^m}| = (2^m)!$ and in fact we did not succeed in running an exhaustive search for $m > 5$. In these cases we simply generated random permutations, but kept the backtracking approach mentioned above.

However in [6], we were able to find very regular looking $(0, 5, 2)$-nets with a minimum toroidal distance of $\sqrt{32}/2^5 \approx 0.17677670$, which is larger than the minimum toroidal distance of all possible digital $(0, 5, 2)$-nets, where the maximum is $\sqrt{29}/2^5 \approx 0.1682864$.

In continuation of these findings we now present a first construction (see Figure 3) of such permutation-based nets. In Table 1 we compare the minimum toroidal distance of different $(0, m, 2)$-nets in base 2. The new permutation construction clearly features the largest minimum toroidal distance. Due to the novelty of the approach, we present two different derivations and provide more interpretations than usually necessary.

3.1 Iterative Construction by Quadrupling Point Sets

Given the search results as displayed in Figure 3, an iterative construction procedure can be inferred. This procedure repeatedly quadruples an initial point set until the

Table 1 Comparison of the Hammersley points, the Larcher-Pillichshammer points, the points resulting from the optimized matrices given in [6], and the new permutation construction with respect to minimum toroidal distance. Note that for easier comparison all distances have been multiplied by the number 2^m of points of a $(0,m,2)$-net in base 2 and squared afterwards.

m	Hammersley	Larcher-Pillichsh.	Opt. matrices	Permutation
2	2	2	2	2
3	2	5	8	8
4	2	8	13	13
5	2	18	29	32
6	2	32	52	53
7	2	72	100	128
8	2	128	208	241
9	2	265	400	512
10	2	512	832	964
11	2	1060	1600	2048
12	2	2048	3328	3856
13	2	4153	6385	8192
14	2	8192	13312	15424
15	2	16612	25313	32768
16	2	32768	53248	61696

desired number of 2^m points is reached. We therefore need to distinguish the cases of even and odd m as depicted by the examples in Figure 4.

We represent the $(0,k,2)$-nets in base 2 by

$$\left\{\frac{1}{2^k}(u \bmod 2^k, v) : (u,v) \in P_k\right\}, \tag{3}$$

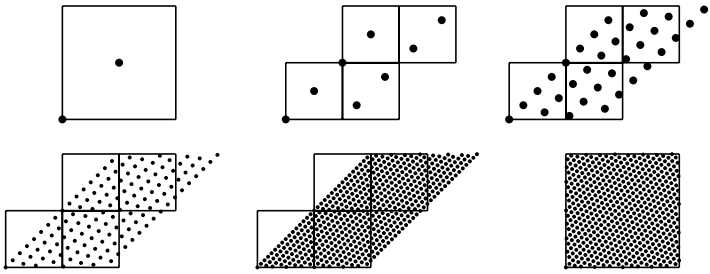
where the set P_k contains 2^k integer coordinates (u,v).

For odd m the initial point set P_1 consists of the points $(0,0)$ and $(1,1)$. The quadrupling rule to construct a point set $P_{k+2} := P_{k,0} \cup P_{k,1} \cup P_{k,2} \cup P_{k,3}$ with 2^{k+2} points from a point set P_k with $|P_k| = 2^k$ points is as follows:

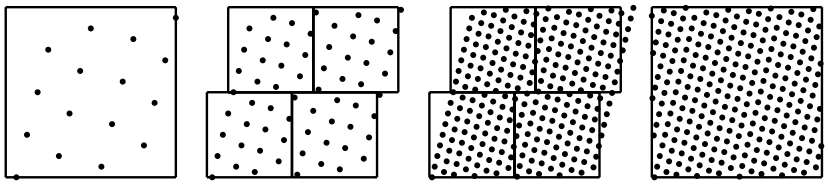
$$\begin{aligned}
\text{Lower left:} \quad & P_{k,0} := \{(2u, 2v) : \ (u,v) \in P_k\} \\
\text{Lower right:} \quad & P_{k,1} := \left\{(2^{k+1}+2u+1, 2v+1) : \ (u,v) \in P_k\right\} \\
\text{Upper left:} \quad & P_{k,2} := \left\{(2^{k+1}+2u, 2^{k+1}+2v) : \ (u,v) \in P_k\right\} \\
\text{Upper right:} \quad & P_{k,3} := \left\{(2^{k+2}+2u+1, 2^{k+1}+2v+1) : \ (u,v) \in P_k\right\}
\end{aligned}$$

For even m, the iterative construction is more involved. We start out with four diagonals, each consisting of four points:

$$\begin{aligned}
P_{2,0} &:= \{(w+1, 4w) : \ w = 0, \ldots, 3\}, \\
P_{2,1} &:= \{(w+5, 4w+2) : \ w = 0, \ldots, 3\}, \\
P_{2,2} &:= \{(w+9, 4w+1) : \ w = 0, \ldots, 3\},
\end{aligned}$$



a) Example for odd m: Construction of a $(0,9,2)$-net in base 2.



b) Example for even m: Construction of a $(0,8,2)$-net in base 2.

Fig. 4 Iterative construction of $(0,m,2)$-nets in base 2 by quadrupling point sets, where the final net results from wrapping the points such that they match the unit square. This wrapping corresponds to the modulo operation in Equation (3).

$$P_{2,3} := \{(w+13, 4w+3) : \ w = 0, \ldots, 3\}.$$

The quadrupling rule for even k constructs $P_{k+2} := P'''_{k,0} \cup P'''_{k,1} \cup P'''_{k,2} \cup P'''_{k,3}$ from sets $P_{k,l}$ for $l = 0, \ldots, 3$ as follows:

1. Multiply all points by two and add an offset for the points in P_1 and P_3:

$$P'_{k,l} := \left\{(2u + (l \bmod 2), 2v) : \ (u,v) \in P_{k,l}\right\} \text{ for } l = 0, \ldots, 3.$$

2. Extend the diagonals:

$$P''_{k,l} := P'_{k,l} \cup \left\{(2^{k-1}+u, 2^{k+1}+v) : \ (u,v) \in P'_{k,l}\right\} \text{ for } l = 0, \ldots, 3.$$

3. Combine diagonals:

$$P'''_{k,0} := P''_{k,0} \cup P''_{k,1},$$
$$P'''_{k,1} := P''_{k,2} \cup P''_{k,3}.$$

4. Append shifted copies:

$$P'''_{k,2} := \left\{(2^{k+1}+u, u+1) : \ (u,v) \in P'''_{k,0}\right\},$$
$$P'''_{k,3} := \left\{(2^{k+1}+v, v+1) : \ (u,v) \in P'''_{k,1}\right\}.$$

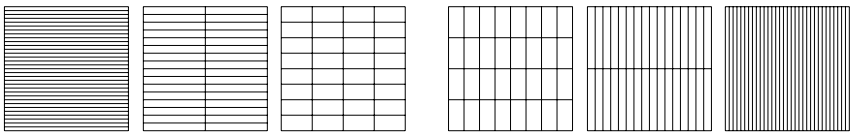


Fig. 5 The two cases in the proof corresponding to certain kinds of elementary intervals, illustrated here for $m = 5$.

3.2 Direct Construction

As these nets consist of points lying on parallel diagonals similar to rank-1 lattices [2], a direct construction is possible by computing a single point on each diagonal. Since the points are evenly spaced along the diagonals, the remaining points follow immediately. These diagonals are to be understood with a modulo 2^m wrap-around for the first coordinate.

3.2.1 Construction for Odd *m*

The position of the point with the smallest second coordinate on the d-th diagonal is given by $(2^m\phi_2(d)+d, d)$, where $0 \le d < 2^{\lfloor \frac{m}{2} \rfloor}$ and $\phi_2(d) \in [0,1)$ denotes the van der Corput radical inverse in base 2.

Theorem 1. *For odd m, the points*

$$\left\{(u_{d,w}, v_{d,w}) : 0 \le d < 2^{\lfloor \frac{m}{2} \rfloor}, 0 \le w < 2^{\lceil \frac{m}{2} \rceil}\right\},$$

where

$$u_{d,w} = \left(2^m\phi_2(d) + d + w \cdot 2^{\lfloor \frac{m}{2} \rfloor}\right) \bmod 2^m, \tag{4}$$

$$v_{d,w} = d + w \cdot 2^{\lfloor \frac{m}{2} \rfloor}, \tag{5}$$

constitute a $(0, m, 2)$*-net in base* 2 *with a minimum toroidal distance of* $\sqrt{2^m}$*. This distance is measured using components multiplied by* 2^m*, i.e. on integer scale.*

Proof. First, we will show that the elementary interval property holds. For each kind $0 \le h \le m$ of elementary intervals, there must be exactly one point in each integer-scaled elementary interval

$$\left[x \cdot 2^h, (x+1)2^h\right) \times \left[y \cdot 2^{m-h}, (y+1)2^{m-h}\right),$$

where $0 \le x < 2^{m-h}$ and $0 \le y < 2^h$. Partitioning the set of elementary intervals as depicted in Figure 5, we need to consider two cases:

1. We first consider the case $\lceil \frac{m}{2} \rceil \leq h \leq m$. The width of these kinds of elementary intervals is larger than their height. Looking at Equation (5), we can see that the $\lfloor \frac{m}{2} \rfloor$ least significant bits of $v_{d,w}$ are equal to the bits of d, while the remaining $\lceil \frac{m}{2} \rceil$ most significant bits of $v_{d,w}$ are equal to the bits of w. Considering one horizontal strip of elementary intervals, y determines the h most significant bits of $v_{d,w}$. Since $h \geq \lceil \frac{m}{2} \rceil$, w is completely determined by y. What remains to show is that for this fixed w, the $m-h$ most significant bits of $u_{d,w}$ differ for suitable values for d. Analyzing Equation (4), we see that the term $w \cdot 2^{\lfloor \frac{m}{2} \rfloor}$ is constant, while the addition of d only modifies the $\lfloor \frac{m}{2} \rfloor$ least significant bits. However

$$\left\{2^m \phi_2(d) : \ 0 \leq d < \left\lfloor \frac{m}{2} \right\rfloor\right\} = \left\{d \cdot \left\lceil \frac{m}{2} \right\rceil : \ 0 \leq d < \left\lfloor \frac{m}{2} \right\rfloor\right\}, \tag{6}$$

   thus due to the addition of $2^m \phi_2(d)$ it is possible to guarantee that the $\lfloor \frac{m}{2} \rfloor \geq m-h$ most significant bits of $u_{d,w}$ are different by selecting suitable values for d. Thus the corresponding points fall into different elementary intervals.
2. Now we consider the case $0 \leq h \leq \lfloor \frac{m}{2} \rfloor$. The width of these kinds of elementary intervals is smaller than their height. We consider one vertical strip of elementary intervals of width 2^h. Looking at Equation (4), we can see that the the only way to achieve consecutive values $u_{d,w}$ equal to $x, x+1, \ldots, x+2^h-1$ is by using consecutive values for d. That is because the terms $2^m \phi_2(d)$ and $w \cdot 2^{\lfloor \frac{m}{2} \rfloor}$ do not modify the $h \leq \lfloor \frac{m}{2} \rfloor$ least significant bits of $u_{d,w}$. However 2^h consecutive values for d mean that the $h \leq \lfloor \frac{m}{2} \rfloor$ most significant bits of the values $2^m \phi_2(d)$ are different for each d (cf. to Equation (6)). In order to stay in the vertical strip of elementary intervals determined by x, the h most significant bits of w thus must be chosen accordingly for each point inside this strip in order to "compensate" using the term $w \cdot 2^{\lfloor \frac{m}{2} \rfloor}$. As a consequence of the different values for w it follows from Equation (5) that the $h \leq \lceil \frac{m}{2} \rceil$ most significant bits of $v_{d,w}$ are different for each point inside this strip, thus they fall into different elementary intervals.

We now consider the achieved minimum toroidal distance of the point set. Points on the diagonals are placed with a multiple of the offset $\left(2^{\lfloor \frac{m}{2} \rfloor}, 2^{\lfloor \frac{m}{2} \rfloor}\right)$, so their squared minimum toroidal distance to each other is

$$2 \cdot \left(2^{\lfloor \frac{m}{2} \rfloor}\right)^2 = 2^m.$$

The squared distance between the diagonals with slope 1 is

$$2 \cdot \left(\frac{2^{\lceil \frac{m}{2} \rceil}}{2}\right)^2 = 2^m.$$

In conclusion, the minimum toroidal distance is $\sqrt{2^m}$. As the diagonals can be tiled seamlessly, the result is identical for the toroidal distance measure. On the

unit square $[0,1)^2$, we need to divide the point coordinates by 2^m, so the minimum toroidal distance on the unit square is $\frac{\sqrt{2^m}}{2^m} = \sqrt{2^{-m}}$, which concludes the proof. □

The theorem allows for an additional interpretation of the structure: Using Equation (4), we can solve for d and w given $u_{d,w}$:

$$d = u_{d,w} \bmod 2^{\lfloor \frac{m}{2} \rfloor},$$
$$w = \frac{(u_{d,w} - 2^m \phi_2(d) - d) \bmod 2^m}{2^{\lfloor \frac{m}{2} \rfloor}}.$$

Inserting these equations into Equation (5) yields

$$\begin{aligned} v_{d,w} &= d + \left(u_{d,w} - 2^m \phi_2\left(u_{d,w} \bmod 2^{\lfloor \frac{m}{2} \rfloor}\right) - d\right) \bmod 2^m \\ &= \left(u_{d,w} - 2^m \phi_2\left(u_{d,w} \bmod 2^{\lfloor \frac{m}{2} \rfloor}\right)\right) \bmod 2^m, \end{aligned} \tag{7}$$

which can be regarded as replicating a $(0, \lfloor \frac{m}{2} \rfloor, 2)$ integer Hammersley-net $2^{\lceil \frac{m}{2} \rceil}$ times horizontally, scaling each one vertically by $-2^{\lceil \frac{m}{2} \rceil}$ and finally adding a linear component to the combination of nets. The resulting points are wrapped around the unit square.

3.2.2 Construction for Even *m*

For even $m \leq 6$ the permutation search did not reveal an improvement of the minimum toroidal distance in comparison to the full matrix search. We would like to note that the permutation search for $m = 6$ did not finish, though. However, when allowing a distance measure that accounts for a slightly irregular tiling, an approach very much resembling the previous construction can be taken, yielding the nets shown in Figure 3.

Using a tiling where each row is shifted by $(2^{m-2}, 0)$ relative to its adjacent row below (see Figure 6), we can get a seamless tiling using $2^{\frac{m}{2}}$ diagonals, each with $2^{\frac{m}{2}}$ points. Again, the points are spaced evenly on each diagonal, this time by the offset vector $(2^{\frac{m}{2}-2}, 2^{\frac{m}{2}})$. The position of the point with the smallest second coordinate on the d-th diagonal, where $d \in \left\{0, \ldots, 2^{\frac{m}{2}} - 1\right\}$, is given by $(2^m \phi_2(d) + \lfloor \frac{d}{4} \rfloor + 2^{\frac{m}{2}-2}, d)$. With the modified tiling described above, these diagonals are continued seamlessly.

These nets cannot be generated via the classical way of using generator matrices as the point $(0,0)$ is not included.

While we believe that a similar proof for the $t = 0$ is possible as for the odd m case, we only verified the $t = 0$ property using a computer program for all even $m \leq 22$. The minimum toroidal distance equals $\sqrt{241}$ for $m = 8$ and $\sqrt{241 \cdot 2^{m-8}}/2^m$ for all even m where $10 \leq m \leq 22$. See Table 1 for minimum toroidal distance values for $m < 8$. We would like to stress that the modified minimum toroidal distance

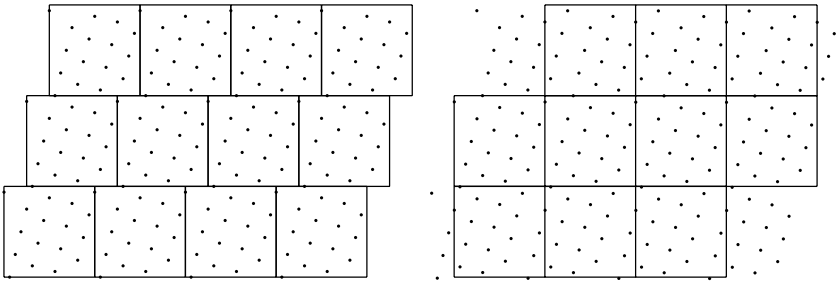


Fig. 6 On the left, the modified tiling for the permutation construction is shown: m is even and each row of the tiling is shifted by $(2^{m-2}, 0)$ relative to its adjacent row below. Using an axis aligned tiling on the right reveals that all tiles in a row share the same sample pattern, while the pattern differs by row by a modulo wrap-around in direction of the x-axis.

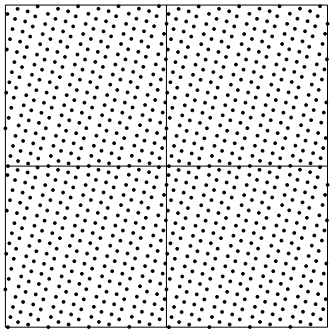


Fig. 7 By taking 4×4 modulo-wrapped copies of the same pattern (see Figure 6), we obtain a larger pattern that is periodic and matches the pixel arrangement. It can be tiled regularly without a decrease in minimum toroidal distance.

measure that respects the shifts of the rows in the tiling has been used for these measurements.

Considering a superimposed axis-aligned tiling as illustrated in Figure 6 reveals that in a row each tile contains the same set of samples, while the set of samples varies by row. This in turn allows for constructing a larger pattern that consists of shifted copies of the original pattern as shown in Figure 7. Four rows and four columns of such modulo-wrapped patterns were combined to generate a larger pattern. This pattern can be tiled regularly without a decrease in minimum toroidal distance.

3.3 Implementation for General m

Exploiting the fact that we are generating nets in base 2, all operations needed to implement the permutation net constructions described above can be implemented very

efficiently without any multiplications. A small C++ class that generates the point sets directly for even and odd $m \geq 4$ is available at http://gruenschloss.org/diag0m2/gendiag0m2.h. In terms of performance, it is comparable to the generation of rank-1 lattice points and also includes code to generate points with the modified tiling explained above.

As an example, we would like to show the implementation of Equation (7). Given the function `phi2` to compute the van der Corput radical inverse in base 2 [9], the implementation `y` has a one line body.

```
inline unsigned int phi2(unsigned int bits) { // for 32 bits
  bits = (bits << 16) | (bits >> 16);
  bits = ((bits & 0x00ff00ff) << 8) | ((bits & 0xff00ff00) >> 8);
  bits = ((bits & 0x0f0f0f0f) << 4) | ((bits & 0xf0f0f0f0) >> 4);
  bits = ((bits & 0x33333333) << 2) | ((bits & 0xcccccccc) >> 2);
  bits = ((bits & 0x55555555) << 1) | ((bits & 0xaaaaaaaa) >> 1);
  return bits;
}

inline unsigned int y(const unsigned int x, const unsigned int m) {
  return (x - (phi2(x & ~-(1 << (m >> 1))) >> (32 - m))) & ~-(1U << m);
}
```

For successive point requests this can be optimized by precomputing the bitmasks and the $2^{\lfloor \frac{m}{2} \rfloor}$ different radical inverse values.

4 Conclusion

We constructed new $(0, m, s)$-nets in base 2 for $s = 2, 3$, which are constrained by maximizing the minimum toroidal distance. Especially the $(0, m, 3)$-net has many applications in computer graphics such as computing anti-aliasing and motion blur in the Reyes architecture [1].

Acknowledgements We would like to thank the reviewers and editors for their very helpful comments. We would also like to thank Sabrina Dammertz, Johannes Hanika, Sehera Nawaz, Matthias Raab, and Daniel Seibert for helpful discussions.

References

1. Cook, R., Carpenter, L., Catmull, E.: The Reyes image rendering architecture. SIGGRAPH Computer Graphics **21**(4), 95–102 (1987)
2. Dammertz, S., Keller, A.: Image synthesis by rank-1 lattices. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 217–236. Springer (2008)
3. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arithmetica **41**(4), 337–351 (1982)

4. Faure, H.: Discrepancy and diaphony of digital $(0, 1)$-sequences in prime base. Acta Arithmetica **117**, 125–148 (2005)
5. Faure, H., Tezuka, S.: Another random scrambling of digital (t, s)-sequences. In: K. Fang, F. Hickernell, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 242–256, Springer-Verlag, Berlin (2002)
6. Grünschloß, L., Hanika, J., Schwede, R., Keller, A.: (t, m, s)-nets and maximized minimum distance. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 397–412. Springer (2008)
7. Hofer, R., Larcher, G.: On existence and discrepancy of certain digital Niederreiter-Halton sequences (2009). To appear in Acta Arithmetica
8. Keller, A.: Myths of computer graphics. In: H. Niederreiter (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 217–243. Springer (2006)
9. Kollig, T., Keller, A.: Efficient multidimensional sampling. Computer Graphics Forum **21**(3), 557–563 (2002)
10. Larcher, G., Pillichshammer, F.: Walsh series analysis of the L_2-discrepancy of symmetrisized point sets. Monatsheft Mathematik **132**, 1–18 (2001)
11. Larcher, G., Pillichshammer, F.: Sums of distances to the nearest integer and the discrepancy of digital nets. Acta Arithmetica **106**, 379–408 (2003)
12. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia (1992)
13. Pharr, M., Humphreys, G.: Physically Based Rendering: From Theory to Implementation, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2004)
14. Sedgewick, R.: Permutation generation methods. ACM Computing Surveys **9**(2), 137–164 (1977)
15. Sobol', I.: On the distribution of points in a cube and the approximate evaluation of integrals. Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki **7**(4), 784–802 (1967)

Automation of Statistical Tests on Randomness to Obtain Clearer Conclusion

Hiroshi Haramoto

Abstract Statistical testing of pseudorandom number generators (PRNGs) is indispensable for their evaluation. A common difficulty among statistical tests is how we consider the resulting probability values (p-values). When we observe a small p-value such as 10^{-3}, it is unclear whether it is due to a defect of the PRNG, or merely by chance. At the evaluation stage, we apply some hundred of different statistical tests to a PRNG. Even a good PRNG may produce some suspicious p-values in the results of a battery of tests. This may make the conclusions of the test battery unclear. This paper proposes an adaptive modification of statistical tests: once a suspicious p-value is observed, the adaptive statistical test procedure automatically increases the sample size, and tests the PRNG again. If the p-value is still suspicious, the procedure again increases the size, and re-tests. The procedure stops when the p-value falls either in an acceptable range, or in a clearly rejectable range. We implement such adaptive modifications of some statistical tests, in particular some of those in the Crush battery of TestU01. Experiments show that the evaluation of PRNGs becomes clearer and easier, and the sensitivity of the test is increased, at the cost of additional computation time.

1 Introduction

Pseudorandom number generators (PRNGs) are computer programs whose purpose is to produce sequences of numbers that seem to behave as if they were generated randomly from a specified probability distribution. Here we consider the case where the outputs of the PRNG imitate independent random variables from the uniform distribution over the interval $[0, 1)$ or over the integers in an interval $\{0, 1, 2, \ldots, N\}$.

Hiroshi Haramoto
Department of General Education, Kure College of Technology, Hiroshima, Japan
e-mail: haramoto@hiroshima-u.ac.jp

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_26,

Since PRNGs have a deterministic and periodic output, it is clear that they do not produce independent random variables in the mathematical sense, and that they cannot pass all possible statistical tests of uniformity and independence. But some of them have huge period lengths and turn out to behave quite well in statistical tests that can be applied in reasonable time. On the other hand, some PRNGs, which are known to be defective, fail very simple tests [7].

Many statistical tests for PRNGs are proposed. Widely used examples are: DIEHARD by Marsaglia [16], the test suite of the National Institute of Standards and Technology (NIST) [24], and TestU01 by L'Ecuyer and Simard [11]. When we use such a test suite for a PRNG, the result is a long list of p-values, each value corresponding to each test. It is often difficult to judge whether or not the existence of a few suspicious but not definitive p-values (say, $10^{-10} < p < 10^{-4}$) implies the defectiveness of the PRNG.

The aim of this paper is to eliminate this uncertainty, by proposing an adaptive modification of (essentially any) statistical test for PRNGs. The adaptive version of a statistical test means that we increase the sample size again and again, until we observe a definitely small p-value, or an acceptable normal p-value. This method is not novel; it is commonly used by hand, with heuristics. Our proposal is to automate this process. The rest of this paper is organized as follows. In Section 2, we review the statistical tests for PRNGs. We give a detailed description of an adaptive statistical test in Section 3. In Section 4, we show results of some adaptive statistical tests on some well known PRNGs.

2 Statistical Tests

Let $(a_1, \ldots, a_n)$ be a sequence in $[0, 1)$ generated by some method, i.e., by a PRNG, which imitates a uniform independent random sequence. Let Y_n be a (test) function of n variables from $[0, 1)^n$ to $\mathbb{R}$. A statistical test of $(a_1, \ldots, a_n)$ by Y_n is a function

$$T_{Y_n} : [0, 1)^n \to [0, 1], \quad (a_1, \ldots, a_n) \mapsto P(Y_n(X_1, \ldots, X_n) \le Y_n(a_1, \ldots, a_n)) \quad (1)$$

where $X_1, \ldots, X_n$ are random variables with identical, independent distribution (i.i.d.) uniform in $[0, 1)$, and $P(Y_n(X_1, \ldots, X_n) \le Y_n(a_1, \ldots, a_n))$ is the probability that $Y_n(X_1, \ldots, X_n) \le Y_n(a_1, \ldots, a_n)$ holds. This probability is called the p-value of the test. If this value is too close to 0 or too close to 1, the null hypothesis that $a_1, \ldots, a_n$ are uniform i.i.d is deemed suspicious. (As the editor pointed out, there are tests where the number n varies according to the values of a_i, but here we treat only the above type of statistical tests.)

If the p-value is extremely small (e.g., less than 10^{-10}), then it is (more or less) clear that the PRNG fails the test. If the p-value is suspicious but does not clearly indicate rejection ($p = 10^{-4}$, for example), it is difficult to judge. When we apply several tests to a PRNG, p-values smaller than 0.01 or larger than 0.99 are often

observed (since such values appear with probability 0.02). Therefore, users of test packages for PRNGs are often troubled by the interpretation of suspicious p-values.

In order to avoid such difficulties, a two-level test is often used, see [1] [4] [5] [15]. In a two-level test, we fix a test function Y_n. At the first level, we apply the test T_{Y_n} to the PRNG to be tested, consecutively k times. Then we obtain k of p-values, $p_1, \ldots, p_k$. At the second level, we test these k values under the null hypothesis of the uniform i.i.d. in the $[0, 1]$ interval, by some statistical test such as Kolmogorov-Smirnov test. The resulting p-value is the result of the two-level test. A merit of the two-level test is that it tends to give a clearer result, by accumulating the possibly existing deviation k times. Even if the first-level tests report moderate p-values, the two-level test may give a definitive p-value such as 10^{-8}. However, the possibility of getting a suspicious but not definitive p-value still remains.

Moreover, one may suffer from accumulated approximation error in computing p-values. We often compute p-values by using approximation formula: for example, the p-value of χ^2-test is computed by using an approximation. Therefore, some computing error exists in every p-value. Thus, if the p-values of the first level tests has 1% error in the same direction, and if the second level test uses a large number of these p-values (say, $k = 10000$ times), then the second level may detect the systematic computing error, which may lead to a false rejection [5] [13].

3 Adaptive Statistical Test

An adaptive statistical test requires a test function of variable sample size. That is, one type of test function Y_n of n variables, where n may vary. This is usually the case: most test statistics for testing PRNGs allow any sample size.

An adaptive statistical test for a PRNG based on Y_n is as follows. Fix a moderately large n. We generate n samples $a_0, \ldots, a_{n-1}$ using the PRNG, and compute the corresponding p-value $p_1 := T_{Y_n}(a_0, \ldots, a_{n-1})$. If p_1 lies in the pre-fixed admissible interval, say, in $[0.1, 0.9]$, then the test ends and does not reject the null hypothesis. Otherwise, we double the sample size, and generate $2n$ new samples $a_n, \ldots, a_{3n-1}$ using the PRNG, and compute the p-value $p_2 := T_{Y_{2n}}(a_n, \ldots, a_{3n-1})$. If p_2 lies in the admissible interval, then we accept. Otherwise, we double the sample size again, namely, we generate the next $4n$ samples using the PRNG, and compute the p-value p_3 for these $4n$ samples using Y_{4n}. We iterate this process, until reaching one of the following three cases:

Rejection: the p-value reaches to a prefixed value for the definitive rejection (e.g. $p = 10^{-8}$),
Acceptance: the p-value falls in the admissible interval,
Give up: the number of iterations reached to a prefixed number (say, 6) to stop the test (considering some limitation of memory and/or computation time).

Merits of the adaptive test are:

1. In most cases, the user obtains clear conclusions. The test takes care of suspicious p-values, until we conclude that they were obtained only by chance, or that they expose a systematic deviation. A final suspicious p-value is obtained only in the "give up" case, which occurs rarely.
2. The approximation errors in computing p-values are not accumulated, contrary to the the two-level tests described in the previous section. In the adaptive test, the larger sample size usually results in the smaller approximation error.

Note that such adaptive tests are appropriate for testing PRNGs, but not for general statistical tests such as census of population, where the sample size is often fixed or limited.

As recognized in the introduction, it is not novel to increase the sample size to resolve the suspicious p-values, and a number of studies exist, which treat delicate issues arising in adaptive tests. There are statistical tests where one need to change the parameters and/or the approximation formula of the distribution, according to the increase of the sample size n. In [9] and [12], the number of the cells in a classical serial test is kept proportional to n. In [10], $n^3/(4k)$ is kept constant. In both cases, these changes are necessary to keep an asymptotic approximation formula. In [12, p. 658], the asymptotic formula is changed according to the sample size n.

Below in §4.1 and §4.2, we treat some toy examples, where we may change the sample size n with keeping other parameters constant. In §4.3, we show a more serious implementation based on TestU01 [11], where appropriate choices of the parameters and approximation formulas are processed in the TestU01 library.

Another difficulty is the choice of the first sample size for the adaptive test. Every PRNG is rejected if the sample size is large enough, but the size depends on the interaction of the type of the PRNG and the test, and there are some thumb-nail rules [10], but we do not discuss here. In §4.1, we treat the case where a risky sample size is known in advance. In §4.3, we are constrained by the sample sizes selected by TestU01; see below.

4 The Results of Tests

4.1 Weight Distribution Test on FSRs

The weight distribution test is a test on the distribution of 1's in a pseudorandom bit sequence $x_1, x_2, \ldots$. We cut the sequence into subsequences of fixed length (here we choose the length 94), and count the number of 1's in each subsequence (i.e., the Hamming weight of each subsequence). The number should conform to the binomial distribution $B(94, 1/2)$. Let n be the sample size, namely, the number of subsequences of length 94 generated to be tested. We categorize the 95 observable values into several categories by merging, and apply the χ^2-test for the goodness-of-fit of the observed values to the binomial distribution. Note that this test is a variant

of the Hamming test treated in the next section, and related tests are included in TestU01 [11, Section 5.2.1].

The tested generator is a trinomial based feedback shift register (FSR) generator, defined by the recurrence

$$x_{j+89} := x_{j+38} + x_j$$

over the two-element field $\mathbb{F}_2$ (i.e., every operation is done modulo 2).

This generator is not an excellent generator; it is a toy-model example to explain how our adaptive test works. Matsumoto-Nishimura [22] computes *risky sample size* of such kind of generators, which means that if the sample size is larger than this size, then a simple weight distribution test will probabilistically reject a generated sequence with significance level 0.01. Thus, we can know an appropriate sample size for the test in advance. The risky sample size of the above generator is reported to be 1.16×10^5, so we choose the initial sample size $n = 50000$ in our adaptive test. Table 1 lists the results of the adaptive test described above (i.e., the acceptable interval is $[0.1, 0.9]$ and the rejection corresponds to the p-value outside $[10^{-8}, 1 - 10^{-8}]$). We apply the same adaptive test to the same generator with three randomly chosen initial values.

Table 1 Weight distribution test on the generator $x_{j+89} = x_{j+38} + x_j$.

sample size	50000	100000	200000	400000	result
1st	4.3×10^{-4}	7.4×10^{-3}	1.1×10^{-3}	1.4×10^{-14}	reject
2nd	3.6×10^{-2}	8.6×10^{-3}	1.6×10^{-5}	1.9×10^{-9}	reject
3rd	2.0×10^{-1}				accept

For example, in the first experiment, the p-value with sample size 50000 is 4.3×10^{-4}. It is suspicious, but not in the clear-rejection area ($< 10^{-8}$). Accordingly, the sample size is doubled, and the same test is applied to the new 100,000 samples, obtaining the p-value 7.4×10^{-3}. After four iterations, the p-value reaches 1.4×10^{-14}, and the bias of the weight distribution of the PRNG becomes clear. The result of the second experiment is similar. The first p-value of the third experiment lies in the acceptance interval, and hence there is no rejection, this time (could be regarded as a false-acceptance).

Table 2 shows the result of the same test on a similar generator based on a 5-term relation $x_{j+89} = x_{j+57} + x_{j+23} + x_{j+15} + x_j$ over $\mathbb{F}_2$. The risky sample size is known to be 6.99×10^7, so we choose the initial sample size to be 7×10^7 [22].

4.2 Hamming Weight Test on LCG

Here we treat a classical linear congruential generator (LCG), defined by the recurrence

$$x_{j+1} = 110351245x_j + 12345 \pmod{2^{31}}.$$

Table 2 Weight distribution test on the generator $x_{j+89} = x_{j+57} + x_{j+23} + x_{j+15} + x_j$.

sample size	7×10^7	1.4×10^8	2.8×10^8	5.6×10^8	result
1st	1.8×10^{-2}	4.2×10^{-5}	1.3×10^{-9}		reject
2nd	7.4×10^{-4}	1.1×10^{-5}	2.4×10^{-10}		reject
3rd	3.4×10^{-2}	2.2×10^{-5}	1.8×10^{-5}	4.4×10^{-33}	reject

The outputs (x_j) of this LCG are considered as 31-bit integers. In the Hamming test, these 31-bit integers are concatenated to be a single bit stream. The bit stream is divided to consecutive subsequences of 60 bits, and the number of 1's in each subsequence is counted. Then, the χ^2-test on the null hypothesis of the binomial distribution $B(60, 1/2)$ is applied, in the same way as the previous section. Unlike the previous generator, we do not have theory that tells us the risky sample size. We choose the initial sample size n to be $1,000,000$. Table 3 shows the results of three experiments, for randomly chosen initial values.

Table 3 Hamming weight test on the generator $x_{j+1} = 1103515245x_j + 12345 \pmod{2^{31}}$.

sample size	10^6	2×10^6	4×10^6	8×10^6	result
1st	0.0×10^0				reject
2nd	2.5×10^{-2}	1.6×10^{-8}	8.1×10^{-7}	0.0×10^0	reject
3rd	7.0×10^{-2}	3.0×10^{-3}	7.2×10^{-10}		reject

4.3 Crush in TestU01 and the Adaptive Crush

TestU01 by L'Ecuyer and Simard [11] is a strong comprehensive suite of statistical tests for uniform random numbers. TestU01 has flexible parameters, and hence is suitable to implement the adaptive version of statistical tests, unlike DIEHARD and NIST where the sample size is fixed.

There are three related batteries of tests in TestU01, using different sample sizes. These are the Small Crush, Crush, and Big Crush batteries. Big Crush is the most serious test battery, containing several statistical tests whose computation time and the memory consumption is near the limit of our computer, so partly it does not fit the adaptive version where the sample size is doubled iteratively. So, we choose the Crush battery as the basis for a battery of adaptive versions.

Among the 144 tests in Crush, 48 are not suitable to create adaptive versions. More precisely, (1) some of 48 are unable to create adaptive versions due to the lack of our computing resources, and (2) the remaining 96 are two-level-test versions of other tests in Crush. We implement adaptive versions of the remaining 96 tests, and call them the *Adaptive Crush* battery.

We apply the Adaptive Crush to the following three generators: an LCG [2] based on the recurrence

$$x_{j+1} = 950706376 x_j \pmod{2^{31} - 1}, \tag{2}$$

a subtract with borrow (SWB) [18] based on the recurrence

$$x_i = (x_{i-22} - x_{i-48} - c_{i-1}) \pmod{2^{32} - 5}, \tag{3}$$

$$c_i = \lfloor (x_{i-22} - x_{i-48} - c_{i-1})/(2^{32} - 5) \rfloor, \tag{4}$$

and TT800 [20].

Table 4, 5, and 6 list the tests for which

- the original Crush (namely the first step of the adaptive version) gives a p-value in the interval $[10^{-8}, 1 - 10^{-8}]$,
- but the p-value given by the Adaptive Crush lies outside $[10^{-8}, 1 - 10^{-8}]$ (the fourth column shows the number of iteration of doubling the sample size until the p-value was outside $[10^{-8}, 1 - 10^{-8}]$).

Table 4 Result on $x_{j+1} = 950706376 x_j \pmod{2^{31} - 1}$.

test name	initial sample size	1st p-value	# of iteration	final p-value
Gap	5×10^6	0.0125	2	$< 10^{-300}$
MaxOft	10^7	0.9863	2	$> 1 - 10^{-15}$
GCD	10^8	0.9805	3	$> 1 - 10^{-15}$
PeriodsInStrings	3×10^8	$1 - 6.6 \times 10^{-8}$	2	$> 1 - 10^{-15}$
PeriodsInStrings	3×10^8	0.9999	2	$1 - 7.3 \times 10^{-9}$

Table 5 Result on a subtract with borrow.

test name	initial sample size	1st p-value	# of iteration	final p-value
SimpPoker	10^7	0.0505	3	6.1×10^{-13}
SimpPoker	10^7	4.9×10^{-4}	3	3.8×10^{-14}
CouponCollector	10^7	0.0217	4	1.5×10^{-11}
CouponCollector	10^7	0.0328	4	6.0×10^{-15}

4.4 Comparison of the Sensitivity

In order to compare the sensitivity between the original Crush and the adaptive Crush, we apply the same 96 tests as in §4.3 to several PRNGs. Table 7 gives the

Table 6 Result on TT800.

test name	initial sample size	1st p-value	# of iteration	final p-value
Gap	5×10^6	1.6×10^{-4}	4	8.9×10^{-10}
RandomWalk1 J	10^6	6.1×10^{-5}	3	5.2×10^{-14}
HammingIndep	10^7	8.5×10^{-3}	2	9.5×10^{-12}

number of tests which report a p-value outside $[10^{-8}, 1-10^{-8}]$ by the original Crush and the adaptive Crush, respectively.

LCG(m, a, c) means the generator which obeys the recurrence $x_{j+1} := (ax_j + c) \pmod m$. LFib$(m, r, k, \mathtt{op})$ uses the recurrence $x_j := x_{j-r} \ \mathtt{op} \ x_{j-k} \pmod m$, where op is an operation which can be $+$ (addition), $-$ (subtraction), $*$ (multiplication), $\oplus$ (bitwise exclusive-or). The ran3 generator of Press and Teukolsky [23] is essentially an LFib$(10^9, 55, 24, -)$. Unix-randoms are PRNGs that are LFib$(2^{32}, 7, 3, +)$, LFib$(2^{32}, 15, 1, +)$, LFib$(2^{32}, 31, 3, +)$, with the least significant bit of each random number dropped. Knuth-ran_array2 [4] is LFib$(2^{30}, 100, 37, -)[100, 1009]$, where $[100, 1009]$ indicates that from each 1009 successive terms x_j from the recurrence, the first 100 outputs are retained and the others are discarded.

The notation GFSR(k, r) means the GFSR generator, with recurrence of the form $x_j := x_{j-r} \oplus x_{j-k}$. T800 and TT800 are twisted GFSR generators proposed by Matsumoto-Kurita [19] [20]. MT19937 is the Mersenne Twister of Matsumoto-Nishimura [21]. LFSR113 and LFSR258 are the combined Tausworthe generators of L'Ecuyer [6] designed for 32-bit and 64-bit computers, respectively. Marsaglia [17] recommends the 3-shift PRNGs Marsa-xor32 and Marsa-xor64.

SWB(m, r, k) means a subtract with borrow generator, employing the recurrence $x_j := (x_{j-r} - x_{j-k} - c_{j-1}) \pmod m$, where $c_j := \lfloor (x_{j-r} - x_{j-k} - c_{j-1}) \rfloor$. SWB$(2^{24}, 10, 24)[24, l]$ is called RANLUX with luxury level ℓ [14] [3]. In its original form, it returns 24-bit output values. For our tests, we use a version with 48-bit of precision obtained by concatenating every pair of the outputs to have a 48-bit integer. All these generators are copied from the library of TestU01 [11], except for the above-mentioned modification for SWB.

5 Conclusion

We introduced the notion of an adaptive statistical test. This method clarifies the conclusion from the test: Suspicious p-values are resolved by doubling the sample size iteratively. Experiments showed that this method works well in almost all cases. The sensitivity of the test increased, at the cost of additional computational time. In the experiments shown in Table 7, the Adaptive Crush consumed 3–5 times longer time than the original Crush for most cases (of course, it heavily depends on the number of iterations.)

Table 7 The number of rejections in the tests.

	original crush	adaptive crush
LCG($2^{31}, 65539, 0$)	86	92
LCG($2^{32}, 69069, 1$)	72	74
LCG($2^{32}, 1099087573, 0$)	75	81
LCG($2^{46}, 5^{13}, 0$)	15	19
LCG($2^{48}, 252144903917, 11$)	8	8
LCG($2^{48}, 5^{19}, 0$)	8	12
LCG($2^{31} - 1, 16807, 0$)	10	19
LCG($2^{31} - 1, 2^{15} - 2^{10}, 0$)	29	33
LCG($2^{31} - 1, 397204094, 0$)	6	14
LFib($2^{31}, 55, 24, +$)	9	10
LFib($2^{31}, 55, 24, -$)	11	11
LFib($2^{48}, 607, 273, +$)	3	3
ran3	10	10
Unix-random-32	86	86
Unix-random-64	38	52
Unix-random-128	13	14
Knuth-ran_array2	2	2
GFSR(250, 103)	8	10
GFSR(521, 32)	6	6
T800	29	33
TT800	13	17
MT19937	2	2
LFSR113	6	6
LFSR258	6	6
Marsa-Xor32	78	88
Marsa-Xor64	8	8
SWB($2^{24}, 10, 24$)	26	28
SWB($2^{24}, 10, 24$)[24,48]	3	4
SWB($2^{24}, 10, 24$)[24,97]	0	0
SWB($2^{24}, 10, 24$)[24,389]	0	0
SWB($2^{32} - 5, 22, 43$)	8	11
SWB($2^{31}, 8, 48$)	10	10

Among the PRNGs tested in this way, we could not find a generator that passes all the Crush tests but fails one of the adaptive Crush tests. This means that the sample size of the original Crush is very well chosen.

Acknowledgements I would like to thank Professor Makoto Matsumoto who helped and encouraged me constantly, Professor Pierre L'Ecuyer who gave useful comments, Professor Hirokazu Yanagihara who pointed out the importance of the power of tests, and the anonymous referee for deep and valuable comments. This research has been supported in part by JSPS Grant-In-Aid #19204002, #18654021, #21654017 and JSPS Core-to-Core Program No.18005.

References

1. G. S. Fishman. *Monte Carlo*. Springer Series in Operations Research. Springer-Verlag, New York, 1996. Concepts, algorithms, and applications.
2. G. S. Fishman and L. R. Moore, III. An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$. *SIAM J. Sci. Statist. Comput.*, 7(3):1058, 1986.
3. F. James. RANLUX: a Fortran implementation of the high-quality pseudorandom number generator of Lüscher. *Computer Physics Communications*, 97:357–357(1), September 1996.
4. D. E. Knuth. *The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.
5. P. L'Ecuyer. Testing random number generators. In *WSC '92: Proceedings of the 24th conference on Winter simulation*, pages 305–313, New York, NY, USA, 1992. ACM.
6. P. L'Ecuyer. Tables of maximally equidistributed combined LFSR generators. *Math. Comput.*, 68(225):261–269, 1999.
7. P. L'Ecuyer. Software for uniform random number generation: distinguishing the good and the bad. In *WSC '01: Proceedings of the 33nd conference on Winter simulation*, pages 95–105, Washington, DC, USA, 2001. IEEE Computer Society.
8. P. L'Ecuyer, J. F. Cordeau, and R. Simard. Close-point spatial tests and their application to random number generators. *Operations Research*, 48(2):308–317, 2000.
9. P. L'Ecuyer and P. Hellekalek. Random number generators: selection criteria and testing. In *Random and Quasi-Random Point Sets*, volume 138 of *Lecture Notes in Statistics*, pages 223–266. Springer, 1998.
10. P. L'Ecuyer and R. Simard. On the interaction of birthday spacings tests with certain families of random number generators. *Mathematics and Computers in Simulation*, 55:131–137, 2001.
11. P. L'Ecuyer and R. Simard. TestU01: a C library for empirical testing of random number generators. *ACM Trans. Math. Software*, 33(4):Art. 22, 40, 2007.
12. P. L'Ecuyer, R. Simard, and S. Wegenkittl. Sparse serial tests of uniformity for random number generators. *SIAM Journal on Scientific Computing*, 24(2):652–668, 2002.
13. P. C. Leopardi. Testing the tests: using pseudorandom number generators to improve empirical tests. Talk in MCQMC 2008, July 2008.
14. M. Lüscher. A portable high-quality random number generator for lattice field theory simulations. *Comput. Phys. Comm.*, 79(1):100–110, 1994.
15. G. Marsaglia. A Current View of Random Number Generators. In *Computer Science and Statistics, Sixteenth Symposium on the Interface*, pages 3–10. Elsevier Science Publishers, 1985.
16. G. Marsaglia. DIEHARD: A battery of tests of randomness. 1996. See `http://stat.fsu.edu/~geo/diehard.html`.
17. G. Marsaglia. Xorshift RNGs. *Journal of Statistical Software*, 8(14):1–6, 2003.
18. G. Marsaglia, B. Narasimhan, and A. Zaman. A random number generator for PCs. *Comput. Phys. Comm.*, 60(3):345–349, 1990.
19. M. Matsumoto and Y. Kurita. Twisted GFSR generators. *ACM Trans. Model. Comput. Simul.*, 2(3):179–194, 1992.
20. M. Matsumoto and Y. Kurita. Twisted GFSR generators II. *ACM Trans. Model. Comput. Simul.*, 4(3):254–266, 1994.
21. M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.
22. M. Matsumoto and T. Nishimura. A nonempirical test on the weight of pseudorandom number generators. In *Monte Carlo and quasi-Monte Carlo methods, 2000 (Hong Kong)*, pages 381–395. Springer, Berlin, 2002.
23. W. H. Press and S. A. Teukolsky. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.

24. A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A Statistical Test Suite for Random and Pseudorandom number Generators for Cryptographic Applications. NIST Special Publication 800-22, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 2001. See http://csrc.nist.gov/rng/.

On Subsequences of Niederreiter-Halton Sequences

Roswitha Hofer

Abstract In this paper we investigate the distribution properties of subsequences of Niederreiter-Halton sequences. A Niederreiter-Halton sequence is generated by joining digital $(\mathbf{T}, s)$-sequences in different prime bases. Thus Niederreiter-Halton sequences are hybrids of digital $(\mathbf{T}, s)$-sequences as mainly introduced by Niederreiter and the van der Corput-Halton sequences, which are joint versions of special $(0, 1)$-sequences in different prime bases. As hybrids of well-known low-discrepancy sequences the distribution properties of such sequences are of great interest. In this paper we give an overview of existing results. Furthermore, we investigate the distribution of special types of subsequences, as for example subsequences indexed by arithmetic progressions or the subsequence indexed by primes.

1 Introduction

We assume that the basic notions of *uniform distribution* of a sequence $(\mathbf{x}_n)_{n\geq 0}$ in the unit cube $[0, 1)^s$ and its *discrepancy* D_N (*star-discrepancy* D_N^*) are known. We just want to emphasize here, that a sequence in the s-dimensional unit cube is called a "low-discrepancy sequence" if its (star-)discrepancy satisfies the following upper bound

$$N D_N = O(\log^s(N)).$$

Excellent introductions to these and related topics can be found in the book of Kuipers & Niederreiter [13] or in the book of Drmota & Tichy [4].

In this paper we focus on a special class of sequences, which are hybrids of digital $(\mathbf{T}, s)$-sequences in prime base q as mainly introduced by Niederreiter and

Roswitha Hofer
Institute of Financial Mathematics, University of Linz, Austria
e-mail: roswitha.hofer(at)jku.at

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_27,

of the van der Corput-Halton sequences in different prime bases. We introduce the notion of "Niederreiter-Halton sequences" to emphasize the hybrid character. We give the detailed definition.

Definition 1. Let $v \in \mathbb{N}$ and $q_1, \ldots, q_v$ be different primes. Furthermore, let $w_1, \ldots, w_v$ be positive integers. For all $l \in \{1, \ldots, v\}$ let $C^{(l,1)}, \ldots, C^{(l,w_l)}$ be $\mathbb{N} \times \mathbb{N}_0$-matrices over $\mathbb{Z}_{q_l}$ (i.e. the finite field of residues modulo q_l). We now define a sequence $(\mathbf{x}_n)_{n \geq 0}$ in $[0,1)^s$ with $s := w_1 + \cdots + w_v$ by

$$\mathbf{x}_n := \left(x_n^{(1,1)}, \ldots, x_n^{(1,w_1)}, \ldots, x_n^{(v,1)}, \ldots, x_n^{(v,w_v)} \right).$$

The component $x_n^{(l,j)}$, for $j \in \{1, \ldots, w_l\}$, $l \in \{1, \ldots, v\}$, is generated as follows.

Let $n = n_0^{(l)} + n_1^{(l)} q_l + n_2^{(l)} q_l^2 + \cdots$ be the q_l-ary representation of n. Then we set

$$C^{(l,j)} \cdot \left(n_0^{(l)}, n_1^{(l)}, \ldots \right)^\top =: \left(y_1^{(l,j)}, y_2^{(l,j)}, \ldots \right)^\top \in \mathbb{Z}_{q_l}$$

and

$$x_n^{(l,j)} := \frac{y_1^{(l,j)}}{q_l} + \frac{y_2^{(l,j)}}{q_l^2} + \cdots.$$

Throughout this paper, if nothing else is said, the integers $v \geq 1$, $w_1, \ldots, w_v \geq 1$ and the different primes $q_1, ..., q_v$ are fixed.

The Niederreiter-Halton sequences contain many well-known low-discrepancy sequences, e.g., digital (t,s)-sequences in prime base, which were introduced by Niederreiter (see [17] and [18]) and generalize sequences introduced by Faure [5] and earlier forms by Sobol [19], or the van der Corput-Halton sequences [7] in different prime bases. If w_l is equal to 1 and $C^{(l,1)}$ is chosen as the unit matrix for all $l \in \{1, \ldots, v\}$ the construction principle above generates a van der Corput-Halton sequence. If we set $v = 1$, Definition 1 is consistent with the definition of digital $(\mathbf{T}, s)$-sequences in the sense of Larcher and Niederreiter (see [14]), which generalizes digital (t,s)-sequences over $\mathbb{Z}_{q_1}$ introduced by Niederreiter. Hybrids similar to Niederreiter-Halton sequences have already been introduced by Faure [6], which are generated by juxtaposing certain components of Faure sequences (i.e. digital $(0,s)$-sequences generated by powers of the Pascal-matrices) in different prime bases. These sequences are also contained in the class of Niederreiter-Halton sequences.

Since Niederreiter-Halton sequences are hybrids of low-discrepancy sequences, the investigation of their distribution properties is of great interest. The primary problem is the classification of the sequences in this family which are uniformly distributed. The special case where the generator matrices consist of rows of finite length exclusively was considered in [10]. (We say a row is of finite length if it contains just finitely many entries not equal to zero). It turned out that the investigation of this special class of Niederreiter-Halton sequences can be reduced to estimates of the number of solutions of systems of congruences. The application of the Chinese Remainder Theorem is a crucial step in the proofs of statements about certain

distribution properties of these special sequences. Hence the proof of the following interesting result is more or less straightforward (see [10]).

Theorem 1. *[10, Theorem 2.3] A Niederreiter-Halton sequence generated by matrices consisting of rows of finite length exclusively is uniformly distributed if and only if for each* $l \in \{1, \ldots, v\}$ *the corresponding digital* $(\mathbf{T}^{(l)}, w_l)$*-sequence is uniformly distributed.*

It is well known, that a digital $(\mathbf{T}, s)$-sequence is uniformly distributed if and only if $\lim_{m\to\infty}(m - \mathbf{T}(m)) = +\infty$, where $\mathbf{T}(m)$ is a quality parameter-function defined on the rank-structure of the matrices (see for example [18] for more details here).

As mentioned already, the method of proof for Theorem 1 used in [10] is based on the application of the Chinese Remainder Theorem. Unfortunately, this method fails for the investigation of the general class of Niederreiter-Halton sequences. Note that the Chinese Remainder Theorem cannot be applied if we have at least one row of infinite length in any of the generator matrices. (We say a row is of infinite length if it contains infinitely many entries not equal to zero). In [10] it was pointed out that this general case is much more difficult. Nevertheless, Theorem 1 has been generalized by the following theorem.

Theorem 2. *[9, Theorem 4] A Niederreiter-Halton sequence is uniformly distributed if and only if for each* $l \in \{1, \ldots, v\}$ *the corresponding digital* $(\mathbf{T}^{(l)}, w_l)$*-sequence is uniformly distributed.*

The statement of Theorem 2 can be reworded by: "Every joint version of uniformly distributed digital $(\mathbf{T}, s)$-sequences in different prime bases remains uniformly distributed." The proof (see [9]) is based on methods elaborated by Kim [12] and further generalized by Drmota and Larcher [3] and by Hofer [8].

By Theorem 2 the classification of the Niederreiter-Halton sequences that are uniformly distributed is done and the next question of course is: Are there any new low-discrepancy sequences amongst the Niederreiter-Halton sequences? Unfortunately, it turns out that this question is a problem of considerable difficulty and it is still open for future research. There exist estimates of the discrepancy of some special Niederreiter-Halton sequences, e.g., an upper bound on the discrepancy for the special case of Niederreiter-Halton sequences generated by matrices consisting of rows of finite length exclusively (see [10, Theorem 3.1]), in [11] this upper bound is studied further and also discrepancy bounds are given for some special Niederreiter-Halton sequences generated by matrices containing certain infinite rows.

In this paper concentration is laid on subsequences of Niederreiter-Halton sequences. The investigation of the distribution properties of subsequences — beyond mere curiosity — is motivated by the following results about the van der Corput-Halton sequences.

Theorem 3. *[10, Theorem 6.1] Let* ω_{vdC} *be the van der Corput sequence in base* q*. Let* $u, w \in \mathbb{Z}$ *with* $u \geq 1$ *and* $\gcd(u, q) = 1$*. Further define* $b_n = un + w$*. Then the sequence* $\omega = (x_{b_n})_{n\geq 0}$ *satisfies*

$$D_N^*(\omega) \leq D_N^*(\omega_{\mathrm{vdC}}).$$

By Theorem 3 we have found subsequences of the van der Corput sequences that beat the full sequence. For the s-dimensional van der Corput-Halton sequences we know subsequences, that are also low-discrepancy sequences.

Theorem 4. *[10, Theorem 6.2] Let $(\mathbf{x}_n)_{n\geq 0}$ be the van der Corput-Halton sequence in relatively prime bases $q_1, \ldots, q_s$. Furthermore, let $u \in \mathbb{N}$ with $\gcd(u, q_i) = 1$ for all $i \in \{1, \ldots, s\}$ and let $k_n = un$. Then the sequence $\omega = (\mathbf{x}_{k_n})_{n\geq 0}$ satisfies*

$$D_N^*(\omega) \leq \prod_{i=1}^{s} \frac{q_i - 1}{2 \log q_i} \frac{(\log N)^s}{N} + O\left(\frac{(\log N)^{s-1}}{N}\right),$$

where the implied factor in the O-notation is independent of N, but depends on $q_1, \ldots, q_s$, u and s.

Of course the constant in this theorem could be improved as it was done for the full sequence. The latest upper bound for the discrepancy of van der Corput-Halton sequences was given by Atanassov [2].

Motivated by these results we investigate the distribution properties of subsequences of Niederreiter-Halton sequences. Again, the primary problem is to classify the subsequences which are uniformly distributed.

For the special class of Niederreiter-Halton sequences, which are generated by matrices consisting of rows of finite length exclusively, useful criteria were found already [10], e.g.:

Theorem 5. *[10, Theorem 4.2] Let $(\mathbf{x}_n)_{n\geq 0}$ be a uniformly distributed Niederreiter-Halton sequence generated by matrices consisting of rows of finite length exclusively.*

(a) Let $(k_n)_{n\geq 0}$ be a sequence of non-negative integers. If for all positive integers d the sequence $(k_n)_{n\geq 0}$ is uniformly distributed modulo $(q_1 \cdots q_v)^d$, then the subsequence $(\mathbf{x}_{k_n})_{n\geq 0}$ is uniformly distributed in $[0, 1)^s$.

(b) The condition given in (a) for $(k_n)_{n\geq 0}$ is also a necessary condition for the uniform distribution of $(\mathbf{x}_{k_n})_{n\geq 0}$, if and only if $w_l = 1$ for all l and $C^{(l,1)}$ is a lower triangular matrix for all l.

In this paper we aim for conditions for uniform distribution of subsequences of Niederreiter-Halton sequences in the general case. In Section 2 we define the special subsequences, which are indexed by the solutions of a system of congruences, and find both sufficient and necessary conditions such that they are uniformly distributed. Furthermore, we investigate the subsequences indexed by primes and generalize [10, Theorem 5.1]. In a short conclusion we summarize the new results obtained in this paper and sketch open problems concerning the distribution properties of Niederreiter-Halton sequences and their subsequences.

2 Subsequences

It is easy to check (see proof of necessity [9, Theorem 4]) that a Niederreiter-Halton sequence is uniformly distributed in $[0,1)^s$ if and only if it is dense in $[0,1)^s$. Hence any subsequence of a not uniformly distributed Niederreiter-Halton sequence cannot be uniformly distributed, since it is not even dense. Therefore, we reduce our investigations in this chapter to uniformly distributed Niederreiter-Halton sequences.

For the proofs in this section we will use the following notation for the generator matrices for any $l \in \{1,\dots,v\}$, $j \in \{1,\dots,w_l\}$

$$C^{(l,j)} := \begin{pmatrix} \gamma_{1,0}^{(l,j)} & \gamma_{1,1}^{(l,j)} & \gamma_{1,2}^{(l,j)} & \cdots \\ \gamma_{2,0}^{(l,j)} & \gamma_{2,1}^{(l,j)} & \gamma_{2,2}^{(l,j)} & \cdots \\ \gamma_{3,0}^{(l,j)} & \gamma_{3,1}^{(l,j)} & \gamma_{3,2}^{(l,j)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \in {\mathbb{Z}_{q_l}}^{\mathbb{N}\times\mathbb{N}_0},$$

where we assume that the r-th row of the matrix $C^{(l,j)}$ is given by $\gamma_r^{(l,j)} = (\gamma_{r,t}^{(l,j)})_{t\geq 0}$ in $\mathbb{Z}_{q_l}$.

From the definition of a Niederreiter-Halton sequence one can see the connection to the integer-weighted q-ary sum-of-digits function modulo q. The "weighted q-ary sum-of-digits function" of a non-negative integer n is defined by

$$s_{q,\gamma}(n) := \sum_{i=0}^{\infty} n_i \gamma_i,$$

where $\gamma = (\gamma_i)_{i\geq 0}$ is a given weight sequence in $\mathbb{R}$ and $n = n_0 + n_1 q + n_2 q^2 + \cdots$ is the q-ary representation of n with $0 \leq n_i \leq q-1$. Especially, if γ is a sequence in $\mathbb{Z}$ then we call $s_{q,\gamma}(n)$ the "integer-weighted q-ary sum-of-digits function".

In the introduction we used the well-established parameter-function $\mathbf{T}(m)$ (for the detailed definition we refer to [18]). For the investigation of the hybrids another parameter-function based on the rank-structures of the generator matrices turns out to be useful.

Definition 2. For each $l \in \{1,\dots,v\}$ and for each choice of non-negative integers $d^{(l,1)},\dots,d^{(l,w_l)}$ let $F^{(l)}(d^{(l,1)},\dots,d^{(l,w_l)}) \in \mathbb{N}$ be minimal such that the $(d^{(l,1)} + \cdots + d^{(l,w_l)}) \times F^{(l)}(d^{(l,1)},\dots,d^{(l,w_l)})$-matrix formed by

the left upper $d^{(l,1)} \times F^{(l)}(d^{(l,1)},\dots,d^{(l,w_l)})$-submatrix of $C^{(l,1)}$ together with
the left upper $d^{(l,2)} \times F^{(l)}(d^{(l,1)},\dots,d^{(l,w_l)})$-submatrix of $C^{(l,2)}$ together with
$\vdots$
the left upper $d^{(l,w_l)} \times F^{(l)}(d^{(l,1)},\dots,d^{(l,w_l)})$-submatrix of $C^{(l,w_l)}$

has rank $d^{(l,1)} + \cdots + d^{(l,w_l)}$. We set $F^{(l)}(d^{(l,1)},\dots,d^{(l,w_l)}) := +\infty$ if this is not satisfied for any finite $F^{(l)}(d^{(l,1)},\dots,d^{(l,w_l)})$.

In the following a Niederreiter-Halton sequence introduced in Definition 1 will often be called "digital $(\mathbf{F}, s)$-sequence in bases $((q_1, w_1), \ldots, (q_v, w_v))$" with $\mathbf{F} := (F^{(1)}, \ldots, F^{(v)})$ and $s = w_1 + \cdots + w_v$.

Using the parameter-function $\mathbf{F}$, Theorem 2 can be reworded by:
A digital $(\mathbf{F}, s)$-sequence in bases $((q_1, w_1), \ldots, (q_v, w_v))$ is uniformly distributed if and only if for each $l \in \{1, \ldots, v\}$ the parameter-function $F^{(l)}$ is finite, i.e. $F^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) < +\infty$ for all integers $d^{(l,j)} \geq 0$, $j \in \{1, \ldots, w_l\}$.

For the investigation of subsequences we introduce the notion of compatibility of matrices and additional rows.

Definition 3. Let $d \in \mathbb{N}$. For any $l \in \{1, \ldots, v\}$ we call the generator matrices $C^{(l,1)}, \ldots, C^{(l,w_l)}$ in $\mathbb{Z}_{q_l}$ and the additional rows given by $(\rho_{i,0}, \rho_{i,1}, \rho_{i,2}, \ldots) \in \mathbb{Z}_{q_l}^{\mathbb{N}_0}$ and $i \in \{1, \ldots, d\}$ "compatible" if the parameter-function $\overline{F}^{(l)}$ defined as follows is finite. Let $\overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) \in \mathbb{N}$ be minimal such that the matrix formed by

the left upper $d^{(l,1)} \times \overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)})$-submatrix of $C^{(l,1)}$ together with
the left upper $d^{(l,2)} \times \overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)})$-submatrix of $C^{(l,2)}$ together with
$\vdots$
the left upper $d^{(l,w_l)} \times \overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)})$-submatrix of $C^{(l,w_l)}$ together with
the d additional rows truncated to length $\overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)})$

has full row-rank. We set $\overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) := +\infty$ if this is not satisfied for any finite $\overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)})$. We call the parameter-function $\overline{F}^{(l)}$ finite if for all $d^{(l,1)}, \ldots, d^{(l,w_l)} \in \mathbb{N}_0$ we get $\overline{F}^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) < +\infty$.

Example 1. The Gray-Code digits of a non-negative integer n in any base q are given by $n_i^g = n_i + n_{i+1} \pmod q$, $i \geq 0$, where the n_i are the digits of the q-ary representation of n. So the van der Corput sequence in prime base q based on Gray-Code digits is generated by the following matrix.

$$\begin{pmatrix} 1\,1\,0\,0\,\ldots \\ 0\,1\,1\,0\,\ldots \\ 0\,0\,1\,1\,\ldots \\ \vdots\,\vdots\,\vdots\,\vdots\,\ddots \end{pmatrix} \in \mathbb{Z}_q^{\mathbb{N}\times\mathbb{N}_0}.$$

It is obvious that the generator matrix and the additional row $(10000\ldots) \in \mathbb{Z}_q^{\mathbb{N}_0}$ are compatible and $\overline{F}(d) = F(d) + 1 = d + 1$.

2.1 Subsequences Indexed by the Solutions of a System of Congruences

Here we consider subsequences of Niederreiter-Halton sequences, for which the indices are given by the solutions (of course in increasing order) of a system of congruences.

Definition 4. For $z \in \mathbb{N}$ let $p_1, \ldots, p_z$ be different primes and let $y_1, \ldots, y_z$ be positive integers. For every pair (i,k) with $i \in \{1,\ldots,z\}, k \in \{1,\ldots,y_i\}$ we are given the following congruence over $\mathbb{Z}_{p_i}^{\mathbb{N}_0}$

$$(\rho_0^{(i,k)}, \rho_1^{(i,k)}, \ldots) \cdot (x_0^{(i)}, x_1^{(i)}, \ldots)^T \equiv b^{(i,k)} \pmod{p_i}$$

with fixed $b^{(i,k)} \in \mathbb{Z}_{p_i}$ and fixed $\rho_h^{(i,k)} \in \mathbb{Z}_{p_i}$ for all $h \geq 0$. We call a non-negative integer n a "solution of this system of congruences" if for each $i \in \{1,\ldots,z\}$ the digit-vector $(n_0^{(i)}, n_1^{(i)}, \ldots)^T$ given by the p_i-ary representation of $n = n_0^{(i)} + n_1^{(i)} p_i + \cdots$ solves all the y_i congruences over $\mathbb{Z}_{p_i}^{\mathbb{N}_0}$.

We demand that the system of congruences is not overdetermined, i.e., for every $i \in \{1,\ldots,z\}$ if we join the y_i rows, $(\rho_0^{(i,k)}, \rho_1^{(i,k)}, \ldots)$, corresponding to the fixed prime p_i to a matrix, then we can find a finite but sufficiently large $t \in \mathbb{N}$ such that the left $y_i \times t$-submatrix has full row-rank.

Example 2. The indices given by an arithmetic progression, $k_b = ub + w, b \in \mathbb{N}_0$ and with fixed integers $u > 0$ and $0 \leq w < u$, can be interpreted as the solutions of a system of congruences.

We consider the unique prime factorization of $u = p_1^{\alpha_1} \cdots p_m^{\alpha_m}$ with $\alpha_i \geq 1$, $1 \leq i \leq m$. For each $i \in \{1,\ldots,m\}$ we choose the first α_i rows of the unit matrix in $\mathbb{Z}_{p_i}^{\mathbb{N}\times\mathbb{N}_0}$, which are denoted by I_{α_i}, and get the condition

$$I_{\alpha_i} \cdot (n_0^{(i)}, n_1^{(i)}, \ldots)^T = (w_0^{(i)}, \ldots, w_{\alpha_i-1}^{(i)})^T$$

over $\mathbb{Z}_{p_i}$, where $(n_j^{(i)})_{j\geq 0}$ are the digits of the p_i-ary representation of any non-negative integer n and $w_0^{(i)}, \ldots, w_{\alpha_i-1}^{(i)}$ are the first α_i digits of the p_i-ary representation of w.

It is easy to check that for all positive integers N we have

$$\{0 \leq n < N : n = ub + v \text{ with } b \in \mathbb{N}_0\} = \{0 \leq n < N : \ n \text{ solves the system above}\}.$$

Using the notion of compatibility we find both sufficient and necessary conditions for uniform distribution of any subsequence determined by indices, which can be interpreted as the solutions of a system of congruences.

Theorem 6. *Let $(\mathbf{x}_n)_{n\geq 0}$ be a uniformly distributed digital $(\mathbf{F}, s)$-sequence in bases $((q_1, w_1), \ldots, (q_v, w_v))$. Any subsequence determined by indices, which can be interpreted as the solutions of a system of congruences given in Definition 4, is*

uniformly distributed if and only if for every $l \in \{1, \dots, v\}$ *with* $q_l = p_i$ *for some* $i \in \{1, \dots, z\}$ *the generator matrices* $C^{(l,1)}, \dots, C^{(l,w_l)}$ *and the* y_i *additional rows,* $(\rho_0^{(i,k)}, \rho_1^{(i,k)}, \dots), k \in \{1, \dots, y_i\}$*, are compatible.*

Proof. Let $S := \{k_b : b \in \mathbb{N}_0\} \subseteq \mathbb{N}_0$ be the set of indices, which determines the subsequence $(\mathbf{x}_{k_b})_{b \geq 0}$ and can be interpreted as the solutions of a system of congruences as given in Definition 4.

We consider elementary intervals of the following form

$$I = \prod_{l=1}^{v} \prod_{j=1}^{w_l} \left[\frac{a^{(l,j)}}{q_l^{d^{(l,j)}}}, \frac{a^{(l,j)}+1}{q_l^{d^{(l,j)}}} \right),$$

where $d^{(l,j)}$ are arbitrary non-negative integers and $a^{(l,j)} \in \{0, 1, \dots, q_l^{d^{(l,j)}} - 1\}$ for $j \in \{1, \dots, w_l\}$ and $l \in \{1, \dots, v\}$. In order to show uniform distribution of the subsequence $(\mathbf{x}_{k_b})_{b \geq 0}$, it suffices to show that the following relation holds for each such interval

$$\lim_{N \to \infty} \frac{1}{B(N)} \#\{0 \leq b < B(N) : \mathbf{x}_{k_b} \in I\} = \lambda(I), \tag{1}$$

where $B(N) := \#\{0 \leq n < N : n \in S\}$.

The set of indices is represented by the solutions of the following system of congruences (not overdetermined).

Let z be a positive integer. Let $p_1, \dots, p_z$ be different primes and $y_1, \dots, y_z$ be positive integers. For $i \in \{1, \dots, z\}, k \in \{1, \dots, y_i\}$ we have

$$(\rho_0^{(i,k)}, \rho_1^{(i,k)}, \dots) \cdot (n_0^{(i)}, n_1^{(i)}, \dots)^T \equiv b^{(i,k)} \pmod{p_i}$$

where $n_0^{(i)} + n_1^{(i)} p_i + \cdots$ is the p_i-ary representation of n and $b^{(i,k)} \in \mathbb{Z}_{p_i}$, $\rho_h^{(i,k)} \in \mathbb{Z}_{p_i}$ for all $h \geq 0$.

In the following we show that

$$\lim_{N \to \infty} \frac{B(N)}{N} = \lim_{N \to \infty} \frac{1}{N} \#\{0 \leq n < N : n \in S\} = \frac{1}{\prod_{i=1}^{z} p_i^{y_i}}$$

and

$$\lim_{N \to \infty} \frac{1}{N} \#\{0 \leq n < N : \mathbf{x}_n \in I \text{ and } n \in S\} = \frac{1}{\prod_{i=1}^{z} p_i^{y_i}} \cdot \lambda(I).$$

We regard the q_l-ary representation of $a^{(l,j)}/q_l^{d^{(l,j)}} = (0.a_1^{(l,j)} a_2^{(l,j)} \dots a_{d^{(l,j)}}^{(l,j)})_{q_l}$ and observe that the following condition is equivalent to $\mathbf{x}_n \in I$:

$$\forall l \in \{1, \dots, v\} \forall j \in \{1, \dots, w_l\} \forall r \in \{1, \dots, d^{(l,j)}\} : s_{q_l, \gamma_r^{(l,j)}}(n) \equiv a_r^{(l,j)} \pmod{q_l}. \tag{2}$$

We can interpret the left hand sides of the congruences, which determine the set of indices S, as integer weighted p_i-ary sum-of-digits functions. Hence the condition $n \in S$ is equivalent to

$$\forall i \in \{1, \ldots, z\} \forall k \in \{1, \ldots, y_i\} : s_{p_i, \rho^{(i,k)}}(n) \equiv b^{(i,k)} \pmod{p_i}, \tag{3}$$

where the sequences $\rho^{(i,k)} := (\rho_h^{(i,k)})_{h \geq 0}$.

We can count the number of solutions of (3) and also the number of common solutions of (2) and (3) as in the proof of [9, Theorem 4]. Since we have assumed that the system of congruences (3) is not overdetermined and that the generator matrices and the additional rows given by the system of congruences are compatible, we get the results as claimed above. (We refer the interested reader to [9].)

Altogether we get:

$$\begin{aligned}
&\lim_{N \to \infty} \frac{1}{B(N)} \#\{0 \leq b < B(N) : \mathbf{x}_{k_b} \in I\} \\
&= \lim_{N \to \infty} \frac{N}{B(N)} \frac{1}{N} \#\{0 \leq n < N : \mathbf{x}_n \in I \text{ and } n \in S\} \\
&= \prod_{i=1}^{z} p_i^{y_i} \frac{1}{\prod_{i=1}^{z} p_i^{y_i}} \cdot \lambda(I),
\end{aligned}$$

which completes the proof of sufficiency.

Conversely, if we assume non-compatibility we can find an elementary interval I such that

$$\frac{1}{N} \#\{0 \leq n < N : \mathbf{x}_n \in I \text{ and } n \in S\} = 0.$$

Non-compatibility implies, we have an $l \in \{1, \ldots, v\}$ such that $q_l = p_i$ for any $i \in \{1, \ldots, z\}$ and that there exist non-negative integers $d^{(l,1)}, \ldots, d^{(l,w_l)}$ (not all zero) such that $F(d^{(l,1)}, \ldots, d^{(l,w_l)}) < +\infty$ but $\overline{F}(d^{(l,1)}, \ldots, d^{(l,w_l)}) = +\infty$.

Let $t \in \mathbb{N}$ be arbitrary but fixed. We consider the system of congruences given by

the left upper $d^{(l,1)} \times t$-submatrix of $C^{(l,1)}$ together with
the left upper $d^{(l,2)} \times t$-submatrix of $C^{(l,2)}$ together with
$\vdots$
the left upper $d^{(l,w_l)} \times t$-submatrix of $C^{(l,w_l)}$

and we consider the system built by the y_i congruences over $\mathbb{Z}_{p_i}$ of course the rows $(\rho_0^{(i,k)}, \rho_1^{(i,k)}, \ldots)$ are also truncated to length t.

Since the corresponding $\overline{F}(d^{(l,1)}, \ldots, d^{(l,w_l)})$ is not finite, we know that there exist a linear combination of the truncated y_i rows and a linear combination of the rows of the submatrices given above such that the results are equal. We can find some parameters on the right of the system of congruences given by the submatrices above, such that there does not exist any common solution. From these parameters we can derive an elementary interval, which remains empty if we consider the first $B(q_l^t)$ points of the subsequence. Since t was arbitrarily chosen we can find an elementary interval, which remains empty no matter how many points of the subsequence we consider. Thus the subsequence is not even dense. □

Remark 1. In the case where $w_l = 1$ for all $l \in \{1, \dots, v\}$ and $C^{(l,1)}$ are lower left triangular matrices with parameter-function $F^{(l)}(d) = d$, Theorem 5 (compare [10, Theorem 4.2 (b)]) provides already sufficient as well as necessary conditions for uniform distribution of a subsequence, namely the sequence of indices $(k_b)_{b\geq 0}$ has to be uniformly distributed modulo $(q_1^d \cdots q_v^d)$ for all positive integers d. It is not so hard to check, that in the special case of both — a subsequence as considered in Theorem 6 and a Niederreiter-Halton sequence as considered in [10, Theorem 4.2 (b)] — the condition "compatibility" is equivalent to the condition "the sequence of indices $(k_b)_{b\geq 0}$ has to be uniformly distributed modulo $(q_1^d \cdots q_v^d)$ for all positive integers d".

Example 3. By Theorem 6 and Example 2 the subsequence indexed by $k_b = ub + w$ with integers $u > 0, 0 \leq w < u$ remains uniformly distributed:

- For the van der Corput-Halton sequence in prime bases $q_1, \dots, q_s$ if and only if $\gcd(u, q_1 \cdots q_s) = 1$.
- For every u.d. digital $(\mathbf{T}, s)$-sequence in prime base q if q does not divide u.
- For the van der Corput-Halton sequence in prime bases $q_1, \dots, q_s$ based on the Gray-Code digits if and only if for all $j \in \{1, \dots, s\}$ we have $\gcd(u, q_j^2) \in \{1, q_j\}$.

2.2 Subsequences Indexed by Primes

We cannot find a system of congruences such that its set of solutions is equal to the set of primes $\mathbb{P}$. Nevertheless, parts of the last subsection and the notion of compatibility can be used to investigate the subsequence indexed by primes of a Niederreiter-Halton sequence. We get the following necessary condition for uniform distribution of subsequences indexed by primes.

Proposition 1. *Let $(\mathbf{x}_n)_{n\geq 0}$ be a uniformly distributed digital $(\mathbf{F}, s)$-sequence in bases $((q_1, w_1), \dots, (q_v, w_v))$. If the sequence $(\mathbf{x}_{p_n})_{n\geq 1}$, where p_n denotes the n-th prime, is uniformly distributed, then for each $l \in \{1, \dots, v\}$ the generator matrices $C^{(l,1)}, \dots, C^{(l,w_l)}$ and the additional row $(1000\dots)$ are compatible.*

Proof. We use here the same notations as in the proof of Theorem 6.

Let assume that there exists $l \in \{1, \dots, v\}$ and $d^{(l,1)}, \dots, d^{(l,w_l)}$ such that for any choice of truncation of the matrix-rows the row $(100\dots0)$ (of course truncated as well) can be written as a linear combination of the corresponding $d^{(l,1)} + \cdots + d^{(l,w_l)}$ truncated rows. We consider the projection of the sequence to the corresponding w_l-dimensional unit-cube and choose the following elementary interval

$$I = \left[0, 1/q_l^{d^{(l,1)}}\right) \times \cdots \times \left[0, 1/q_l^{d^{(l,w_l)}}\right).$$

By our assumption that $(100\dots)$ can be written as a linear combination of the corresponding $d^{(l,1)} + \cdots + d^{(l,w_l)}$ rows it is not so hard to check that the condition $\mathbf{x}_{p_n} \in I$ implies $q_l | p_n$.

From the construction principle we get $\mathbf{x}_{p_n} \in I$ is equivalent to

$$\begin{pmatrix} \gamma_{1,0}^{(l,j)} & \gamma_{1,1}^{(l,j)} & \gamma_{1,2}^{(l,j)} & \cdots \\ \vdots & \vdots & \vdots & \cdots \\ \gamma_{d^{(l,j)},0}^{(l,j)} & \gamma_{d^{(l,j)},1}^{(l,j)} & \gamma_{d^{(l,j)},2}^{(l,j)} & \cdots \end{pmatrix} \cdot \begin{pmatrix} p_{n,0}^{(l)} \\ p_{n,1}^{(l)} \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

modulo q_l for all $j \in \{1, \ldots, w_l\}$. We take the mentioned linear combination of the $d^{(l,1)} + \cdots + d^{(l,w_l)}$ rows and get the condition

$$\left(1\ 0\ 0\ 0\ \ldots\right) \cdot \left(p_{n,0}^{(l)}, p_{n,1}^{(l)}, p_{n,2}^{(l)}, \ldots\right)^{\top} = \left(0\right) \in \mathbb{Z}_{q_l}.$$

This implies $q_l | p_n$. Hence I contains at most one point, $\mathbf{x}_{q_l}$. Thus $(\mathbf{x}_{p_n})_{n \geq 1}$ is not uniformly distributed. □

In the case where the generator matrices consist of rows of finite length exclusively the necessary condition in Theorem 1 is also sufficient.

Theorem 7. *Let $(\mathbf{x}_n)_{n \geq 0}$ be a uniformly distributed digital $(\mathbf{F}, s)$-sequence in bases $((q_1, w_1), \ldots, (q_v, w_v))$ generated by matrices consisting of rows of finite length exclusively. The sequence $(\mathbf{x}_{p_n})_{n \geq 1}$, where p_n is the n-th prime, is uniformly distributed if and only if for each $l \in \{1, \ldots, v\}$ the generator matrices $C^{(l,1)}, \ldots, C^{(l,w_l)}$ and the additional row $(1000\ldots)$ are compatible.*

For the proof of this theorem we use the prime number theorem for arithmetic progressions (compare e.g. [1, p.154]).

Lemma 1. *For different primes $q_1, \ldots, q_v$, given integers $d_1, \ldots, d_v > 0$ and r with $\gcd(r, q_1 \cdots q_v) = 1$, we have*

$$\lim_{N \to \infty} \frac{1}{N} \# \left\{ 1 \leq n \leq N : p_n \equiv r \pmod{q_1^{d_1} \cdots q_v^{d_v}} \right\} = \frac{1}{\varphi\left(q_1^{d_1} \cdots q_v^{d_v}\right)},$$

where $\varphi : \mathbb{N} \to \mathbb{N}$ denotes Euler's totient function.

Proof of Theorem 7: Necessity of compatibility follows from Proposition 1.

For the proof of sufficiency we use the notation as in the proof of Theorem 6. In order to prove sufficiency we will deduce for each arbitrary elementary interval I the following relation

$$\lim_{N \to \infty} \frac{1}{N} \#\{1 \leq n \leq N : \mathbf{x}_{p_n} \in I\} = \lambda(I).$$

We choose positive integers $d^{(1,1)}, \ldots, d^{(1,w_1)}, \ldots, d^{(v,1)}, \ldots, d^{(v,w_v)}$ arbitrarily but fixed and set

$$I_a := \prod_{l=1}^{v} \prod_{j=1}^{w_l} \left[\frac{a^{(l,j)}}{q_l^{d^{(l,j)}}}, \frac{a^{(l,j)} + 1}{q_l^{d^{(l,j)}}} \right)$$

with arbitrarily chosen $a^{(l,j)} \in \{0, 1, \ldots, q_l^{d^{(l,j)}} - 1\}$ for all $1 \le j \le w_l$, $1 \le l \le v$.

We define a further parameter-function $\mathbf{L} := (L^{(1)}, \ldots, L^{(v)})$.

For every $l \in \{1, \ldots, v\}$ and arbitrary non-negative integers $d^{(l,1)}, \ldots, d^{(l,w_l)}$, let $L^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) \in \mathbb{N}$ be minimal such that for all $j \in \{1, \ldots, w_l\}$ each of the first $d^{(l,j)}$ rows of $C^{(l,j)}$ has length less than or equal to $L^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)})$. By the "length" of a row $(c_1, c_2, c_3, \ldots)$ we mean $\sup\{k : c_k \neq 0\}$.

From the conditions in Theorem 7 we know that for each $l \in \{1, \ldots, v\}$ we have $L^{(l)} := L^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) < +\infty$,

$$F^{(l)} := F^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) \le L^{(l)}(d^{(l,1)}, \ldots, d^{(l,w_l)}) < +\infty$$

and compatibility of the generator matrices $C^{(l,1)}, \ldots, C^{(l,w_l)}$ and $(1000\ldots)$ for each $l \in \{1, \ldots, v\}$.

From the construction principle of a Niederreiter-Halton sequence and the condition $L^{(l)} < +\infty$ we get the following condition, which is equivalent to $\mathbf{x}_n \in I_a$:

For all $l \in \{1, \ldots, v\}$, $j \in \{1, \ldots, w_l\}$, n solves

$$\begin{pmatrix} \gamma_{1,0}^{(l,j)} & \gamma_{1,1}^{(l,j)} & \gamma_{1,2}^{(l,j)} & \gamma_{1,3}^{(l,j)} & \cdots & \gamma_{1,L^{(l)}-1}^{(l,j)} \\ \gamma_{2,0}^{(l,j)} & \gamma_{2,1}^{(l,j)} & \gamma_{2,2}^{(l,j)} & \gamma_{2,3}^{(l,j)} & \cdots & \gamma_{2,L^{(l)}-1}^{(l,j)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ \gamma_{d^{(l,j)},0}^{(l,j)} & \gamma_{d^{(l,j)},1}^{(l,j)} & \gamma_{d^{(l,j)},2}^{(l,j)} & \gamma_{d^{(l,j)},3}^{(l,j)} & \cdots & \gamma_{d^{(l,j)},L^{(l)}-1}^{(l,j)} \end{pmatrix} \cdot \begin{pmatrix} n_0^{(l)} \\ n_1^{(l)} \\ \vdots \\ n_{L^{(l)}-1}^{(l)} \end{pmatrix} = \begin{pmatrix} a_1^{(l,j)} \\ a_2^{(l,j)} \\ \vdots \\ a_{d^{(l,j)}}^{(l,j)} \end{pmatrix},$$

where the matrix above is the left upper $d^{(l,j)} \times L^{(l)}$-submatrix of $C^{(l,j)}$, $n_0^{(l)} + n_1^{(l)} q_l + \cdots$ is the q_l-ary representation of n and $(0.a_1^{(l,j)} a_2^{(l,j)} \ldots a_{d^{(l,j)}}^{(l,j)})_{q_l}$ is the q_l-ary representation of $a^{(l,j)}/(q_l^{d^{(l,j)}})$.

From the Chinese Remainder Theorem it follows, that whether $\mathbf{x}_n \in I_a$ or not is determined by the residue r of n modulo $Q := \prod_{l=1}^{v} q_l^{L^{(l)}}$.

We define $R := \{r \in \mathbb{Z} : 0 \le r < Q\}$. The set of residues r such that $(\mathbf{x}_{Qb+r})_{b \ge 0} \in I_a$ is denoted by

$$J(I_a) := \{r \in R : (\mathbf{x}_{Qb+r})_{b \ge 0} \in I_a\}.$$

Note that we have $|J(I_a)| = Q/A$, where $A := \prod_{l=1}^{v} q_l^{\sum_{j=1}^{w_l} d^{(l,j)}}$ is the number of different elementary intervals of the same order. Furthermore, the sets $J(I_a)$, where I_a varies over all elementary intervals of fixed order, produce a partition of the set R.

Note that if the sequence of indices $(b_n)_{n \ge 0}$ of a subsequence $(\mathbf{x}_{b_n})_{n \ge 0}$ is uniformly distributed among the residue classes modulo $q_1^d \cdots q_v^d$ for all integers $d \ge 1$, then it is not so hard to prove that the subsequence $(\mathbf{x}_{b_n})_{n \ge 0}$ of a Niederreiter-Halton sequence generated by matrices consisting of rows of finite length exclusively is uniformly distributed modulo one (see proof of [10, Theorem 4.2 (a)]).

Unfortunately, the sequence of primes $(p_n)_{n \ge 1}$ is not uniformly distributed modulo $q_1^{d_1} \cdots q_v^{d_v}$ for any integers $d_l \ge 1$, $l \in \{1, \ldots, v\}$, but it is uniformly distributed

among all residues r with $\gcd(r, q_1 \cdots q_v) = 1$ (see Lemma 1). We will use this fact to deduce uniform distribution of $(\mathbf{x}_{p_n})_{n\geq 1}$ under the conditions in Theorem 7.

In the following we partition the set R again into pairwise disjoint subsets of equal cardinal number. For each $(b_1, \ldots, b_v) \in \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_v}$ we define

$$K(b_1, \ldots, b_v) := \{r \in R : r \equiv b_j \pmod{q_j} \text{ for all } j \in \{1, \ldots, v\}\}.$$

For the cardinality we have

$$|K(b_1, \ldots, b_v)| = \frac{Q}{q_1 \cdots q_v}$$

for each $(b_1, \ldots, b_v) \in \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_v}$.

The set of residues r modulo Q with $\gcd(r, q_1 \cdots q_v) = \gcd(r, Q) = 1$ is given by

$$\bigcup_{b_1=1}^{q_1-1} \cdots \bigcup_{b_v=1}^{q_v-1} K(b_1, \ldots, b_v).$$

Since we have presumed compatibility of the generator matrices $C^{(l,1)}, \ldots, C^{(l,w_l)}$ and $(1000\ldots)$ for each $l \in \{1, \ldots, v\}$ the subsets defined by

$$J'(I_a, b_1, \ldots, b_v) := K(b_1, \ldots, b_v) \cap J(I_a),$$

where $(b_1, \ldots, b_v)$ varies over all possible values in $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_v}$ represent a partition of $J(I_a)$ into $q_1 \cdots q_v$ pairwise disjoint subsets of equal cardinal number $Q/(A \cdot q_1 \cdots q_v)$.

We now compute

$$\begin{aligned}
&\frac{1}{N}\#\{1 \leq n \leq N : \mathbf{x}_{p_n} \in I_a\} \\
&= \frac{1}{N}\#\{1 \leq n \leq N : p_n \equiv r \pmod{Q} \text{ with } r \in J(I_a)\} \\
&= \sum_{b_1=0}^{q_1-1} \cdots \sum_{b_v=0}^{q_v-1} \frac{1}{N}\#\{1 \leq n \leq N : p_n \equiv r \pmod{Q}, r \in J'(I_a, b_1, \ldots, b_v)\} \\
&= \sum_{b_1=0}^{q_1-1} \cdots \sum_{b_v=0}^{q_v-1} \sum_{r \in J'(I_a, b_1, \ldots, b_v)} \underbrace{\frac{1}{N}\#\{1 \leq n \leq N : p_n \equiv r \pmod{Q}\}}_{(**)}.
\end{aligned}$$

By Lemma 1 we have

$$\lim_{N\to\infty} \frac{1}{N}\#\{1 \leq n \leq N : p_n \equiv r \pmod{Q}\} = \begin{cases} \frac{1}{\varphi(Q)} & \text{if } \gcd(r, Q) = 1, \\ 0 & \text{else.} \end{cases}$$

We have $\gcd(r, Q) = 1$ is equivalent to $r \in K(b_1, \ldots, b_v)$ with some $(b_1, \ldots b_v) \in \prod_{l=1}^{v} \{1, \ldots, q_l - 1\}$. Hence

$$\lim_{N\to\infty}\frac{1}{N}\#\{1\le n\le N:\mathbf{x}_{p_n}\in I_a\}=\underbrace{(q_1-1)\cdots(q_v-1)}_{|\prod_{l=1}^{v}\{1,\dots,q_l-1\}|}\cdot\underbrace{\frac{Q}{A\cdot q_1\cdots q_v}}_{|J'(I_a,b_1,\dots,b_v)|}\cdot\underbrace{\frac{1}{\varphi(Q)}}_{\lim_{N\to\infty}(**)}.$$

Since q_l are pairwise different primes we have

$$\varphi(Q)=Q\cdot\frac{q_1-1}{q_1}\cdots\frac{q_v-1}{q_v}.$$

This yields

$$\lim_{N\to\infty}\frac{1}{N}\#\{1\le n\le N:\mathbf{x}_{p_n}\in I_a\}=\frac{1}{A}=\lambda(I_a)$$

which completes the proof of Theorem 7 since I_a was arbitrarily chosen. □

Example 4. By Theorem 7 the van der Corput-Halton sequence in different prime bases indexed by primes is not uniformly distributed. However the subsequence indexed by primes of the van der Corput-Halton sequence in different prime bases based on Gray-Code digits is uniformly distributed. The digital $(0,q)$-sequence in any prime base q as introduced by Faure [5] (also by Sobol [19] in base 2) indexed by primes is not uniformly distributed.

3 Conclusion

In this paper we investigated the distribution properties of subsequences of Niederreiter-Halton sequences. In detail, we considered the primary question, which subsequences are uniformly distributed? We found sufficient as well as necessary conditions for uniform distribution of subsequences determined by indices, which can be interpreted as the solutions of a certain system of congruences. Furthermore, we found a necessary condition for the uniform distribution of subsequences indexed by the set of primes. For the special case where the generator matrices consist of rows of finite length exclusively, we were able to prove sufficiency of this condition as well. Whether this condition is also sufficient in the general case or not is an open question. The method for the investigation of the distribution of Niederreiter-Halton sequences in general is linked to the joint distribution of the integer-weighted sum-of-digits function in different prime bases (see [9]). Until now the subsequence indexed by primes of the joint distribution of the integer-weighted sum-of-digits function in different prime bases is investigated just for the very special one-dimensional and unweighted case (see [15]). Investigation of the joint and weighted case seems to be rather difficult.

The question, if there are new low-discrepancy sequences amongst the Niederreiter-Halton sequences and its subsequences, is still open for future research. First steps concerning the investigation of the discrepancy of Niederreiter-Halton sequences are done in [10] and [11]. It seems that the discrepancy of Niederreiter-Halton sequences and their subsequences can vary significantly with different

choices of the bases and the matrices. The following example points out this behavior.

Example 5. In Reference [10] certain subsequences indexed by arithmetic progressions of the van der Corput-Halton sequences were identified as low-discrepancy sequences (see Theorem 3 and 4). By contrast the digital $(0, 2)$-sequence in base 2 $(\mathbf{x}_n)_{n\geq 0} =: \omega$ as introduced by Sobol [19] and later generalized by Faure [5] is a well known low-discrepancy sequence, i.e.,

$$N D_N(\omega) = O\big(\log^2(N)\big).$$

But for the subsequence $\omega' = (\mathbf{x}_{3n})_{n\geq 0}$ we have the following lower bound

$$N D_N(\omega') \geq c N^{\lambda} \tag{4}$$

for all $N \in \mathbb{N}$ with $0 < \lambda = \log_4(3) < 1$ and certain $c > 0$.

This lower bound is a consequence of the following result of Newman [16] on the sum-of-digits function in base 2:

$$c_1 N^{\lambda} \leq \#\{0 \leq n < N : 2|(s_2(3n))\} - \#\{0 \leq n < N : 2|(s_2(3n) - 1)\} \leq c_2 N^{\lambda}, \tag{5}$$

for all $N \in \mathbb{N}$ where c_1, c_2 are fixed positive constants.

The relation between (4) and (5) becomes clear if we consider that the second component of the sequence ω is generated by the Pascal-matrix in base 2,

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \dots \\ 0 & 1 & 0 & 1 & \dots \\ 0 & 0 & 1 & 1 & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \in \mathbb{Z}_2^{\mathbb{N}\times\mathbb{N}_0}.$$

Hence, a point of the sequence $\omega' = (\mathbf{x}_{3n})_{n\geq 0}$ is included in $[0, 1) \times [0, 1/2)$ iff $s_2(3n)$ is even and it is included in $[0, 1) \times [1/2, 1)$ iff $s_2(3n)$ is odd. Note that (5) yields the same lower bound on the discrepancy of the hybrid sequences introduced in [6] for $s \geq 3$.

Altogether it turned out to be very difficult to give a detailed and complete analysis of the discrepancy of Niederreiter-Halton sequences and their subsequences.

Acknowledgements This research has been supported by a DOC-fFORTE-fellowship of the Austrian Academy of Sciences.

References

1. Apostol, T.M.: Introduction to Analytic Number Theory. Undergraduate Texts in Mathematics. Springer-Verlag, New York (1976)
2. Atanassov, E.: On the discrepancy of the Halton sequences. Mathematica Balkanica **18**, 15–32 (2004)
3. Drmota, M., Larcher, G.: The sum-of-digits function and uniform distribution modulo 1. Journal of Number Theory **89**, 65–96 (2001)
4. Drmota, M., Tichy, R.F.: Sequences, Discrepancies and Applications. Lecture Notes in Mathematics 1651. Springer-Verlag, Berlin (1997)
5. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arithmetica **XLI**, 337–351 (1982)
6. Faure, H.: Méthode quasi-Monte-Carlo multidimensionelles. Theoretical Computer Science **123**, 131–137 (1994)
7. Halton, J.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Mathematik **2**, 84–90 (1960)
8. Hofer, R.: Note on the joint distribution of the weighted sum-of-digits function modulo one in case of pairwise coprime bases. Uniform Distribution Theory **2**(2), 35–47 (2007)
9. Hofer, R.: On the distribution properties of Niederreiter-Halton sequences. Journal of Number Theory **129**, 451–463 (2009)
10. Hofer, R., Kritzer, P., Larcher, G., Pillichshammer, F.: Distribution properties of generalized van der Corput-Halton sequences and their subsequences. International Journal of Number Theory **5**(4), 719–746 (2009)
11. Hofer, R., Larcher, G.: On existence and discrepancy of certain digital Niederreiter-Halton sequences. Acta Arithmetica to appear
12. Kim, D.H.: On the joint distribution of q-additive functions in residue classes. Journal of Number Theory **74**, 307–336 (1998)
13. Kuipers, L., Niederreiter, H.: Uniform Distribution of Sequences. John Wiley & Sons, New York (1974)
14. Larcher, G., Niederreiter, H.: Generalized (t, s)-sequences, Kronecker-type sequences, and diophantine approximations of formal Laurent series. Transactions of the American Mathematical Society **347**, 2051–2073 (1995)
15. Mauduit, C., Rivat, J.: Sur un problème de Gelfond: la somme des chiffres des nombres premiers. Annals of Mathematics to appear
16. Newman, D.J.: On the number of binary digits in a multiple of three. Proceedings of the American Mathematical Society **21**, 719–721 (1969)
17. Niederreiter, H.: Point sets and sequences with small discrepancy. Monatshefte für Mathematik **104**, 273–337 (1987)
18. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia, PA (1992)
19. Sobol, I.: On the distribution of points in a cube and the approximation evaluation of integrals. USSR Computational Mathematics and Mathematical Physics **7**, 86–112 (1967)

Correcting the Bias in Monte Carlo Estimators of American-style Option Values

K.H. Felix Kan, R. Mark Reesor, Tyson Whitehead, and Matt Davison

Abstract Existing Monte Carlo estimators of American option values are consistent but biased. This article presents a general bias reduction technique which corrects the bias due to making suboptimal exercise decisions. The derived asymptotic expression for the bias is independent of dimensionality, holds for very general underlying processes and option payoffs, and is easily evaluated. The bias is subtracted from the estimators at each exercise opportunity in order to produce bias-corrected estimators. We illustrate how to apply this technique to three methods of generating estimators — stochastic tree, stochastic mesh and least-squares Monte Carlo. Numerical results demonstrate that for a fixed sample size this technique significantly reduces the relative error for both high- and low-biased estimators.

1 Introduction

Pricing American options is difficult because the option holder has the right to exercise before the maturity date. This is an optimal stopping-time problem in the sense that the owner chooses the exercise time to maximize option value. Hence, given that the option has not yet been exercised at time t, its value is

$$B_t = \sup_{t \le \tau \le T} \mathbb{E}[\, P_\tau \,|\, \mathcal{F}_t \,], \tag{1}$$

K.H. Felix Kan
Department of Applied Mathematics, The University of Western Ontario, Canada
e-mail: kkan8@uwo.ca

R. Mark Reesor, and Matt Davison
Departments of Applied Mathematics and of Statistical and Actuarial Sciences, The University of Western Ontario, Canada
e-mail: mreesor@uwo.ca, e-mail: mdavison@uwo.ca

Tyson Whitehead
SHARCNET, Canada
e-mail: tyson@sharcnet.ca

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_28,

where the supremum is taken over all possible stopping times with values in the interval $[t, T]$, T is the maturity, P_t is the discounted exercise value at time t and $\mathcal{F}_t$ is the filtration generated by the underlying processes. We focus on valuation, so work with an equivalent martingale measure. Without loss of generality and with a gain in clarity, the discounting factor is suppressed throughout this article.

Numerically, an American option value can be approximated by a Bermudan option value, i.e.,

$$B_k = \max_{\tau \in [k,\ldots,N]} \mathbb{E}[\, P_\tau \,|\, \mathcal{F}_k \,], \tag{2}$$

where the maximum is taken over all possible discrete stopping times from k to N and time N is the option maturity. Without ambiguity, we use the term American option instead of Bermudan option. Several Monte Carlo (MC) approaches have been developed to estimate the American option value, ranging from earlier work that focused on parameterizing the early-exercise region based on the form of the option payoff [17, 4, 10], to recent work on a dual formulation of the problem [1, 12, 16]. In this article, we concentrate on techniques for computing the continuation value of the contingent claim, which include Broadie and Glasserman's stochastic tree and mesh [5, 7], and a modified version of Longstaff and Schwartz's least-squares Monte Carlo (LSM) [14] (earlier variants were done by Carrière [8], and Tsitsiklis and Van Roy [18]).

Valuation of American options can be broken down into two recursive equations

$$H_k = \mathbb{E}[\, B_{k+1} \,|\, \mathcal{F}_k \,] \quad \text{and} \tag{3}$$

$$B_k = \max(H_k, P_k), \tag{4}$$

where H_k is the value of holding the option until at least the next exercise opportunity and B_k is the current value of the option (i.e., the greater of the value of holding or exercising). The terminal condition is $H_N = 0$ since there is no value in holding the option past expiry.

In MC valuation algorithms an estimator of the continuation value, $\tilde{H}_k$, is used in (3) and (4), giving an option-value estimator of $\tilde{B}_k = \max(\tilde{H}_k, P_k)$. MC estimators of the continuation value from the stochastic tree, stochastic mesh, and LSM are biased but consistent [5, 7, 3, 14, 9]. Furthermore, estimators are either high- or low-biased and have a close relative with a bias of opposite sign. Though there has been much work on improving the efficiency of MC estimators through the use of variance-reduction techniques, very little work has been done on reducing estimator bias. The obvious approach of averaging the final high- and low- biased estimators is unsuccessful since the estimators are asymmetric about the true value. A more successful (ad hoc) approach is to recursively average the high- and low-biased estimators at each exercise opportunity. This is shown to be reasonably successful for the stochastic mesh [2], though it has not been shown to be effective for the stochastic tree and LSM. Two other works that focus on estimator bias are [8] and [6]. In the former a corrected regression-based estimator is generated at the cost of a significant increase in estimator variance. In [6] a nonparametric bootstrap is used to estimate the bias and this is subtracted from the uncorrected stochastic tree

estimators. The cost of this correction is a significant increase in computational time and complexity.

Some recent work uses large-sample theory to derive and rigorously justify an approximation to the stochastic tree estimator bias [20, 21, 22]. In [20] and [21] heuristic derivations for the approximation to the bias are given for both high- and low-biased stochastic tree estimators. [21] contains an extensive numerical study showing the significantly-increased convergence rate of the corrected estimators. The approximation to the bias is rigorously justified for the high-biased stochastic tree estimator in [20] and [22]. Furthermore, in [20], corrected versions of the high- and low-biased stochastic mesh estimators are presented along with substantial numerical results showing the efficacy of the method.

In Section 2 of this paper, the heuristic derivation for the bias approximation is generalized to accommodate each of the high-biased stochastic tree, mesh and LSM estimators. The high-biased estimators considered here are those in which the determiner (estimator used for exercise decision) and the propagator (estimator passed on to preceding exercise opportunity) are the same. A corresponding derivation for the low-biased case in which the continuation value estimators and the recursive equations are different from those of the high-biased estimators exists but is not presented here due to space constraints. The approximation for the bias is used to construct bias-corrected estimators for the stochastic tree, stochastic mesh, and LSM in Section 3. Numerical results obtained by applying the bias-correction technique to a well-studied multivariate pricing problem via these three pricing methods are presented in Section 4. The numerical results presented here for the tree and mesh are a small subset of those given in [21] and [20], respectively. The bias-corrected LSM estimators presented here are unique, as are the numerical results showing the increased convergence rate of the corrected LSM estimators. Section 5 concludes the paper.

2 Bias Correction

In this section, a heuristic derivation of an approximation to the time-k estimator bias is presented. This relies on a normal approximation to the distribution of the hold-value estimator. The result is then applied in Section 3 to estimators from the stochastic tree, stochastic mesh and LSM methods. Arguments similar to these for stochastic tree estimators appear in [21] and [20].

To begin, let $\bar{H}_k = \mathbb{E}[\tilde{H}_k|\mathcal{F}_k]$. The time-$k$ bias is defined as $\bar{H}_k - H_k = \mathbb{E}[\tilde{B}_{k+1} - B_{k+1}|\mathcal{F}_k]$. An estimator is high-biased if $\bar{H}_k - H_k > 0$ and is low-biased if $\bar{H}_k - H_k < 0$. Expanding the inner terms of $\mathbb{E}[\tilde{B}_{k+1} - B_{k+1}|\mathcal{F}_k]$ gives

$$\mathbb{E}\left[\max(\tilde{H}_{k+1}, P_{k+1}) - \max(H_{k+1}, P_{k+1})\middle|\mathcal{F}_k\right]. \tag{5}$$

Adding and subtracting $\mathbb{E}[\max(\bar{H}_{k+1}, P_{k+1})|\mathcal{F}_k]$ splits this expression into a local (6) and a global (7) component

$$\mathbb{E}\left[\max(\tilde{H}_{k+1}, P_{k+1}) - \max(\bar{H}_{k+1}, P_{k+1})\middle|\mathcal{F}_k\right] \tag{6}$$

$$+\mathbb{E}\left[\max(\bar{H}_{k+1}, P_{k+1}) - \max(H_{k+1}, P_{k+1})\middle|\mathcal{F}_k\right]. \tag{7}$$

We return to the global component, which represents accumulated bias, at the end of this section and focus on the local component for now. Let $\mathbb{1}_A$ be an indicator function which is equal to one on the set A and is equal to zero otherwise. The $\mathcal{F}_{k+1}$-conditional expectation (and, by nested expectation, the $\mathcal{F}_k$-conditional expectation) of $\mathbb{1}_{\bar{H}_{k+1}>P_{k+1}}(\tilde{H}_{k+1} - \bar{H}_{k+1})$ is zero as $\mathbb{1}_{\bar{H}_{k+1}>P_{k+1}}$ is $\mathcal{F}_{k+1}$-measurable and $\mathbb{E}[\tilde{H}_{k+1}|\mathcal{F}_{k+1}] = \bar{H}_{k+1}$. Therefore, this term can be subtracted inside the local bias expectation (6) without altering the expected value. Doing this, and expressing the max with indicator functions, gives

$$\mathbb{E}\Big[\mathbb{1}_{\bar{H}_{k+1}>P_{k+1}}\mathbb{1}_{\tilde{H}_{k+1}\le P_{k+1}}(P_{k+1} - \tilde{H}_{k+1}) + \mathbb{1}_{\bar{H}_{k+1}\le P_{k+1}}\mathbb{1}_{\tilde{H}_{k+1}>P_{k+1}}(\tilde{H}_{k+1} - P_{k+1})\Big|\mathcal{F}_k\Big] \tag{8}$$

as an equivalent local bias expression. Rewritten using $\tilde{Y}_{k+1} = \tilde{H}_{k+1} - P_{k+1}$ and $\bar{Y}_{k+1} = \bar{H}_{k+1} - P_{k+1}$, this is

$$\mathbb{E}\left[\mathbb{1}_{\bar{Y}_{k+1}>0}\mathbb{1}_{\tilde{Y}_{k+1}\le 0}(-\tilde{Y}_{k+1}) + \mathbb{1}_{\bar{Y}_{k+1}\le 0}\mathbb{1}_{\tilde{Y}_{k+1}>0}(\tilde{Y}_{k+1})\middle|\mathcal{F}_k\right]. \tag{9}$$

Table 1 summarizes (9). It is evident that the local bias component is solely due to exercising incorrectly (i.e., making the wrong choice between holding and exercising). This implies that significant contributions are limited to the region about the exercise boundary as even poor estimators are unlikely to result in incorrect exercising away from the boundary. It is also evident why the estimator is high-biased.

Table 1 Local error in the time-$(k+1)$ hold-value estimator.

	Held: $\tilde{Y}_{k+1} > 0$	Exercised: $\tilde{Y}_{k+1} \le 0$
Should Hold: $\bar{Y}_{k+1} > 0$	0	$-\tilde{Y}_{k+1}$
Should Exercise: $\bar{Y}_{k+1} \le 0$	$\tilde{Y}_{k+1}$	0

Equation (9) is not too valuable numerically as $\bar{Y}_{k+1}$ is not directly observable and replacing it with an estimator immediately collapses the expression to zero. It is necessary to incorporate additional distributional knowledge. Monte Carlo estimators are (possibly weighted) averages of a typically large number of random variables. Under very general conditions, the distribution of a normalized and centered

average or weighted average tends towards that of a normal random variable as the sample size approaches infinity. Many versions of the central limit theorem (CLT) provide such a result, ranging from averages of independent, identically distributed random variables to averages of dependent, non-identically distributed random variables [19]. Thus, the normal distributional approximation for the distribution of MC estimators of the hold value is reasonable. We provide further discussion of this in Section 3 for each of the tree, mesh and LSM estimators.

Returning to the derivation, let $\bar{V}_{k+1}$ be the $\mathcal{F}_{k+1}$-conditional variance of $\tilde{B}_{k+2}$ and assume that $\tilde{Y}_{k+1}$ in (9) can be replaced by $\tilde{Y}^*_{k+1}$, where the latter is a normally distributed random variable with mean $\bar{Y}_{k+1}$ and variance $\bar{V}_{k+1}/M$ (conditional on $\mathcal{F}_{k+1}$), and M is the sample size used to calculate $\tilde{Y}_{k+1}$. The bias expression then becomes

$$\mathbb{E}\left[\mathbb{1}_{\bar{Y}_{k+1}>0}\mathbb{1}_{\tilde{Y}^*_{k+1}\le 0}(-\tilde{Y}^*_{k+1}) + \mathbb{1}_{\bar{Y}_{k+1}\le 0}\mathbb{1}_{\tilde{Y}^*_{k+1}>0}(\tilde{Y}^*_{k+1})\middle|\mathcal{F}_k\right]. \tag{10}$$

Let $f_{\bar{Y}_{k+1},\bar{V}_{k+1}|\mathcal{F}_k}$ and $f_{\tilde{Y}^*_{k+1},\bar{V}_{k+1}|\mathcal{F}_k}$ denote the $\mathcal{F}_k$-conditional joint density function of $(\bar{Y}_{k+1},\bar{V}_{k+1})$ and $(\tilde{Y}^*_{k+1},\bar{V}_{k+1})$, respectively. Writing the bias expression in integral form gives

$$\int_0^\infty\int\int_D |\tilde{y}^*|\,\frac{1}{\sqrt{\bar{v}/M}}\phi\left(\frac{\tilde{y}^*-\bar{y}}{\sqrt{\bar{v}/M}}\right) f_{\bar{Y}_{k+1},\bar{V}_{k+1}|\mathcal{F}_k}(\bar{y},\bar{v})\,\mathrm{d}\tilde{y}^*\mathrm{d}\bar{y}\,\mathrm{d}\bar{v}, \tag{11}$$

where $D=(0,\infty)\times(-\infty,0]\cup(-\infty,0]\times(0,\infty)$ and ϕ is the standard normal density function. There are two underlying scales in this integral — the distribution of $\bar{Y}_{k+1}$ is nearly invariant as M changes, whereas the conditional distribution of $\tilde{Y}^*_{k+1}$ converges towards a delta function centered at $\bar{Y}_{k+1}$. Substituting $\bar{z}=\bar{y}\sqrt{M}$ and $\tilde{z}^*=\tilde{y}^*\sqrt{M}$ separates these two, giving

$$\frac{1}{M}\int_0^\infty\int\int_D |\tilde{z}^*|\,\frac{1}{\sqrt{\bar{v}}}\phi\left(\frac{\tilde{z}^*-\bar{z}}{\sqrt{\bar{v}}}\right) f_{\bar{Y}_{k+1},\bar{V}_{k+1}|\mathcal{F}_k}\left(\frac{\bar{z}}{\sqrt{M}},\bar{v}\right)\mathrm{d}\tilde{z}^*\mathrm{d}\bar{z}\,\mathrm{d}\bar{v}. \tag{12}$$

We see the time-k local bias is $O(1/M)$ due to the combined ($O(1/\sqrt{M})$) effects of a decreasing probability of making incorrect stopping decisions and an increasing probability of those that are made being less significant.

Note that $\tilde{Y}^*_{k+1}$, $\bar{Y}_{k+1}$ and $\bar{V}_{k+1}$ converge to Y_{k+1}, Y_{k+1} and V_{k+1}, respectively, where V_{k+1} is the $\mathcal{F}_{k+1}$-conditional variance of B_{k+2}. Therefore, assume both $f_{\bar{Y}_{k+1},\bar{V}_{k+1}|\mathcal{F}_k}(\bar{z}/\sqrt{M},\bar{v})$ and $f_{\tilde{Y}^*_{k+1},\bar{V}_{k+1}|\mathcal{F}_k}(\tilde{z}^*/\sqrt{M},\bar{v})$ converge to $f_{Y_{k+1},V_{k+1}|\mathcal{F}_k}(0,\bar{v})$ [20, 22]. Assuming $\lim_M$ commutes with the integration, (12) then becomes asymptotically equivalent to

$$\frac{1}{M}\int_0^\infty\int\int_D |\tilde{z}^*|\,\frac{1}{\sqrt{\bar{v}}}\phi\left(\frac{\tilde{z}^*-\bar{z}}{\sqrt{\bar{v}}}\right) f_{\tilde{Y}^*_{k+1},\bar{V}_{k+1}|\mathcal{F}_k}\left(\frac{\tilde{z}^*}{\sqrt{M}},\bar{v}\right)\mathrm{d}\tilde{z}^*\mathrm{d}\bar{z}\,\mathrm{d}\bar{v}. \tag{13}$$

Undoing the $\tilde{z}^*$ and $\bar{z}$ substitutions gives

$$\int_0^\infty \int \int_D |\tilde{y}^*| \, \frac{1}{\sqrt{\bar{v}/M}} \phi\left(\frac{\tilde{y}^* - \bar{y}}{\sqrt{\bar{v}/M}}\right) f_{\tilde{Y}^*_{k+1}, \bar{V}_{k+1}|\mathcal{F}_k}(\tilde{y}^*, \bar{v}) \, \mathrm{d}\tilde{y}^* \mathrm{d}\bar{y} \, \mathrm{d}\bar{v}. \tag{14}$$

This expression is special because the $\bar{y}$ integral can be performed. This yields

$$\int_0^\infty \int_{-\infty}^\infty |\tilde{y}^*| \, \Phi\left(\frac{-|\tilde{y}^*|}{\sqrt{\bar{v}/M}}\right) f_{\tilde{Y}^*_{k+1}, \bar{V}_{k+1}|\mathcal{F}_k}(\tilde{y}^*, \bar{v}) \, \mathrm{d}\tilde{y}^* \, \mathrm{d}\bar{v}, \tag{15}$$

where Φ is the standard normal cumulative distribution function. In expectation form, this is

$$\mathbb{E}\left[|\tilde{Y}^*_{k+1}| \, \Phi\left(\frac{-|\tilde{Y}^*_{k+1}|}{\sqrt{\bar{V}_{k+1}/M}}\right) \middle| \mathcal{F}_k \right]. \tag{16}$$

In order to utilize (16) it is necessary to assume that sample quantities $(\tilde{Y}_{k+1}, \tilde{V}_{k+1})$ can be substituted for the idealized quantities $(\tilde{Y}^*_{k+1}, \bar{V}_{k+1})$. Doing so yields

$$\mathbb{E}\left[|\tilde{Y}_{k+1}| \, \Phi\left(\frac{-|\tilde{Y}_{k+1}|}{\sqrt{\tilde{V}_{k+1}/M}}\right) \middle| \mathcal{F}_k \right]. \tag{17}$$

Subtracting (17) from the bias gives

$$\mathbb{E}[\tilde{H}_k|\mathcal{F}_k] - H_k - \mathbb{E}\left[|\tilde{Y}_{k+1}| \, \Phi\left(\frac{-|\tilde{Y}_{k+1}|}{\sqrt{\tilde{V}_{k+1}/M}}\right) \middle| \mathcal{F}_k \right] \tag{18}$$

$$= \mathbb{E}\left[\max(\tilde{H}_{k+1}, P_{k+1}) - \max(\bar{H}_{k+1}, P_{k+1}) \middle| \mathcal{F}_k \right] \tag{19}$$

$$+ \mathbb{E}\left[\max(\bar{H}_{k+1}, P_{k+1}) - \max(H_{k+1}, P_{k+1}) \middle| \mathcal{F}_k \right] \tag{20}$$

$$- \mathbb{E}\left[|\tilde{H}_{k+1} - P_{k+1}| \, \Phi\left(\frac{-|\tilde{H}_{k+1} - P_k|}{\sqrt{\tilde{V}_{k+1}/M}}\right) \middle| \mathcal{F}_k \right], \tag{21}$$

where (19), (20) and (21) are the local bias, global bias and correction components, respectively.

The local and correction components asymptotically cancel as the sample size gets large, leaving just the global component. Applying Jensen's inequality to move the absolute value inside the expectation and applying the inequality $|\max(x, y) - \max(u, v)| \leq |x - u| + |y - v|$ to the absolute value of the global component gives

$$\left| \mathbb{E}\left[\max(\bar{H}_{k+1}, P_{k+1}) - \max(H_{k+1}, P_{k+1}) \middle| \mathcal{F}_k \right] \right| \tag{22}$$

$$\leq \mathbb{E}\left[\left| \bar{H}_{k+1} - H_{k+1} \right| \middle| \mathcal{F}_k \right] = \mathbb{E}\left[\left| \mathbb{E}\left[\tilde{H}_{k+1} \middle| \mathcal{F}_{k+1} \right] - H_{k+1} \right| \middle| \mathcal{F}_k \right], \tag{23}$$

which shows it to be bound by the time-$(k+1)$ bias. Similarly the time-$(k+1)$ bias is bound by the time-$(k+2)$ bias. Continue in this fashion through to the next-to-last exercise opportunity $(N-1)$ and note that the time-$(N-1)$ hold-value estimator is unbiased. Thus, the global bias is also accounted for. Specifically, the propagation

of bias across exercise opportunities is at most of the same order as the difference between the local bias and the correction component.

As a result, the corrected option-value estimator is obtained by subtracting the term that approximates the bias from the original estimator in (4), namely

$$\tilde{B}_k = \max(\tilde{H}_k, P_k) - |\tilde{H}_k - P_k| \, \Phi\left(\frac{-|\tilde{H}_k - P_k|}{\sqrt{\tilde{V}_k/M}}\right). \tag{24}$$

This general expression is applicable to stochastic tree, stochastic mesh, and LSM estimators. For a given sample size, its effectiveness at correcting the bias relies on the accuracy of (i) the normal distributional approximation and (ii) the sample variance estimator used in place of the true variance.

3 Applications

3.1 Stochastic Tree

This is the most intuitive method to approximate the conditional expectation defining H_k in (3). From each node, a finite number of iid paths are simulated using the underlying asset-price processes. The option values can then be evaluated at each node based on the simulated values. Working backward from the maturity date, the continuation values are estimated at each node by averaging the option values at successive nodes across sample paths, hence determining the optimal stopping times. More details about the stochastic tree estimator are given in [5].

Let each path be identified by a vector of indices $\mathbf{i} = i_1, ..., i_N$ recording its branching history from the root, and let each node along a path $\mathbf{i}$ be identified by its depth k. The recursive equations for the high-biased stochastic tree estimator are

$$\tilde{H}_k^{\mathbf{i}} = \frac{1}{M} \sum_{i_{k+1}=1}^{M} \tilde{B}_{k+1}^{\mathbf{i}} \quad \text{and} \tag{25}$$

$$\tilde{B}_k^{\mathbf{i}} = \max(\tilde{H}_k^{\mathbf{i}}, P_k^{\mathbf{i}}), \tag{26}$$

where $\tilde{H}_k^{\mathbf{i}}$ is the time-k, path-$\mathbf{i}$ continuation value estimator, $\tilde{B}_k^{\mathbf{i}}$ is the time-k, path-$\mathbf{i}$ option-value estimator, $P_k^{\mathbf{i}}$ is the time-k, path-$\mathbf{i}$ exercise value, and M is the number of branches emanating from each node. We also have the terminal condition that $\tilde{H}_N^{\mathbf{i}} = 0$ for all $\mathbf{i}$.

The normal approximation to the distribution of the time-k continuation value estimator follows from the condition that $\mathrm{Var}(\tilde{B}_{k+1}^{\mathbf{i}} | \mathcal{F}_k) < \infty$. The branches emanating from each node of the tree give option-value estimators, $\tilde{B}_{k+1}^{\mathbf{i}}$, that are iid random variables given $\mathcal{F}_k$. In this case the conditions for the standard CLT are satisfied, hence validating the distributional approximation used in going from (9) to (10). Further, note that $\tilde{V}_k^{\mathbf{i}}$ is an unbiased and consistent estimator of $V_k^{\mathbf{i}}$, as the $\tilde{B}_k^{\mathbf{i}}$

are iid random variables given $\mathcal{F}_k$. Thus, the corrected estimator is

$$\tilde{B}_k^{\mathbf{i}} = \max(\tilde{H}_k^{\mathbf{i}}, P_k^{\mathbf{i}}) - |\tilde{H}_k^{\mathbf{i}} - P_k^{\mathbf{i}}| \, \Phi\left(\frac{-|\tilde{H}_k^{\mathbf{i}} - P_k^{\mathbf{i}}|}{\sqrt{\tilde{V}_k^{\mathbf{i}}/M}}\right), \tag{27}$$

where the sample variance is

$$\tilde{V}_k^{\mathbf{i}} = \frac{1}{M-1} \sum_{i_{k+1}=1}^{M} \left(\tilde{B}_{k+1}^{\mathbf{i}} - \frac{1}{M} \sum_{i_{k+1}=1}^{M} \tilde{B}_{k+1}^{\mathbf{i}} \right)^2. \tag{28}$$

When the corrected option-value estimators for the stochastic tree are averaged to obtain the continuation value estimator at the previous step, an estimate of (21) is formed by the law of large numbers that asymptotically cancels with the local bias (19). This gives rise to the following key result which is formally proven in [20] and [22] for the high-biased estimator.

Theorem 1. (*Bias Correction to* $o(1/M)$)
Suppose the following conditions are satisfied: For all k,

1. *the fourth moment of* P_{k+1} *is finite;*
2. V_{k+1} *is bounded about the exercise boundary;*
3. *the* $\mathcal{F}_k$*-conditional density function of* Y_{k+1} *is bounded about the exercise boundary and is continuous there;*
4. *the* $\mathcal{F}_k$*-conditional joint density function of* Y_{k+1} *and* V_{k+1} *exists about the exercise boundary (possibly in a singular form) and is continuous there.*

Then, as $M \to \infty$,

$$M|\mathbb{E}\left[\tilde{H}_k^c \middle| \mathcal{F}_k\right] - H_k| \to 0 \tag{29}$$

almost surely, where $\tilde{H}_k^c$ *is the bias-corrected time-k hold-value estimator.*

3.2 Stochastic Mesh

Although the stochastic tree method is very easy to understand, it is impractical since the number of sample paths increases exponentially with the number of exercise opportunities. The stochastic mesh method overcomes this problem by simulating a finite set of M sample paths of the underlying and performing dynamic programming on this fixed set of paths. At each of the time-k nodes in this mesh, the option values at each of the time-$(k+1)$ nodes are used to construct the continuation values. In this construction, weights that describe a change-of-measure are applied to each of the time-$(k+1)$ values to conform with the assumed probability model for the underlying.

The recursive equations for the high-biased stochastic mesh estimator are

$$\tilde{H}_k^i = \frac{1}{M}\sum_{j=1}^{M} \tilde{B}_{k+1}^j \omega_{k+1}^{i,j} \quad \text{and} \tag{30}$$

$$\tilde{B}_k^i = \max(\tilde{H}_k^i, P_k^i), \tag{31}$$

where $\tilde{H}_k^i$ is the time-k, path-i continuation value estimator, $\tilde{B}_k^i$ is the time-k, path-i option-value estimator, P_k^i is the time-k, path-i exercise value, M is the mesh size or the number of sample paths, and $\omega_{k+1}^{i,j}$ are weights describing transition probabilities between the path i at time k and the path j at time $(k+1)$. The terminal condition is $\tilde{H}_N^i = 0$ for all i.

Note that there are a number of ways to generate the mesh [11]. No matter the method of mesh construction, $\tilde{H}_k^i$ is a weighted average of dependent, identically distributed random variables, hence the standard CLT does not apply. This dependence also poses a challenge when estimating the variance of $\tilde{H}_k^i$ since it is non-trivial to estimate the covariances of the weighted option-value estimators. Current research focuses on computing an unbiased and consistent variance estimator for $\tilde{H}_k^i$ and on developing rigorous arguments justifying the normal distributional approximation.

Given the generality under which CLT's apply we propose the corrected estimator

$$\tilde{B}_k^i = \max(\tilde{H}_k^i, P_k^i) - |\tilde{H}_k^i - P_k^i|\, \Phi\left(\frac{-|\tilde{H}_k^i - P_k^i|}{\sqrt{\tilde{V}_k^i/M}}\right), \tag{32}$$

where the sample variance is

$$\tilde{V}_k^i = \frac{1}{M-1}\sum_{j=1}^{M}\left(\tilde{B}_{k+1}^j \omega_{k+1}^{i,j} - \frac{1}{M}\sum_{j=1}^{M} \tilde{B}_{k+1}^j \omega_{k+1}^{i,j}\right)^2. \tag{33}$$

This variance estimator neglects the dependence between the option value estimators and is, in general, biased. It is difficult to determine whether the option-value estimators are positively or negatively correlated. In the former case (33) underestimates the true variance and the bias correction will be too small, whereas in the latter case, (33) overestimates the true variance and the bias correction will be too large. The numerical results presented in Section 4 indicate that significant improvements in the estimator convergence rate are realized using (32) and (33).

3.3 LSM

LSM is similar to the stochastic mesh method. As with the stochastic mesh, dynamic programming is done on a fixed set of M simulated sample paths. The LSM estimator is similar to the stochastic mesh estimator described in [7] except that the weights implied by a regression are used to replace the likelihood ratio mesh weights. Compared with the stochastic mesh, LSM is more popular among practitioners because it can be implemented more efficiently. We modify Longstaff and

Schwartz's algorithm by using the continuation values estimated by regression for both determiners and propagators, where the latter are discounted cash flows in their original algorithm [14]. This modification usually generates high-biased estimates. A proper choice of basis functions in the regression is critical to the success of the LSM method.

The conditional expectations defining the continuation values are approximated by the fitted values of a regression, where the regression coefficients are estimated by the least-squares method (hence the name least-squares Monte Carlo). Specifically, discounted approximate option values at time $(k+1)$ are regressed against a set of basis functions evaluated at time k, which should be related to the underlying processes and the payoff function. Consider the linear regression

$$\tilde{B}^i_{k+1} = x^i_k \beta_k + \epsilon^i_k, \qquad i = 1, 2, ..., M, \tag{34}$$

where x^i_k is a $(1 \times p)$ vector of basis functions evaluated at time k for path i, β_k is a $(p \times 1)$ vector of regression coefficients, ϵ^i_k is the time-k, path-i error term, $\tilde{B}^i_{k+1}$ is the time-$(k+1)$, path-i option-value estimator, M is the number of sample paths and p is the number of basis functions. In matrix form this becomes

$$\tilde{\mathbf{B}}_{k+1} = X_k \beta_k + \epsilon_k, \tag{35}$$

where $\tilde{\mathbf{B}}_{k+1} = (\tilde{B}^1_{k+1}, \ldots, \tilde{B}^M_{k+1})'$, $X_k = ((x^1_k)', \ldots, (x^M_k)')'$, $\epsilon_k = (\epsilon^1_k, \ldots, \epsilon^M_k)'$ and $'$ denotes transpose.

We use standard assumptions on the errors, namely that $\mathbb{E}[\epsilon_k | \mathcal{F}_k] = \mathbf{0}$ and $\mathbb{E}[\epsilon_k \epsilon_k' | \mathcal{F}_k] = \text{diag}(\sigma^2_{k,1}, \ldots, \sigma^2_{k,M}) \equiv W_k$, where $\mathbf{0}$ is the column vector of zeros, $\text{diag}(a_1, \ldots, a_M)$ is the diagonal matrix with entries $(a_1, \ldots, a_M)$ and $\sigma_{k,i}$'s are constants which could be different for different values of i. Note that the assumption of independence can be relaxed and 2-stage least squares can be used to obtain consistent regression estimators having the desired large-sample distributional properties. This approach allows one to incorporate dependence among the option-value estimators at the cost of increased computational work. Such a study is the focus of ongoing work.

The ordinary least-squares regression estimators are

$$\tilde{\beta}_k = \left(X_k' X_k\right)^{-1} X_k' \tilde{\mathbf{B}}_{k+1}. \tag{36}$$

With the above assumptions on the errors it is seen that

$$\mathbb{E}\left[\tilde{\beta}_k \middle| \mathcal{F}_k\right] = \beta_k \quad \text{and} \tag{37}$$

$$\text{Var}\left[\tilde{\beta}_k \middle| \mathcal{F}_k\right] = \left(X_k' X_k\right)^{-1} X_k' W_k X_k \left(X_k' X_k\right)^{-1} \equiv \frac{\bar{V}_k}{M}. \tag{38}$$

Under general conditions, standard regression theory dictates a multivariate normal approximation to the distribution of $\tilde{\beta}_k$ (conditional on $\mathcal{F}_k$) [19]. Specifically,

$$\tilde{\beta}_k|\mathcal{F}_k \sim \mathcal{MVN}\left(\beta_k, \frac{\bar{V}_k}{M}\right), \tag{39}$$

where $\mathcal{MVN}(\tilde{\mu}, \Sigma)$ denotes a multivariate normal random vector with mean vector $\tilde{\mu}$ and variance-covariance matrix Σ. An application of the Cramer-Wold device yields the approximate conditional distribution of the time-k, path-i hold-value estimator

$$\tilde{H}_k^i = x_k^i \tilde{\beta}_k|\mathcal{F}_k \sim \mathcal{N}\left(x_k^i \beta_k, \frac{x_k^i \bar{V}_k (x_k^i)'}{M}\right), \tag{40}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal random variable with mean μ and variance σ^2. For LSM estimators, these arguments provide the intuition for the substitution that takes (9) to (10).

The recursive equations for the LSM high-biased estimators are

$$\tilde{H}_k^i = x_k^i \tilde{\beta}_k \quad \text{and} \tag{41}$$

$$\tilde{B}_k^i = \max(\tilde{H}_k^i, P_k^i) \tag{42}$$

with the terminal condition $\tilde{H}_N^i = 0$ for all i.

The corrected LSM estimator is

$$\tilde{B}_k^i = \max(\tilde{H}_k^i, P_k^i) - |\tilde{H}_k^i - P_k^i| \, \Phi\left(\frac{-|\tilde{H}_k^i - P_k^i|}{\sqrt{x_k^i \tilde{V}_k (x_k^i)'/M}}\right), \tag{43}$$

where $\tilde{V}_k/M = \left(X_k' X_k\right)^{-1} X_k' \tilde{W}_k X_k \left(X_k' X_k\right)^{-1}$, $\tilde{W}_k = \text{diag}(\tilde{\epsilon}_{k,1}^2, \ldots, \tilde{\epsilon}_{k,M}^2)$ and $\tilde{\epsilon}_{k,i}^2 = (\tilde{B}_{k+1}^i - x_k^i \tilde{\beta}_k)^2$. The assumption of independent errors is not generally appropriate as $\tilde{H}_k^i$ is a weighted average of dependent, non-identically distributed random variables. Thus, a similar comment to that given at the end of Section 3.2 concerning the effect of using a biased and inconsistent variance estimator in the correction term applies here.

3.3.1 Remark

In deriving the approximate bias expression for the LSM estimator, it is assumed that the uncorrected estimator is consistent given a finite set of basis functions. In general, this is not true. In fact, the estimator is only consistent for the true approximation value for this set of basis functions. If the true approximation value is equal to the true option value, then the estimator is consistent for the true option value. Otherwise, there exists another bias outside the scope of our method. As a consequence, this bias reduction method is not designed to address the choice of basis functions, a separate problem beyond the scope of this article.

4 Numerical Results

The bias-corrected estimators are tested on a well-studied example of [7] — an American-style max-max-call option with five underlying assets and a maturity of three years. Its payoff function is

$$(\max(S_T^1, ..., S_T^5) - K)_+ \tag{44}$$

where K is the strike price, $(a)_+$ denotes $\max(a, 0)$, and $S^1, ..., S^5$ are the underlying asset-price processes. These processes are modeled with uncorrelated geometric Brownian motions, pay a continuous dividend of 10% and have a volatility of 20%. The initial prices are all \$90, the strike price is \$100 and the risk-free interest rate is 5%. The option has three exercise opportunities, one per year.

The numerical example includes the uncorrected and corrected low-biased estimators. Standard path estimators are adopted as the low-biased estimators. A standard path estimator uses one sample path for the propagator that is independent of another set of sample paths used to obtain the determiner. The readers are referred to [11] for a more detailed discussion of the standard path estimator for the stochastic tree, stochastic mesh and LSM estimators. The correction terms can be derived in a similar fashion as that for the high-biased estimators [21], but their final expressions are not reported in this article due to space constraints.

For each sample size, M, a number of independent repeated valuations are performed. The option-value estimators for each repeated valuation are averaged to give the sample-size-M option value estimator. The combinations of number of repeated valuations and sample size are chosen for each method to (i) keep the computational time approximately the same across combinations; and (ii) yield an estimator standard error of less than 0.02. The combinations used are given in Table 2. For a given sample size, the estimator standard error is the standard deviation of the hold value estimators divided by the square root of the number of repeated valuations. Note that averaging repeated valuations does not change estimator bias, it only affects estimator variance.

The method of mesh construction used here is the same as in [7]. For LSM twelve basis functions are used consisting of a constant, the first three Hermite polynomials in the maximum of the values of the five assets, the values of four assets, the product of the highest and second highest, second highest and third highest, etc.

Figures 1–3 plot the average relative error with approximate pointwise 95% confidence intervals against sample size for both the original and corrected estimators. The relative errors are $(\tilde{B}_0 - B_0)/B_0$ and $(B_0 - \tilde{B}_0)/B_0$ for the high- and low-biased estimators, respectively, where B_0 is the best estimate taken from [21] — an average of high- and low-biased uncorrected stochastic mesh estimators computed with a large sample size. Since the estimators are consistent, correcting for bias has no effect on the relative errors at the largest sample sizes shown here. For all other cases the correction significantly reduces the bias, with the reduction varying across estimation scheme, estimator type, and sample size. In particular, the bias reduction technique seems to work best for the tree and LSM estimators and is more effec-

Table 2 Number of repeated valuations and the sample size (M) required to compute an option-value estimator with standard error of less than 0.02.

Tree		Mesh		LSM	
Valuations	Sample Size (M)	Valuations	Sample Size (M)	Valuations	Sample Size (M)
64000	10	640	2000	6400	200
32000	20	320	4000	3200	400
16000	40	160	8000	1600	800
8000	80	80	16000	800	1600
4000	160	40	32000	400	3200
2000	320	20	64000	200	6400
1000	640	10	128000	100	12800

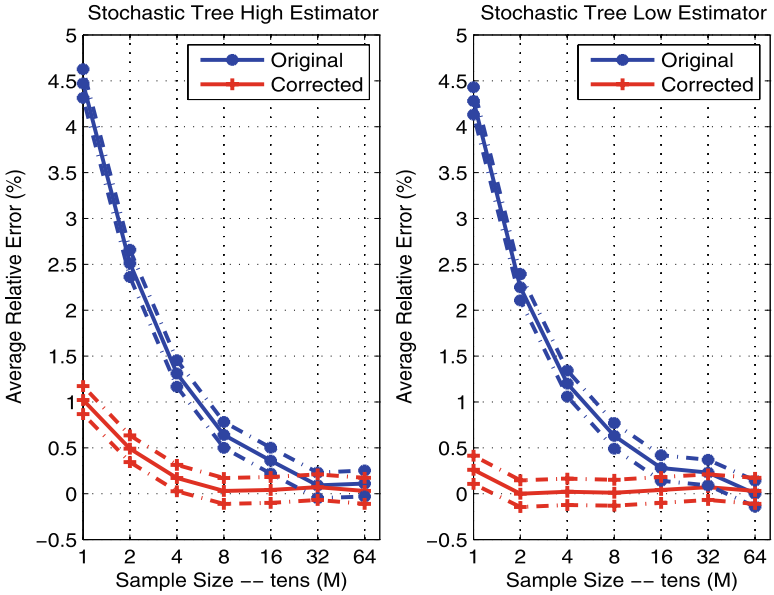


Fig. 1 Average relative error with approximate pointwise 95% confidence intervals of the original and corrected stochastic tree estimators against sample size (M).

tive for low-biased estimators. The correction terms for the stochastic tree and LSM high-biased estimators reduce the relative errors by up to a factor of four, while the relative error is reduced by a factor of two for the high-biased mesh estimators. This could indicate that the large-sample distributional approximation to the hold-value estimator is not as accurate for high-biased mesh estimators or that the hold-value dependency (ignored here) significantly affects the variance estimator. On the other hand, the relative errors for low-biased estimators are very close to zero with small

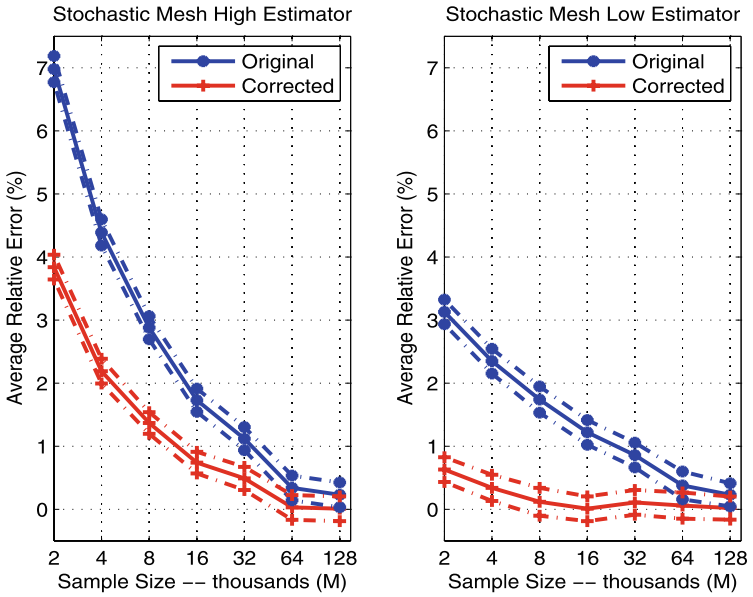


Fig. 2 Average relative error with approximate pointwise 95% confidence intervals of the original and corrected stochastic mesh estimators against sample size (M).

sample sizes. Furthermore, Figure 3 reveals that the corrected and uncorrected low-biased LSM estimators do not seem to converge to the true value as the average relative error is significantly different from zero. This indicates the importance of the choice of basis functions as discussed in Section 3.3.1. Nonetheless, the bias-corrected LSM estimators converge faster to the true approximation value given a finite set of basis functions.

For a given sample size, bias-corrected estimators have relative error less than half of the corresponding uncorrected estimators. This implies that the corrected estimator gives the same level of accuracy as the uncorrected estimator with less than half the number of sample paths used in each valuation. Since the same level of accuracy is obtained with a smaller mesh size for the corrected estimator, the computational speed of the corrected estimator can be further increased by running repeated valuations of this smaller size (each of which takes less time) in parallel (on a cluster of workstations). That means that a combination of this bias reduction method with parallel computing more than doubles (and, in some cases, more than quadruples) the convergence speed.

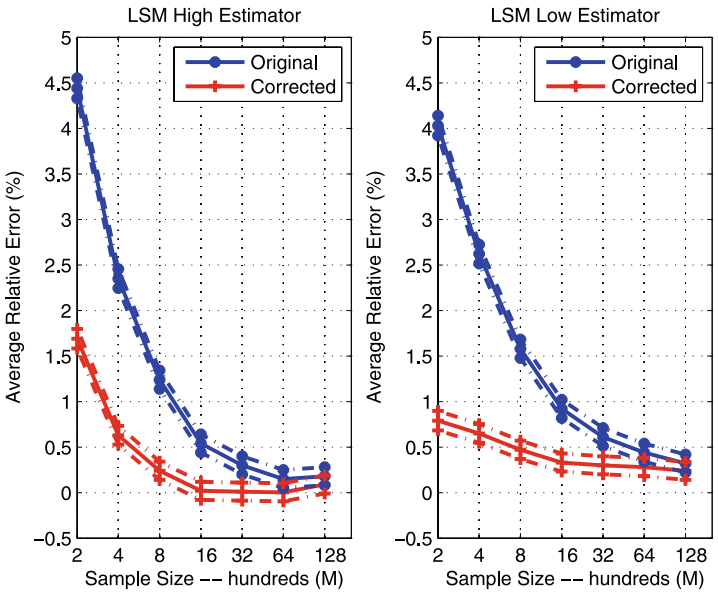


Fig. 3 Average relative error with approximate pointwise 95% confidence intervals of the original and corrected LSM estimators against sample size (M).

5 Conclusion

We introduced a general technique for reducing the bias of Monte Carlo estimators of American-style options. A five-dimensional max-max call option is used to test the effects of this bias reduction technique on the stochastic tree, stochastic mesh and LSM high- and low-biased estimators. Results show that all corrected estimators significantly outperform their uncorrected counterparts in terms of convergence speed. Other advantages of this technique include that it applies equally well in an arbitrary number of dimensions, with virtually any reasonable underlying asset-price process and payoff function, and that it contributes little incremental implementation (altering a few lines of code) and computational costs.

We continue to work on the rigorous justification for convergence of the corrected stochastic mesh and LSM estimators, including the development of an unbiased and consistent variance estimator that allows for dependence. Current work also includes the development of higher order corrections for each of the three valuation techniques. Additionally, interest lies in applications to duality methods, optimal-switching time problems, multiple-exercise options and in sensitivity estimators (e.g., the Greeks). We continue to explore the use of high-performance computing techniques and variance reduction methods to generate further computational efficiencies.

Acknowledgements We thank Western, OGS, NSERC, MITACS, and the Bank of Canada for financial support. We also thank SHARCNET for computational resources. Various versions of this work were presented at MITACS and INFORMS in 2005; ARC and SSC in 2006; SONAD, SSC, CAIMS, EURO, ICIAM, and the University of Waterloo Computational Methods in Finance Conference in 2007; CORS and MCQMC in 2008. We thank the audiences for their feedback.

References

1. L. Anderson, and M. Broadie. A primal-dual simulation algorithm for pricing multi-dimensional American options. *Management Science* 50(9): 1222–1234, 2004.
2. A.N. Avramidis, and P. Hyden. Efficiency improvements for pricing American options with a stochastic mesh. *Proceedings of the 1999 Winter Simulation Conference* 344–350, 1999.
3. A.N. Avramidis, and H. Matzinger. Convergence of the stochastic mesh estimator for pricing Bermudan options. *Journal of Computational Finance* 7(4): 73–91, 2004.
4. J. Barraquand, and D. Martineau. Numerical valuation of high-dimensional multivariate American securities. *Journal of Financial and Quantitative Analysis* 30: 383–405, 1995.
5. M. Broadie, and P. Glasserman. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control* 21(8/9): 1323–1352, 1997.
6. M. Broadie, and P. Glasserman. A pruned and bootstrapped American option simulation. *Proceedings of the 1995 Winter Simulation Conference* 229–235, 1995.
7. M. Broadie, and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. *The Journal of Computational Finance* 7(4): 35–72, 2004.
8. J.F. Carrière. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics* 19(1): 19–30, 1996.
9. E. Clément, D. Lamberton, and P. Protter. An analysis of a least squares regression method for American option pricing. *Finance and Stochastics* 6(4): 449–471, 2002.
10. M.C. Fu, S.B. Laprise, D.B. Madan, Y. Su, and R.W. Wu. Pricing American options: a comparison of Monte Carlo simulation approaches. *The Journal of Computational Finance* 4(3): 39–88, 2001.
11. P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2004.
12. M. Haugh, and L. Kogan. Pricing American options: A duality approach. *Operations Research* 52(2): 258–270, 2004.
13. C. Lemieux, and J. La. A study of variance reduction techniques for American option pricing. *Proceedings of the 37th conference on Winter simulation* 1884–1891, 2005.
14. F.A. Longstaff, and E.S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies* 14, 113–147, 2001.
15. N. Moreni. A variance reduction technique for American option pricing. *Physica A: Statistical Mechanics and its Applications* 338(1-2), 292–295, July 2004.
16. L.C.G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance* 12(3): 271–286, 2002.
17. J.A. Tilley. Valuing American options in a path simulation model. *Transactions of the Society of Actuaries* 45: 83–104, 1993.
18. J. Tsitsiklis, and B. Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks* 12(4): 694–703, 2001.
19. H. White. *Asymptotic Theory for Econometricians: Revised Edition*. Academic Press, 2001.
20. T. Whitehead. Correcting the Monte Carlo optimal-stopping bias. PhD Thesis. University of Western Ontario, 2008.
21. T. Whitehead, M. Davison, and R.M. Reesor. A bias-reduction technique for Monte Carlo pricing of early-exercise options. Submitted, 2008.
22. T. Whitehead, M. Davison, and R.M. Reesor. Correcting the Monte Carlo optimal-stopping bias. Submitted, 2008.

Fast Principal Components Analysis Method for Finance Problems With Unequal Time Steps

Jens Keiner and Benjamin J. Waterhouse

Abstract The use of the Principal Components Analysis (PCA) method as a variance reduction technique when evaluating integrals from mathematical finance using quasi-Monte Carlo point sets suffers from a distinct disadvantage in that it requires a dense matrix-vector multiplication with $\mathcal{O}(s^2)$ computations for an s-dimensional problem. It was shown by Scheicher [18] that the cost of this matrix-vector multiplication could be reduced to $\mathcal{O}(s \log s)$ arithmetic operations for problems where the time steps are equally sized. In this paper we show how we may drop this requirement and perform the matrix-vector multiplication in $\mathcal{O}(s \log s \log(1/\varepsilon))$ arithmetic operations for any desired accuracy $\varepsilon > 0$.

1 Background

1.1 The Use of Monte Carlo in Finance

Many problems in mathematical finance involve calculating the expected value of some function G under a Gaussian density and may be formulated as the evaluation of an integral of the form

$$\mathbb{E}(G) = \int_{\mathbb{R}^s} G(\mathbf{w}) \frac{1}{\sqrt{(2\pi)^s \det C}} \exp\left(-\frac{1}{2}\mathbf{w}^T C^{-1}\mathbf{w}\right) \mathrm{d}\mathbf{w},$$

Jens Keiner
Institut für Mathematik, Universität zu Lübeck, Lübeck, Germany
e-mail: keiner@math.uni-luebeck.de

Benjamin J. Waterhouse
School of Mathematics and Statistics, University of New South Wales, Sydney, Australia
e-mail: ben.waterhouse@unsw.edu.au

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_29,

where C is the symmetric positive definite covariance matrix of the Brownian motion discretized at timepoints $0 < t_1 < t_2 < \cdots < t_s$, given by

$$C = \left(\min(t_i, t_j)\right)_{i,j=1}^{s}. \tag{1}$$

It is not difficult to conceive of real-world problems where the dimension s can grow large. The integral may be written as an integral over the s-dimensional unit cube

$$\mathbb{E}(G) = \int_{[0,1]^s} G(A\Phi^{-1}(\mathbf{x}))\,\mathrm{d}\mathbf{x} \tag{2}$$

where $\Phi^{-1}(\cdot)$ is the componentwise inverse cumulative density function of the Normal distribution, and A is any matrix where $AA^T = C$.

Following Boyle [3], high-dimensional integration problems of the form of (2) have commonly been approximated using the *Monte Carlo* (MC) method. Under the MC method, an integral is approximated by the n-point equal-weight rule

$$\int_{[0,1]^s} G(A\Phi^{-1}(\mathbf{x}))\,\mathrm{d}\mathbf{x} \approx \frac{1}{n}\sum_{k=1}^{n} G(A\Phi^{-1}(\mathbf{x}_k)) \tag{3}$$

for a set of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ chosen iid from the s-dimensional unit cube.

Recently there has been a great deal of research into *quasi-Monte Carlo* (QMC) integration. The QMC method has the same form as the MC method as shown in (3) except that the points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are chosen deterministically from the unit cube.

In the literature there are three choices of A such that $AA^T = C$ which are commonly used, namely the "standard" or Cholesky construction, the Brownian Bridge (BB) construction and the Principal Components Analysis (PCA) construction. In this paper we will focus on the efficient use of the PCA construction, as first proposed by Acworth, Broadie, and Glasserman [1].

Under the PCA method, A^{PCA} is taken to be the scaled eigenvector matrix

$$A^{\mathrm{PCA}} = [\sqrt{\lambda_1}\mathbf{v}_1, \ldots, \sqrt{\lambda_s}\mathbf{v}_s] = V\Lambda^{1/2},$$

where $\lambda_1, \ldots, \lambda_s$ are the eigenvalues of C in decreasing order (comprising the entries in the diagonal matrix Λ) and $\mathbf{v}_1, \ldots, \mathbf{v}_s$ the corresponding unit-length eigenvectors (which form the columns of the matrix V).

Several papers have appeared which suggest that the PCA method works well for various problems from mathematical finance. (See for example Acworth et al. [1], Giles et al. [9] and L'Ecuyer et al. [15].) However, even though the PCA construction may, for particular problems, require fewer function evaluations to calculate the value of an integral to within a certain error tolerance, each of these function evaluations is more computationally expensive since there is no obvious way to avoid the matrix-vector multiplication $A^{\mathrm{PCA}}\Phi^{-1}(\mathbf{x})$ in the way it is avoided by the standard and BB constructions. Under the standard and BB constructions, the matrix A has structure which may be exploited to allow the matrix-vector multiplication $A\Phi^{-1}(\mathbf{x})$ to be performed in $\mathcal{O}(s)$ operations. Hence the use of the PCA method

increases the computational cost of each function evaluation to $\mathcal{O}(s^2)$. Additionally, the eigenvalues and eigenvectors of C need to be calculated, which requires, in general, $\mathcal{O}(s^3)$ operations, although this is only calculated once, rather than for each function evaluation.

It is therefore tempting to relegate the PCA method to the category of "theoretically interesting" but impractical. This paper is devoted to assuaging this temptation.

The remainder of this paper is structured as follows. Section 2 proves that the matrix-vector multiplication for the PCA method may be performed for unequal time-steps in $\mathcal{O}(s \log s \log(1/\varepsilon))$ operations. It also contains an algorithm to demonstrate how to implement the method. Section 3 contains some numerical testing.

2 Fast Matrix-Vector Product for the Principal Components Analysis Method

We will demonstrate that it is possible to exploit the structure of the problem and to reduce the cost of the matrix-vector multiplication $A^{\mathrm{PCA}}\Phi^{-1}(\mathbf{x})$ from $\mathcal{O}(s^2)$ operations to $\mathcal{O}(s \log s \log(1/\varepsilon))$ operations for any given accuracy $\varepsilon > 0$.

The problem of efficiently computing $A^{\mathrm{PCA}}\Phi^{-1}(\mathbf{x})$ has been considered by Scheicher [18] for the special case where the time steps are equal in length. That is, $t_j - t_{j-1} = \Delta t$ for $j = 1, \ldots, s$ and $t_0 = 0$. The method relies on the fact, first shown by Åkesson and Lehoczky [2], that for the equal time step case, there exist closed form expressions for both the eigenvalues and eigenvectors of C.

The matrix-vector multiplication of $A^{\mathrm{PCA}}\Phi^{-1}(\mathbf{x})$ is then re-written as a Discrete Sine Transformation and may be performed with $\mathcal{O}(s \log s)$ operations by means of a Fast Fourier Transform (FFT). No further pre-computation is necessary since all quantities are known explicitly. See Scheicher [18] for details of this method.

In reality, time steps in most finance problems are not equally spaced. Often they will be close to equally spaced, but public holidays, weekends, different numbers of days in months cause many of the time steps to differ slightly. This means that the closed form of the eigenvalues and eigenvectors mentioned above no longer hold, and the FFT cannot be used.

However, in the following section, we will demonstrate that it is still feasible to efficiently calculate the eigendecomposition of C as well as to perform the corresponding matrix-vector multiplication $A^{\mathrm{PCA}}\Phi^{-1}(\mathbf{x})$. This is done by observing that, even in the general case, the matrix C as defined in (1) does in fact have structure that we can exploit. As we will see below, C is a semi-separable matrix, and matrices of this type allow for fast calculation of eigenvalues and eigenvectors as well as matrix-vector multiplication of scaled eigenvector matrices.

2.1 Symmetric Semi-Separable Matrices

In this section we will state the definition of symmetric semi-separable matrices and demonstrate how their structure can be exploited to efficiently obtain their eigenvalues and eigenvectors. In the definition below we use the Matlab notation diag(·), triu(·) and tril(·) to define the diagonal, strictly upper triangular and strictly lower triangular parts of a matrix, respectively.

Definition 1. An $s \times s$ matrix C is said to be a *diagonal plus symmetric generator representable semi-separable matrix* if it can be written in the form

$$C = \operatorname{diag}(\mathbf{d}) + \operatorname{triu}(\mathbf{u}\mathbf{v}^T) + \operatorname{tril}(\mathbf{v}\mathbf{u}^T) \quad \text{for some } \mathbf{d}, \mathbf{u}, \mathbf{v} \in \mathbb{R}^s.$$

For the rest of this text, a *diagonal plus symmetric generator representable semi-separable* matrix will be just called *symmetric semi-separable* matrix.

Any symmetric matrix C has an eigendecomposition of the form $C = V \Lambda V^T$ with an orthogonal eigenvector matrix V and a real diagonal eigenvalue matrix Λ. To find the eigenvalues and eigenvectors of an $s \times s$ matrix generally requires $\mathcal{O}(s^3)$ operations. However, Chandrasekaran and Gu [5] and Mastronardi et al. [17] developed divide-and-conquer algorithms for symmetric semi-separable matrices to reduce this computational cost.

2.1.1 Divide-and-Conquer Algorithm

Given a symmetric semi-separable matrix $C = \operatorname{diag}(\mathbf{d}) + \operatorname{triu}(\mathbf{u}\mathbf{v}^T) + \operatorname{tril}(\mathbf{v}\mathbf{u}^T)$, we would like to write this in the form of several smaller symmetric semi-separable matrices. This can be done in the following way. Take $\rho = \pm 1$ to be a freely chosen scalar. Split each of the vectors $\mathbf{d}$, $\mathbf{u}$, $\mathbf{v}$ into two vectors with the first $\lfloor s/2 \rfloor$ components in the first vector, and the remaining components in the second vector. That is, define $\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{v}_1$, $\mathbf{v}_2$, an additional vector $\mathbf{a}$ and write C such that

$$\mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} \rho\,\mathbf{u}_1 \\ \mathbf{v}_2 \end{pmatrix}, \quad C = \begin{pmatrix} \hat{C}_1 & 0 \\ 0 & \hat{C}_2 \end{pmatrix} + \rho\,\mathbf{a}\mathbf{a}^T,$$

where $\hat{C}_1$ and $\hat{C}_2$ are defined to the symmetric semi-separable matrices

$$\begin{aligned} \hat{C}_1 &= \operatorname{diag}\left(\mathbf{d_1} - \rho \operatorname{diag}(\mathbf{u}_1\mathbf{u}_1^T)\right) + \operatorname{triu}\left(\mathbf{u}_1(\mathbf{v}_1 - \rho\mathbf{u}_1)^T\right) + \operatorname{tril}\left((\mathbf{v}_1 - \rho\mathbf{u}_1)\mathbf{u}_1^T\right), \\ \hat{C}_2 &= \operatorname{diag}\left(\mathbf{d_2} - \rho \operatorname{diag}(\mathbf{v}_2\mathbf{v}_2^T)\right) + \operatorname{triu}\left((\mathbf{u}_2 - \rho\mathbf{v}_2)\mathbf{v}_2^T\right) + \operatorname{tril}\left(\mathbf{v}_2(\mathbf{u}_2 - \rho\mathbf{v}_2)^T\right). \end{aligned}$$

This result can be easily verified. Each of the matrices $\hat{C}_1$ and $\hat{C}_2$ may now be themselves decomposed into a similar pattern.

In the conquer phase we re-group the subproblems into larger problems. Suppose that two symmetric semi-separable matrices $\hat{C}_1$ and $\hat{C}_2$, obtained from the divide phase of a symmetric semi-separable matrix C, have the eigendecomposition

$$\hat{C}_1 = V_1 \Lambda_1 V_1^T \quad \text{and} \quad \hat{C}_2 = V_2 \Lambda_2 V_2^T,$$

with diagonal eigenvalue matrices Λ_1 and Λ_2 and orthogonal eigenvector matrices V_1 and V_2. Then C has the representation

$$C = \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix} (\Lambda + \rho \mathbf{y}\mathbf{y}^T) \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix}^T, \quad \text{where } \Lambda = \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix}, \mathbf{y} = \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix}^T \mathbf{a},$$

and where ρ is defined as it was for the divide phase. Now suppose that we can efficiently compute the eigendecomposition of the symmetric rank-one modified diagonal matrix $\Lambda + \rho \mathbf{y}\mathbf{y}^T$, which is written as $\Lambda + \rho \mathbf{y}\mathbf{y}^T = U \Omega U^T$. We can then write the eigendecomposition of C as

$$C = \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix} (U \Omega U^T) \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix}^T = V \Omega V^T, \quad \text{with } V = \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix} U.$$

More details are found in Chandrasekaran and Gu [5]. The critical step in the implementation is the eigendecomposition computation of the rank-one modified diagonal matrix $\Lambda + \rho \mathbf{y}\mathbf{y}^\mathbf{T}$ and the efficient application of the eigenvector matrix U. The divide-and-conquer technique relies on being able to efficiently handle these problems.

2.1.2 Symmetric Rank-one Modified Diagonal Eigenvalue Problem

The problem of determining the eigendecomposition of a rank-one modified diagonal matrix $\Lambda + \rho \mathbf{y}\mathbf{y}^\mathbf{T}$ was formulated by Golub [10] and subsequently investigated by Bunch et al. and others [4, 6, 7, 14, 19, 13].

It is valid to assume that all diagonal entries of Λ are numerically distinct and that all entries in $\mathbf{y}$ are bounded away from zero. If not, we can use the deflation procedure from Dongarra and Sorensen [7] with the criterion from Gu and Eisenstat [13] to arrange this. Moreover, we require that by permutations we have ordered the diagonal entries of Λ to be increasing. The following theorem, restating results found in Golub [10] and Bunch et al. [4], characterizes the structure of the desired eigendecomposition.

Theorem 1. *Let Λ be a diagonal matrix with entries $\lambda_1 < \lambda_2 < \cdots < \lambda_s$, $\mathbf{y}$ a vector with non-zero entries $y_1, y_2, \ldots, y_s$, and $\rho \neq 0$. Then for the symmetric rank-one modified diagonal matrix $\Lambda + \rho \mathbf{y}\mathbf{y}^T$ the following results hold:*

1. *The eigenvalues $\omega_1, \omega_2, \ldots, \omega_s$ have the interlacing property*

$$\begin{cases} \lambda_1 < \omega_1 < \lambda_2 < \omega_2 < \cdots < \lambda_s < \omega_s < \lambda_s + \rho \mathbf{y}^T\mathbf{y}, & \text{if } \rho > 0, \\ \lambda_1 + \rho \mathbf{y}^T\mathbf{y} < \omega_1 < \lambda_1 < \omega_2 < \lambda_2 < \cdots < \omega_s < \lambda_s, & \text{if } \rho < 0. \end{cases}$$

2. *The eigenvalues $\omega_1, \omega_2, \ldots, \omega_s$ are solutions to the secular equation*

$$1+\rho\sum_{j=1}^{s}\frac{y_j^2}{\lambda_j-\omega}=0. \tag{4}$$

3. For each eigenvalue ω_j, a corresponding unit-length eigenvector $\mathbf{u}_j$ is given by

$$\mathbf{u}_j=\pm\left(\sum_{m=1}^{s}\frac{y_m^2}{\left(\lambda_m-\omega_j\right)^2}\right)^{-1/2}\cdot\left(\frac{y_1}{\lambda_1-\omega_j},\frac{y_2}{\lambda_2-\omega_j},\ldots,\frac{y_s}{\lambda_s-\omega_j}\right)^T. \tag{5}$$

The eigenvalues ω_j, $j=1,\ldots,s$ can be efficiently obtained as the zeros of the rational equation (4) by iterative methods (see Bunch et al. [4] and Li [16]). The eigenvectors $\mathbf{u}_j$ have explicit expressions in terms of the entries of the vector $\mathbf{y}$, the diagonal entries λ_j and the eigenvalues ω_j. It is imperative to use the technique from Gu and Eisenstat [13] to recompute $\mathbf{y}$ for numerical orthogonality.

2.1.3 Efficient Application of the Eigenvector Matrix

An efficient method is needed to apply the eigenvector matrix U, obtained from the symmetric rank-one modified system $\Lambda+\rho\,\mathbf{y}\mathbf{y}^{\mathbf{T}}$, to an arbitrary vector. We find such a method by observing that U has the form of a Cauchy-like matrix.

Definition 2. A matrix U is called a *Cauchy-like matrix* if it is of the form

$$U=\left(\frac{a_i b_j}{d_i-c_j}\right)_{i,j=1}^{s},\quad \text{for } \mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}\in\mathbb{R}^s. \tag{6}$$

It is easily verified that the eigenvectors $\mathbf{u}_1,\ldots,\mathbf{u}_s$ of the rank-one modified diagonal matrix $\Lambda+\mathbf{y}\mathbf{y}^T$ from Theorem 1 form a Cauchy-like matrix $U=[\mathbf{u}_1,\ldots,\mathbf{u}_s]$ since

$$U=\left(\frac{y_i z_j}{\lambda_i-\omega_j}\right)_{i,j=1}^{s},\qquad \text{with } z_j=\left(\sum_{m=1}^{s}\frac{y_m^2}{(\lambda_m-\omega_j)^2}\right)^{-1/2}, \tag{7}$$

where $\lambda_1<\lambda_2<\cdots<\lambda_s$ are the entries of the diagonal matrix Λ and $\omega_1,\omega_2,\ldots,\omega_s$ are the eigenvalues of $\Lambda+\mathbf{y}\mathbf{y}^T$ corresponding to the eigenvectors $\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_s$.

For the divide-and-conquer method, we require efficient methods to apply a Cauchy-like matrix to a vector. These methods are commonly subsumed under the name *Fast Multipole Method (FMM)* and were introduced by Greengard and Rokhlin [11]. The principle common to all implementations is that a Cauchy-like matrix can be decomposed into tiles that are well approximated by low-rank matrices. This is done by imposing a tree structure on the nodes d_i and c_j from (6). Efficient organization of the computation leads to an algorithm for the matrix-vector multiplication requiring $\mathcal{O}(s\log(1/\varepsilon))$ computations for an $s\times s$ matrix, where ε is the desired accuracy. It has been shown by Gu and Eisenstat [12] that the method can be made numerically as stable as the direct computation of the matrix-vector product.

2.1.4 Complexity of the Divide-and-Conquer Algorithm

Using plain matrix-vector multiplications, the divide-and-conquer method needs $O(s^2)$ arithmetic operations and memory to compute the full eigendecomposition of an $s \times s$ matrix. To apply the eigenvector matrix to a vector clearly takes $O(s^2)$ operations. However, if we use the FMM to accelerate the calculation of matrix-vector products, we only need to pre-compute the data that defines the eigenvector matrix (7) of every rank-one modified system encountered plus a number of small eigenvector matrices from the bottom of the computation tree. This leads to an $\mathcal{O}(s \log s \log(1/\varepsilon))$ algorithm for the application of the eigenvector matrix that needs only $\mathcal{O}(s \log s \log(1/\varepsilon))$ of pre-computed data. The FMM can even be used to lower time and memory requirements of the pre-computation part to the same order (see Chandrasekaran and Gu [5]).

2.2 Fast PCA Method and Semi-Separable Matrices

The divide-and-conquer strategy for symmetric semi-separable matrices can be used to perform the matrix-vector multiplication for the PCA method. The application of $A^{\mathrm{PCA}} = V \Lambda^{1/2}$ to any vector can be done by scaling the input by the diagonal entries of $\Lambda^{1/2}$ and then using the divide-and-conquer algorithm to apply V. This is established in the following theorem.

Theorem 2. *For $0 < t_1 < t_2 < \cdots < t_s$, an arbitrary $\mathbf{x} \in [0,1]^s$ and the matrix $C = \left(\min(t_i, t_j)\right)_{i,j=1}^{s}$, with eigendecomposition $C = V \Lambda V^T$, where the entries of the diagonal eigenvalue matrix Λ are decreasing, the following hold for any desired but fixed accuracy ε:*

1. *The eigenvector matrix V may be applied to any vector in $\mathcal{O}(s \log s \log(1/\varepsilon))$ arithmetic operations. The (one-off) cost for pre-computation can be made $\mathcal{O}(s \log s \log(1/\varepsilon))$.*
2. *The matrix-vector multiplication $A^{PCA}\Phi^{-1}(\mathbf{x})$ where $A^{PCA} = V \Lambda^{1/2}$ may be computed in $\mathcal{O}(s \log s \log(1/\varepsilon))$ operations.*

Proof. Observe that for $\mathbf{t} = (t_1, t_2, \ldots, t_s)^T$ and $\mathbf{1} = (1, 1, \ldots, 1)^T$, the matrix C can be written in the form

$$C = \operatorname{diag}(\mathbf{t}) + \operatorname{triu}(\mathbf{t}\,\mathbf{1}^T) + \operatorname{tril}(\mathbf{1}\,\mathbf{t}^T).$$

Hence C is a symmetric semi-separable matrix. The statements are then a direct consequence of the results in the last section. For the second part, note that $A^{\mathrm{PCA}} = V \Lambda^{1/2}$. That is, for the matrix-vector multiplication $A^{\mathrm{PCA}}\Phi^{-1}(\mathbf{x})$, we cheaply compute $\mathbf{z} = \Lambda^{1/2}\Phi^{-1}(\mathbf{x})$ with $\mathcal{O}(s)$ operations and then $V\mathbf{z}$ in $\mathcal{O}(s \log s \log(1/\varepsilon))$ operations as explained in the last section. The total cost is $\mathcal{O}(s \log s \log(1/\varepsilon))$.

Algorithm 1 Divide-and-conquer method (precomputation)

% Computes the eigendecomposition of a symmetric semi-separable matrix

$$\mathbf{A} = \operatorname{diag}(\mathbf{d}) + \operatorname{triu}(\mathbf{u}\,\mathbf{v}^{\mathrm{T}}) + \operatorname{tril}(\mathbf{v}\,\mathbf{u}^{\mathrm{T}}).$$

Input: The vectors $\mathbf{d}, \mathbf{u}, \mathbf{v} \in \mathbb{R}^s$.

$t := 2$, $J := \lceil \log(s/t) \rceil$ *% J = Number of levels -1, where t is the size threshold*

$r := \sqrt{\|\mathbf{v}\|_2 / \|\mathbf{u}\|_2}$, $\mathbf{u} := r\,\mathbf{u}$, $\mathbf{v} := r^{-1}\,\mathbf{v}$ *% Balance norms to prevent overflow.*

$\hat{A}_{(0,1)} := A$, $\mathbf{d}_{(0,1)} := \mathbf{d}$, $\mathbf{u}_{(0,1)} := \mathbf{u}$, $\mathbf{v}_{(0,1)} := \mathbf{v}$ *% Naming convention for decomposition tree.*

% Define auxilliary matrices and vectors $W_{(0,1)}$, $H_{(0,1)}$, $\mathbf{h}_{(0,1)}$, $\mathbf{g}_{(0,1)}$ such that

$$\begin{aligned}\hat{A}_{(0,1)} = {} & \operatorname{diag}\left(\mathbf{d}_{(0,1)}\right) + \operatorname{diag}\left(\operatorname{diag}\left(W_{(0,1)} H_{(0,1)} W_{(0,1)}^{\mathrm{T}}\right)\right) \\ & + \operatorname{triu}\left(W_{(0,1)}\,\mathbf{h}_{(0,1)}\,\mathbf{g}_{(0,1)}^{\mathrm{T}}\,W_{(0,1)}^{\mathrm{T}}\right) + \operatorname{tril}\left(W_{(0,1)}\,\mathbf{g}_{(0,1)}\,\mathbf{h}_{(0,1)}^{\mathrm{T}}\,W_{(0,1)}^{\mathrm{T}}\right).\end{aligned}$$

$W_{(0,1)} := \left(\mathbf{u}_{(0,1)}\ \mathbf{v}_{(0,1)}\right)$, $H_{(0,1)} := \left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right)$, $\mathbf{h}_{(0,1)} := \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$, $\mathbf{g}_{(0,1)} := \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$

% Divide phase

for $j = 1, \ldots, J$ **do** *% Traverse all levels.*

  for $k = 1, \ldots, 2^{j-1}$ **do** *% Split matrices on current level.*

    $\rho_{(j,k)} := \pm 1$ *% Can be chosen arbitrarily.*

    % For $r = (j, 2k-1), (j, 2k)$, determine W_r, H_r, h_r, and g_r such that

$$\hat{A}_r = \operatorname{diag}(\mathbf{d}_r) + \operatorname{diag}(\operatorname{diag}(W_r H_r W_r^{\mathrm{T}})) + \operatorname{triu}(W_r\,\mathbf{h}_r\,\mathbf{g}_r^{\mathrm{T}}\,W_r^{\mathrm{T}}) + \operatorname{tril}(W_r\,\mathbf{g}_r\,\mathbf{h}_r^{\mathrm{T}}\,W_r^{\mathrm{T}}).$$

    % Notation: $(\cdot)_{1/2}$ means first/second half of vector.

$\mathbf{d}_{(j,2k-1)} := (\mathbf{d}_{(j-1,k)})_1$	$\mathbf{d}_{(j,2k)} := (\mathbf{d}_{(j-1,k)})_2$
$\mathbf{u}_{(j,2k-1)} := (\mathbf{u}_{(j-1,k)})_1$	$\mathbf{u}_{(j,2k)} := (\mathbf{u}_{(j-1,k)})_2$
$\mathbf{v}_{(j,2k-1)} := (\mathbf{v}_{(j-1,k)})_1$	$\mathbf{v}_{(j,2k)} := (\mathbf{v}_{(j-1,k)})_2$
$W_{j,2k-1} := \left(\mathbf{u}_{(j,2k-1)}\ \mathbf{v}_{(j,2k-1)}\right)$	$W_{(j,2k)} := \left(\mathbf{u}_{(j,2k)}\ \mathbf{v}_{(j,2k)}\right)$
$\mathbf{h}_{(j,2k-1)} := \mathbf{h}_{(j-1,k)}$	$\mathbf{h}_{(j,2k)} := \mathbf{h}_{(j-1,k)} - \rho_{(j,k)}\mathbf{g}_{(j-1,k)}$
$\mathbf{g}_{(j,2k-1)} := \mathbf{g}_{(j-1,k)} - \rho_{(j,k)}\mathbf{h}_{(j-1,k)}$	$\mathbf{g}_{(j,2k)} := \mathbf{g}_{(j-1,k)}$
$H_{(j,2k-1)} := H_{(j-1,k)} - \rho_{(j,k)}\mathbf{h}_{(j-1,k)}\mathbf{h}_{(j-1,k)}^{\mathrm{T}}$	$H_{(j,2k)} := H_{(j-1,k)} - \rho_{(j,k)}\mathbf{g}_{(j-1,k)}\mathbf{g}_{(j-1,k)}^{\mathrm{T}}$

  end for

end for

for $k = 1, \ldots, 2^J$ **do** *% Conquer phase – Solve smallest problems directly.*

  $\hat{A}_{(J,k)} = V_{(J,k)}\,\Lambda_{(J,k)}\,V_{(J,k)}^{\mathrm{T}}$ *% Determine eigendecomposition.*

  $\mathbf{u}_{(J,k)} := V_{(J,k)}^{\mathrm{T}}\mathbf{u}_{(J,k)}$, $\mathbf{v}_{(J,k)} := V_{(J,k)}^{\mathrm{T}}\mathbf{v}_{(J,k)}$ *% Update vectors using inverse eigenvector matrix.*

end for

for $j = J-1, \ldots, 0$ **do** *% Conquer phase – Combine solutions.*

  for $k = 1, \ldots, 2^j$ **do**

    $\mathbf{y}_{(j,k)} := \begin{pmatrix} \rho\mathbf{u}_{(j+1,2k-1)} \\ \mathbf{v}_{(j+1,2k)} \end{pmatrix}$, $\tilde{\Lambda}_{(j,k)} := \begin{pmatrix} \Lambda_{(j+1,2k-1)} & 0 \\ 0 & \Lambda_{(j+1,2k-1)} \end{pmatrix}$

    % Note that $\mathbf{u}_{(j+1,2k-1)}$ and $\mathbf{v}_{(j+1,2k)}$ have already been multiplied by $\mathbf{V}_{(j+1,2k-1)}^{\mathrm{T}}$ and
    % $\mathbf{V}_{(j+1,2k)}^{\mathrm{T}}$, respectively. Eigendecomposition of rank-one modified diagonal matrix:

    $\tilde{\Lambda}_{(j,k)} + \rho_{(j,k)}\mathbf{y}_{(j,k)}\mathbf{y}_{(j,k)}^{\mathrm{T}} = U_{(j,k)}\,\Lambda_{(j,k)}\,U_{(j,k)}^{\mathrm{T}}$

    $\mathbf{u}_{j,k} = U_{j,k}^{\mathrm{T}}\begin{pmatrix} \mathbf{u}_{j+1,2k-1} \\ \mathbf{u}_{j+1,2k} \end{pmatrix}$, $\mathbf{v}_{j,k} = U_{j,k}^{\mathrm{T}}\begin{pmatrix} \mathbf{v}_{j+1,2k-1} \\ \mathbf{v}_{j+1,2k} \end{pmatrix}$ *% Updated vectors for next level.*

  end for

end for

Output: The matrices $V_{(J,k)}$ for $k = 1, \ldots, 2^J$, $U_{(j,k)}$ for $j = 0, \ldots, J-1$; $k = 1, \ldots, 2^j$, and the diagonal eigenvalue matrix $\Lambda = \Lambda_{(0,1)}$

Algorithm 2 Divide-and-conquer method (fast matrix-vector product)

% Computes the matrix-vector product

$$\mathbf{y} = V\,\Lambda^{1/2}\,\mathbf{x},$$

% where V and Λ are from the eigendecomposition of a symmetric semi-separable matrix

$$\mathbf{A} = \operatorname{diag}(\mathbf{d}) + \operatorname{triu}(\mathbf{u}\,\mathbf{v}^{\mathrm{T}}) + \operatorname{tril}(\mathbf{v}\,\mathbf{u}^{\mathrm{T}}) = V\,\Lambda\,V^{\mathrm{T}}.$$

Input: The matrices $V_{(J,k)}$ for $k = 1,\ldots,2^J$, $U_{(j,k)}$ for $j = 0,\ldots,J-1$; $k = 1,\ldots,2^j$, and the diagonal eigenvalue matrix $\Lambda = \Lambda_{(0,1)}$ as computed by Algorithm 1, and the vector $\mathbf{x}$.
$\mathbf{z}_{(0,1)} := \Lambda^{1/2}\mathbf{x}$ *% Cheap product with diagonal matrix.*
for $j = 0,\ldots,J-1$ **do** *% Traverse all levels*
  for $k = 1,\ldots,2^j$ **do** *% Process all matrices on current level*
    $\mathbf{z}_{(j,k)} := U_{(j,k)}\mathbf{z}_{(j,k)}$ *% FMM-accelerated matrix-vector products.*
    $\mathbf{z}_{(j+1,2k-1)} := \left(\mathbf{z}_{(j+1,2k-1)}\right)_1$
    $\mathbf{z}_{(j+1,2k)} := \left(\mathbf{z}_{(j+1,2k-1)}\right)_2$
  end for
end for
for $k = 0,\ldots,2^J$ **do** *% Process all matrices on smallest level*
  $\mathbf{z}_{(J,k)} := V_{(J,k)}\mathbf{z}_{(J,k)}$ *% Non-accelerated products with small matrices.*
end for
Output: The vector $\mathbf{y}$, that is, the concatenation of the vectors $\mathbf{z}_{(J,1)}, \mathbf{z}_{(J,2)}, \ldots, \mathbf{z}_{(J,2^J)}$.

Some sample pseudo-code for the divide-and-conquer method is provided in Algorithms 1 and 2[1]. Pseudo-code for the FMM can be found, for example, in [8].

3 Numerical Tests

We have implemented and tested the divide-and-conquer algorithm in double precision C on an Intel Xeon 3.00 GHz system to compare the FMM accelerated divide-and-conquer algorithm (F) with the direct matrix-vector multiplication (D). The parameters for the FMM are chosen to yield maximum accuracy in working precision. We do not use the FMM to accelerate the pre-computation part. We use the LAPACK routine `dspev` when eigendecompositions need to be calculated explicitly. This is similar to the routine `dsyev` called by Matlab internally. In the divide-and-conquer method, problems smaller than 256×256 are not decomposed further but solved directly using `dspev`. This threshold is specifically chosen to maximize performance for matrix-vector multiplications. The results are shown in Table 1. The divide-and-conquer method reaches parity with the direct method at a size s between 512 and 1024 and is faster for every larger problem. Also, the divide-and-conquer method needs substantially less time for pre-computation for $s \geq 512$.

An additional side benefit not shown in the table is the more benign memory requirements as mentioned in Section 2.1. Explicit calculation of the matrix A^{PCA} for $s = 32768$ would yet require roughly 8GB of memory which might render the

[1] A MATLAB implementation of these two algorithms may be downloaded from http://sourceforge.net/projects/fastpca/

Table 1 Computation time for the matrix-vector product $\mathbf{w} = A^{\mathrm{PCA}} \Phi^{-1}(\mathbf{x})$ with the direct method (D) and the FMM accelerated divide-and-conquer method (F). The time in seconds for precomputation $\left(t_{\mathrm{p}}^{\mathrm{D}}, t_{\mathrm{p}}^{\mathrm{F}}\right)$ and for a single matrix-vector product $\left(t_{\mathrm{a}}^{\mathrm{D}}, t_{\mathrm{a}}^{\mathrm{F}}\right)$, and that value divided by the respective cost bound are shown for different problems sizes s. Smaller values are better. The gaps marked with $*$ are unavailable due to excessive time and memory requirements.

s	$t_{\mathrm{p}}^{\mathrm{D}}$	$t_{\mathrm{a}}^{\mathrm{D}}$	$t_{\mathrm{a}}^{\mathrm{D}}/s^2$	$t_{\mathrm{p}}^{\mathrm{F}}$	$t_{\mathrm{a}}^{\mathrm{F}}$	$t_{\mathrm{a}}^{\mathrm{F}}/(s \log s)$
16	2.0E-01	4.7E-07	1.8E-09	2.0E-01	4.7E-07	1.1E-08
32	2.0E-01	2.4E-06	2.4E-09	2.0E-01	2.2E-06	2.0E-08
64	2.1E-01	8.1E-06	2.0E-09	2.1E-01	8.3E-06	3.1E-08
128	2.5E-01	3.1E-05	1.9E-09	2.5E-01	3.1E-05	5.0E-08
256	4.2E-01	1.6E-04	2.5E-09	4.2E-01	1.8E-04	1.3E-07
512	1.7E+00	8.0E-04	3.1E-09	8.7E-01	8.7E-04	2.7E-07
1024	8.9E+00	2.7E-03	2.5E-09	1.0E+00	2.5E-03	3.5E-07
2048	6.2E+01	1.2E-02	2.9E-09	2.6E+00	6.6E-03	4.2E-07
4096	4.5E+02	4.1E-02	2.4E-09	8.2E+00	1.6E-02	4.8E-07
8192	3.7E+03	1.6E-01	2.4E-09	4.2E+01	4.0E-02	5.4E-07
16384	2.9E+04	6.9E-01	2.6E-09	1.8E+02	8.9E-02	5.6E-07
32768	*	*	*	6.9E+02	2.0E-01	5.8E-07

problem infeasible to solve with the direct method on many platforms. Under the accelerated FMM the memory requirement for such a calculation is only 900MB.

Acknowledgements The support of the Australian Research Council under its Centres of Excellence and Linkage programs as well as the helpful comments of the referees and editors are gratefully acknowledged.

References

1. P. Acworth, M. Broadie, and P. Glasserman. A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. In P. Hellekalek, G. Larcher, H. Niederreiter, and P. Zinterhof, editors, *Monte Carlo and Quasi-Monte Carlo Methods 1996*, volume 127 of *Lecture Notes in Statistics*, pages 1–18. Springer-Verlag, New York, NY, 1998.
2. F. Åkesson and J. Lehoczky. Discrete eigenfunction expansion of multi-dimensional Brownian motion and the Ornstein-Uhlenbeck process. Technical report, Department of Statistics, Carnegie-Melon University, Pittsburgh, PA, 1998.
3. P. P. Boyle. Options: A Monte Carlo approach. *Journal of Financial Economics*, 4:323–338, May 1977.
4. J. R. Bunch, C. P. Nielsen, and D. C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numer. Math.*, 31:31–48, 1978.
5. S. Chandrasekaran and M. Gu. A divide-and-conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semiseparable matrices. *Numer. Math.*, 96:723–731, 2004.
6. J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numer. Math.*, 36:177–195, 1981.
7. J. J. Dongarra and D. C. Sorensen. A fully parallel algorithm for the symmetric eigenvalue problem. *SIAM J. Sci. Stat. Comput.*, 8:139–154, 1987.

8. A. Dutt, M. Gu, and V. Rokhlin. Fast algorithms for polynomial interpolation, integration and differentiation. *SIAM J. Numer. Anal.*, 33:1689–1711, 1996.
9. M. B. Giles, F. Y. Kuo, I. H. Sloan, and B. J. Waterhouse. Quasi-Monte Carlo for finance applications. In G. N. Mercer and A. J. Roberts, editors, *Proceedings of the 14th Biennial Computational Techniques and Applications Conference, CTAC-2008*, volume 50 of *ANZIAM J.*, pages C308–C323, 2008.
10. G. H. Golub. Some modified matrix eigenvalue problems. *SIAM Rev.*, 15:318–334, 1973.
11. L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.
12. M. Gu and S. C. Eisenstat. A stable and fast algorithm for updating the singular value decomposition. Technical Report YALE/DCS/TR966, Department of Computer Science, Yale University, New Haven, CT, 1993.
13. M. Gu and S. C. Eisenstat. A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM J. Matrix Anal. Appl.*, 15:1266–1276, 1994.
14. W. Kahan. Rank-1 perturbed diagonal's eigensystem. unpublished manuscript, Department Computer Science, Stanford University, CA, 1989.
15. P. L'Ecuyer, J.-S. Parent-Chartier, and M. Dion. Simulation of a Lévy process by PCA sampling to reduce the effective dimension. In *WSC '08: Proceedings of the 40th Conference on Winter Simulation*, pages 436–443. Winter Simulation Conference, 2008.
16. R. Li. Solving secular equations stably and efficiently. Technical report CSD-94-851, University of California at Berkeley, Berkeley, CA, 1993.
17. N. Mastronardi, E. van Camp, and M. van Barel. Divide and conquer algorithms for computing the eigendecomposition of symmetric diagonal-plus-semiseparable matrices. *Numer. Algorithms*, 39:379–398, 2005.
18. K. Scheicher. Complexity and effective dimension of discrete Lévy areas. *J. Complexity*, 23:152–168, 2007.
19. D. C. Sorensen and P. T. P. Tang. On the orthogonality of eigenvectors computed by divide-and-conquer techniques. *SIAM J. Numer. Anal.*, 28:1752–1775, 1991.

Adaptive Monte Carlo Algorithms for General Transport Problems

Rong Kong, Martin Ambrose, and Jerome Spanier

Abstract Recently there has been a concerted effort to develop adaptively modified Monte Carlo algorithms that converge geometrically to solutions of the radiative transport equation. We have concentrated on algorithms that extend to integral equations methods first proposed for matrix equations by Halton in 1962 [Halton, J., Proc. Camb. Phil. Soc., 58, 57–78 (1962)]. Geometric convergence has been rigorously demonstrated [Kong, R., and Spanier, J., J. Comp. Phys., 227(23), 9762–9777 (2008)] for these "first generation" (G1) algorithms but their practical utility is limited by computational complexities resulting from the expansion. Recently, we have developed new adaptive algorithms that overcome most of the computational restrictions of the earlier algorithms and we have also established the geometric convergence of these "second generation" (G2) algorithms [Kong, R. and Spanier, J.: Geometric convergence of second generation adaptive Monte Carlo algorithms for general transport problems based on sequential correlated sampling. In review]. In this paper we outline the main ideas involved and indicate how the resulting G2 algorithm might be optimized using information drawn from simulations of both the RTE and the dual RTE. Simple examples will illustrate these ideas and the gains in computational efficiency that the new methods can achieve.

Jerome Spanier
Beckman Laser Institute
University of California, Irvine
Irvine, California
url: http://www.bli.uci.edu

Rong Kong, Martin Ambrose, Jerome Spanier
Claremont Graduate University,
Claremont, California
url: http://www.cgu.edu

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_30,

1 Introduction

Monte Carlo (MC) simulations have provided a "gold standard" of computational support for many important problems of science and engineering that are modeled using the radiative transport equation (RTE). This situation has persisted even though MC converges very slowly when implemented in conventional ways and in spite of the availability of many faster methods based on approximations to the RTE (e.g., the diffusion approximation). No doubt one of the reasons for MC's lasting prominence in the field is that it is capable of describing the transport medium as accurately as the underlying physics is known. That is, when the sources of radiation, the material properties and physical characteristics of the medium are described in detail as input variables and parameters, the MC simulation produces a solution of the RTE with a precision that is limited only by the total number, W, of independent random walks generated. Thus, if a method could be devised to dramatically accelerate the convergence of MC simulations, its usefulness as a computational standard would certainly increase.

Computing costs associated with Monte Carlo simulation can be reduced by applying standard variance reduction methods such as correlated sampling [42, 38], importance sampling [42, 34, 35, 22, 15] and others [42, 14, 17], but conventional use of such methods does nothing to alter the underlying *rate* of convergence. As well, the success of such methods often relies on the use of auxiliary information (such as an approximate importance function) and on the user's skill at applying it to the MC simulation. This is the case, for example, with the popular weight-windows scheme [20, 4] used in the MCNP code [21] developed at Los Alamos National Laboratory. This method attempts to control both the largest and the smallest values of each estimating random variable by a procedure that depends on generating a crude estimate of a problem importance function, and by applying splitting and Russian roulette [42, 12, 11] when estimator values fall outside a pre-established "window". In some cases a deterministic calculation supplies this auxiliary information [45, 13] and in others, a series of simulations in which increasingly focused conventional MC solutions are used to control the individual contributions to the final tallies. Los Alamos staff regularly offer workshops on the use of variance reduction techniques in their code MCNP. The manual [4] is used in conjunction with these workshops and it illustrates a wide variety of variance reduction strategies that are applied to a sample problem specifically constructed to be computationally challenging. Gains in efficiency ranging from a few per cent to 1-2 orders of magnitude are reported in [4] when skillful use is made of variance reduction methods chosen with the sample problem in mind. In [10], gains reported range from a factor of 1.9 to 75 when weight windows is applied to a deep penetration problem, while more modest gains (of less than a factor of 10) are described in [37]. Examination of references [45, 13, 32] confirms that gains resulting from the use of weight windows cluster between 10 and 50 with only rare exceptions, even when auxiliary calculations are used to determine approximate importance functions for each problem. While gains such as these often mean the difference between running MCNP for only a few hours or needing weeks or longer of information processing, they offer

little hope of producing accuracies of a fraction of 1% across the broad spectrum of radiative transport problems. Furthermore, *ad hoc* procedures for variance reduction involve substantial human costs and can greatly increase the investment of both time and labor in solving a single transport problem in practical situations. **What is badly needed is an automated, highly efficient MC solution algorithm that "tunes" itself to the specific needs of each RTE problem and requires minimal or no user intervention.**

In 1962, John Halton [16] introduced methods for exponentially accelerating Monte Carlo solutions of matrix problems by making use of either correlated sampling or importance sampling, sequentially applied. In a series of papers [3, 5, 6, 7], Booth examined the possibility that adaptive Monte Carlo methods might also be applied to continuous transport problems by successively improving estimates of zero variance importance sampling estimators. In related work, Kollman's Stanford dissertation [23] showed that the methods Booth was exploring could produce exponential acceleration of convergence, and in [24] this was proved for finite state spaces.

Beginning in about 1996, Booth and his co-workers at Los Alamos National Laboratory (LANL) [8, 7, 4, 9] and Spanier and his co-workers at Claremont Graduate University (CGU) [18, 25, 29, 28, 33, 44, 36, 41, 43] succeeded in extending Halton's ideas to Monte Carlo solutions of continuous radiative transport problems. While the LANL work focused mainly on importance sampling as the variance reduction mechanism and the CGU group concentrated more on correlated sampling methods, both of these variance reduction techniques were studied by the two groups. In [40] the authors used a control variates mechanism adaptively to accelerate the convergence of estimators for discrete state spaces and showed that exponential acceleration could be achieved under special conditions. Recently [31, 27] we have demonstrated that the methods discussed in Sections 2 and 3 of this paper converge geometrically for general continuous radiation transport problems. By geometric convergence we mean methods that produce strict error reduction (with probability 1)

$$E_m \leq \lambda E_{m-1} \quad \text{for} \quad 0 < \lambda < 1$$

where E_m is the error after m adaptive stages. Then we have

$$E_m \leq \lambda^m E_0$$

where E_0 is the error produced by any initial estimate; e.g., a modest conventional MC estimate.

An additional aim of ours has been to devise geometrically convergent algorithms that are *fully automated* and *general* - algorithms that can be implemented with little or no user intervention and apply, in principle, to very general transport problems. Subject only to very mild conditions[1] that guarantee the existence and uniqueness of the RTE solution [42], the algorithms described here can accommo-

[1] The main restriction is that the RTE kernel defines an integral operator whose L_∞ norm is strictly less than 1 (see [31] and [42] for a slightly weaker condition.)

date very general geometries, boundary conditions and material heterogeneities that arise in fields as diverse as nuclear reactor design, atmospheric physics, biomedical applications and financial modeling. To date we have developed two different sequential correlated sampling strategies for achieving geometric convergence for general RTE problems and established rigorously their geometric convergence [31, 27]. The sampling methods used and the random variables that provide the estimates themselves are described in some detail in [26]. In this paper our purpose is to present the main ideas involved to a broad mathematical audience, exhibit the convergence characteristics of our adaptive algorithms using a very simple model RTE problem and outline plans for future research. Briefly, having established the geometric convergence of these algorithms in other papers, our goal now is to apply the new algorithms to increasingly realistic problems making use of major production Monte Carlo codes (e.g., MCNP [21] at Los Alamos National Laboratory and the Virtual Tissue Simulator [19] (VTS) currently under development at the University of California, Irvine). These are very versatile Monte Carlo codes that are designed for widespread use in the broad research communities they serve.

Our plan to design fast and accurate MC simulations that require little or no user intervention or special knowledge places severe demands on algorithm design. This is because real applications are very diverse, often incorporating severe heterogeneities in all of the phase space variables. For steady state transport these are the scalar energy E (or speed v), the spatial variables x, y, z and the directional variables μ and ϕ that describe unit vectors on the surface of the unit sphere in 3 dimensions ($\mu = \cos\theta$, where $\theta =$ polar angle, $\phi =$ azimuthal angle). The solutions of such problems can vary by many orders of magnitude over the phase space and display steep gradients that place extreme demands on algorithms designed to perform well irrespective of individual problem details. In this paper our purpose is to describe the evolution of our thinking about adaptive algorithms that are capable of meeting these diverse needs, and apply them to a very simple model transport problem to illustrate their potential. In other papers we will test the methods on prototype problems that are intended to pose specific challenges that arise in practice. For example, in [2] we apply the methods discussed here to model reactor problems featuring severe heterogeneities in the energy variable.

2 Problem Setting

Whereas the scalar energy is a key independent variable in RTE problems dealing with neutron and electron transport, both spatial and directional variation must be accommodated in nearly all transport problems. Indeed, it is because the RTE solution tends to be highly anisotropic at locations close to sources and detectors, as well as near internal material interfaces, that approximations such as those based on diffusion theory prove to be inadequate. Another feature prominent in many transport problems is the dramatic fall in radiation intensity with distance from radiation sources. Thus, when detectors are very distant from sources of radiation, as in neu-

tron shielding problems and in many biophotonics problems, if one is to solve such problems with relative precisions of 1% or less in realistic time frames, it is absolutely essential to achieve effective variance reduction without increasing the computational costs unduly. The utility of our adaptive methods will be severely tested when our algorithms are fully incorporated into Monte Carlo code systems such as MCNP and VTS. Here we will only describe a very simple problem with input parameters chosen to be typical of those encountered in the biomedical applications. This problem is so simple that one can easily solve it exactly without resorting to Monte Carlo methods at all, so it provides an excellent first example to study since relative precisions achieved with our adaptive algorithms can be computed exactly for it.

A typical optical probe is an instrument that introduces light into tissue from a laser source at the tissue surface and also collects light reemitted from the surface using one or more detectors positioned at fixed distances from the source. Light scattering in tissue is predominantly forward-directed so that many individual scattering events are required to produce an isotropic light distribution (deep in the interior of tissue) from an initial distribution that is collimated at the light source. The anisotropic nature of the light distribution in tissue is critical for making accurate predictions in biomedical problems, especially when the distances between the sources and detectors of the light are small, as in making measurements in small animal models. Both the forward RTE problem (modeling the light field produced by the source throughout the tissue) and the inverse RTE problem (characterizing the tissue properties from information encoded in the reemitted light) are important in understanding and interpreting light-tissue interactions. Highly accurate forward representations of the light field are needed to solve the inverse problems, which are frequently very ill-conditioned.

To introduce our adaptive Monte Carlo methods to researchers who use Monte Carlo in other than transport settings, we will apply them here to a very simple transport problem modeling a homogeneous slab of tissue of finite thickness T which is infinite in extent in the other two dimensions. A related one-dimensional family of

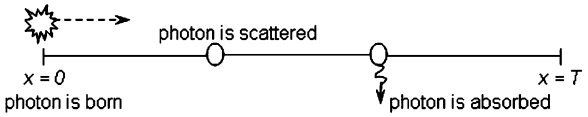


Fig. 1 Schematic of the model problem physics.

biomedical examples is studied in [26], while in [31] and [27] we apply the methods to two dimensional RTE problems.

A light source is introduced at one boundary (prescribed by $x = 0$) of the tissue and only forward photon scattering and absorption are permitted throughout the slab (see Figure 1). In real tissue, the scattering of photons is often modeled using the

Henyey-Greenstein probability density function that is characterized by specifying the average cosine of the scattering angle, g :

$$f_{HG}(\cos\theta) = \frac{1}{2}\frac{1-g^2}{\left(1-2g\cos\theta+g^2\right)^{3/2}} \tag{1}$$

where θ is the angle of deflection of the photon following collision. In tissue, g ranges from 0.7-.97, whereas in our model problem, $g = 1$. However, this assumption allows us to model this problem using a single spatial variable $0 \le x \le T$ and yet incorporate some realistic practical challenges such as those described earlier. For this purpose we designate a light source of strength Q_0 at $x = 0$ and describe the interactions of the light with the tissue chromophores by means of the constants μ_a and μ_s, the optical absorption and scattering coefficients, respectively. The total attenuation coefficient then becomes $\mu_t = \mu_a + \mu_s$ and characterizes the exponentially distributed distances traveled by photons between their successive collisions within the tissue. Thus, the function

$$T(x;y) = \begin{cases} \mu_t \exp[-\mu_t(x-y)] & 0 \le y \le x \le T \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

is sampled to determine transport from position y to position x. That is, distances d are sampled from the exponential density function $\mu_t \exp[-\mu_t d]$ $(0 \le d < \infty)$ to determine the distance d from position y to the next collision position x, and points x that lie beyond T are rejected because they lie outside the tissue under investigation. The ratio μ_s/μ_t defines the probability that the collision at x produces (forward) scattering rather than absorption of the photon. This leads to the integral form of the governing RTE

$$\begin{aligned} \Psi(x) &= Q_0 \exp(-\mu_t x) + \int_0^T K(x;y)\Psi(y)dy,\ 0 \le x \le T \\ &= S(x) + \int_0^T K(x;y)\Psi(y)dy,\ 0 \le x \le T \end{aligned} \tag{3}$$

where

$$K(x;y) = \begin{cases} \mu_s \exp[-\mu_t(x-y)] & 0 \le y \le x \le T \\ 0 & \text{otherwise.} \end{cases}$$

The solution, $\Psi(x)$, is the expected photon density at x

$$\Psi(x) = Q_0 \exp[-\mu_a x] \qquad 0 \le x \le T. \tag{4}$$

In the most general (steady-state) case, photon transport is modeled in a 5 dimensional phase space $\Gamma = V \times S^2$ consisting of vectors $\mathbf{P} = (\mathbf{r}, \Omega)$, where $\mathbf{r} = (x, y, z) \in V$, a closed, bounded subset of $\mathcal{R}^3$ and $\Omega \in S^2$ is the space of unit

direction vectors that describe the direction of photon transport at each position (x, y, z). It can be shown [27] that the integral equation

$$\Psi(\mathbf{P}) = \int_{V \times S^2} \Psi(\mathbf{P}') K(\mathbf{P}; \mathbf{P}') d\mathbf{P}' + S(\mathbf{P}) \tag{5}$$

generalizes (3), and this establishes the notation we will adopt throughout the balance of this paper.

3 First Generation (G1) Methods: Solution Expansion

During the period 1996-2003 we developed adaptive first generation (G1) Monte Carlo algorithms capable of unlimited precision based on expansion of the RTE solution or adjoint solution in an infinite series of orthonormal basis functions. That is, the solution of (5) is represented by

$$\Psi(\mathbf{P}) = \sum_{i=1}^{\infty} a_i B_i(\mathbf{P})$$

and is then truncated after M terms to produce an approximate solution

$$\tilde{\Psi}(\mathbf{P}) = \sum_{i=1}^{M} a_i B_i(\mathbf{P}) \approx \Psi(\mathbf{P}).$$

The G1 algorithm then estimates, using conventional MC methods, the expansion coefficients

$$a_i = \int_{\Gamma} B_i(\mathbf{P}) \Psi(\mathbf{P}) \tag{6}$$

in a sequence of adaptive stages of ever-increasing accuracy. Details about our sequential correlated sampling (SCS) implementation of this strategy can be found in [28, 31].

Although we have developed G1 algorithms that make use of both correlated sampling and importance sampling as variance reduction techniques, in recent years we have concentrated primarily on correlated sampling methods and we restrict our attention in this paper to these. The basic idea of correlated sampling is to find an approximation to the solution and develop an equation whose solution provides an additive correction to that approximation. Then an unbiased estimator of the difference should have a smaller variance than that of the initial approximation. This basic variance reduction strategy is then applied recursively in our sequential correlated sampling (SCS) implementations and leads to a sequence of ever-smaller additive corrections that provide increasingly accurate reconstructions of the full RTE solution when added to the initial approximation.

To implement this idea for integral equations, we begin with an initial approximation, $\tilde{\psi}^0(\mathbf{P})$, of the solution and introduce a first correction, $\psi^1(\mathbf{P})$, to this solution by setting

$$\Psi(\mathbf{P}) = \tilde{\psi}^0(\mathbf{P}) + \psi^1(\mathbf{P}). \tag{7}$$

Substituting (7) into (5) produces an equation for ψ^1

$$\psi^1(\mathbf{P}) = \int_\Gamma K(\mathbf{P},\mathbf{Q})\psi^1(\mathbf{Q})d\mathbf{Q} + S^1(\mathbf{P}) \tag{8}$$

where

$$S^1(\mathbf{P}) = S(\mathbf{P}) + \int_\Gamma K(\mathbf{P},\mathbf{Q})\tilde{\psi}^0(\mathbf{Q})d\mathbf{Q} - \tilde{\psi}^0(\mathbf{P}). \tag{9}$$

Equation (8) can be solved by conventional MC to produce an approximate solution $\tilde{\psi}^1(\mathbf{P})$. The reduced source (9) describes the residual - that is, the error made when the approximation $\tilde{\psi}^0(\mathbf{P})$ is substituted into the RTE (5). Although this "source" has only mathematical, not physical relevance to the original biomedical problem (for example, it will not, in general, be of one sign for all $\mathbf{P} \in \Gamma$), the function $S^1(\mathbf{P})$ may nevertheless be used to initiate random walks throughout the phase space by conventional Monte Carlo methods.

Continuing in this way to produce approximate corrections $\tilde{\psi}^j$, $j = 1, ..., n$, we define the approximate full solution after stage n by setting

$$\tilde{\Psi}^n(\mathbf{P}) = \tilde{\psi}^0(\mathbf{P}) + \tilde{\psi}^1(\mathbf{P}) + \cdots + \tilde{\psi}^n(\mathbf{P}). \tag{10}$$

The critical step in implementing this strategy is to define a reduced source for adaptive stage n as

$$\begin{aligned} S^n(\mathbf{P}) &= S^{n-1}(\mathbf{P}) + \int_\Gamma K(\mathbf{P},\mathbf{Q})\tilde{\psi}^{n-1}(\mathbf{Q})d\mathbf{Q} - \tilde{\psi}^{n-1}(\mathbf{P}) \\ &= S(\mathbf{P}) + \int_\Gamma K(\mathbf{P},\mathbf{Q})\tilde{\Psi}^{n-1}(\mathbf{Q})d\mathbf{Q} - \tilde{\Psi}^{n-1}(\mathbf{P}). \end{aligned} \tag{11}$$

We have recently established the geometric convergence of $\tilde{\Psi}^n(\mathbf{P})$ to $\Psi(\mathbf{P})$ in the sense of the following theorem: [31].

Theorem 1. *Let $\tilde{\Psi}^n(\mathbf{P})$ be the approximation to the solution of (5) from adaptive stage n. Then, under suitable conditions on the source S and kernel K, of the RTE, for any $\varepsilon > 0$ and any $0 < \lambda < 1$, there is a threshold number of random walks, W_0, per stage, that is independent of the stage number n, that assures that the inequality*

$$\Pr\left\{\left\|\Psi - \tilde{\Psi}^n\right\|_\infty \le \lambda \left\|\Psi - \tilde{\Psi}^{n-1}\right\|_\infty\right\} > 1 - \varepsilon$$

where

$$\left\|\Psi - \tilde{\Psi}^n\right\|_\infty = \max_{\mathbf{P}\in\Gamma}\left|\Psi - \tilde{\Psi}^n\right|$$

is satisfied for any $W \ge W_0$.

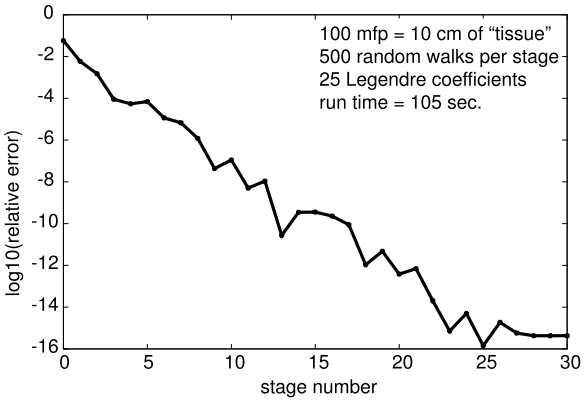


Fig. 2 Geometric reduction in error, G1 solution.

To illustrate how this G1 algorithm performs, we used it to solve the simple 1D tissue problem described earlier. We chose $Q_0 = 1$, $\mu_a = 0.01/\text{mm}$, $\mu_s = 0.99/\text{mm}$, $T = 100$ to specify the problem and the goal was to estimate the average value of the solution (4) over the final 10 mean free paths of the tissue; i.e., we estimate

$$\frac{1}{10}\int_{90}^{100} \Psi(x)dx = .386902186.$$

Figure 2 exhibits the clear geometric decrease of the logarithm of the relative error in this estimate as the number of adaptive stages increases. We have used a basis set of Legendre polynomials and truncated the RTE solution expansion after 25 terms. Notice that the error reaches machine precision in about 25 adaptive stages and that the full 30 stage run required less than 2 minutes on a 2.80GHz Pentium 4 computer.

This example is presented just to illustrate typical geometric convergence and the power of the SCS algorithm in solving very simple problems. While very high precision can be achieved very quickly with this G1 algorithm in such simple homogeneous model problems, that efficiency cannot be maintained when more complex and heterogeneous problems are attempted. This is because of an explosive growth in computational complexity caused by the need to calculate vastly increased numbers of expansion coefficients in order to achieve useful error levels. Thus, there is a geometric growth in expansion coefficients with the number of independent phase space variables. When this reaches 3 or more, and even in 1 or 2 dimensional problems when the solution deviates sufficiently from a globally defined polynomial, computational efficiency is severely compromised. This should not surprise us since the "correct" basis set for an efficient global representation of *any* transport problem is related to the spectral analysis of the integral operator defined by the transport kernel; thus, it depends on the details of each problem. *No fixed* basis will

suffice to provide accurate and economical representations of the RTE solution for *all* RTE problems.

4 Second Generation (G2) Methods: Avoiding Expansion

For reasons such as those just reviewed, we began in 2003 to move away from the idea of identifying a *globally* defined RTE function based on truncated infinite series expansions. Instead, we concentrated on representing RTE solutions *locally* in terms of low degree polynomials. That is, we look for simple, spline-like approximations that give rise to efficient MC implementation methods. Instead of striving for unlimited accuracy at each point of the phase space, our new goal became to obtain only the required accuracy at those locations where radiation measurements are to be made. In other words, we want to model each problem taking into account only those features essential for accurate estimation of the radiation measurements planned.

We pose the question: Can we relax the condition of achieving infinite precision in the RTE solution at every point of the phase space Γ to the more modest one of estimating only a small number of linear functionals of the transport solution (the "measurements") with sufficient accuracy? That is, can we estimate integrals such as $\int_\Gamma S^*(\mathbf{P})\Psi(\mathbf{P})d\mathbf{P}$ accurately *without* estimating $\Psi(\mathbf{P})$ for all $\mathbf{P} \in \Gamma$ (or even *without representing* $\Psi(\mathbf{P})$ *globally* in basis functions)? Our hope in doing so would be to restore the low computational cost that was the initial promise of geometrically converging algorithms.

We define a fixed decomposition $\Pi = \{\Gamma_i\}_{i=1}^R$ of the underlying phase space $\Gamma = \cup_{i=1}^R \Gamma_i$ where $\Gamma_i \cap \Gamma_j = \emptyset$. The simplest local representation of $\Psi(\mathbf{P})$ would be as a histogram; i.e., a piecewise constant approximation $\Psi_a(\mathbf{P})$ whose value in each region is the average of the RTE solution in that region:

$$\Psi_a(\mathbf{P}) = \begin{cases} \Psi_{ai} = \frac{1}{vol(\Gamma_i)} \int_{\Gamma_i} \Psi(\mathbf{P})d\mathbf{P}, & \mathbf{P} \in \Gamma_i \\ 0, & \mathbf{P} \notin \Gamma_i. \end{cases}$$

The second generation (G2) method we have developed based on this idea also uses correlated sampling as its variance reduction mechanism. We simply obtain an initial piecewise constant approximating RTE solution $\tilde{\Psi}_a^0(\mathbf{P})$ by any means at our disposal (e.g., from a conventional Monte Carlo simulation using track lengths in each subregion to estimate regionwise averages of the solution) and then define a reduced source iteratively by mimicking the formulas we used for our G1 SCS algorithm.

We define a reduced source for our *averaged* SCS (ASCS) G2 algorithm by

$$\begin{aligned} S_a^n(\mathbf{P}) &= S_a^{n-1}(\mathbf{P}) + \int_\Gamma K(\mathbf{P},\mathbf{Q})\tilde{\psi}_a^{n-1}(\mathbf{Q})d\mathbf{Q} - \tilde{\psi}_a^{n-1}(\mathbf{P}) \\ &= S(\mathbf{P}) + \int_\Gamma K(\mathbf{P},\mathbf{Q})\tilde{\Psi}_a^{n-1}(\mathbf{Q})d\mathbf{Q} - \tilde{\Psi}_a^{n-1}(\mathbf{P}) \end{aligned} \tag{12}$$

with

$$S_a^0(\mathbf{P}) = S(\mathbf{P})$$

where

$$\tilde{\Psi}_a^{n-1}(\mathbf{P}) = \tilde{\psi}_a^0(\mathbf{P}) + \tilde{\psi}_a^1(\mathbf{P}) + \cdots + \tilde{\psi}_a^{n-1}(\mathbf{P})$$

and we follow the same basic algorithm strategy outlined in the previous section and detailed in [26].

The resulting algorithm has also been shown [27] to produce geometric convergence to a histogram approximation of the RTE solution whose accuracy is determined by the phase space decomposition Π chosen. The statement follows:

Theorem 2. *Let $\tilde{\Psi}_a^n(\mathbf{P})$ be the approximation to the solution from adaptive stage n produced by the G2 algorithm relative to a phase space decomposition Π. Then, under suitable conditions on the source S and kernel K, for any $\varepsilon > 0$ and any $0 < \lambda < 1$, there is a threshold number of random walks, W_0, per stage, that is independent of the stage number n, that assures that the inequality*

$$\Pr\left\{\left\|\Psi - \tilde{\Psi}_a^n\right\|_\infty \leq \lambda \left\|\Psi - \tilde{\Psi}_a^{n-1}\right\|_\infty + r_\Pi\right\} > 1 - \varepsilon$$

where

$$r_\Pi = \sup_{\mathbf{P}\in\Gamma} |\Psi(\mathbf{P}) - \Psi_a(\mathbf{P})|$$

is satisfied for any $W \geq W_0$.

Both Theorem 1 and Theorem 2 only establish the *existence* of threshold numbers, W_0, of random walks to generate in each adaptive stage to guarantee geometric convergence. The proofs require only that the kernel $K(\mathbf{P}, \mathbf{Q})$ is norm-reducing (see footnote 1), a condition that is naturally satisfied for the transport problems under investigation. The bounds provided by the proofs of the theorems are quite conservative so that, in practice, a few inexpensive sample runs need to be made currently to decide on reasonable choices for W_0. This is the approach we have taken in our work so far; it is explored experimentally (numerically) in the context of the model 2 dimensional problems studied in [31]. In practical applications that make use of these adaptive methods we expect to provide simple guidelines for the user to make sensible choices for the key input parameters, such as the number of subregions of the phase space decomposition and the number of random walks to generate per subregion. Eventually we hope that spectral analysis of the RTE kernel will yield more precise estimates of these parameters.

In Figure 3 we illustrate the geometric convergence that results from applying this G2 ASCS algorithm to the same model slab RTE problem described earlier. Again we notice the rapid geometric decrease of the logarithm of the relative error as the number of adaptive stages increases. While only four G2 adaptive stages are shown in Fig. 3, we actually ran 8 stages of the G2 algorithm, with the tissue decomposed into 1000 uniform subintervals. Only the first 4 stages are shown in the figure since there is no appreciable change in the estimates after about the third stage. We initiated 5 random walks uniformly in each subinterval (making use of a technique we have described in [31] as "backward sampling", based on a simulation

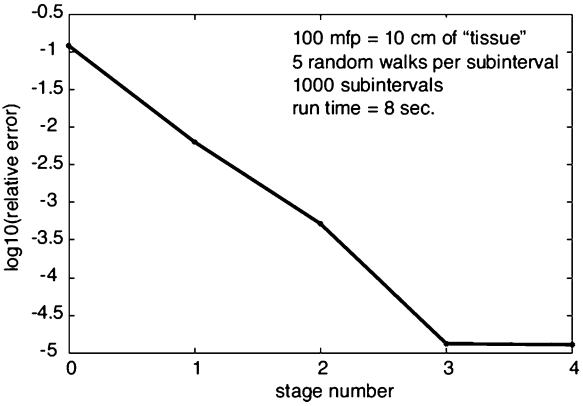


Fig. 3 Geometric reduction in error, G2 solution.

of the RTE that is adjoint to the original RTE) and the total run required only 8 sec. on the same 2.80GHz Pentium 4 computer used to illustrate the G1 algorithm performance.

The G2 convergence theorem states that the ASCS algorithm converges geometrically to an estimate whose accuracy depends on the chosen decomposition of the phase space Π, as measured by

$$r_\Pi = \max_{\mathbf{P}\in\Gamma} |\Psi(\mathbf{P}) - \Psi_a(\mathbf{P})| .$$

Evidently, if Π_1 is a decomposition of Γ that is a refinement of Π, then $r_{\Pi_1} \leq r_\Pi$. This raises a number of interesting questions.

Q1. Can the phase space Γ be refined *intelligently* to extend the G2 geometric convergence to lower error levels?
Q2. If Q1 can be answered affirmatively, what is an optimal strategy for phase space refinements?

As we will see, we believe that an approach based on accumulating combined information derived from simulations of both the original and the adjoint RTEs holds the key to developing a strategy for intelligent *non-uniform* refinements of Γ. We next outline our third generation (G3) algorithm design based on this idea.

5 Third Generation (G3) Algorithms: Intelligent Mesh Refinement

The idea underlying our G3 algorithm design is intuitively very plausible. Given a single source of radiation described by the source function $S(\mathbf{P})$ and a single "detec-

tor" of radiation, described by a detector function $S^*(\mathbf{P})$ (adjoint source), we define an *information density function (IDF)*

$$I(\mathbf{P}) = \Psi(\mathbf{P})\Psi^*(\mathbf{P})$$

as the product of the solutions of the RTE equation with source $S(\mathbf{P})$ and the adjoint RTE equation with source $S^*(\mathbf{P})$. This product function, which has been called the *contributon* in older literature [47, 46, 1, 39] satisfies an RTE with a number of interesting properties that are easily derived from the RTE equations for Ψ and Ψ^* :

1. The information defined by $I(\mathbf{P})$ is neither absorbed nor does it leak from the phase space Γ [47, 46]. That is, the RTE equation that is satisfied by $I(\mathbf{P})$ incorporates a kernel that does not allow for absorption, only for scattering from one direction to another. In [46], this conservation law for information is interpreted as characterizing a lossless "flow" of information from the source (defined by the function $S(\mathbf{P})$) to the "sink" (defined by the function $S^*(\mathbf{P})$)
2. The scattering of $I(\mathbf{P})$, as described by the RTE it satisfies, directs contributon "flow" from regions of low importance to regions of higher importance, where importance is measured by the importance function $\Psi^*(\mathbf{P})$.

Furthermore, because $\Psi(\mathbf{P})$ may be described as the intensity of radiation at $\mathbf{P}$ from the source S while $\Psi^*(\mathbf{P})$ is the expected contribution to the detector from a unit weight particle initiated at $\mathbf{P}$, the function $I(\mathbf{P})$ may be interpreted as the (relative) importance of the point $\mathbf{P}$ in transmitting radiation from the source to the detector. It is this attractive interpretation that suggests that the function $I(\mathbf{P})$ should be involved intrinsically in designing optimized transport of radiation from source(s) to detector(s).

We have tested algorithms that make use of this idea on some simple RTE problems and it seems very promising so far. Our strategy is to initiate a G2 algorithm based on a relatively crude initial subdivision of the phase space. Theorem 2 assures that this algorithm will converge to a histogram representation $\tilde{\Psi}_a(\mathbf{P})$ of the RTE solution $\Psi(\mathbf{P})$, provided a sufficient number W of random walks is used in each stage. The function $\tilde{\Psi}_a(\mathbf{P})$ can be used to form an initial estimate of the detector response $\int_\Gamma S^*(\mathbf{P})\tilde{\Psi}_a(\mathbf{P})d\mathbf{P}$. The accuracy in this initial estimate will, according to Theorem 2, depend on the crudeness of the phase space decomposition Π used as determined by $r_\Pi = \max_{\mathbf{P}\in\Gamma} |\Psi(\mathbf{P}) - \Psi_a(\mathbf{P})|$. Using the same initial decomposition of Γ, the G2 algorithm is also applied to the adjoint RTE with source S^* and produces a histogram approximation, $\tilde{\Psi}^*_a(\mathbf{P})$, to the solution of the adjoint RTE, $\Psi^*(\mathbf{P})$. The components of these two vector-valued functions consist of estimates of the average values of the two solutions over the subregions of the phase space decomposition.

Our refinement strategy examines the product of these two functions across the initial decomposition of the phase space Π and identifies those subregions Γ_j where the integral of the product $\tilde{\Psi}_{aj} \cdot \tilde{\Psi}^*_{aj}$ is large and those where it is small. It then uniformly subdivides the regions in which the integrals are large while leaving the regions with very small integrals either undivided or combined, if necessary, to increase their IDF values. The overall goal of the G3 learning is to create a phase

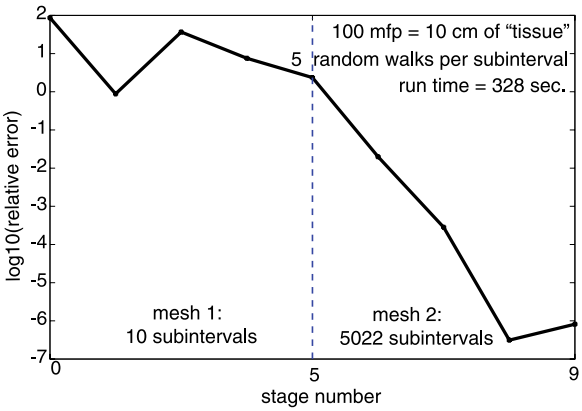


Fig. 4 Geometric reduction in error, G3 solution.

Table 1 Comparison of efficiencies of each method.

Method	S	Est	$\|E\|$	T	Eff
exact	0	.3869022	-	-	-
CMC	1	.3400000	5.77×10^{-2}	2.5	1.0
G2	8	.3869062	1.04×10^{-5}	16	4.8×10^{6}
G3	8	.3869025	8.16×10^{-7}	328	3.8×10^{8}

space refinement Π of Γ such that the integrals of the products $\tilde{\Psi}_{aj} \cdot \tilde{\Psi}^*_{aj}$ are approximately independent of j across all subregions defined by Π. A description of the G3 algorithm strategy may be found in [26].

We have applied this technique to the same model tissue problem discussed earlier and the results are shown in Fig. 4. Here we have applied a strategy that might be used in a practical problem. Starting with a very crude initial phase space decomposition Π_0 of Γ (10 subintervals over the entire 100 mean free paths of tissue) and initiating only 5 random walks in each subinterval again, the first 4 stages of G2 learning produce only about 2 orders of magnitude error reduction because of the coarseness of the initial decomposition of Γ. However, the mesh refinement Π_1 produced by one application of the G3 learning resulted in a total of 5022 subintervals that took advantage of variations in IDF values, as sketched above. This step added another 4 or so decades of reduction in error. The total G3 run time increased to about 5.5 minutes because the number of random walks in each subinterval was maintained at 5, so there was a large increase in the total number of random walks for the second set of G2 adaptive stages based on the decomposition Π_1.

In Table 1 we compare the computational efficiencies of a conventional Monte Carlo (CMC) simulation, our G2 ASCS algorithm, and the G3 mesh refinement algorithm we just described applied to the model 1D tissue problem. The numbers in the final column of this table were found by using the central limit theorem to predict how many independent random walks would be needed by a CMC simulation to

achieve the error reduction levels of the two adaptive algorithms. In deriving these estimates we make use of the fact that the relative error (column 3 of the table) behaves like the square root of the variance of the Monte Carlo estimator [30] so we can use its square as a measure of the relative variance for each estimator. In general we would use the variance estimate itself in this computation of efficiencies since the exact value of the error would not be known.

Taking into account the uncertainties inherent in the mechanisms that supply and detect radiation in real physical systems, 3-4 digits of precision would be adequate to distinguish true signals from noise, and therefore suffice in practice. We believe that the significance of the accuracy achieved with the G2 and G3 methods is that it is obtainable with relatively simple algorithms that incorporate features "tuned" to each specific RTE problem. Thus, while it is certainly to be expected that the number of subdivisions of the phase space will increase with the dimensionality of the phase space (which is five for the biomedically relevant examples discussed here), this increase is not controlled by purely geometric factors in our algorithm design. That is, the growth is governed by the variations in the RTE and adjoint RTE solutions over the phase space, which does not, in general, lead to a product space decomposition, as in the case of multidimensional quadrature. The simple refinement strategy described above that maintains a fixed number of random walks in each subregion can also be improved significantly, we believe, with algorithm optimization.

6 Summary, Conclusions and Future Research Directions

We have demonstrated that expansion-free geometric learning is possible. The rigorous geometric convergence of the G1 and G2 algorithms has recently been established, and we have obtained numerical evidence that supports the theory and illustrates the latent power in our G2/G3 strategy over existing conventional and adaptive MC methods. We believe that adaptive MC algorithms of the sort we have developed here hold the key to making RTE modeling truly practical. We expect that the new methods will support accurate RTE modeling even in cases of complex geometric heterogeneity and subtleties in angular variation near sources and detectors. Our ongoing research is now focused on questions about how to "tune" the G2/G3 algorithm strategy to problem specifics and make use of the IDF to optimize the geometric and energy/angular details through intelligent phase space refinements.

Acknowledgements The authors gratefully acknowledge support for this work from the National Science Foundation grant NSF/DMS 0712853, the Laser Microbeam and Medical Program (LAMMP) grant NIH RR0192, and the University of California Office of the President grant UCOP 41730. We also thank Dr. Katherine Bhan for assistance in the preparation of this manuscript.

References

1. Aboughantous, C.: A Contributon Monte Carlo method. Nucl. Sci. and Eng. **108**, 160–177 (1994)
2. Bhan, K., Kong, R., Spanier, J.: Adaptive Monte Carlo Algorithms Applied to Heterogenous Transport Problems. In: Monte Carlo and Quasi-Monte Carlo Methods 2008. Springer-Verlag (to appear)
3. Booth, T.: Exponential convergence for Monte Carlo particle transport. Trans. Amer. Nucl. Soc. **50**, 267–268 (1985)
4. Booth, T.: LA-10363-MS, A sample problem for variance reduction in MCNP. Tech. rep., Los Alamos National Laboratory (1985)
5. Booth, T.: A Monte Carlo learning/biasing experiment with intelligent random numbers. Nucl. Sci. and Eng. **92**, 465–481 (1986)
6. Booth, T.: The intelligent random number technique in mcnp. Nucl. Sci. and Eng. **100**, 248–254 (1988)
7. Booth, T.: Zero-variance solutions for linear Monte Carlo. Nucl. Sci. and Eng. **102**(4), 332–340 (1989)
8. Booth, T.: Exponential convergence on a continuous Monte Carlo transport problem. Nucl. Sci. Eng. **127**, 338–345 (1997)
9. Booth, T.: Adaptive importance sampling with a rapidly varying importance function. Nucl. Sci. and Eng. **136**(3), 399–408 (2000)
10. Booth, T., Hendricks, J.: Importance estimation in forward Monte Carlo calculations. Nucl. Technol./Fusion **5**(1), 90–100 (1984)
11. Burn, K., Gualdrini, G., Nava, E.: Variance reduction with multiple responces. In: Advanced Monte Carlo for Radiation Physics, Particle Transport Simulation and Applications (eds. A. Kling, F. Barao, M. Nakagawa, L. Tavora and P. Vaz), Proc. MC2000 Conference, Lisbon, Portugal, 23-26 October 2000, pp. 687–695 (2002)
12. Burn, K., Nava, E.: Optimization of variance reduction parameters in Monte Carlo transport calculations to a number of responces of interest. In: Proc. Int. Conf. Nuclear Data for Science and Technology, Trieste, Italy, Italian Physical Society (1997)
13. Cooper, M., Larsen, E.: Automated weight windows for global Monte Carlo particle transport calculations. Nucl. Sci. and Eng. **137**(1), 1–13 (2001)
14. Goertzel, G., Kalos, M.: Monte Carlo methods in transport problems. Progess in Nuclear Energy, Series I **2**, 315–369 (1958)
15. eds. Greenspan, H., Kelber, C., Okrent, D.: Computing methods in reactor physics. Gordon and Breach, New York (1968)
16. Halton, J.: Sequential Monte Carlo. Proc. Camb. Phil. Soc. **58**, 57–78 (1962)
17. Hammersley, J., Handscomb, D.: Monte Carlo methods. Methuen and Co., Ltd., London (1964)
18. Hayakawa, C.K., Spanier, J.: Comparison of Monte Carlo algorithms for obtaining geometric convergence for model transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 214–226. Springer (1999)
19. Hayakawa, C.K., Spanier, J., Venugopalan, V.: Computational Engine for a Virtual Tissue Simulator. In: Monte Carlo and Quasi-Monte Carlo Methods 2006, A. Keller, S. Heinrich and H. Niederreiter (Eds.), pp. 431–444. Springer-Verlag (2007)
20. Hendricks, J.: A code-generated Monte Carlo importance function. Trans. Amer. Nucl. Soc. **41**, 307 (1982)
21. J. Briesmeister, E.: MCNP4C, a Monte Carlo N-Particle Transport Code System, Report CCC-660. Tech. rep., Los Alamos National Laboratory (1999)
22. Kalos, M.: Importance sampling in Monte Carlo shielding calculations. Nucl. Sci & Eng. **16**, 227–234 (1963)
23. Kollman, C.: Rare event simulation in radiation transport. Ph.D. thesis, University of California, Berkley (1993)

24. Kollman, G., Baggerly, K., Cox, D., Picard, R.: Adaptive importance sampling on discrete Markov chains. The Ann. Appl. prob. **9**, 391–412 (1999)
25. Kong, R.: Transport problems and monte carlo methods. Ph.D. thesis, Claremont Graduate University (1999)
26. Kong, R., Ambrose, M., Spanier, J.: Efficient, automated Monte Carlo methods for radiation transport. J. Comp. Phys. **227**(22), 9463–9476 (2008)
27. Kong, R., Spanier, J.: Geometric convergence of second generation adaptive Monte Carlo algorithms for general transport problems based on sequential correlated sampling. in review
28. Kong, R., Spanier, J.: Error analysis of sequential Monte Carlo methods for transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, H. Niederreiter, J. Spanier (Eds.). Springer-Verlag (1999)
29. Kong, R., Spanier, J.: Sequential correlated sampling algorithms for some transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 238–251. Springer (1999)
30. Kong, R., Spanier, J.: Residual Versus Error in Transport Problems. In: Monte Carlo and Quasi-Monte Carlo Methods 2000, K.-T. Fang, F. J. Hickernell and H. Niederreiter (Eds.), pp. 306–317. Springer-Verlag (2002)
31. Kong, R., Spanier, J.: A new proof of geometric convergence for general transport problems based on sequential correlated sampling methods. J. Comp. Phys. **227**(23), 9762–9777 (2008)
32. L. Liu, R.G.: A geometry-independent fine-mesh-based Monte Carlo importance generator. Nucl. Sci. and Eng. **125**(2), 188–195 (1997)
33. Lai, Y., Spanier, J.: Adaptive importance sampling algorithms for transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 273–283. Springer (1999)
34. Leimdorfer, M.: On the transformation of the transport equation for solving deep penetration problems by the Monte Carlo methods. Trans. Chalmers Univ. of Tech., Goteborg, Sweden **286** (1964)
35. Leimdorfer, M.: On the use of Monte Carlo methods for calculating the deep penetration of neutrons in shields. Trans. Chalmers Univ. of Tech., Goteborg, Sweden **287** (1964)
36. Li, L., Spanier, J.: Approximation of transport equations by matrix equations and sequential sampling. Monte Carlo Methods and Applications **3**, 171–198 (1997)
37. Mickael, M.: A fast, automated, semideterministic weight windows generator for mcnp. Nucl. Sci. and Eng. **119**(1), 34–43 (1995)
38. Rubinstein, R.: Simulation and the Monte Carlo Method. Wiley (1981)
39. Serov, I.V., John, T.M., Hoogenboom, J.E.: A new effective Monte Carlo midway coupling method in MCNP applied to a well logging problem. Appl. Radiat. Isot. **49**(12), 1737–1744 (1998)
40. S.H. Henderson, B. Simon: Adaptive simulation using perfect control variates. J. Appl. Prob. **41**, 859–876 (2004)
41. Spanier, J.: Geometrically convergent learning algorithms for global solutions of transport problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 98–113. Springer (1999)
42. Spanier, J., Gelbard, E.: Monte Carlo Principles and Neutron Transport Problems. Addison-Wesley (1969)
43. Spanier, J., Kong, R.: A new adaptive method for geometric convergence. In: Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 439–449. Springer (2004)
44. Spanier, J., Li, L.: General sequential sampling techniques for Monte Carlo simulations: Part 1 matrix problems. In: Monte Carlo and Quasi-Monte Carlo Methods 1996, pp. 382–397. Springer (1998)
45. Wagner, J., Haghighat, A.: Automated variance reduction of Monte Carlo shielding calculations using the discrete ordinates adjoint function. Nucl. Sci. and Eng. **128**(2), 186–208 (1998)
46. Williams, M.L.: Generalized Contributon Response Theory. Nucl. Sci. and Eng. **108**, 355–383 (1991)
47. Williams, M.L., Engle, W.W.: The concept of spatial channel theory applied to reactor shielding analysis. Nucl. Sci. Eng. **62**, 160–177 (1977)

On Array-RQMC for Markov Chains: Mapping Alternatives and Convergence Rates

Pierre L'Ecuyer, Christian Lécot, and Adam L'Archevêque-Gaudet

Abstract We study the convergence behavior of a randomized quasi-Monte Carlo (RQMC) method for the simulation of discrete-time Markov chains, known as array-RQMC. The goal is to estimate the expectation of a smooth function of the sample path of the chain. The method simulates n copies of the chain in parallel, using highly uniform point sets randomized independently at each step. The copies are sorted after each step, according to some multidimensional order, for the purpose of assigning the RQMC points to the chains. In this paper, we provide some insight on why the method works, explain what would need to be done to bound its convergence rate, discuss and compare different ways of realizing the sort and assignment, and report empirical experiments on the convergence rate of the variance and of the mean square discrepancy between the empirical and theoretical distribution of the states, as a function of n, for various types of discrepancies.

1 Introduction

Quasi-Monte Carlo (QMC) and randomized QMC (RQMC) methods can be quite effective to estimate an integral when the integrand is reasonably smooth and has low effective dimension [11, 16, 20]. But when we simulate a system (modeled as a Markov chain) that evolves over several time steps, and the integrand is a function

Pierre L'Ecuyer and Adam L'Archevêque-Gaudet
Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal, H3C 3J7, Canada
e-mail: lecuyer@iro.umontreal.ca
e-mail: larcheva@iro.umontreal.ca

Christian Lécot
Laboratoire de Mathématiques, UMR 5127 CNRS and Université de Savoie, 73376 Le Bourget-du-Lac Cedex, France
e-mail: Christian.Lecot@univ-savoie.fr

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_31,

of the sample path, the dimension is typically very large, the effective dimension can also be large, and RQMC is often not very effective.

A different type of QMC and RQMC methodology, whose RQMC version is called array-RQMC, has been introduced and developed in [8, 9, 13, 14]. This array-RQMC algorithm simulates n copies of the chain in parallel. It advances all copies by one step at each iteration, using an RQMC point set of cardinality n to generate the transitions of these chains at the given step, and a clever matching of the RQMC points to the chains. This matching is done by sorting both the chains and the points according to their successive coordinates. The idea is (loosely speaking) to induce negative dependence between the n copies, so that the empirical distribution of the n states at any given step provides a much more accurate approximation of the true distribution than if the n copies were simulated independently [14]. Empirical experiments have shown that this can improve the simulation efficiency for Markov chains simulated over several hundred steps, sometimes by factors of over 1000. Potential applications include queueing systems, option pricing in finance, reliability and risk assessment models, image generation in computer graphics, and more [2, 12, 14, 21].

The aim of this paper is to provide further insight on why the method works, examine and compare alternative ways of matching the RQMC points to the chains at each step, and report empirical experiments on the convergence rate of the variance and of the mean square discrepancy between the empirical and theoretical distribution of the states, as a function of n, for various types of discrepancies.

The remainder is organized as follows. The Markov chain setting and the estimation problems are defined in Section 2. In Section 3, we explain the array-RQMC algorithm, provide (heuristic) arguments for why an how the variance of the resulting estimator could converge faster than the Monte Carlo rate of $O(1/n)$, and discuss what would be the required ingredients to bound this convergence rate. In Section 4, we examine how to map the chains to the RQMC points at each step. Empirical investigations of the convergence rate of the variance and the mean square discrepancy are reported in Section 5. A conclusion is given in Section 6.

2 A Markov Chain Setting

We consider a Markov chain model with state space $\mathcal{X} \subseteq \mathbb{R}^\ell$, whose state evolves according to the stochastic recursion

$$X_0 = x_0, \qquad X_j = \varphi_j(X_{j-1}, \mathbf{U}_j), \ j \geq 1,$$

where x_0 is fixed, $\mathbf{U}_1, \mathbf{U}_2, \ldots$ are i.i.d. uniform random variables over the unit hypercube $(0,1)^d$, and $\varphi_j : \mathcal{X} \times (0,1)^d \to \mathcal{X}$ is a measurable mapping for each j. As usual, we assume that the uniform random variables never take the value 0 or 1, to avoid infinite realizations after they are transformed by inversion to normals, exponentials, etc. We want to estimate

$$\mu = \mathbb{E}[Y], \quad \text{where} \quad Y = \sum_{j=1}^{\tau} c_j(X_j)$$

for some measurable *cost functions* $c_j : \mathcal{X} \to \mathbb{R}$, and τ is a fixed positive integer. This can in fact be generalized to the case where τ is a random stopping time, for example the first (smallest) time when X_j hits a given subset of states. The array-RQMC also works in that case, but its performance (in terms of variance reduction) is usually not as good as when τ is fixed, according to our experiments (see also [11] for one example).

To estimate μ by ordinary Monte Carlo (MC), we proceed as follows. Given a large integer n, for each i, $i = 0, \ldots, n-1$, we generate a sample path of the chain via

$$X_{i,0} = x_0, \qquad X_{i,j} = \varphi_j(X_{i,j-1}, \mathbf{U}_{i,j}), \quad j = 1, \ldots, \tau, \tag{1}$$

where $\mathbf{U}_{i,1}, \ldots, \mathbf{U}_{i,\tau}$ are i.i.d. uniform over $(0,1)^d$, and we compute $Y_i = \sum_{j=1}^{\tau} c_j(X_{i,j})$, the realization number i of Y. These sample paths are independent. The MC estimator of μ is then

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=0}^{n-1} Y_i. \tag{2}$$

For the classical RQMC method, let $s = \tau d$ and put $\mathbf{V}_i = (\mathbf{U}_{i,1}, \mathbf{U}_{i,2}, \ldots, \mathbf{U}_{i,s/d})$. Let $P_n = \{\mathbf{V}_0, \ldots, \mathbf{V}_{n-1}\} \subset (0,1)^s$ be an s-dimensional *RQMC point set*, defined as a point set with the following properties [15, 17]: (a) each point $\mathbf{V}_i$ has the uniform distribution over $(0,1)^s$, and (b) P_n has low discrepancy in some sense (the precise meaning would depend on the definition of discrepancy that one would adopt, and this may depend on the problem context). The RQMC estimator of μ is defined as in (2):

$$\hat{\mu}_{\text{rqmc},n} = \frac{1}{n} \sum_{i=0}^{n-1} Y_i = \frac{1}{n} \sum_{i=0}^{n-1} \sum_{j=1}^{\tau} c_j(X_{i,j}), \tag{3}$$

where the $X_{i,j}$ are also defined as in the MC estimator. One difficulty here is that the dimension s can be very large when the chain has many steps.

3 The Array-RQMC Algorithm

With the array-RQMC method introduced in [13, 14], we simulate n chains in parallel, and use a d-dimensional RQMC point set P_n at each step to advance all the chains by one step, in a way that at each step j, the empirical distribution of the set of states $S_{n,j} = \{X_{0,j}, \ldots, X_{n-1,j}\}$ is a very accurate approximation of the theoretical distribution of X_j, hopefully more accurate than with standard Monte Carlo. We want the discrepancy between these two distributions to be as small as possible, for an appropriate measure of discrepancy whose choice may depend on the application.

To explain what this means and why we want to do that, let $\mu_j = \mathbb{E}[c_j(X_j)]$ be the expected cost at step j, and

$$\hat{\mu}_{\text{rqmc},j,n} = \frac{1}{n}\sum_{i=0}^{n-1} c_j(X_{i,j}), \tag{4}$$

the sample average cost over the n chains at step j. The methods considered in this paper estimate μ_j by $\hat{\mu}_{\text{rqmc},j,n}$ and are unbiased: $\mathbb{E}[\hat{\mu}_{\text{rqmc},j,n}] = \mu_j$ (for array-RQMC, see Proposition 1 below). Our goal is to reduce the variance $\text{Var}[\hat{\mu}_{\text{rqmc},j,n}]$, which in this case is the same as the mean square error $\mathbb{E}[(\hat{\mu}_{\text{rqmc},j,n} - \mu_j)^2]$. It would also be nice if we could show (under appropriate conditions) that this variance converges faster than $O(1/n)$, which is the ordinary MC rate. In the remainder of this section, we explain (with heuristic arguments) why the array-RQMC appears a sensible way to achieve that.

Let us assume for now that X_j has the uniform distribution over $\mathcal{X} = (0,1)^\ell$ for each j. This assumption is in force up to the statement of Proposition 1; after that we will relax it to cover the case of a more general distribution of X_j over $\mathbb{R}^\ell$. As is usually done to bound the mean square error for RQMC schemes [3, 4, 5, 11], we can select a reproducing kernel Hilbert space (RKHS) of functions $c_j : (0,1)^\ell \to \mathbb{R}$, from which we obtain a definition of function variation V and a corresponding definition of discrepancy D for randomized point sets in $(0,1)^\ell$, such that

$$\mathbb{E}[(\hat{\mu}_{\text{rqmc},j,n} - \mu_j)^2] \le \mathbb{E}[D^2(S_{n,j})]\, V^2(c_j), \tag{5}$$

which provides a variance bound whenever $V(c_j) < \infty$. The next step would be to make sure that $\mathbb{E}[D^2(S_{n,j})]$ is small for all j, and (ideally) that it converges faster than $O(1/n)$ for any fixed j.

In the RKHS case, $D(S_{n,j})$ is equal to the integration error of some representer function ξ_j (say) that depends on $S_{n,j}$, and such that $V(\xi_j) < \infty$. If (1) holds for all i, for some points $\mathbf{U}_{i,j} \in (0,1)^d$, then $D(S_{n,j})$ can also be written as the integration error of $\xi_j \circ \varphi_j$ by the (randomized) point set $Q_n = \{(X_{0,j-1}, \mathbf{U}_{0,j}), \ldots, (X_{n-1,j-1}, \mathbf{U}_{n-1,j})\}$. To bound this integration error, we may select another discrepancy $D_{(2)}$ defined over the $(\ell+d)$-dimensional unit hypercube, with corresponding variation $V_{(2)}$, such that for any function $g : (0,1)^{\ell+d} \to \mathbb{R}$ with $V_{(2)}(g) < \infty$, the mean square integration error of g by Q_n is bounded by $\mathbb{E}[D^2_{(2)}(Q_n)] \cdot V^2_{(2)}(g)$. This discrepancy measure $D_{(2)}$ can of course be different from D. Its role is to measure the departure of the empirical distribution of Q_n from the uniform distribution over $(0,1)^{\ell+d}$. If we can show that $V_{(2)}(\xi_j \circ \varphi_j) < \infty$ and that $\mathbb{E}[D^2_{(2)}(Q_n)] = O(n^{-\alpha+\epsilon})$ for any $\epsilon > 0$ for some constant $\alpha > 1$, then this would imply that $\text{Var}[\hat{\mu}_{\text{rqmc},j,n}]$ converges faster than $O(1/n)$, which is what we are trying to achieve. Of course, this may work only if $\xi_j \circ \varphi_j$ has sufficient "smoothness."

Note that in the points $(X_{i,j-1}, \mathbf{U}_{i,j})$ of Q_n, the last d coordinates (the $\mathbf{U}_{i,j}$) can be defined via some RQMC scheme, but the $X_{i,j-1}$ cannot be chosen; they are determined by the previous history of the chains. The aim is to select (or generate) the $\mathbf{U}_{i,j}$ in a way that $\mathbb{E}[D^2_{(2)}(Q_n)]$ is small.

In the array-RQMC algorithm defined below, we (try to) achieve this in the following way. We select an $(\ell + d)$-dimensional point set

$$\tilde{Q}_n^0 = \{(\mathbf{w}_0, \tilde{\mathbf{u}}_0), \ldots, (\mathbf{w}_{n-1}, \tilde{\mathbf{u}}_{n-1})\}$$

having low-discrepancy with respect to $D_{(2)}$, where $\mathbf{w}_i \in [0,1)^\ell$ and $\tilde{\mathbf{u}}_i \in [0,1)^d$ (these points are allowed to have zero coordinates). Then we define a randomization of $\tilde{P}_n^0 = \{\tilde{\mathbf{u}}_0, \ldots, \tilde{\mathbf{u}}_{n-1}\}$ with the property that if $P_n = (\mathbf{U}_0, \ldots, \mathbf{U}_{n-1})$ is (a realization of) the randomized version and if $\tilde{Q}_n$ is the version of $\tilde{Q}_n^0$ in which $\tilde{P}_n^0$ is replaced by its randomized version P_n, then: (a) each $\mathbf{U}_i$ is uniformly distributed over $(0,1)^d$ and (b) $\tilde{Q}_n$ has low discrepancy, in the sense that $\mathbb{E}[D_{(2)}^2(\tilde{Q}_n)]$ is small. Note that $\tilde{Q}_n^0$ does not have to be the same at all steps j, but taking the same point set (with independent randomizations at the different steps) is more convenient and works fine in practice.

Then we define a permutation π_j over $\{0, \ldots, n-1\}$, for which $X_{\pi_j(i),j-1}$ is close to $\mathbf{w}_i$ for each i, as much as possible, so that there is not much difference (loosely speaking) between the point sets $\tilde{Q}_n$ and

$$Q_{n,j}^\pi = \{(X_{\pi_j(0),j-1}, \mathbf{U}_{0,j}), \ldots, (X_{\pi_j(n-1),j-1}, \mathbf{U}_{n-1,j})\}.$$

The motivation is that if these two point sets are close to each other, then $Q_{n,j}^\pi$ should also have low discrepancy. This RQMC point set $Q_{n,j}^\pi$ is the one that turns out to be used to approximate the integral of $\xi_j \circ \varphi_j$ at step j of the algorithm. The $\mathbf{w}_i$ are fixed once for all and are the same at all steps; their role is only to define the mapping between the chains and the points of $\tilde{Q}_n$. In the case of a one-dimensional state space ($\ell = 1$), we usually take $\mathbf{w}_i = (i + 1/2)/n$ and then the best permutation π_j is the one for which the states $X_{\pi_j(i),j-1}$ are sorted in increasing order, because the $\mathbf{w}_i$ are sorted in increasing order. The choice of permutations for higher-dimensional state spaces is less obvious. We discuss it in Section 4.

The array-RQMC algorithm simulates (in parallel) n copies of the chain; it can be summarized as follows.

Array-RQMC algorithm:
For $i = 0, \ldots, n-1$, let $X_{i,0} = x_0$;
For $j = 1, 2, \ldots, \tau$ {
  Randomize $\tilde{P}_n^0$ afresh (independently of the previous
    randomizations) into a new $P_n = P_{n,j} = \{\mathbf{U}_{0,j}, \ldots, \mathbf{U}_{n-1,j}\}$;
  For $i = 0, \ldots, n-1$, let $X_{i,j} = \varphi_j(X_{\pi_j(i),j-1}, \mathbf{U}_{i,j})$;
  Compute the permutation π_{j+1} for the next step;
  }
Estimate μ by the same average $\bar{Y}_n = \hat{\mu}_{\text{rqmc},n}$ as in (3).

This can be replicated m times to estimate the variance and compute a confidence interval on μ. The following is proved in [14]:

Proposition 1. *(a) $\bar{Y}_n$ is an unbiased estimator of μ and (b) the empirical variance of the m copies of $\bar{Y}_n$ is an unbiased estimator of* $\text{Var}[\bar{Y}_n]$.

So far we have assumed that X_j has the uniform distribution over $(0,1)^\ell$, which is of course unrealistic for practical applications. In the case where X_j has a more general distribution over $\mathbb{R}^\ell$, the array-RQMC algorithm operates in exactly the same way. The only changes are in how to define the mappings π_j of chains to points and in the interpretation of the discrepancies.

It is standard in QMC studies to use discrepancies for the uniform distribution over the unit hypercube $(0,1)^\ell$, with the understanding that more general distributions over $\mathbb{R}^\ell$ can be transformed to the uniform distribution, usually via a change of variables. To follow this path, we assume that X_j has a continuous distribution and that for each j, there is a bijection $\psi_j : \mathcal{X} \to (0,1)^\ell$ such that $\psi_j(X_j)$ has the uniform distribution over $(0,1)^\ell$. We then define the discrepancy of the states at step j as

$$D_j = D_j(S_{n,j}) = D_j(X_{0,j}, \ldots, X_{n-1,j}) \stackrel{\text{def}}{=} D(\psi_j(X_{0,j}), \ldots, \psi_j(X_{n-1,j})),$$

where D is the same as earlier. In (5), $V(c_j)$ also needs to be replaced by $V(c_j \circ \psi_j^{-1})$.

We emphasize that there is no need to know ψ_j to run the algorithm. For a one-dimensional state space, the most natural definition is obviously the standard *probability integral transformation*, $\psi_j(x) = F_j(x)$, where F_j is the cumulative distribution function (CDF) of X_j. With this definition, the permutation π_j will simply sort the states by increasing order, at each step. In more than one dimension, this can be generalized as follows [19]: Given $X_j = (X_j^{(1)}, \ldots, X_j^{(\ell)})$, let $U_j^{(1)} = F_{j,1}(X_j^{(1)})$ where $F_{j,1}$ is the CDF of $X_j^{(1)}$, then let $U_j^{(2)} = F_{j,2}(X_j^{(2)} \mid X_j^{(1)})$ where $F_{j,2}(\cdot \mid X_j^{(1)})$ is the CDF of $X_j^{(2)}$ conditional on $X_j^{(1)}$, and so on. Then put $\psi_j(X_j) = U_j = (U_j^{(1)}, \ldots, U_j^{(\ell)})$. When the distribution of X_j is not continuous, this does not define a bijection, but one could still define ψ_j by taking $U_j^{(1)}$ as some solution of $U_j^{(1)} = F_{j,1}(X_j^{(1)})$, and so on.

It would be nice if we could show, under appropriate smoothness assumptions on the φ_j and ψ_j, and with proper choices of discrepancies D and $D_{(2)}$, that

$$\mathbb{E}[D_j^2] \le \kappa_j n^{-\alpha+\epsilon} \tag{6}$$

for any $\epsilon > 0$, for some $\alpha > 1$, where κ_j does not depend on n and grows only very slowly (or not at all) with j. From this, assuming that the $V(c_j \circ \psi_j^{-1}) < \infty$, it would follow that $\mathrm{Var}[(Y_0 + \cdots + Y_{n-1})/n]$ converges as $O(n^{-\alpha+\epsilon})$. A natural path to establish such a result would be to show that low mean-square discrepancy $\mathbb{E}[D_j^2]$ is preserved from one step j to the next.

At this time, we do not have a proof. We only have empirical evidence. In our numerical experiments reported in Section 5, we observed a convergence rate of $O(n^{-2})$ for the variance. On the other hand, the convergence rate of the mean square discrepancy $\mathbb{E}[D_j^2]$ (which we estimated only for one-dimensional examples) depends on the choice of discrepancy D. For example, if D is defined as the $\mathcal{L}_2$-star discrepancy, the rates observed empirically are (approximately) $O(n^{-3/2})$, whereas with D equal to the discrepancy defined in Eq. (15) of [5], we observe $O(n^{-2})$.

4 Mapping the Chains to the Points

We now discuss how to define and implement the one-to-one mapping of the n points to the n chains in the array-RQMC algorithm, so that each state is assigned to a representative point that is close to it. As in the previous section, we start with the simplified case where the chain's state X_j has the uniform distribution over $(0,1)^\ell$.

We consider the following way of mapping the chains to the points, called a *multivariate sort* [2, 7]. Select some positive integers $n_1, \ldots, n_\nu$ such that $\nu \geq \ell$ and $n_1 \cdots n_\nu = n$. Sort the states (i.e., the chains) by their first coordinate, in n_1 packets of size n/n_1. This means that any state in a given packet will have its first coordinate smaller or equal to the first coordinate of any other state in the next packet. Then sort each packet by the second coordinate, in n_2 packets of size $n/n_1 n_2$, and so on. When we reach coordinate ℓ, we sort each packet in n_ℓ packets of size $n/n_1 \cdots n_\ell$ by the last coordinate. If $\nu > \ell$, then at the next step we sort each packet into $n_{\ell+1}$ packets according to the first coordinate, and so on. As a special case of this, one can take $n_j = 2$ for all j, with n equal to a power of 2. This corresponds to splitting each packet of states in two with respect to the next coordinate, and doing this for each coordinate in a round-robin fashion.

If ℓ is deemed too large, we can map the state space to a lower-dimensional space as follows. Define a *sorting function* $v : \mathcal{X} \to [0,1)^c$, for $c < \ell$, and apply the multivariate sort to the transformed points $v(X_{i,j})$, in c dimensions. The function v should be selected so that two states mapped to nearby values in $[0,1)^c$ should be approximately equivalent in terms of the probability distribution of future costs when we are in these two states. In [14], it was assumed that such a mapping was always used, with $c = 1$, so v uniquely determined the sort, whence the appellation "sorting function."

Figure 1 illustrates the mappings obtained for two choices of n_1, namely $n_1 = n$ and $n_1 = n^{1/2}$, for an example with $\ell = 2$ and $n = 16$.

In the more general (and realistic) case where the state space is not $[0,1)^\ell$ but $\mathbb{R}^\ell$ (or a subset) and the ψ_j cannot be computed explicitly, a reasonable heuristic is to simply sort the states in the real space in exactly the same way as in the unit hypercube. This is what we will do in our examples.

Further discussion and suggestions for the mapping between the points and states can be found in [21]. On page 675, the authors assume that the points lie in a predefined two-dimensional grid, exactly one point per square of the grid (in our understanding), and use what they call a Z-curve to order the points. This would work fine to sort the points of a digital net in base 2, for example. However, sorting the states (or chains) with this scheme seems problematic, because there is generally not exactly one state per square of the grid. It would also need to be adapted in some way for the (usual) case where the state space is unbounded and ψ_j is unknown (X_j has an unknown distribution).

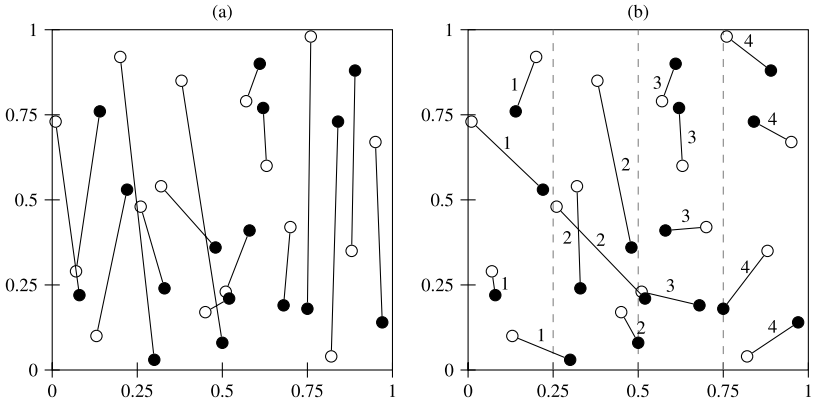


Fig. 1 Two mappings between points and states, in $\ell = 2$ dimensions, with $n = 16$. The black dots represent the states of the chains, and the white dots are the first 16 points of the two-dimensional Sobol' sequence, with a random digital shift. The lines indicate the mapping between the two sets of points. In the left picture, we have $n_1 = n$, so we sort according to the first coordinate only: the leftmost state is mapped to the leftmost point, the second leftmost state is mapped to the second leftmost point, and so on. The right picture is for $n_1 = n^{1/2} = 4$: we first sort both the points and the states in four packets according to the first (horizontal) coordinate. The numbers from 1 to 4 indicate the packet number in which each pair ended up in this first sort. Within each packet, the states are mapped to the points according to the second (vertical) coordinate. The dashed vertical lines at 1/4, 1/2, and 3/4 separate the Sobol' points in packets of four, but not the states. These dashed lines are only for visual intuition; they are not used by the sorting procedure.

5 Empirical Investigations of the Convergence Rate

We now show how the variance and the mean square discrepancy $\mathbb{E}[D_j^2]$ (for different definitions of D) behave as functions of j and n, for small examples. All mean square discrepancies and variances were estimated from 100 independent replications of the array-RQMC estimator.

5.1 Example 1: An Autoregressive Process

Consider a Markov chain defined over the real line by

$$Y_0 = 0, \qquad Y_1 = Z_1, \qquad Y_j = \frac{\beta Y_{j-1} + Z_j}{\sqrt{\beta^2 + 1}} \text{ for } j \geq 2, \tag{7}$$

where $\beta \geq 0$ (a constant) and $Z_1, Z_2, \ldots$ are i.i.d. $N(0,1)$ (standard normal). This is a simple autoregressive process of order one. We have that $Y_j \sim N(0,1)$ and $X_j = \Phi(Y_j) \sim U(0,1)$, where Φ is the standard normal CDF. The transformed state X_j has the uniform distribution over $(0,1)$ at each step j, so here we are able

to compute explicitly the mean square discrepancy $\mathbb{E}[D_j^2]$ and to see how it evolves with j and n. This can be done for this small academic example, but cannot be done in general for more realistic examples. The Markov chain can also be defined directly in terms of a stochastic recurrence for X_j, namely $X_1 = U_1$ and

$$X_j = \varphi_j(X_{j-1}, U_j) = \Phi\left(\frac{\beta\Phi^{-1}(X_{j-1}) + \Phi^{-1}(U_j)}{\sqrt{\beta^2+1}}\right) \text{ for } j \geq 2,$$

where $U_1, U_2, \ldots$ are i.i.d. $U(0,1)$. Note that for $\beta = 0$ we have $Y_j = Z_j$ for all j, whereas for $\beta \to \infty$ (in the limit), we have $Y_j = Z_1$ for all j.

Our primary interest is in how the variance of $\hat{\mu}_{\text{rqmc},j,n}$ behaves as a function of j and n for various choices of the cost function c_j. In view of our discussion in Section 3, we are also interested in the behavior of $\mathbb{E}[D_j^2]$ as a function of j and of n, for various choices of the discrepancy D. These different discrepancies correspond to different assumptions on the smoothness of c_j and/or different choices of the RQMC point set $\tilde{Q}_n$.

To fix ideas, we consider two specific choices of D. The first one is the $\mathcal{L}_2$-*star discrepancy* [3]. In one dimension, its square value is the same as the Cramer-von Mises statistic:

$$D^2(u_0, \ldots, u_{n-1}) = D^2_{2,*}(u_0, \ldots, u_{n-1}) = \frac{1}{12n^2} + \frac{1}{n}\sum_{i=0}^{n-1}(w_i - u_i)^2,$$

where $w_i = (i + 1/2)/n$ and $0 \leq u_0 \leq u_1 \leq \cdots \leq u_{n-1} \leq 1$.

We name our second example of D the *shift-baker2* discrepancy. In one dimension, its square is given by

$$D^2_{\text{shb}}(u_0, \ldots, u_{n-1}) = \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left[\frac{16}{45}\left[B_6(\{u_i - u_j - 1/2\}) - B_6(\{u_i - u_j\})\right] + \frac{1}{9}\left[10B_4(\{u_i - u_j - 1/2\}) - 19B_4(\{u_i - u_j\})\right]\right],$$

where the bold braces mean "mod 1", and B_4 and B_6 are the Bernoulli polynomials

$$B_4(u) = u^4 - 2u^3 + u^2 - \frac{1}{30} \quad \text{and} \quad B_6(u) = u^6 - 3u^5 + \frac{5}{2}u^4 - \frac{1}{2}u^2 + \frac{1}{42}.$$

This is the discrepancy given in Eq. (15) of [5], without the weights and with a correction on the coefficient of $B_4(\{u_i - u_j - 1/2\})$. This discrepancy represents the worst-case mean square error for a class of functions with square integrable second derivative, with the point set $\{u_0, \ldots, u_{n-1}\}$ randomized by a random shift modulo 1 followed by a baker's transformation [5]. Strictly speaking, this discrepancy would be appropriate only if we would apply the baker's transformation to the states X_j before computing the average cost $\hat{\mu}_{\text{rqmc},j,n}$ at each step, and we do not do that.

We nevertheless examine D_{shb} as an example to illustrate how the convergence rate might depend on the choice of discrepancy.

We also consider two choices for the two-dimensional RQMC point set used at each stage. In the first choice, we take the first n points of the two-dimensional Sobol' sequence, where the second coordinate of the points is randomized by a (random) left matrix scramble followed by a random digital shift [18]. For our second choice, we take a Korobov lattice rule with a random shift modulo 1 followed by a baker's transformation [5]. For the Korobov rule, for each n, we took the parameter a (in the usual notation) that gave the smallest shift-baker2 discrepancy in a random search over 1000 different values. The simulations were done using SSJ [10].

Our first results are for the $\mathcal{L}_2$-star $D_{2,*}$ and shift-baker2 D_{shb} discrepancies, for the Sobol' point sets. Figure 2 shows our estimate of $\mathbb{E}[D_j^2]$ (the sample average of D_j^2 over 100 independent replicates of the algorithms, as said earlier) as a function of j, with $n = 4096$ points, for $\beta = 0.1$, $\beta = 1$, and $\beta = 10$. The mean square discrepancy turns out to be quite stable even when we simulate this chain over a large number of steps. We observed the same behavior as a function of j for other discrepancies, other RQMC point sets, and also for the variance. This is very encouraging. For the shift-baker2 discrepancy and $\beta = 10$ (or any large β), $\mathbb{E}[D_j^2]$ increases in a visible way toward an horizontal asymptote as a function of j. Due to the nature of the recurrence (7), $\mathbb{E}[D_j^2]$ turns out to be an exponential smoothing of the discrepancies of previous steps, plus an additional term.

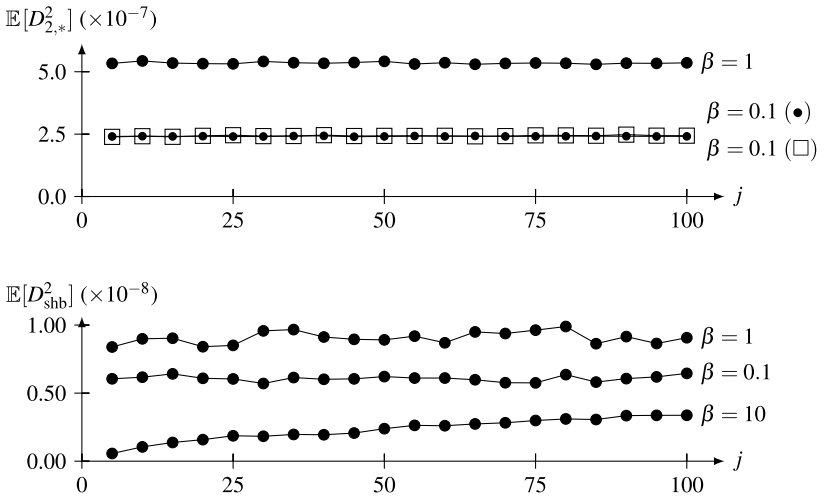


Fig. 2 Estimate of $\mathbb{E}[D_j^2]$ as a function of j for Example 5.1 with $n = 4096$, for $D = D_{2,*}$ (above) and $D = D_{\text{shb}}$ (below).

For $D_{2,*}$, the behavior for $\beta = 10$ is essentially the same as for $\beta = 0.1$. This also occurs more generally for any pair $(\beta, 1/\beta)$ where $\beta > 1$, and could be explained by the fact that it gives a linear combination of two independent standard normals

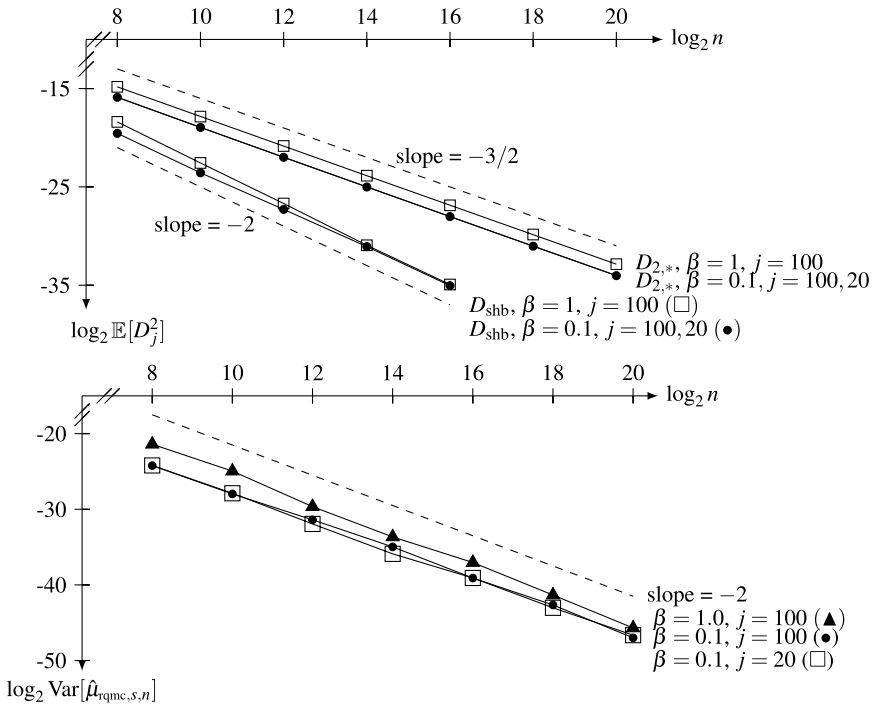


Fig. 3 Above: Estimate of $\log_2 \mathbb{E}[D_j^2]$ as a function of $\log_2 n$ for Example 5.1, for $D = D_{2,*}$ and $D = D_{\text{shb}}$. Below: Estimate of $\log_2 \text{Var}[\hat{\mu}_{\text{rqmc},j,n}]$ as a function of $\log_2 n$.

with the coefficients swapped in (7). For D_{shb}, however, the discrepancy is smaller for $\beta = 10$ than for $\beta = 0.1$. In what follows we will only report results for $\beta = 0.1$ and $\beta = 1$.

Figure 3 shows our estimate of $\log_2 \mathbb{E}[D_j^2]$ as a function of $\log_2 n$, for the same values of β, for $j = 20$ and $j = 100$, again for the Sobol' points, for $D_{2,*}$ and D_{shb}. The expectation was still estimated by the average over 100 independent replications. For $D_{2,*}$, $\mathbb{E}[D_j^2]$ seems to converge approximately as $O(n^{-3/2})$ as a function of n, and appears independent of j, as in the previous figure. With a Korobov lattice, the results are almost the same. They are also almost the same for other similar types of discrepancies such as unanchored $\mathcal{L}_2$-discrepancies defined in [4], for example. For the shift-baker2 discrepancy D_{shb}, the convergence rate differs and is (empirically) quite close to $O(n^{-2})$. The bottom part of Figure 3 shows our estimate of $\log_2 \text{Var}[\hat{\mu}_{\text{rqmc},j,n}]$ as a function of $\log_2 n$, for cost function $c_j(x) = x$, for $j = 20$ and $j = 100$. The slope indicates a convergence rate of approximately $O(n^{-2})$. This corresponds to the convergence rate of the mean square shift-baker2 discrepancy. We also tried other smooth cost functions such as $c_j(x) = x^2$, $\sqrt{x}$ and $\ln(x)$ and the observed rate was the same. The variance reduction factor compared to MC was also roughly the same for x, x^2, and $\sqrt{x}$, but it was approximately ten times smaller

for $\ln(x)$. Here, the n chains were simulated for j steps, the cost was then averaged over the n chains (at step j) to get one realization of the estimator $\hat{\mu}_{\text{rqmc},j,n}$. This was repeated $m = 100$ times, and the variance shown is the empirical variance of those m observations. The variance reduction factor, defined as the Monte Carlo variance divided by the array-RQMC variance when the two estimators are based on an average for n chains, is very roughly $600n$ when $\beta = 0.1$ and $j = 100$ (although there is significant fluctuation around this value when we change n and especially the RQMC point set that is used). The variance is also practically independent of j.

We also tried $c_j(x) = \mathbb{I}(x > 0.5)$, where $\mathbb{I}$ is the indicator function. The (empirical) convergence rate then dropped to approximately $O(n^{-3/2})$, and the variance reduction factor with respect to MC was about a hundred times smaller than for x for $n = 2^{10}$, and a thousand times smaller for $n = 2^{20}$ (that is, about half a million instead of 500 million). It is encouraging to see that even with such a discontinuous indicator function, the variance is much smaller than for MC and its convergence rate is faster (empirically).

5.2 Example 2: An Asian Option

In this example, let $0 < t_1 < t_2 < \cdots < t_s = T$ be fixed numbers (observation times), r and σ be positive constants, $S_0 = s_0$ (a constant), and

$$S_j = S_{j-1} \exp[(r - \sigma^2/2)(t_j - t_{j-1}) + \sigma (t_j - t_{j-1})^{1/2} \Phi^{-1}(U_j)] \tag{8}$$

where $U_j \sim U(0,1)$, for $j = 1, \ldots, s$. Define

$$\bar{S}_j = \frac{1}{j} \sum_{i=1}^{j} S_i .$$

We want to estimate

$$\mu = \mathbb{E}\left[\max\left(0, \bar{S}_s - K\right)\right].$$

This estimation problem occurs in pricing an Asian call option for a single asset whose price evolves as a geometric Brownian motion [6]. Note that μ is then multiplied by a constant discount factor, which we ignore here.

To put this model in our framework, we define a Markov chain with state $X_j = (S_j, \bar{S}_j)$ at step j, and whose transitions obey $(S_j, \bar{S}_j) = \varphi_j(S_{j-1}, \bar{S}_{j-1}, U_j)$ where φ_j is defined via (8) and $\bar{S}_j = [(j-1)\bar{S}_{j-1} + S_j]/j$. The function c_j is zero for $j < s$ and we have $c_s(S_s, \bar{S}_s) = \max\left(0, \bar{S}_s - K\right)$. The estimator is defined by (3) as usual. Here, $\tau = s$, we have a two-dimensional state space ($\ell = 2$), and we use a two-dimensional sort at each step: we first sort the states in n_1 packets of size n/n_1 based on $S(t_j)$, then we sort the packets based on $\bar{S}_j$.

In contrast with the previous example, we have no explicit mapping ψ_j available to transform the state into a uniform point over the unit square, so we cannot compute the discrepancy D_j explicitly. However, we can estimate the variance and

examine its convergence speed as a function of n. Our RQMC point set at each step is the first n points of a Sobol' sequence, this time in three dimensions, with coordinates 2 and 3 randomized by a left matrix scramble followed by a random digital shift.

For a numerical example, we take $S(0) = 100$, $K = 90$, $T = 240/365$, $t_1 = T - (s-1)/365$, $t_j - t_{j-1} = 1/365$, $r = \ln 1.09$, $\sigma = 0.2$, and $s = 10$ and 60.

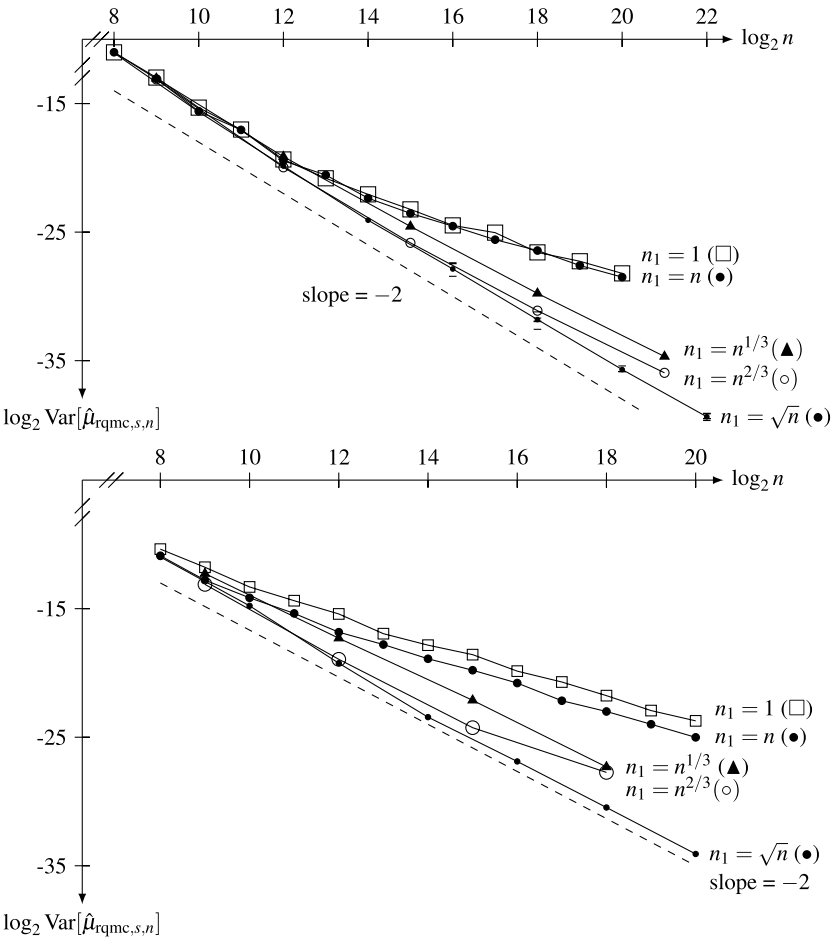


Fig. 4 Estimate of $\log_2 \mathrm{Var}[\hat{\mu}_{\mathrm{rqmc},n}]$ as a function of $\log_2 n$, for Example 5.2 with $K = 90$, for $s = 10$ (above) and $s = 60$ (below). For $s = 10$, $n \geq 2^{16}$, and $n_1 = \sqrt{n}$, we made four independent replicates of the experiment and their results are indicated by small horizontal bars on the graphs. The line goes through the log of the average variance over these four replicates.

Figures 4 show the variance as a function of n, again in a log-log scale, for different choices of n_1, for $s = 10$ and $s = 60$, respectively. The best results are with $n_1 \approx n^{1/2}$, for which the variance seems to converge approximately as $O(n^{-2})$.

For $n_1 \approx n^{1/3}$ and $n_1 \approx n^{2/3}$, the variance is larger (by a factor of about 10 for $s = 60$, $n \approx 2^{18}$, and $n_1 \approx n^{1/3}$, for example). The results are even worse if we take $n_1 = 1$ or $n_1 = n$, which corresponds to sorting the states by one of their two coordinates (this is the strategy that was used for this example in [14]). For $s = 60$ and $n \approx 2^{18}$, the variance with the best two-dimensional sort adopted here is about 400 times smaller than with a sort based on the second coordinate only. We emphasize that not only the convergence rate of the variance is better than for MC, but the variance is also much smaller for the range of values of n shown in the figure. For example, with $s = 10$, $K = 90$, and the best sorting strategy ($n_1 = \sqrt{n}$), the variance reduction factor is approximately $5n$. Thus, for $n = 2^{20}$, the variance with array-RQMC is about five million times smaller than with MC.

Of course, the variance behavior depends on the option and model parameters. For example, the probability p of a nonzero final payoff becomes very small when K is large, and the relative error (the standard deviation divided by the mean) increases without bound. This is a case of *rare event simulation*, for which RQMC is not the right tool. In that situation, we should first apply an appropriate technique such as *importance sampling* [1] to smooth out the estimator. Then we can apply RQMC for further improvement. On the other hand, the convergence *rate* of the variance for either MC or array-RQMC does not depend on K or p. With $K = 90$ and $s = 10$ as in Figure 4, we have $p \approx 0.87$. If we change to $K = 111$, for example, we get $p \approx 0.23$ and the variance reduction factor of array-RQMC over MC turns out to be about four times smaller than with $K = 90$, but we still have (approximately) an $O(n^{-2})$ convergence rate for the variance.

We also experimented with the discontinuous payoff function $c_s(S_s, \bar{S}_s) = \max(0, \bar{S}_s - K)$ for $\bar{S}_s \geq S_s$, and 0 otherwise. In this case, the convergence rate drops (empirically) to $O(n^{-1.3})$ with $s = 10$ and $K = 90$, and the variance reduction factor becomes much more modest (5 for $n = 2^{10}$ and 50 for $n = 2^{20}$). Nevertheless, these factors are non-negligible and this is encouraging.

6 Future Work and Conclusion

The array-RQMC algorithm is a promising methodology for reducing the variance in the simulation of Markov chains. We believe that plenty of interesting results on its convergence are waiting to be established, in particular for multidimensional state spaces, under various sets of assumptions on the transition and cost functions. Further empirical experimentation is also needed, with large examples, alternative sorting strategies, and various classes of applications.

Acknowledgements This research has been supported by NSERC-Canada grant No. ODGP0110050 and a Canada Research Chair to the first author, and an NSERC scholarship to

the third author. We thank the two anonymous reviewers and the Editor Art B. Owen for their helpful suggestions.

References

1. Asmussen, S., Glynn, P.W.: Stochastic Simulation. Springer-Verlag, New York (2007)
2. El Haddad, R., Lécot, C., L'Ecuyer, P.: Quasi-Monte Carlo simulation of discrete-time Markov chains on multidimensional state spaces. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 413–429. Springer-Verlag, Berlin (2008)
3. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Mathematics of Computation **67**, 299–322 (1998)
4. Hickernell, F.J.: What affects the accuracy of quasi-Monte Carlo quadrature? In: H. Niederreiter, J. Spanier (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 16–55. Springer-Verlag, Berlin (2000)
5. Hickernell, F.J.: Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules. In: K.T. Fang, F.J. Hickernell, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 274–289. Springer-Verlag, Berlin (2002)
6. Hull, J.C.: Options, Futures, and Other Derivatives, sixth edn. Prentice-Hall, Upper Saddle River, N.J. (2006)
7. Lécot, C., Coulibaly, I.: A quasi-Monte Carlo scheme using nets for a linear Boltzmann equation. SIAM Journal on Numerical Analysis **35**(1), 51–70 (1998)
8. Lécot, C., Ogawa, S.: Quasirandom walk methods. In: K.T. Fang, F.J. Hickernell, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 63–85. Springer-Verlag, Berlin (2002)
9. Lécot, C., Tuffin, B.: Quasi-Monte Carlo methods for estimating transient measures of discrete time Markov chains. In: H. Niederreiter (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 329–343. Springer-Verlag, Berlin (2004)
10. L'Ecuyer, P.: SSJ: A Java Library for Stochastic Simulation (2008). Software user's guide, available at http://www.iro.umontreal.ca/~lecuyer
11. L'Ecuyer, P.: Quasi-Monte Carlo methods with applications in finance. Finance and Stochastics, **13**(3), 307–349 (2009)
12. L'Ecuyer, P., Demers, V., Tuffin, B.: Rare-events, splitting, and quasi-Monte Carlo. ACM Transactions on Modeling and Computer Simulation **17**(2), Article 9 (2007)
13. L'Ecuyer, P., Lécot, C., Tuffin, B.: Randomized quasi-Monte Carlo simulation of Markov chains with an ordered state space. In: H. Niederreiter, D. Talay (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 331–342. Springer-Verlag, Berlin (2006)
14. L'Ecuyer, P., Lécot, C., Tuffin, B.: A randomized quasi-Monte Carlo simulation method for Markov chains. Operations Research **56**(4), 958–975 (2008)
15. L'Ecuyer, P., Lemieux, C.: Variance reduction via lattice rules. Management Science **46**(9), 1214–1235 (2000)
16. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia, PA (1992)
17. Owen, A.B.: Latin supercube sampling for very high-dimensional simulations. ACM Transactions on Modeling and Computer Simulation **8**(1), 71–102 (1998)
18. Owen, A.B.: Variance with alternative scramblings of digital nets. ACM Transactions on Modeling and Computer Simulation **13**(4), 363–378 (2003)
19. Rosenblatt, M.: Remarks on a multivariate transformation. The Annals of Mathematical Statistics **23**(3), 470–472 (1952)
20. Sloan, I.H., Joe, S.: Lattice Methods for Multiple Integration. Clarendon Press, Oxford (1994)

21. Wächter, C., Keller, A.: Efficient simultaneous simulation of Markov chains. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 669–684. Springer-Verlag, Berlin (2008)

Testing the Tests: Using Random Number Generators to Improve Empirical Tests

Paul Leopardi

Abstract The implementer of an empirical test for random number generators is faced with some difficult problems, especially if the test is based on a statistic which is known only approximately: How can the test be tested? How can the approximation be improved? When is it good enough? A number of principles can be applied to these problems. These principles are illustrated using implementations of the overlapping serial "Monkey" tests of Marsaglia and Zaman.

1 Introduction

For many empirical tests of random number generators (RNGs), the distribution of the test statistic is known only approximately or asymptotically. The use of two-level testing with such empirical tests and "known good" random number generators can reveal the goodness of fit between the empirical distribution of the test statistic and the approximate theoretical distribution [10, Section 3.1] [11, Section 3]. Two-level testing with a battery of tests can reveal which of the tests use approximations which give a better fit for the size of the test used in the battery.

This paper describes the improvement of the implementation of the overlapping serial "Monkey" tests of Marsaglia and Zaman [18, 17] in the TestU01 suite [13, 14].

The remainder of this paper is organized as follows. Section 2 describes some of the principles of two-level testing with a battery of tests. Section 3 describes how the PseudoDIEHARD battery of TestU01 was tested. Section 4 describes Marsaglia and Zaman's Monkey tests, shows how various differences with the TestU01 implementation were detected, gives some new results for the theoretical moments for these tests, and describes the improvements made to the tests in TestU01. Section 5 summarizes the results of the tests using the improved version of the Pseudo-

Mathematical Sciences Institute, Australian National University, Building 27, Canberra ACT 0200, Australia
url: http://www.maths.anu.edu.au/~leopardi

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_32,

DIEHARD battery of TestU01. Section 6 gives a brief summary of the computation of the variances which are used in the Monkey tests.

2 Two-Level Testing with a Battery of Tests

The general idea of two-level testing with a battery of tests is that a battery may yield b p-values; when the battery is repeated r times on disjoint subsequences generated by an RNG, this yields a set of br p-values. This set is then tested using one or more statistical tests for uniformity.

Two-level testing with a battery of tests is not a good idea in general if the intention is to test RNGs, since the individual tests must be short so that the battery can be repeated enough times to give a meaningful result within a reasonable runtime. A short test is less likely to give significant results on an individual RNG than a longer test of the same type. Even worse, since the statistics on which a test is based may be only approximate, performing a two-level test can lead to false rejection of an RNG [10, Section 3.1] [11, Section 3]. For example, a series of papers by Kao and Tang apparently wrongly rejects generators on the basis of failing two-level tests [7, 25, 26].

The key to understanding repeated testing with a battery of empirical tests is to realize that the outcome involves two independent null hypotheses.

$\mathcal{H}_0$: The RNG under test generates a $U(0, 1)$ sequence.
$\mathcal{H}_1$: Each test of the battery, when applied to a $U(0, 1)$ sequence, yields a p-value from $U(0, 1)$.

These can be combined into the hypothesis:

$\mathcal{H}_2$: The battery, when repeatedly applied to the RNG under test, yields a sequence of independent p-values from $U(0, 1)$.

For a battery of tests on a single RNG, if $\mathcal{H}_2$ fails it may be hard to distinguish failures of $\mathcal{H}_0$ from failures of $\mathcal{H}_1$. A possible solution is to look for consistent failures of tests across multiple different "known good" RNGs. This approach may not always work, since no RNG actually satisfies $\mathcal{H}_0$ [10, Section 3.4], but there are enough RNGs which work well enough to make this approach feasible.

Once a failure of $\mathcal{H}_1$ for a test battery is detected, the next step is to repeat testing but use each type of test in isolation, or equivalently, to extract from each sequence of p-values produced by a run of the battery those p-values produced by each type of test.

Some tests of a battery may produce multiple correlated p-values. If this causes $\mathcal{H}_1$ to detectably fail for the battery as a whole, this failure will also be detected for the individual test.

For tests using a statistic with a discrete distribution, $\mathcal{H}_1$ is never strictly true, but this type of failure may only be revealed when the number of repetitions of the test is large relative to the number of different p-values which the test can yield.

Models of an empirical test are given by L'Ecuyer and Hellekalek [12, Section 3.1] and by L'Ecuyer and Simard [14, Section 3]. Essentially, a test is a two-step process.

1. The test generates a value y taken by a test statistic Y, where Y is a real-valued function of a number of values generated by the RNG.
2. The test computes a p-value $p := 1 - f(y)$ by using an approximation f to the theoretical distribution function F of the test statistic Y, where F is defined by

$$F(y) := 1 - P[Y \geq y].$$

Possible causes of the failure of $\mathcal{H}_1$ for a particular test may therefore include:

1. The implementation of the test does not match its description in the literature, and actually generates a test statistic $Y' \neq Y$;
2. The function f is not a good approximation to F for the particular parameters used by the test;
3. The implementation of the test actually computes a function $f' \neq f$, giving a different approximation to F from the one described in the literature.

Deeper investigation of the cause of the failure of $\mathcal{H}_1$ for a particular test may therefore require examination of the source code of the test, as well as its description.

3 Initial Testing of PseudoDIEHARD in TestU01

TestU01 is a collection of "Utilities for empirical statistical testing of uniform random number generators" [14]. It contains a library of empirical tests, arranged into batteries. Typical use of TestU01 is to test an RNG using the Small Crush, Crush and Big Crush batteries in succession. TestU01 also includes the PseudoDIEHARD battery, which is based on Marsaglia's DIEHARD battery [16].

To investigate the PseudoDIEHARD battery of TestU01, two high quality generators were used: Mersenne Twister mt19937 [20, 19], and Brent Xorgens xor4096 [1, 2]. For the remainder of this paper, the Mersenne Twister mt19937 generator is referred to as MT and the Brent Xorgens xor4096 generator is referred to as BX. Both generators pass all tests of the Small Crush battery. BX passes all tests of the Crush and Big Crush batteries while MT fails Crush and Big Crush in tests of linear complexity [14].

The first testing method used is to perform 64 repetitions of the PseudoDIEHARD battery using an RNG with each of 4 different seeds, yielding a sequence of $N = 32\,256$ p-values, and submit this sequence to a second-level test. The two-sided one sample Kolmogorov test is used to compare the empirical cumulative distribution function (CDF) of the sequence of p-values to the CDF for $U(0, 1)$. Hypothesis $\mathcal{H}_2$ is rejected if the second-level test yields a p-value less than 0.001.

The results using TestU01 version 0.6.1 are as follows.

	D	p
BX:	0.0118	0.0002466
MT:	0.0136	1.408×10^{-5}

We see that hypothesis $\mathcal{H}_2$ is rejected for both generators, throwing suspicion on hypothesis $\mathcal{H}_1$.

One way to gain more confidence in these results is to increase the number of repetitions. Under hypothesis $\mathcal{H}_0$ this should yield an empirical distribution of p-values which more closely approximates the distribution which is produced by the PseudoDIEHARD test battery. The results using TestU01 0.6.1 PseudoDIEHARD repeated 1024 times, giving 129 024 p-values each, are as follows.

	D	p
BX:	0.0113	1.221×10^{-14}
MT:	0.011	5.951×10^{-14}

Figure 1 plots the difference between the p-values from 1024 repetitions of TestU01 0.6.1 PseudoDIEHARD using BX, sorted in increasing order, and the corresponding values of the $U(0, 1)$ distribution, in this case the numbers $(k - 1/2)/N$, for k from 1 to $N = 129\,024$. A systematic pattern resembling a sideways S is easily visible.

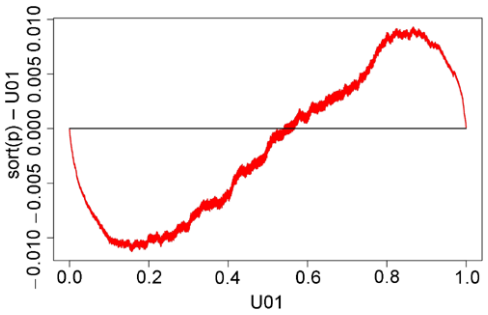


Fig. 1 1024 × PseudoDIEHARD 0.6.1 (BX).

When the p-values for each type of test are extracted from a run of 1024 repetitions of TestU01 0.6.1 PseudoDIEHARD, the tests which produce failures of hypothesis $\mathcal{H}_1$ are seen to be the Run test, the OQSO test and the DNA test.

The Run test implemented in TestU01 0.6.1 is essentially the test described in the 1981 version of Knuth [8], with a slight difference. The RNG is called $n + 1$ times rather than n times. The improvement to the Run test, which was incorporated into TestU01 version 1.2.1, essentially consists of implementing the description given in the 1998 version of Knuth [9, pp. 66–69]. Details of this improvement are omitted.

4 Overlapping Serial Tests

Of the 126 p-values generated by PseudoDIEHARD, 82 come from three overlapping serial (Monkey) tests [18]: 23 OPSO tests, 28 OQSO tests, and 31 DNA tests.

These Monkey tests use an alphabet of size α, form a string of length $n = 2^{21}$ by taking $n \times \log_2 \alpha$ bits from an RNG, and examine the $n - t + 1$ overlapping words of length t. According to [18], the number of missing words should be approximately normal with expected value μ and variance σ^2 as given by Table 1.

Table 1 Marsaglia and Zaman's 1993 Monkey test means and standard deviations.

	α	t	μ	σ
OPSO:	2^{10}	2	141 909.4653	290.27
OQSO:	2^5	4	141 909.4737	290
DNA:	4	10	141 910.5378	290

In TestU01 the Monkey tests are treated as instances of the overlapping Collision sparse serial tests [13, p. 106, pp. 121–122] [15].

The two-level tests for the Monkey tests use the same method as the Run test. The PseudoDIEHARD battery is repeated 1024 times. From each subsequence of 126 p-values generated by each repetition, the corresponding p-values are extracted: 28 for the OQSO tests and 31 for the DNA tests.

The Kolmogorov test results using the corresponding sequences of p-values extracted from 1024 repetitions of the TestU01 0.6.1 PseudoDIEHARD battery are as follows.

		D	p
OQSO	BX:	0.0085	0.033 11
	MT:	0.0061	0.24
DNA	BX:	0.0389	$< 2.2 \times 10^{-16}$
	MT:	0.0374	$< 2.2 \times 10^{-16}$

Hypothesis $\mathcal{H}_2$ is rejected for the DNA test for both generators. Figure 2 shows the difference between the sorted p-values and the $U(0, 1)$ distribution for BX. The distinct sideways S shaped curve of the graph casts suspicion on the variance used to calculate the p-values for the DNA test.

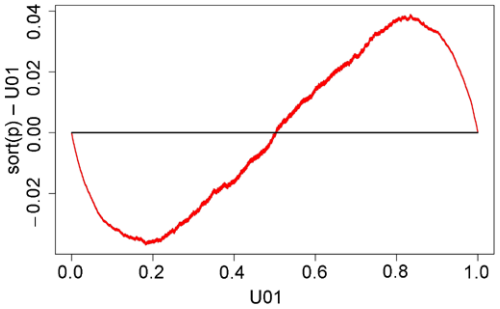


Fig. 2 DNA tests from 1024 × PseudoDIEHARD 0.6.1 (BX).

For all three Monkey tests, TestU01 0.6.1 calculates the values $k := \alpha^t$ and $\lambda := n/\alpha^t = 2$ and then obtains

$$\mu = k\mathrm{e}^{-\lambda} = 2^{20}\mathrm{e}^{-2} \simeq 141\,909.329955,$$
$$\sigma = k\mathrm{e}^{-\lambda}(1 - 3\mathrm{e}^{-\lambda}) = 2^{10}\sqrt{\mathrm{e}^{-2} - 3\mathrm{e}^{-4}} \simeq 290.3331.$$

These values of μ and σ agree with those of Table 1 to the nearest integer, except for the expected value for the DNA test.

As part of the project which produced the DIEHARD battery of tests [16], Marsaglia in 1995 produced a revised version of his joint paper with Zaman [18]. The newer paper [17] revises the values of σ for the OQSO and DNA tests to 295 and 339 respectively. These revised values were obtained by simulation.

The author submitted a patch to TestU01 to use the revised values of σ for the OQSO and DNA tests. The patch also sets the number of words used to calculate λ to $n - t + 1$ so that $\lambda = (n - t + 1)/k$, matching the description of the overlapping collision test in the User's Guide [13, Version 0.6.1 p. 121, Version 1.2.1 p. 131]. This patch is used in TestU01 1.2.1.

The Kolmogorov test results, using the corresponding sequences of p-values extracted from 1024 repetitions of the TestU01 1.2.1 PseudoDIEHARD battery, are as follows.

		D	p
OQSO	BX:	0.0109	0.002 0921
	MT:	0.0093	0.0141
DNA	BX:	0.0127	6.802×10^{-5}
	MT:	0.0109	0.001 052

Hypothesis $\mathcal{H}_2$ is still rejected for the DNA test for BX, and the p-values for DNA for MT and OQSO for BX are suspiciously low.

Figure 3 shows a U shaped curve which represents the difference between the sorted p-values for the patched DNA test using BX, and the $U(0, 1)$ distribution. The sideways S shaped curve from Figure 2 is also shown for comparison. It appears

that at least for the DNA test, TestU01 1.2.1 no longer uses an inaccurate variance but instead uses an inaccurate mean.

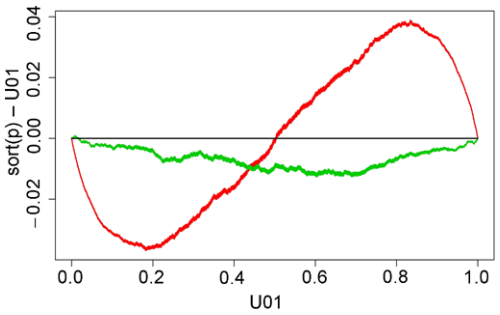


Fig. 3 DNA tests from 1024 × PseudoDIEHARD 1.2.1 (BX).

A deeper investigation into the source code of TestU01 1.2.1 reveals two key differences between this implementation and Marsaglia and Zaman's description of the Monkey tests [18, 17].

1. Different test.
   The OPSO, OQSO and DNA tests as described by Marsaglia and Zaman [18, 17] each use a string of length n. Their implementations in TestU01 use words on a *cycle* of length n rather than a string, yielding n not $n-t+1$ words of length t [13, p. 106, pp. 121–122] [15]. This produces a different test with a different expected outcome.
   The expected number of missing words in the cyclic case is different from the non-cyclic case. Using the methods of Guibas and Odlyzko [6], and Rivals and Rahmann [23], and Maple code provided by Edlin and Zeilberger [3], the corresponding μ for the cyclic versions of the OPSO, OQSO and DNA tests was calculated to be:

OPSO:	141 909.194 619 723 81
OQSO:	141 909.194 525 907 72
DNA:	141 909.184 583 083 19

   The corresponding σ is not yet known.
2. Different number of words.
   In the Marsaglia and Zaman [18, 17] (string) versions of the OPSO, OQSO and DNA tests, the number of words in the sequence is $n-t+1$. In the TestU01 1.2.1 (cyclic) implementation of these tests, the number of words in the sequence is n, yet the number of words used to calculate the expected value μ and the variance σ^2 is $n-t+1$. As a result, TestU01 1.2.1 sets μ to the following values.

OPSO:	141 910.329 955
OQSO:	141 912.329 955
DNA:	141 918.329 955

The effect of the combination of differences 1 and 2 results in an inaccurate expected value being used to calculate the p-value.

Precise values of the expected value and variance for the number of missing words in the Monkey tests were calculated using the methods of Guibas and Odlyzko [6], and Rivals and Rahmann [23, 22], with the help of Maple code provided by Noonan and Zeilberger [21], and Edlin and Zeilberger [3]. Calculation of the variance for the OPSO test needs the calculation of 6 generating functions, the OQSO test needs 55, and the DNA test needs 4592. The methods used for the calculation are described in Section 6 below. The resulting values of μ and σ are listed in Table 2 to 20 decimal places.

Table 2 Improved Monkey test means and standard deviations.

	μ	σ
OPSO:	141 909.329 955 006 918 91	290.462 263 403 751 797 69
OQSO:	141 909.600 532 131 639 00	294.655 872 365 832 448 93
DNA:	141 910.402 604 762 935 66	337.290 150 690 427 643 65

An improved patch for the TestU01 implementations of the Monkey tests eliminates differences 1 and 2 above, and uses the improved means and standard deviations listed in Table 2. The Kolmogorov test results using the corresponding sequences of p-values extracted from 1024 repetitions of the improved TestU01 PseudoDIEHARD battery are as follows.

		D	p
OQSO	BX:	0.008	0.05186
	MT:	0.006	0.2456
DNA	BX:	0.0038	0.7527
	MT:	0.0034	0.8589

5 Final TestU01 Results

The summarized Kolmogorov test results for 1024 repetitions of the PseudoDIEHARD battery, for different versions of TestU01 for the generators MT and BX, are as follows.

Version		D	p
0.6.1	BX:	0.0113	1.221×10^{-14}
	MT:	0.011	5.951×10^{-14}
1.2.1	BX:	0.0063	7.01×10^{-5}
	MT:	0.0056	0.000 6376
Improved	BX:	0.0032	0.1352
	MT:	0.0025	0.3982

The improved version no longer results in rejection of hypothesis $\mathcal{H}_2$.

6 Computation of the Variances for the Monkey Tests

The following analysis is based on that of Rahmann and Rivals [22]. The problem is to find the mean and variance of the distribution of the number of missing words in a random string. A random string S of length n is formed from an alphabet of size α, with each character equally likely. The string S contains $n - t + 1$ overlapping words of length t. There are therefore α^n possible strings S_i, and α^t possible words W_j.

For the remainder of this section, consider α and t to be fixed. Define the indicator variable $v_{i,j}$ to be 1 if word W_j is missing from string S_i, and 0 otherwise. The number of words missing from string S_i is thus

$$X_i := \sum_j v_{i,j}.$$

The probability that both words W_j and W_k are missing from a random string S of length n is

$$a_{j,k}^{(n)} := \alpha^{-n} \sum_i v_{i,j} v_{i,k}.$$

Define the generating functions $A_{j,k}(z) := \sum_n a_{j,k}^{(n)} z^n$ and $A_j := A_{j,j}$. Define the random variable $X^{(n)}$ to be the number of words missing from a random string S of length n. The expected value of $X := X^{(n)}$ is then

$$\mathbb{E}[X] = \alpha^{-n} \sum_i X_i = \alpha^{-n} \sum_i \sum_j v_{i,j} = \sum_j a_{j,j}^{(n)}.$$

The variance is $\mathrm{Var}[X] = \mathbb{E}[X^2 - X] + \mathbb{E}[X] - (\mathbb{E}[X])^2$, with

$$\mathbb{E}[X^2 - X] = \alpha^{-n} \sum_i \sum_{j \neq k} v_{i,j} v_{i,k} = \sum_{j \neq k} a_{j,k}^{(n)}.$$

Given words B and C of length t, with $B = B_0 \dots B_{t-1}$ etc. define the (word overlap) correlation vector BC by $BC_s = 1$ if $B_{r+s} = C_r$ for $r \in \{0, \dots, t-s-1\}$, and $BC_s = 0$ otherwise. Figure 4 shows an example of a correlation vector. The correlation vectors BB, CC are called autocorrelations [5, 23].

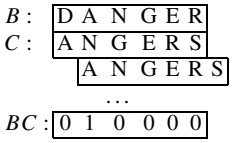


Fig. 4 The correlation vector for the words B=DANGER, C=ANGERS.

For the correlation vector v, define the correlation polynomial

$$P_v(z) := v_0 + v_1 z + \ldots + v_{t-1} z^{t-1}.$$

For $P_j := P_{W_j W_j}$, the generating function A_j is given by Guibas and Odlyzko [6, Theorem 1.1], and Rahmann and Rivals [22, Lemma 2.1] as

$$A_j(z) = \frac{P_j(z/\alpha)}{(z/\alpha)^t + (1-z)P_j(z/\alpha)}.$$

For $P_{g,h} := P_{W_g, W_h}$, the correlation matrix is $M_{j,k}(z) := \begin{bmatrix} P_{j,j}(z) & P_{j,k}(z) \\ P_{k,j}(z) & P_{k,k}(z) \end{bmatrix}$.

Given $M := \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$, define $M^V := \begin{bmatrix} m_{22} & m_{21} \\ m_{12} & m_{11} \end{bmatrix}$ and

$R(M) := m_{11} + m_{22} - m_{12} - m_{21}$.

Define the equivalence class $[M] := \{M,\ M^T,\ M^V,\ M^{TV}\}$, so that

$$[M_{j,k}(z)] = \{M_{j,k}(z),\ M^T_{j,k}(z),\ M_{k,j}(z),\ M^T_{k,j}(z)\}.$$

Note that $M' \in [M]$ implies $\det M' = \det M$ and $R(M') = R(M)$.

The generating function $A_{j,k}$ for the pair W_j, W_k is given by Rahmann and Rivals [22, Lemma 3.2] as

$$A_{j,k}(z) = \frac{Q_{j,k}(z/\alpha)}{(1-z)Q_{j,k}(z/\alpha) + (z/\alpha)^t R_{j,k}(z/\alpha)},$$

where $Q_{j,k}(z) := \det M_{j,k}(z)$, and $R_{j,k}(z) := R(M_{j,k}(z))$. See also [6, 21, 24].

Standard methods are used to obtain $a^{(n)}_{j,k} = [z^n] A_{j,k}(z)$ from each $A_{j,k}(z)$. See e.g. Graham, Knuth and Patashnik [4, Section 7.3].

We could simply sum $a_{j,k}^{(n)}$ for all $\alpha^{2t} - \alpha^t$ word pairs $W_j \neq W_k$, but for Marsaglia's tests, $\alpha^{2t} = 2^{40}$. So instead we enumerate correlation classes and count the word pairs for each class.

Each word pair W_j,W_k containing β distinct letters yields a partition of the set $\{0, \ldots, 2t-1\}$ into β nonempty subsets, which is equivalent to a restricted growth string of length $2t$ having exactly β distinct letters. The string S of length $2t$ is a restricted growth string if $S_k \leqslant S_j + 1$ for each j from 0 to $k-1$, for k from 1 to $2t-1$.

Each permutation of the alphabet preserves the correlation matrix. The set of word pairs having β distinct letters splits under the symmetry group $\mathbb{S}_\alpha$ into orbits of size $\alpha!/(\alpha-\beta)!$.

Define $N[M](\alpha) = \sharp\{(j,k) \mid M_{j,k} = [M]\}$, the number of word pairs associated to the correlation class $[M]$. For $\alpha \leqslant 2t$, the following algorithm is used to determine all correlation classes $[M]$, and find $N[M](\alpha)$ for each one.

For each β from 1 to α, for each restricted growth string of length $2t$ having exactly β distinct letters:

1. Find the correlation class for the corresponding word pair.
2. Add $\frac{\alpha!}{(\alpha-\beta)!}$ to the count for the correlation class.

For each correlation class $[M]$, $N[M](\alpha)$ is a polynomial in α of maximum degree $2t$. For $\alpha > 2t$, to find $N[M](\alpha)$, first find $N[M](\gamma)$ for γ from 1 to $2t$, and then interpolate the resulting polynomial.

Acknowledgements This paper is the result of correspondence with Richard Simard of the University of Montreal, and with Eric Rivals at LIRMM, Montpellier. Art Owen of Stanford encouraged this paper and the presentation on which it is based. Jörg Arndt assisted in programming and proofreading. The combinatorial, statistical and programming techniques used were refined through various discussions with Sacha van Albada, Peter Drysdale, and others at Complex Systems, School of Physics, University of Sydney; and Jörg Arndt, Sylvain Forêt, John Maindonald, Judy-anne Osborn, and others at the Mathematical Sciences Institute, Australian National University.

References

1. Brent, R.P.: Some uniform and normal random number generators (2006–2008). URL http://wwwmaths.anu.edu.au/~brent/random.html
2. Brent, R.P.: Some long-period random number generators using shifts and xors. ANZIAM Journal **48 (CTAC2006)**(1), C188–C202 (2007). URL http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/40
3. Edlin, A.E., Zeilberger, D.: The Goulden-Jackson cluster method for cyclic words. Advances in Applied Mathematics **25**, 228–232 (2000)

4. Graham, R.L., Knuth, D.E., Patashnik, O.: Concrete Mathematics, second edn. Addison-Wesley, New Jersey (1994)
5. Guibas, L.J., Odlyzko, A.M.: Periods in strings. Journal of Combinatorial Theory Series A **30**, 19–43 (1981)
6. Guibas, L.J., Odlyzko, A.M.: String overlaps, pattern matching, and nontransitive games. Journal of Combinatorial Theory Series A **30**, 183–208 (1981)
7. Kao, C., Tang, H.C.: Several extensively tested multiple recursive random number generators. Comput. Math. Appl. **36**(6), 129–136 (1998)
8. Knuth, D.E.: Seminumerical Algorithms, *The Art of Computer Programming*, vol. 2, second edn. Addison-Wesley, Reading, Massachusetts (1981)
9. Knuth, D.E.: Seminumerical Algorithms, *The Art of Computer Programming*, vol. 2, third edn. Addison-Wesley, Reading, Massachusetts (1998)
10. L'Ecuyer, P.: Testing random number generators. In: WSC '92: Proceedings of the 24th conference on Winter simulation, pp. 305–313. ACM, New York, NY, USA (1992). DOI 10.1145/167293.167354
11. L'Ecuyer, P.: Random number generators and empirical tests. In: Monte Carlo and quasi-Monte Carlo methods 1996 (Salzburg), *Lecture Notes in Statistics*, vol. 127, pp. 124–138. Springer, New York (1998)
12. L'Ecuyer, P., Hellekalek, P.: Random number generators: selection criteria and testing. In: Random and quasi-random point sets, *Lecture Notes in Statistics*, vol. 138, pp. 223–265. Springer, New York (1998)
13. L'Ecuyer, P., Simard, R.: TestU01: a software library in ANSI C for empirical testing of random number generators: User's guide, detailed version. Département d'Informatique et de Recherche Opérationnelle Université de Montréal (2005). URL http://www.iro.umontreal.ca/~simardr/testu01/tu01.html
14. L'Ecuyer, P., Simard, R.: TestU01: a C library for empirical testing of random number generators. ACM Trans. Math. Software **33**(4), Art. 22, 40 (2007)
15. L'Ecuyer, P., Simard, R., Wegenkittl, S.: Sparse serial tests of uniformity for random number generators. SIAM Journal on Scientific Computing **24**(2), 652–668 (2002). DOI 10.1137/S1064827598349033
16. Marsaglia, G.: The Marsaglia random number CDROM including the Diehard battery of tests of randomness (1995). URL http://www.stat.fsu.edu/pub/diehard/
17. Marsaglia, G.: Monkey tests for random number generators, (revised extract, 1995). Journal of Statistical Software **14**(13), 1–10 (2005). URL http://www.jstatsoft.org/v14/i13/supp/1
18. Marsaglia, G., Zaman, A.: Monkey tests for random number generators. Computers and Mathematics with Applications **26**(9), 1–10 (1993)
19. Matsumoto, M.: Mersenne Twister with improved initialization (2002). URL http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/MT2002/emt19937ar.html
20. Matsumoto, M., Nishimura, T.: Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation **8**(1), 3–30 (1998). DOI 10.1145/272991.272995
21. Noonan, J., Zeilberger, D.: The Goulden-Jackson cluster method: Extensions, applications, and implementations. J. Difference Eq. Appl. **5**, 355–377 (1999)
22. Rahmann, S., Rivals, E.: On the distribution of the number of missing words in random texts. Combinatorics, Probability and Computing **12**, 73–87 (2003)
23. Rivals, E., Rahmann, S.: Combinatorics of periods in strings. Journal of Combinatorial Theory Series A **104**(1), 95–113 (2003)
24. Rukhin, A.L.: Distribution of the number of words with a prescribed frequency and tests of randomness. Advances in Probability **34**(4), 775–797 (2002)
25. Tang, H.C.: A statistical analysis of the screening measure of multiple recursive random number generators of orders one and two. J. Statist. Comput. Simulation **71**(4), 345–356 (2001)
26. Tang, H.C., Kao, C.: Searching for good multiple recursive random number generators via a genetic algorithm. INFORMS J. Comput. **16**(3), 284–290 (2004)

Stochastic Spectral Formulations for Elliptic Problems

Sylvain Maire and Etienne Tanré

Abstract We describe new stochastic spectral formulations with very good properties in terms of conditioning. These formulations are built by combining Monte Carlo approximations of the Feynman-Kac formula and standard deterministic approximations on basis functions. We give error bounds on the solutions obtained using these formulations in the case of linear approximations. Some numerical tests are made on an anisotropic diffusion equation using a tensor product Tchebychef polynomial basis and one random point schemes quantized or not.

1 Introduction

The Feynman-Kac formula is a very powerful tool to achieve stochastic representations of the pointwise solution of numerous partial differential equations like diffusion or transport equations [6, 12]. For instance, let us first consider the Poisson equation in a bounded domain $D \subset \mathbb{R}^d$ with a sufficiently smooth boundary ∂D

$$\begin{cases} \frac{1}{2}\Delta u(x) = -f(x) & x \in D, \\ u(x) = g(x) & x \in \partial D. \end{cases} \tag{1}$$

This equation models the temperature u in a domain D with a source term f in D and a prescribed temperature g on the boundary ∂D. The operator Δ is the Laplacian, that is

Sylvain Maire
ISITV, Université de Toulon et du Var, avenue G. Pompidou, BP 56, 83262 La Valette du Var CEDEX France
e-mail: maire@univ-tln.fr

Etienne Tanré
INRIA, Equipe Projet TOSCA, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis, France
e-mail: Etienne.Tanre@inria.fr

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_33,

$$\Delta u(x) = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2} u(x) \quad \forall x (= (x_1, \cdots, x_d)) \in D,$$

which models an isotropic diffusion.

The Laplacian can be replaced by a more general second order linear operator L

$$Lu(x) = \sum_{i=1}^{d} b_i(x) \frac{\partial}{\partial x_i} u(x) + \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \xi_{i,j}(x) \frac{\partial^2}{\partial x_i \partial x_j} u(x)$$

to take into account advection and anisotropic diffusions. The functions b_i are assumed to be smooth and the matrix $\xi(x)$ with elements $\xi_{i,j}(x)$ is assumed to satisfy $\xi(x) = \sigma(x)\sigma(x)^t$ for a $d \times \tilde{d}$ matrix σ. We also assume standard assumptions on the uniform ellipticity of the operator.

In the following, we consider the general Dirichlet boundary value problem

$$\begin{cases} Lu(x) = -f(x) & x \in D \\ u(x) = g(x) & x \in \partial D \end{cases} \tag{2}$$

whose solution u admits the classical stochastic representation: $\forall x \in D$

$$u(x) = \mathbb{E}_x \left[g(X_{\tau_D}) + \int_0^{\tau_D} f(X_s) ds \right], \tag{3}$$

where $\{X_t, t \geq 0\}$ is the stochastic process (starting at $X_0 = x$) solution of the stochastic differential equation relative to the operator L

$$X_t = x + \int_0^t b(X_s) ds + \int_0^t \sigma(X_s) dW_s$$

and where τ_D is the exit time of this process from the domain D. The notation $\mathbb{E}_x$ is used for the expectation given that $X_0 = x$. In the particular case of the Poisson equation, the process $\{X_t, t \geq 0\}$ is just the d dimensional Brownian motion.

Gobet and Maire have introduced in [8] sequential Monte Carlo algorithms to compute global approximations of the solutions by combining this formula and deterministic linear approximations. This has led to a geometric reduction up to a threshold of both the bias and the variance involved in the Monte Carlo computation of the Feynman-Kac representations [9]. This threshold appears because we use only a finite number of terms in our deterministic approximation and it is linked to the error between this approximation and the exact solution. In order to improve the speed of convergence of the sequential algorithms, we have described new schemes for the evaluation of the source terms based on the one-random-point method and quantization techniques [15]. In the case of the Poisson equation, we have made a new interpretation of the algorithm which has led to a direct spectral formulation with almost perfect properties in terms of conditioning.

This article is organized as follows. We first recall in Section 2 the main tools we have introduced in [15] and then, we show how to extend them in the case of a general elliptic operator. In Section 3, we give some confidence intervals on the

solution if the approximations of this solution are linear. Finally, we give in Section 4 some numerical results on an anisotropic diffusion over a square domain using an approximation based on tensor product Tchebychef polynomial interpolation and either Monte Carlo simulations or quantization tools.

2 The Stochastic Spectral Formulation

2.1 One Random Step Schemes

The goal of this section is to remind of the tools introduced in [15] to compute Feynman-Kac representations at a numerical cost which is similar for the boundary and source terms. We also assume that f and g are bounded. Representation (3) is computed using a Monte Carlo method which requires generally the simulation of the process $\{X_t, t \geq 0\}$ using an approximation scheme $\{\hat{X}^{\delta}_{k\delta}, k \in \mathbb{N}\}$ like the Euler scheme [2] with a time step δ. If we estimate that the process leaves the domain D at time $\hat{\tau}_D$ with $n\delta < \hat{\tau}_D < (n+1)\delta$ (for instance $\hat{X}^{\delta}_{(n+1)\delta} \notin D$ or $\hat{X}^{\delta}_{n\delta}$ and $\hat{X}^{\delta}_{(n+1)\delta}$ are very close to the boundary), X_{τ_D} is approximated by $\hat{X}^{\delta}_{\hat{\tau}_D}$ using these last two points. The simplest way to obtain $\hat{X}^{\delta}_{\hat{\tau}_D}$ is the orthogonal projection of $\hat{X}^{\delta}_{n\delta}$ on ∂D. We approximate $g(X_{\tau_D})$ by $g(\hat{X}^{\delta}_{\hat{\tau}_D})$. The standard approximation of $\int_0^{\tau_D} f(X^x_s)ds$ by the rectangle method is $\delta \sum_{i=1}^{n} f(\hat{X}^{\delta}_{i\delta})$, whose bias is of order δ. For each simulated trajectory, we can see that many evaluations of the function f are required but we need only one evaluation of the function g. Thanks to the representation

$$\mathbb{E}_x\left[\int_0^{\tau_D} f(X_s)ds\right] = \mathbb{E}_x\left[\int_0^1 \tau_D f(X_{y\tau_D})dy\right] = \mathbb{E}_x\left[\tau_D f(X_{U\tau_D})\right]$$

introduced in [15], we can rewrite the Feynman-Kac formula as

$$u(x) = \mathbb{E}_x\left[g(X_{\tau_D}) + \tau_D f(X_{U\tau_D})\right],$$

where U is a random variable with uniform law on $[0, 1]$, independent of the process $\{X_t, t \geq 0\}$. We replace the standard approximation by $n\delta f(\hat{X}^{\delta}_{J\delta})$ where J is a discrete uniform random variable on the set $\{1, \cdots, n\}$. This new estimator uses now only one evaluation of f and we have showed in [15] that in most situations the increase of its variance is compensated by the decay of its computational cost. This is especially true when δ is small, x is away from the boundary and the evaluation of f is costly.

Remark 1. 1. In the algorithms developed in [9, 15], the evaluation of f is naturally very costly because f is a sum of a large number of terms of a finite expansion on basis functions.
2. We generally need to simulate the whole trajectory (for instance with an Euler scheme) to obtain a realisation of $(\tau_D, X_{U\tau_D})$.

3. This approach has the advantage to be adapted to quantization: it is easier to quantify a random variable in $\mathbb{R}_+ \times D$ than to quantify the law of the whole trajectory.

When the operator L is $\frac{1}{2}\Delta$, the stochastic process to simulate is the Brownian motion $\{B_t, t \geq 0\}$ for which different methods of simulation are available. The Euler scheme with discretization parameter δ writes

$$\begin{cases} B_0 = x \\ B_{(n+1)\delta} = B_{n\delta} + \sqrt{\delta} Y_n \end{cases}$$

where the $\{Y_n, n \in \mathbb{N}\}$ are independent standard Gaussian random variables. The crude version makes the simulation stops once $B_{(n+1)\delta} \notin D$. This leads to approximations that are of weak order $\sqrt{\delta}$. It is possible to take into account the possibility for the Brownian motion to leave the domain between step n and $n+1$ and be back into it at time $(n+1)\delta$, to obtain a scheme of weak order δ using the half-space approximation [7]. Some faster schemes can be used like the walk on rectangles [5] or the walk on spheres method [20].

We have developed in [15] a one-random-point version of the modified walk on spheres method introduced in [11]: exactly as for the Euler scheme, we evaluate the source term at only one random point (in one sphere) of the trajectory. The sphere is picked at random, proportionally to the square of its radius then the point is picked in this sphere according to a conditional Green function.

This method has been tested in [15] and appeared as the most efficient in all the examples we have tried. The one-random-point method has been also used successfully in [19] for the exact simulation of prices and hedges in the financial mathematics context.

2.2 Quantization

In some situations like spectral methods [4] or in the sequential Monte Carlo methods developed in earlier works [9, 10], the points $(x_1, \cdots, x_N)$ where the solution is computed are fixed. We shall describe what can be done in that case for a diffusion equation in a general bounded domain D with a sufficiently smooth boundary. For a fixed point $x_i \in D$, we can already use a Monte Carlo simulation to build a quadrature formula

$$u(x_i) \simeq \sum_{k=1}^{M} \frac{1}{M} g(Z_{i,k}^b) + \sum_{k=1}^{M} \frac{\tau_D^{(k)}}{M} f(Z_{i,k}^s) \tag{4}$$

at M random points $(Z_{i,1}^b, \cdots, Z_{i,M}^b)$ of the boundary and $(Z_{i,1}^s, \cdots, Z_{i,M}^s)$ of the interior of the domain. In order to increase the rate of convergence of this kind of formula, we can furthermore optimize the locations of the points of evaluations of both f and g by using quantization techniques [17].

We explain in detail how these techniques work when one wants to quantify q points of the boundary term

$$\mathbb{E}_{x_i}\left(g(X_{\tau_D})\right) = \int_{\partial D} g(z) w^b_{x_i}(y)dy,$$

where $w^b_{x_i}(y)$ is the density of the exit position of the stochastic process $\{X_t, t \geq 0\}$ starting at the point x_i. This density is usually unknown but we need only to sample from it or from an approximation of it.

Optimal quantization in the quadratic case consists in finding the q points $(z^b_{i,1}, \cdots, z^b_{i,q})$ on ∂D minimizing the functional

$$\int_{\partial D} \inf_{1 \leq k \leq q} d^2(z, z^b_{i,k}) w^b_{x_i}(z)dz,$$

where $d(z, z')$ is the geodesic distance on ∂D, that is the minimal length of a path on ∂D between z and z'. This problem is solved numerically using the competitive learning vector quantization algorithm (see [3]) which can be described as follows. We first simulate q independent points according to the density $w^b_{x_i}(y)$. We simulate one more point $Y^b_{i,1}$ from the same density and then move a little the closest (among the q ones) point $Z^b_{i,min}$ along the minimal path between $Z^b_{i,min}$ and $Y^b_{i,1}$. The new position $\hat{Z}^b_{i,min}$ satisfies

$$d(Y^b_{i,1}, \hat{Z}^b_{i,min}) = (1 - \varepsilon_1) d(Y^b_{i,1}, Z^b_{i,min})$$

where $\varepsilon_1 > 0$ is a small parameter. The point $Y^b_{i,1}$ is then removed, another point $Y^b_{i,2}$ is simulated, ε_1 is replaced by ε_2 and so on. The sequence $(\varepsilon_n)_{n \geq 0}$ is decreasing. The numerical aspects of the algorithm are discussed in [18] in the case of multidimensional Gaussian densities.

For each quantization point $z^b_{i,j}$ a tessel

$$C_{i,j} = \left\{ u \in \partial D \,|\, d\left(z^b_{i,j}, u\right) < \inf_{k \neq j} d\left(z^b_{i,k}, u\right) \right\}$$

is associated. Then the approximation of $\int_{\partial D} g(y) w^b_{x_i}(y)dy$ is given by the quadrature formula

$$\sum_{j=1}^{q} a_{i,j} g(z^b_{i,j})$$

with $a_{i,j} = \left(\int_{C_{i,j}} w^b_{x_i}(y)dy\right)$. The weights $a_{i,j}$ are computed using Monte Carlo simulations after convergence of the previous algorithm. These quadrature formulae are more accurate than Monte Carlo integration in rather low dimensions [17].

For the source term, we have to quantify p points of the joint law of $(\tau_D, X_{U\tau_D})$. We now use the Euclidean distance on $\mathbb{R}_+ \times D$ to perform the same algorithm which leads after convergence to the quantization points $((t_{i,1}, z^s_{i,1}), \cdots, (t_{i,p}, z^s_{i,p}))$ and the

corresponding weights $(\tilde{b}_{i,1}, \cdots, \tilde{b}_{i,p})$. The approximation of the source term can be written

$$\sum_{j=1}^{p} \tilde{b}_{i,j} t_{i,j} f\left(z_{i,j}^{s}\right).$$

Letting $b_{i,j} = \tilde{b}_{i,j} t_{i,j}$, the final approximation of the solution is

$$u(x_i) \simeq \sum_{j=1}^{q} a_{i,j} g\left(z_{i,j}^{b}\right) + \sum_{j=1}^{p} b_{i,j} f\left(z_{i,j}^{s}\right).$$

Remark 2. 1. For the sake of simplicity, we have not explained how the points $Z_{i,k}^{b}$, $Z_{i,k}^{s}$ and the times $t_{i,k}$ involved in (4) are generated. In general, we shall use the one-random-point version of the Euler scheme $\hat{X}^{\delta}$ introducted in Sec. 2.1 to obtain these quantities.

2. We point out that the quantization described here minimizes the mean square dispersion (see [16]) of the points. The definition of the quantization does not depend on the functions f of g.

2.3 Formulation and Asymptotic Properties

We want to compute a global approximation of the solution u and we assume that it can be written in a linear form

$$P_N u(x) = \sum_{i=1}^{N} u(x_i) \Psi_i(x) \tag{5}$$

for some functions $(\Psi_1, \cdots, \Psi_N)$ that are at least twice continuously differentiable and some points $(x_1, \cdots, x_N) \in D^N$. The choice of the points x_i and the functions Ψ_i is obviously crucial for the accuracy of the approximation procedure. We do not focus here on this hard task but we use standard approximation methods. In the case of pure interpolation, we have $\Psi_i(x_j) = \delta_{i,j}$, where δ is the Kronecker delta. The Lagrange polynomials associated to $x_1, \cdots, x_N$ are a possible choice for these functions.

We also assume that for every point x_i, we can approximate $u(x_i)$ via for instance a numerical approximation of the Feynman-Kac formula by

$$\widetilde{u}(x_i) = \sum_{j=1}^{q} a_{i,k} g(z_{i,j}^{\delta,b}) + \sum_{j=1}^{p} b_{i,j} f(z_{i,j}^{\delta,s})$$

where this approximation is such that

$$\lim_{p,q \to \infty, \delta \to 0} \widetilde{u}(x_i) = u(x_i).$$

The points $z_{i,j}^{\delta,b}$ are located on the boundary ∂D and the points $z_{i,j}^{\delta,s}$ in D. We now let $r_N(x) = u(x) - P_N u(x)$ and write the partial differential equation solved by $r_N(x)$. We have

$$\begin{cases} Lr_N(x) = Lu(x) - LP_N u(x) = -f(x) - LP_N u(x) & x \in D, \\ r_N(x) = g(x) - P_N u(x) & x \in \partial D. \end{cases}$$

This equation has the same form as (2). Thanks to (3), we have

$$r_N(x) = \mathbb{E}_x \left[(g - P_N u)\left(X_{\tau_D}\right) + \tau_D \left(f + LP_N u\right)\left(X_{U\tau_D}\right) \right]$$

and hence the approximation

$$r_N(x_i) \simeq \sum_{j=1}^{q} a_{i,j}(g(z_{i,j}^{\delta,b}) - P_N u(z_{i,j}^{\delta,b})) + \sum_{j=1}^{p} b_{i,j}(f(z_{i,j}^{\delta,s}) + LP_N u(z_{i,j}^{\delta,s})). \quad (6)$$

Let $\hat{U}_i = \hat{u}(x_i)$, the desired approximation of U. The function $P_N\hat{u}$ is obviously defined by $P_N\hat{u}(x) = \sum_i \hat{u}(x_i)\Psi_i(x)$. We denote by $\hat{r}_N$ the right-hand side of (6), where $P_N u$ is replaced by $P_N\hat{u}$. The N equations $\hat{r}_N(x_i) = \hat{u}(x_i) - P_N\hat{u}(x_i)$ lead to the linear system $C\hat{U} = d$ with

$$\begin{cases} c_{i,i} = \sum_{j=1}^{q} a_{i,j}\Psi_i\left(z_{i,j}^{\delta,b}\right) - \sum_{j=1}^{p} b_{i,j} L\Psi_i\left(z_{i,j}^{\delta,s}\right) + 1 - \Psi_i(x_i) \\ c_{i,k} = \sum_{j=1}^{q} a_{i,j}\Psi_k\left(z_{i,j}^{\delta,b}\right) - \sum_{j=1}^{p} b_{i,j} L\Psi_k\left(z_{i,j}^{\delta,s}\right) - \Psi_k(x_i) \qquad \text{for } i \neq k \\ d_i = \sum_{j=1}^{q} a_{i,j} g\left(z_{i,j}^{\delta,b}\right) + \sum_{j=1}^{p} b_{i,j} f\left(z_{i,j}^{\delta,s}\right). \end{cases}$$

As we have done in [15], we can look at the asymptotic system we obtain when $p, q \to \infty$ and $\delta \to 0$. The term

$$\sum_{j=1}^{q} a_{i,j}\Psi_k(z_{i,j}^{\delta,b}) - \sum_{j=1}^{p} b_{i,j} L\Psi_k(z_{i,j}^{\delta,s})$$

is the approximation at point x_i of the solution of the equation

$$\begin{cases} Lv(x) = L\Psi_k(x) & x \in D, \\ v(x) = \Psi_k(x) & x \in \partial D \end{cases}$$

that is $v(x_i) = \Psi_k(x_i)$. We deduce immediately that the matrix of the asymptotic system converges towards the identity matrix of size N. Our goal is now to give, for fixed values of p, q and δ, a bound on the error we get by using this stochastic spectral formulation.

3 Error Bounds Based on Confidence Intervals

3.1 The Unbiased Monte Carlo Case

We first consider that we use Monte Carlo estimators with exact simulations schemes (which is equivalent to take the time discretization parameter $\delta = 0$) for the two terms of the Feynman-Kac representations. The Monte Carlo estimator of

$$\mathbb{E}_{x_i}\left[(g - P_N u)(X_{\tau_D})\right] + \mathbb{E}_{x_i}\left[\tau_D (f + L P_N u)(X_{U\tau_D})\right] - u(x_i) + P_N u(x_i)$$

using q_i independent sample values for the first term and p_i independent sample values for the second term can be written as

$$\overline{Y_i^{(b)}} + \overline{Y_i^{(s)}} - u(x_i) + P_N u(x_i)$$

where

$$\overline{Y_i^{(b)}} = \frac{1}{q_i}\sum_{k=1}^{q_i}(g(z_{i,k}^b) - P_N u(z_{i,k}^b)),\ \overline{Y_i^{(s)}} = \frac{1}{p_i}\sum_{l=1}^{p_i}\tau_D^{(i,l)}(f(z_{i,l}^s) + L P_N u(z_{i,l}^s)),$$

the random points $z_{i,1}^b, \cdots, z_{i,q_i}^b$ are independent copies of the exit position X_{τ_D} of the process X starting at x_i. The random points and time $(z_{i,1}^s, \tau_D^{(i,1)}), \cdots, (z_{i,p_i}^s, \tau_D^{(i,p_i)})$ are independent copies of $(X_{U\tau_D}, \tau_D)$. We are exactly in the situation of the previous linear system $C\hat{U} = d$ by letting $a_{i,k} = \frac{1}{q_i}$ and $b_{i,l} = \frac{1}{p_i}\tau_D^{(i,l)}$. Indeed, the law of large numbers shows that the random matrix C converges to the identity matrix. To give a confidence interval for the solution based on the central limit theorem, we give confidence intervals for each of the equations of the linear system. To do this for the equation relative to index i, we define the confidence interval

$$A_i = \left[\overline{Y_i^{(b)}} + \overline{Y_i^{(s)}} - r_N(x_i) - C_\alpha\left(\frac{\sigma_i^{(b)}}{\sqrt{q_i}} + \frac{\sigma_i^{(s)}}{\sqrt{p_i}}\right),\right.$$
$$\left.\overline{Y_i^{(b)}} + \overline{Y_i^{(s)}} - r_N(x_i) + C_\alpha\left(\frac{\sigma_i^{(b)}}{\sqrt{q_i}} + \frac{\sigma_i^{(s)}}{\sqrt{p_i}}\right)\right] \quad (7)$$

with $\mathbb{P}(U_i = u(x_i) \in A_i) \geq \left(1 - \alpha - \frac{\beta_i}{\sqrt{q_i}}\right)\left(1 - \alpha - \frac{\gamma_i}{\sqrt{p_i}}\right)$, where

$$\left(\sigma_i^{(b)}\right)^2 = \mathrm{Var}((g - P_N u)(X_{\tau_D}^{x_i})), \qquad \left(\sigma_i^{(s)}\right)^2 = \mathrm{Var}(\tau_D(f + L P_N u)(X_{U\tau_D}^{x_i})),$$

C_α corresponds to the level of confidence α in the Gaussian case and

$$\beta_i = \frac{0.7655\mathbb{E}_{x_i}(|(g - P_N u)(X_{\tau_D})|^3)}{(\sigma_i^{(b)})^{\frac{3}{2}}},\ \gamma_i = \frac{0.7655\mathbb{E}_{x_i}(|\tau_D(f + L P_N u)(X_{U\tau_D})|^3)}{(\sigma_i^{(s)})^{\frac{3}{2}}}.$$

The constants β_i and γ_i are obtained thanks to the Berry-Esseen inequality [21] which holds true even if p_i and q_i are small. This inequality requires the existence of third moments for $(g - P_N u)(X^{x_i}_{\tau_D})$ and $\tau_D(f + LP_N u)(X^{x_i}_{U\tau_D})$ which is verified since $g - P_N u,\ f + LP_N u$ are bounded and $\mathbb{E}(\tau_D^3) < \infty$ (see [6]). Note that for the sake of simplicity, we have included in the sets A_i the contribution of both source and boundary terms. We should mention that there is no guaranty that the constants β_i and γ_i are for instance lower than one. So if p_i and q_i are very small, our bounds may not be meaningful. Hence we obtain with probability

$$\mathbb{P}(A_1 A_2 \cdots A_N) \geq \prod_{i=1}^{N} \left(1 - \alpha - \frac{\beta_i}{\sqrt{q_i}}\right)\left(1 - \alpha - \frac{\gamma_i}{\sqrt{p_i}}\right) = \eta \tag{8}$$

a system of inequalities

$$d - h_d \leq CV \leq d + h_d \tag{9}$$

where C and d are defined as previously, $V \in \mathbb{R}^N$ and where

$$h_{d_i} = C_\alpha \left(\frac{\sigma_i^{(b)}}{\sqrt{q_i}} + \frac{\sigma_i^{(s)}}{\sqrt{p_i}}\right).$$

The solution $U = (u(x_1), \cdots, u(x_N))$ satisfies (9) with probability at least η. If C is non-singular, we have hence

$$\left\|\hat{U} - U\right\| \leq \left\|C^{-1}\right\| \left\|C\hat{U} - CU\right\| \leq \left\|C^{-1}\right\| \|h_d\|$$

with probability at least η for any matrix norm. Furthermore

$$(h_{d_i})^2 \leq 2C_\alpha^2 \left(\frac{\mathbb{E}_{x_i}[(g - P_N u)^2(X_{\tau_D})]}{q_i} + \frac{\mathbb{E}_{x_i}[\tau_D^2 (f + LP_N u)^2(X_{U\tau_D})]}{p_i}\right)$$

which gives

$$(h_{d_i})^2 \leq 2C_\alpha^2 \left(\frac{\sup_{x\in\partial D}(g - P_N u)^2(x)}{q_i} + \frac{\mathbb{E}_{x_i}[\tau_D^2] \sup_{x\in D}(f + LP_N u)^2(x)]}{p_i}\right).$$

From a practical point of view, it can be efficient to choose the values p_i and q_i to be different and adapted to the variances of the source and boundary terms. For the theoretical study, we assume now that $p_i = q_i = M$, and we have furthermore

$$\|h_d\|_2^2 \leq \frac{2NC_\alpha^2}{M}\left[\sup_{x\in\partial D}(g - P_N u)^2(x) + \max_{i=1,\cdots,N} \mathbb{E}_{x_i}[\tau_D^2] \sup_{x\in D}(f + LP_N u)^2(x)\right]. \tag{10}$$

We have finally with probability η (defined in (8)),

$$\left\|\hat{U} - U\right\|_2 \leq \left\|C^{-1}\right\|_2 \|h_d\|_2 .$$

We now study if C is regular and how to find a bound on $\left\|C^{-1}\right\|_2$. We write $C = Id - F$ with

$$F_{i,j} = -\sum_{k=1}^{M} a_{i,k}\Psi_j(z_{i,k}^{\delta,b}) - \sum_{l=1}^{M} b_{i,l}L\Psi_j(z_{i,l}^{\delta,s}) + \Psi_j(x_i)$$

and $\forall i, j$ we obviously have $\mathbb{E}[F_{i,j}] = 0$. We choose $0 < \beta < 1$ and we take M large enough such that $\mathbb{P}(\|F\|_1 \leq \beta)$ with probability η. We have with the same probability

$$\left\|C^{-1}\right\|_1 \leq \frac{1}{1-\beta}$$

and finally

$$\left\|\hat{U} - U\right\|_2 \leq \frac{\sqrt{N}}{1-\beta}\|h_d\|_2. \tag{11}$$

3.2 A Basic One Dimensional Example

The goal of this section is to explain on a trivial example the meaning of the error bounds we have just obtained. We consider the Laplace equation $u'' = 0$ on the interval $D = [0, 1]$ with boundary conditions $u(0) = 0$, $u(1) = 1$ which solution is $u(x) = x$. There is no source term and so the solution is

$$u(x) = \mathbb{P}_x(X_{\tau_D} = 1)u(1) + \mathbb{P}_x(X_{\tau_D} = 0)u(0) = \mathbb{P}_x(X_{\tau_D} = 1).$$

The solution is computed at points $x = \frac{1}{3}$ and $x = \frac{2}{3}$. The law of X_{τ_D} given that $X_0 = \frac{1}{3}$ is a Bernoulli random variable W such that $\mathbb{P}(W = 0) = \frac{2}{3}$ and the law of X_{τ_D} given that $X_0 = \frac{2}{3}$ is a Bernoulli random variable Z such that $\mathbb{P}(Z = 0) = \frac{1}{3}$. We sample directly and exactly from W and Z to obtain our estimates. We denote by p and q the Monte Carlo approximations of these probabilities using M samples. We choose for basis functions the Lagrange polynomials $\Psi_1(x) = -3(x - \frac{2}{3})$ and $\Psi_2(x) = 3(x - \frac{1}{3})$. The exact solution is in the approximation space (that is $u = P_2u$, see (5)). Thanks to (10), we have $h_d = 0$. So, if the spectral matrix C is regular, (11) ensures there is no error on the solution. As we have $\Psi_1(0) = 2, \Psi_1(1) = -1$ and $\Psi_2(0) = -1, \Psi_2(1) = 2$, the linear system to solve is $CU = d$ with $C = \begin{pmatrix} 3p-1 & 2-3p \\ 3q-1 & 2-3q \end{pmatrix}$ and $d = \begin{pmatrix} 1-p \\ 1-q \end{pmatrix}$. The vector $(\frac{1}{3}, \frac{2}{3})$ is always a solution of this system but is the unique solution only if C is regular, that is when $\det(C) = 3(p - q) \neq 0$. When M increases, $p \to \frac{2}{3}$ and $q \to \frac{1}{3}$ at a Monte Carlo speed. So the probability that $p = q$ decreases quickly with M. Moreover, even if $M = 1$, the solution is unique as soon as $W \neq Z$. The probability that the matrix is singular is

$$p_M = \mathbb{P}_M(W = Z) = \sum_{k=0}^{M} \binom{M}{k} \left(\frac{1}{3}\right)^{M-k} \left(\frac{2}{3}\right)^{k} \binom{M}{k} \left(\frac{2}{3}\right)^{M-k} \left(\frac{1}{3}\right)^{k}$$

$$= \left(\frac{2}{9}\right)^M \sum_{k=0}^{M} \binom{M}{k}^2,$$

where $\binom{M}{k}$ are the binomial coefficients. For instance, we have $p_1 = \frac{4}{9}$, $p_{10} \simeq 0,054$, $p_{20} \simeq 0.01$ and $p_{50} \simeq 0.0002$. We can conclude that we obtain an exact solution with a probability $p_M \geq 0.99$ as soon as $M \geq 20$.

3.3 The Biased Monte Carlo Case

We now assume that the process $\{X_t, t \geq 0\}$ is approximated by another process $\{X_t^\delta, t \geq 0\}$ built using a simulation scheme like the Euler scheme or the walk on spheres method with a discretization time step δ. We also assume that $p_i = q_i = M$. In this situation we have to compute error bounds for expressions of the form

$$\mathbb{E}_x[g(X_{\tau_D}) + \tau_D f(X_{U\tau_D})] - \mathbb{E}_x[g(X^\delta_{\tau_D}) + \tau^\delta_D f(X^\delta_{U\tau_D})] = e_1 + e_2$$

letting

$$e_1 = \mathbb{E}_x[g(X_{\tau_D}) - g(X^\delta_{\tau_D})] \quad \text{and} \quad e_2 = \mathbb{E}_x[\tau_D f(X_{U\tau_D}) - \tau^\delta_D f(X^\delta_{U\tau_D})].$$

First, we can notice that for any process $\{X_s^\delta, t \geq 0\}$, we always have

$$|e_1| \leq 2 \sup_{x \in \partial D} |g(x)|$$

and if $\mathbb{E}_x[\tau_D^\delta] < \infty$,

$$|e_2| \leq (\mathbb{E}_x[\tau_D] + \mathbb{E}_x[\tau_D^\delta]) \sup_{x \in D} |f(x)|.$$

If we now really use that $\{X_t^\delta, t \geq 0\}$ is an approximation of $\{X_t, t \geq 0\}$, we can expect to have error bounds of the form

$$|e_1| \leq C\delta^\alpha \sup_{x \in \partial D} |g(x)|$$

and also

$$|e_2| \leq C_1 \delta^\beta \sup_{x \in D} |f(x)|$$

where α, β, C and C_1 are non-negative constants. In both cases, we have

$$|e_1 + e_2| \leq \mu_\delta \sup_{x \in \partial D} |g(x)| + \nu_\delta \sup_{x \in D} |f(x)|$$

where μ_δ and ν_δ are positive constants which may or may not (if α or β is zero) go to zero as $\delta \to 0$. As in Sec. 3.1, we need $\mathbb{E}\left[\left(\tau_D^\delta\right)^3\right] < \infty$ to apply the Berry-Esseen

inequality. This condition is satisfied for most of the classical schemes like Euler scheme or the walk on sphere method (see the Appendix of [9]). If we go back to our problem, we obtain a new system of inequalities

$$d_\delta - h_{d_\delta} \leq C_\delta V_\delta \leq d_\delta + h_{d_\delta}$$

where C_δ and d_δ are defined using X_s^δ and where

$$\left\| h_{d_\delta} \right\|_2^2 \leq \theta(\delta, N, M) \sup_{x \in \partial D} (g - P_N u)^2(x) + \kappa(\delta, N, M) \sup_{x \in D} (f + L P_N u)^2(x)$$

with

$$\theta(\delta, N, M) = 4C_\alpha^2 \left(\frac{N}{M} + \mu_\delta^2 \right),$$
$$\kappa(\delta, N, M) = 4C_\alpha^2 \left(\frac{N}{M} \max_{i=1,N} \mathbb{E}_{x_i}[(\tau_D^\delta)^2] + \nu_\delta^2 \right).$$

We have finally with probability η,

$$\left\| \hat{U}_\delta - U \right\|_2 \leq \left\| C_\delta^{-1} \right\|_2 \left\| h_{d_\delta} \right\|_2,$$

where $\hat{U}_\delta$ is our approximated solution. We observe that in fact the quality of the simulation scheme and the number of simulations have not such a big impact on $\left\| h_{d_\delta} \right\|_2$ as they influence only the constants $\theta(\delta, N, M)$ and $\kappa(\delta, N, M)$. This means that we can have a good enough control on $\left\| h_{d_\delta} \right\|_2$ even with a very bad simulation scheme and few simulations. Conversely, we cannot expect a good convergence of C_δ towards the identity matrix in this last situation so there is a lack of control on $\left\| C_\delta^{-1} \right\|_2$. This can lead to very large values for $\left\| C_\delta^{-1} \right\|_2$ when using polynomial bases of high degree in the approximation of f and g. This is similar to the bad conditioning of spectral methods for elliptic problems. We could also find an upper bound for $\left\| C_\delta^{-1} \right\|_2$ as we did in the unbiased case when M is large enough and δ is small enough.

3.4 Other Cases

Instead of making Monte Carlo approximations of the Feynman-Kac representations, it might be possible to use other approximation methods like quasi-Monte Carlo methods or quantization which may have increased rates of convergence. In such cases, the error bounds do not depend on the central limit theorem via the variance but on other estimates via the discrepancy or the distortion. Some work has been done to simulate diffusions at a quasi-Monte Carlo speed first for the heat equation in $\mathbb{R}^d$ see [13] and then for elliptic problems in bounded domains in the context of domain decomposition [1]. This last approach is very promising but we do not

know yet how to combine it with the one random point approximation of the source term. The remaining problem is how to compute a quasi-Monte Carlo approximation of $\mathbb{E}_x[\tau_D f(X_{U\tau_D})]$. In Section 2.2, we have described how the quantization method works in our context and especially how to deal with the source term evaluated by the one random point method. In both situations, error bounds in the case of zero-bias schemes will be deterministic and the speed of convergence is likely to be faster than the Monte Carlo speed.

4 Numerical Results

We describe our method on equation (2) in the unit square $D = [-1, 1]^2$ where

$$L = \frac{1}{2}\left(\frac{\partial^2}{\partial x^2} + 4\frac{\partial^2}{\partial x \partial y} + 5\frac{\partial^2}{\partial y^2}\right).$$

This problem corresponds to a simple example of an anisotropic diffusion. We solve this equation for three different source terms f_i and boundary conditions g_i for which the exact solutions are $u_1(x, y) = (1-x^2)(1-y^2)$, $u_2(x, y) = (1-x^3)(1-y^3)$ and $u_3(x, y) = \sin\left((1-x^2)(1-2y^2)\right)$. For example, $f_1(x, y) = 6 - 5x^2 - y^2 - 8xy$ and $g_1(x, y) = 0$.

The stochastic process associated to L is solution to the SDE

$$\begin{cases} dX_t = dB_t^1 \\ dY_t = 2dB_t^1 + dB_t^2, \end{cases}$$

where $\{B_t^1, t \geq 0\}$ and $\{B_t^2, t \geq 0\}$ are two independent one dimensional Brownian motions. In all the following results, we use an Euler scheme with time step δ and the half-space approximation [7]. For either Monte Carlo simulations or quantization and for each of the grid points, we take the same number of points M on the boundary and in the domain that is $p_i = q_i = M$. We approximate our solution using Tchebychef interpolation polynomials. Hence the N basis functions are the two dimensional Lagrange polynomials ψ_i associated to the Tchebychef grid. We use either Monte Carlo simulations or quantization tools.

We denote by $\hat{u}_i$ our approximation of u_i $(i = 1, 2, 3)$. We give some criteria to study our method on these examples. These criteria are the error on the solution $\text{err}_i = \sup|u_i - \hat{u}_i|$ (where the supremum is taken over the points of the Tchebychef grid), the condition number $\kappa(C)$ and the spectral radius of the Jacobi $\rho(J)$ and Gauss-Seidel $\rho(GS)$ iteration matrices. We summarize the results in Tables 1 and 2.

In Figure 1, we plot the quantization points on the boundary for point $(x_0, y_0) = (0, 0)$ and we compare them to the ones obtained in the Brownian case.

We can notice that when the solution is in the approximation space there is almost no error on the solution. The condition number of the corresponding deterministic collocation methods is a $O(N^4)$ [4]. Here the condition number is very small especially when using quantization points. When $N = 121$, we observe that the quan-

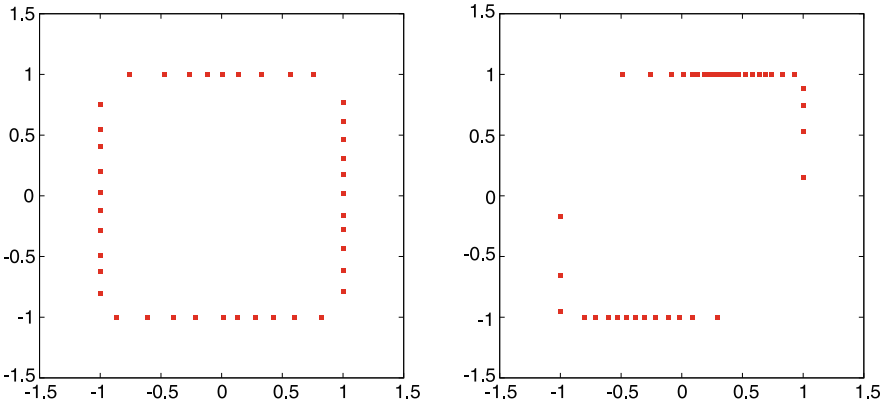
Table 1 Numerical Results for the Monte Carlo procedure with 1000 realisations and a time step $\delta = 10^{-3}$.

N	err_1	err_2	err_3	$\kappa(C)$	$\rho(J)$	$\rho(GS)$
9	3.9×10^{-16}	3.8×10^{-2}	2.7×10^{-2}	1.1	4.9×10^{-3}	4.4×10^{-2}
16	1.5×10^{-15}	3.8×10^{-15}	1.3×10^{-2}	1.4	1.2×10^{-1}	1.6×10^{-2}
121	8.5×10^{-14}	1.1×10^{-13}	1.4×10^{-4}	974	1.74	3.39

Table 2 Numerical Results for quantization procedure with M quantifiers and a time step $\delta = 10^{-4}$.

N	M	err_1	err_2	err_3	$\kappa(C)$	$\rho(J)$	$\rho(GS)$
9	80	4.4×10^{-16}	2.1×10^{-2}	6.3×10^{-3}	1.06	1.5×10^{-2}	2.4×10^{-3}
16	80	5.6×10^{-16}	2.6×10^{-15}	1.7×10^{-2}	1.13	3.2×10^{-2}	7.3×10^{-3}
121	200	2.6×10^{-15}	8.4×10^{-15}	1.3×10^{-5}	8.5	0.46	0.28

tization method is a lot more efficient: it provides a more accurate solution with a smaller condition number and the Jacobi and Gauss-Seidel method are convergent.



(a) Brownian Motion (b) Anisotropic Process

Fig. 1 Quantization points $(x_0, y_0) = (0, 0)$.

5 Conclusion

We have introduced and studied stochastic versions of the collocation method for the solution of elliptic problems in a bounded domain. We have given asymptotic properties of the stochastic spectral matrix and error bounds on the approximate solutions in the very general context of linear approximations. We have proved the

convergence of the spectral matrix toward the identity matrix when increasing the number of simulations and decreasing the stepsize of the simulation schemes involved in the approximations of the Feynman-Kac representations. We have also proved that very accurate solutions can be obtained even when using a small number of simulations with a poor simulation scheme. Numerical results have confirmed the efficiency of the method on the Poisson equation [15] and on an anisotropic diffusion in the unit square. We have also paid a special attention to the optimization of the computation of the Feynman-Kac formula via one random step schemes and quantization tools. In the spirit of what has been done in [14] for numerical integration, the combination of stochastic tools and deterministic approximations has led to stochastic spectral methods which are asymptotically perfect in terms of conditioning. Further numerical examples should be performed on more complex domains or partial differential equations to emphasize the simplicity and efficiency of this new approach.

Acknowledgements The authors wish to thank the anonymous referee and the associated editor for their useful suggestions and remarks, which contributed greatly to clarify this paper.

References

1. Acebrón, J.A., Busico, M.P., Lanucara, P., Spigler, R.: Domain decomposition solution of elliptic boundary-value problems via Monte Carlo and quasi-Monte Carlo methods. SIAM J. Sci. Comput. **27**(2), 440–457 (electronic) (2005)
2. Bally, V., Talay, D.: The law of the Euler scheme for stochastic differential equations. I. Convergence rate of the distribution function. Probab. Theory Related Fields **104**(1), 43–60 (1996)
3. Benveniste, A., Métivier, M., Priouret, P.: Adaptive algorithms and stochastic approximations, *Applications of Mathematics (New York)*, vol. 22. Springer-Verlag, Berlin (1990). Translated from the French by Stephen S. Wilson
4. Bernardi, C., Maday, Y.: Approximations spectrales de problèmes aux limites elliptiques, *Mathématiques & Applications (Berlin) [Mathematics & Applications]*, vol. 10. Springer-Verlag, Paris (1992)
5. Deaconu, M., Lejay, A.: A random walk on rectangles algorithm. Methodol. Comput. Appl. Probab. **8**(1), 135–151 (2006)
6. Freidlin, M.: Functional integration and partial differential equations, *Annals of Mathematics Studies*, vol. 109. Princeton University Press, Princeton, NJ (1985)
7. Gobet, E.: Weak approximation of killed diffusion using Euler schemes. Stochastic Process. Appl. **87**(2), 167–197 (2000)
8. Gobet, E., Maire, S.: A spectral Monte Carlo method for the Poisson equation. Monte Carlo Methods Appl. **10**(3-4), 275–285 (2004)
9. Gobet, E., Maire, S.: Sequential control variates for functionals of Markov processes. SIAM J. Numer. Anal. **43**(3), 1256–1275 (electronic) (2005)
10. Gobet, E., Maire, S.: Sequential Monte Carlo domain decomposition for the Poisson equation. In: Proceedings of the 17th IMACS World Congress, Scientific Computation, Applied Mathematics and Simulation (2005)
11. Hwang, C.O., Mascagni, M., Given, J.A.: A Feynman-Kac path-integral implementation for Poisson's equation using an h-conditioned Green's function. Math. Comput. Simulation **62**(3-6), 347–355 (2003). 3rd IMACS Seminar on Monte Carlo Methods—MCM 2001 (Salzburg)

12. Lapeyre, B., Pardoux, É., Sentis, R.: Méthodes de Monte-Carlo pour les équations de transport et de diffusion, *Mathématiques & Applications (Berlin) [Mathematics & Applications]*, vol. 29. Springer-Verlag, Berlin (1998)
13. Lécot, C., El Khettabi, F.: Quasi-Monte Carlo simulation of diffusion. J. Complexity **15**(3), 342–359 (1999). Dagstuhl Seminar on Algorithms and Complexity for Continuous Problems (1998)
14. Maire, S., De Luigi, C.: Quasi-Monte Carlo quadratures for multivariate smooth functions. Appl. Numer. Math. **56**(2), 146–162 (2006)
15. Maire, S., Tanré, E.: Some new simulations schemes for the evaluation of Feynman-Kac representations. Monte Carlo Methods Appl. **14**(1), 29–51 (2008)
16. Niederreiter, H.: Random number generation and quasi-Monte Carlo methods, *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1992)
17. Pagès, G.: A space vector quantization method for numerical integration. J. Computational and Applied Mathematics **89**, 1–38 (1998)
18. Pagès, G., Printems, J.: Optimal quadratic quantization for numerics: the Gaussian case. Monte Carlo Methods Appl. **9**(2), 135–165 (2003)
19. Reutenauer, V., Tanré, E.: Exact simulation of prices and greeks: application to CIR (2008). URL http://hal.inria.fr/inria-00319139/en/
20. Sabelfeld, K.K.: Monte Carlo methods in boundary value problems. Springer Series in Computational Physics. Springer-Verlag, Berlin (1991). Translated from the Russian
21. Shiganov, I.: Refinement of the upper bound of the constant in the central limit theorem. J. Sov. Math. **35**, 2545–2550 (1986)

Adaptive (Quasi-)Monte Carlo Methods for Pricing Path-Dependent Options

Roman N. Makarov

Abstract We study a recently developed adaptive path-integration technique for pricing financial derivatives. The method is based on the rearrangement and splitting of path-integral variables to apply a combination of bridge sampling, adaptive methods of numerical integration, and the quasi-Monte Carlo method. We study the subregion adaptive Vegas-type method Suave from the CUBA library and propose a new variance reduction method with a multivariate piecewise constant sampling density. Two models of asset pricing are considered: the constant elasticity of variance diffusion model and the variance gamma Lévy model. Numerical tests are done for Asian-type options.

1 Introduction

The idea of using bridge sampling to reduce the effective dimension of a multivariate integration problem was proposed in [2] for pricing path-dependent options under the geometric Brownian motion. Such an approach followed by application of the quasi-Monte Carlo method becomes a classical example of variance reduction and is widely used in computational finance. For example, it is successfully applied to pricing options under the variance gamma model [1]. In reality, the use of bridge sampling for modern asset price models becomes more complicated and computationally time consuming. The application of the quasi-Monte Carlo method and some other variance reduction techniques often requires computations of the inverse of a probability distribution function. For many interesting models that arise in mathematical finance (see, e.g., [4]), the inverse of a distribution function can be computed only by numerically solving a corresponding differential equation. Therefore, it is reasonable to consider the minimization of computational cost of a Monte Carlo algorithm rather than only the minimization of the variance of a random esti-

Department of Mathematics, Wilfrid Laurier University, Waterloo, Ontario, Canada
e-mail: rmakarov@wlu.ca

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_34,

mator. As usual, the computational cost is defined here as a product of the variance of a random estimator and the average computing time per one sample.

Our approach is based on combining the bridge sampling method with adaptive Monte Carlo techniques and randomized quasi-Monte Carlo methods (see [3]). Starting with a path-integral representation of the value of a discretely-monitored path-dependent option, we rearrange the integral variables and then partition the state space into two subspaces so that the variance of a standard Monte Carlo estimator of the path integral depends mostly on variables of the subspace of smaller dimension. Such a subspace can be called the effective subspace. Variables of the effective subspace explain most of the variability of a sample path and are used to construct a path "skeleton". A bridge sampling method is then applied to fill out gaps between the nodes of the path "skeleton". The option value is represented as a double mathematical expectation with respect to the two subspaces, so the Monte Carlo method can be applied for estimating it. A combination of variance reduction techniques is applied only in the effective subspace. The main motivation of such an approach is that variance reduction methods are usually more computationally time-consuming than a plain Monte Carlo simulation method. As a result, we can achieve almost the same efficiency in variance reduction as if the variance reduction methods were applied to the entire state space but for less cost. We illustrate efficiency improvements with numerical examples of pricing Asian options under the variance gamma (VG) exponential Lévy model [11] or the constant variance of elasticity (CEV) diffusion model [5].

Note that this approach can be also used for the evaluation of multidimensional integrals, whose variables admit a decomposition into two subsets followed by application of conditional Monte Carlo methods.

2 Path-Integral Decomposition Approach

Let us start with a continuous-time stochastic process $\{S(t)\}_{t\geq 0} \in \mathbb{R}_+$ that models the price of some financial asset, such as a stock, a commodity, or the value of a financial index. We assume that there exists a risk-neutral probability measure $\mathbb{P}$ so that the discounted asset price process $\tilde{S}(t) = e^{-(r-q)t} S(t)$ is a $\mathbb{P}$-martingale, that is, $\mathsf{E}^{\mathbb{P}}|\tilde{S}(t)| < \infty$ and $\mathsf{E}^{\mathbb{P}}[\tilde{S}(t+\tau) \mid \tilde{S}(t)] = \tilde{S}(t)$ holds for all $t, \tau \geq 0$. Here $r \geq 0$ is the constant risk-free interest rate, and $q \geq 0$ is the dividend yield rate. We also assume that $S(t)$ is a solvable Markov process meaning that its transition probability density function (PDF) p, defined below, is given in closed form:

$$p(t_1, s_1, t_2, s_2)\mathrm{d}s_2 = \Pr\{S(t_2) \in \mathrm{d}s_2 \mid S(t_1) = s_1\}, \ 0 \leq t_1 < t_2, \ s_1, s_2 \in \mathbb{R}_+.$$

The problem of interest is pricing discretely-monitored path-dependent options. Let the total time interval $[0, T]$, $T > 0$, be discretized with partition $\mathcal{T} = \{t_i, i = 0, 1, \ldots, n\}$, so that $0 = t_0 < t_1 < \ldots < t_n = T$. Denote the values of the price process $S(t)$ at the time points $t = t_i$ by S_i for all $i = 0, 1, \ldots, n$. The no-arbitrage value V

of a path-dependent option characterized by its payoff function $\Lambda(s_1, s_2, \ldots, s_n)$ is given in the form of a mathematical expectation:

$$V = V(S_0, \mathcal{T}, \Lambda) = \mathsf{E}^{\mathbb{P}}\left[e^{-rT}\Lambda(S_1, S_2, \ldots, S_n) \mid S(0) = S_0\right]. \tag{1}$$

Using the transition PDF p and exploiting the Markov property of the process S, one can construct a multidimensional PDF $\mathbf{f}$ of the discretized path $\mathbf{S} := \{S_1, S_2, \ldots, S_n\}$. For example, assuming that the values S_i are sampled sequentially we have:

$$\mathbf{f}(\mathbf{s}, \mathcal{T}; s_0) = \prod_{i=1}^{n} p(t_{i-1}, s_{i-1}, t_i, s_i), \quad \mathbf{s} := \{s_i\}_{i=1}^{N}. \tag{2}$$

As is well known, the mathematical expectation (1) can then be written in the form of a path-integral:

$$V = \int_{\mathbb{R}_+^n} e^{-rT}\Lambda(\mathbf{s})\,\mathbf{f}(\mathbf{s}, \mathcal{T}; S_0)\,\mathrm{d}\mathbf{s}. \tag{3}$$

Consider a decomposition of the set of integration variables $\mathbf{s}$ into two disjoint subsets, $\mathbf{s}^1$ and $\mathbf{s}^2$, containing d, $1 \le d \le n$, and $n-d$ variables, respectively:

$$\mathbf{s} = \mathbf{s}^1 \cup \mathbf{s}^2 := \{s_{n/d}, s_{2n/d}, \ldots, s_n\} \cup \{s_i \mid i \ne kn/d,\ 1 \le i \le n,\ k \ge 1\}. \tag{4}$$

Here, we assume that n/d is an integer. According to this partition, the n-dimensional integral (3) can be written as an integral with respect to $\mathbf{s}^1$ and $\mathbf{s}^2$:

$$V = \int_{\mathbb{R}_+^d} \left(\mathbf{f}_1(\mathbf{s}^1, \mathcal{T}^1; S_0) \int_{\mathbb{R}_+^{n-d}} \mathbf{f}_2(\mathbf{s}^2, \mathcal{T}^2; S_0, \mathbf{s}^1, \mathcal{T}^1)\,\Lambda(\mathbf{s}^1, \mathbf{s}^2)\,\mathrm{d}\mathbf{s}^2 \right) \mathrm{d}\mathbf{s}^1, \tag{5}$$

where $\mathcal{T}^1$ and $\mathcal{T}^2$ are two disjoint subsets of the time partition $\mathcal{T}$ that correspond to $\mathbf{s}^1$ and $\mathbf{s}^2$, respectively. The path density $\mathbf{f}$ is represented as a product of the marginal PDF $\mathbf{f}_1$ of $\mathbf{S}^1 := \{S_{kn/d}\}_{k=1}^{d}$ and the PDF $\mathbf{f}_2$ of $\mathbf{S}^2 := \mathbf{S} \setminus \mathbf{S}^1$ conditional on $\mathbf{S}^1$. Alternatively, one can represent the option value V as a double expectation of the payoff function

$$V = \mathsf{E}_{\mathbf{S}^1}^{\mathbb{P}}\left[\mathsf{E}_{\mathbf{S}^2}^{\mathbb{P}}\left[e^{-rT}\Lambda(\mathbf{S}^1, \mathbf{S}^2) \mid \mathbf{S}^1\right] \mid S(0) = S_0\right]. \tag{6}$$

The multidimensional PDFs $\mathbf{f}_1$ and $\mathbf{f}_2$ can be represented as products of one-dimensional transitions density functions depending on algorithms used to sample $\mathbf{S}^1$ and $\mathbf{S}^2$. Suppose that the variables of $\mathbf{S}^1$ are sampled in a sequential manner. Then we have

$$\mathbf{f}_1(\mathbf{s}^1, \mathcal{T}^1; s_0) = \prod_{k=1}^{d} p\left(t_{\frac{(k-1)n}{d}}, s_{\frac{(k-1)}{n}d}, t_{\frac{kn}{d}}, s_{\frac{kn}{d}}\right). \tag{7}$$

The variables of $\mathbf{S}^2$ are sampled conditionally on values S_i obtained previously. For example, being given $\mathbf{S}^1$ we sample $S_1 \in \mathbf{S}^2$ conditional on the values S_0 and $S_{n/d}$. After that we sample S_2 conditional on S_1 and $S_{n/d}$, and so on, and

so forth. At the end, we sample S_{n-1} conditional on S_{n-2} and S_n. The PDF $\mathbf{f}_2 = \mathbf{f}_2(\mathbf{s}^2, \mathcal{T}^2; s_0, \mathbf{s}^1, \mathcal{T}^1)$ that corresponds to this sampling technique is (under the Markov assumption) given by

$$\mathbf{f}_2 = \prod_{k=1}^{d} \prod_{j=\frac{(k-1)n}{d}+1}^{\frac{kn}{d}-1} \frac{p\,(t_{j-1}, s_{j-1}, t_j, s_j)\; p\,(t_j, s_j, t_{kn/d}, s_{kn/d})}{p\,(t_{j-1}, s_{j-1}, t_{kn/d}, s_{kn/d})}. \tag{8}$$

Note that the ratio of transition PDFs in the above equation is a bridge probability density of S_j conditional on $S_{j-1} = s_{j-1}$ and $S_{kn/d} = s_{kn/d}$. It is easy to show that the product of the PDFs $\mathbf{f}_1$ and $\mathbf{f}_2$ defined in (7) and (8), respectively, is exactly the PDF $\mathbf{f}$ given by (2).

To estimate the option value V given by (1) or (6) one can use the Monte Carlo method. Suppose that the variables of $\mathbf{S}^1$ give much larger contributions to the variance of a random estimator of V in comparison with the variables of $\mathbf{S}^2$. We say that the path integral has effective dimension d in proportion p_d if the variables of $\mathbf{S}^1$ account for at least $100 \cdot p_d\%$ of the variance (see [2] for details). For pricing path-dependent options, it is a fairly typical situation where p_d is very close to one for values $d \ll n$. So it is reasonable to apply variance reduction techniques only to the variables of $\mathbf{S}^1$ rather than to all variables. Another reason is that the dimension of the problem may be very large. For a common option pricing problem, the dimension n is of order of several hundreds. The application of some popular variance reduction methods may slow down considerably the computation algorithm, since they are usually more time-consuming than a plain Monte Carlo simulation method. By applying variance reduction methods to the "effective" variables of $\mathbf{S}^1$, we can achieve almost the same efficiency in variance reduction as if these methods were applied to the whole integral but for less cost. Finally, we note that most methods of generating quasirandom point sets (or at least their software implementations) have a restriction on the dimension of points. Typically, the maximum allowable dimension is 40 (e.g., see [8]).

Let us change the region of integration in (5). Let $\mathbf{F}_1(\mathbf{S}^1, \mathcal{T}^1; S_0)$ be the joint cumulative distribution function (CDF) corresponding to the PDF $\mathbf{f}_1$. By applying the change of variables defined by $\mathbf{F}_1(\mathbf{S}^1, \mathcal{T}^1; S_0) = \mathbf{u}$, where $\mathbf{u} \in (0,1)^d$, the integral (5) is transformed as follows:

$$V = \int_{(0,1)^d} \left(\int_{\mathbb{R}_+^{n-d}} \mathbf{f}_2(\mathbf{s}^2, \mathcal{T}^2; S_0, \mathbf{s}^1, \mathcal{T}^1)\, \Lambda(\mathbf{s}^1, \mathbf{s}^2,)\mathrm{d}\mathbf{s}^2 \right) \mathrm{d}\mathbf{u}, \tag{9}$$

where we set $\mathbf{s}^1 = \mathbf{F}_1^{-1}(\mathbf{u}; S_0, \mathcal{T}^1)$, and $\mathbf{F}_1^{-1}$ is the inverse function. As a result, we obtain a problem of numerical integration over a unit d-dimensional hypercube. The internal $(n-d)$-dimensional integral can be estimated by the Monte Carlo method. Therefore, the value of V can be estimated using the double randomization technique (with possible branching).

3 Adaptive Integration Methods

3.1 Adaptive Importance and Stratified Samplings

In the Monte Carlo method, one of the most fundamental variance reduction techniques is the importance sampling principle. This principle tells us that to decrease the variance of a Monte Carlo estimator of a definite integral the sampling density has to be chosen close enough to the integrand. Of great interest are adaptive numerical integration methods that are problem-independent and allow us to construct a sampling density during the Monte Carlo simulation.

There are two well known multidimensional adaptive numerical integration algorithms: Vegas and Miser. The Vegas algorithm, invented by Lepage [10, 12], is a Monte Carlo integration routine that applies adaptive importance sampling. Vegas iteratively builds up a multidimensional sampling PDF as a product of one-dimensional piecewise constant functions. As a result, Vegas can exhibit significant improvements, but only as far as the integrand's characteristic regions are aligned with the coordinate axes. The Miser integration routine [12] is based on the recursive stratified sampling on a rectangular grid. Until the requested accuracy is reached, the region with the largest error at the time is bisected in the dimension in which the fluctuations of the integrand are reduced most. The number of new samples in each half is prorated for the fluctuation in that half.

In [7], CUBA—a library for multidimensional numerical integration, is presented. The CUBA library provides new implementations of four general-purpose multidimensional integration algorithms: Vegas, Suave, Divonne, and Cuhre. Suave (short for **su**bregion-**a**daptive **Ve**gas) uses Vegas-like importance sampling combined with a globally adaptive subdivision strategy similar to that implemented in Miser. Suave uses global error estimation and terminates when the requested relative or absolute accuracy is attained. Divonne works by stratified sampling, where the partitioning of the integration region is aided by methods from numerical optimization. Cuhre employs a cubature rule for subregion estimation in a globally adaptive subdivision scheme.

In this paper we are specifically interested in the implementation of the Suave method since it is an immediate successor of the Vegas routine. The other two routines do not fit well with our approach.

3.2 Sampling from a Multivariate Piecewise Constant PDF

In this subsection, we present a new approach, when the sampling PDF is a multivariate piecewise constant (within rectangles) function. The problem of our interest is the efficient sampling from such a PDF. In fact, this approach is somewhat analogous to the stratified sampling method, where the strata correspond to the regions of constancy of the piecewise constant probability density. Since the PDF is con-

structed adaptively by sampling points in each stratum, this approach can also be referred to adaptive methods.

Introduce a partition of a d-dimensional unit hypercube $[0,1]^d$ by slicing the cube into m_j equal parts along each coordinate $j = 1, 2, \ldots, d$. For each $j = 1, 2, \ldots, d$, introduce $I_j(i) := \left[\frac{i-1}{m_j}, \frac{i}{m_j}\right]$, $i = 1, 2, \ldots, m_j$, $m_j \geq 1$. The unit hypercube can be then represented as a union of $m_1 \cdot m_2 \cdots m_d$ disjoint smaller hyper-parallelepipeds of the form $C_{i_1,i_2,\ldots,i_d} := I_1(i_1) \times I_2(i_2) \times \cdots \times I_d(i_d)$, with $1 \leq i_j \leq m_j$.

Introduce a piecewise constant probability density $f(x_1, x_2, \ldots, x_d)$ that is constant in each subcube $C_{i_1,i_2,\ldots,i_d}$. Let $f(x_1, x_2, \ldots, x_d) \equiv f_{i_1,i_2,\ldots,i_d} > 0$ in $C_{i_1,i_2,\ldots,i_d}$ for $1 \leq i_j \leq m_j$, $j = 1, 2, \ldots, d$. This density can be written in terms of indicator functions as follows:

$$f(x_1, x_2, \ldots, x_d) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_d=1}^{m_d} f_{i_1,i_2,\ldots,i_d} \cdot \mathbb{1}_{I_1(i_1)}(x_1) \cdots \mathbb{1}_{I_d(i_d)}(x_d), \tag{10}$$

where $\mathbb{1}_A(x) := 1$ if $x \in A$ and $\mathbb{1}_A(x) := 0$ otherwise.

To sample from a piecewise constant PDF, one may use the following algorithm. First sample a subdomain from the partition of the cube. In the above setting, a subparallelepiped $C_{i_1,i_2,\ldots,i_d}$ is selected with a probability proportional to $f_{i_1,i_2,\ldots,i_d}$. After that, a point is sampled uniformly in the subdomain selected. This means that $d+1$ uniform random numbers are used for each sample value: one random number is used to choose a subdomain and d variates are used to sample a point in the subdomain selected.

Here we study another approach that employs the inverse of a distribution function. This gives grounds for the use of the quasi-Monte Carlo method. Let us represent a multivariate PDF $f(x_1, x_2, \ldots, x_d)$ as a product of one-dimensional marginal and conditional densities:

$$f(x_1, x_2, \ldots, x_d) = f_1(x_1) f_2(x_2|x_1) \cdots f_d(x_d|x_1, \ldots, x_{d-1}). \tag{11}$$

Now, one can first sample the first coordinate x_1 from the marginal density $f_1(x_1) = \int_{(0,1)^{d-1}} f(x_1, x_2, \ldots, x_d)\, \mathrm{d}x_2 \cdots \mathrm{d}x_d$ and then sample x_j for each $j = 2, \ldots, n$ from the density function $f_j(x_j|x_1, \ldots, x_{j-1})$ conditional on the coordinates $x_1, \ldots, x_{j-1}$.

Introduce the following notation:

$$\begin{aligned}
\Sigma_1(i_1) &= \sum_{i_2,\ldots,i_d} f_{i_1,i_2,\ldots,i_d}, \\
\Sigma_2(i_1, i_2) &= \sum_{i_3,\ldots,i_d} f_{i_1,i_2,\ldots,i_d}, \\
&\ldots \\
\Sigma_{d-1}(i_1, \ldots, i_{d-1}) &= \sum_{i_d} f_{i_1,i_2,\ldots,i_d}, \\
\Sigma_d(i_1, \ldots, i_d) &= f_{i_1,i_2,\ldots,i_d},
\end{aligned}$$

where $i_j = 1, \ldots, m_j$ for each $j = 1, 2, \ldots, m$. Using this notation, we represent the marginal and conditional density functions $f_1, f_2, \ldots, f_d$ as follows:

$$
\begin{aligned}
f_1(x_1) &= \sum_{i_1=1}^{m_1} \frac{\Sigma_1(i_1)}{m_2 \cdots m_d} \mathbb{1}_{I_1(i_1)}(x_1), \\
f_2(x_2|x_1) &= \sum_{i_2=1}^{m_2} m_2 \frac{\Sigma_2(\hat{i}_1, i_2)}{\Sigma_1(\hat{i}_1)} \mathbb{1}_{I_2(i_2)}(x_2), \\
&\cdots \\
f_d(x_d|x_1, \ldots x_{d-1}) &= \sum_{i_d=1}^{m_d} m_d \frac{\Sigma_d(\hat{i}_1, \ldots, \hat{i}_{d-1}, i_d)}{\Sigma_{d-1}(\hat{i}_1, \ldots, \hat{i}_{d-1})} \mathbb{1}_{I_d(i_d)}(x_d),
\end{aligned}
$$

where for each $j = 1, \ldots, m-1$ we define $\hat{i}_j := \arg\{i_j : \mathbb{1}_{I_j(i_j)}(x_j) = 1\}$. As we can see, the multivariate PDF $f(x_1, x_2, \ldots, x_d)$ is represented as a product of d piecewise constant univariate density functions. Therefore, random sampling from f reduces to sampling successively from d univariate probability distributions. Note that the ordering of variables $x_1, \ldots, x_d$ is arbitrary. Therefore, in the implementation of this method for estimating the integral (9), we can use the ordering from the bridge sampling method: $x_1 = u_1 \mapsto S_{n/d}$, $x_2 = u_2 \mapsto S_{n/2d}$, and so on; or the ordering from the sequential sampling method: $x_1 = u_1 \mapsto S_d$, $x_2 = u_2 \mapsto S_{2d}, \ldots, x_d = u_d \mapsto S_n$.

4 Asset Pricing Models

4.1 The CEV Diffusion Model

The constant elasticity of variance (CEV) diffusion process $\{S(t)\}_{t \geq 0}$ obeys the stochastic differential equation $dS(t) = \nu S(t)dt + \delta S(t)^{\beta+1} dW(t)$, $t \geq 0$, $S(0) = S_0 > 0$, where ν, δ, β are real parameters and $\{W(t)\}_{t \geq 0}$ is a standard Wiener process. Here we set $\nu = r - q$ and assume that $\delta > 0$ and $\beta \neq 0$.

When $\beta < 0$, the boundary $s = 0$ of the state space $(0, \infty)$ is regular. Here we consider the case when the endpoint $s = 0$ is a killing boundary. The transition PDF $p_0(s_0, t_0, s, t + t_0) = u_0(s, s_0, t)$, $s, s_0 > 0$, $t > 0$, $t_0 \geq 0$, for the CEV process $S^{(0)}(t)$ with zero drift ($\nu = 0$) takes the form

$$
u_0(s, s_0, t) = \frac{s^{-2\beta - \frac{3}{2}} s_0^{\frac{1}{2}}}{\delta^2 |\beta| t} \exp\left(-\frac{s^{-2\beta} + s_0^{-2\beta}}{2\delta^2 \beta^2 t} \right) \mathcal{I}_{\frac{1}{2|\beta|}} \left(\frac{s^{-\beta} s_0^{-\beta}}{\delta^2 \beta^2 t} \right), \tag{12}
$$

where $\mathcal{I}$ denotes the modified Bessel function of the second order. The density $u_0(s, s_0, t)$ does not integrate (with respect to s) to unity for $t > 0$, since $s = 0$ is an absorbing point, but the driftless CEV process obeys the martingale property: $E[S^{(0)}(t + \tau) \mid S^{(0)}(t)] = S^{(0)}(t)$ for all $t, \tau \geq 0$.

Note that if the reflecting boundary conditions is imposed at $s = 0$, then there is no absorption at the endpoint. The corresponding transition density (for the case with $\beta < -0.5$) is given by (12) with the replacement $\mathcal{I}_{\frac{1}{2|\beta|}} \to \mathcal{I}_{\frac{1}{2\beta}}$. The process $S^{(0)}(t)$ becomes a strict submartingale.

A CEV process $S^{(\nu)}(t)$ with nonzero drift parameter ν is obtained from $S^{(0)}(t)$ by means of scale and time transformation:

$$S^{(\nu)}(t) = e^{\nu t} S^{(0)}(\tau(t)), \text{ where } \tau(t) = \begin{cases} \frac{1}{2\nu\beta}(e^{2\nu\beta t} - 1), & \nu \neq 0, \\ t, & \nu = 0. \end{cases} \tag{13}$$

The resulting transition density $p_\nu(s_0, t_0, s, t + t_0) = u_\nu(s, s_0, t)$ with nonzero drift ν is given by

$$u_\nu(s, s_0, t) = e^{-\nu t} u_0(e^{-\nu t} s, s_0, \tau(t)). \tag{14}$$

Since the driftless process $S^{(0)}(t)$ obeys the martingale property, we have that $E[S^{(\nu)}(t)|S^{(\nu)}(0) = S_0] = e^{\nu t} E[S^{(0)}(\tau(t))|S^{(0)}(0) = S_0] = e^{\nu t} S_0$. Hence $S^{(\nu)}(0)$ drifts at a rate ν, and under the risk neutral measure the forward price $e^{-\nu t} S^{(\nu)}(t)$ is a true martingale.

The Monte Carlo simulation of the CEV diffusion is based on the reduction of the transition PDF (12) to that of the non-central chi-square distribution. Another approach presented in [3] relates to the so-called randomized gamma distributions of the first and second kinds (see [13]). Let $\mathcal{G}(\alpha, \beta)$ denote the gamma distribution with mean $\alpha\beta$ and variance $\alpha\beta^2$ (i.e., α is the shape parameter, and β is the scale parameter). The randomized gamma distribution of the *first type* is the mixture distribution $\mathcal{G}(\mu + \eta + 1, \theta)$, where $\theta > 0$, $\mu > -1$ are constants, and η has the Poisson distribution $\mathcal{P}(\alpha)$ with rate $\alpha > 0$. The randomized gamma distribution of the *second type* is the mixture distribution $\mathcal{G}(\mu + \eta_1 + 2\eta_2 + 1, \theta)$, where $\theta > 0$ and $\mu > -1$ are constants, η_1 and η_2 are independent random variables having the Poisson and Bessel distributions, respectively. A nonnegative integer random variable Y is said to be a Bessel random variable $\mathcal{BES}(\mu, b)$ with parameters $\mu > -1$ and $b > 0$ if

$$\Pr\{Y = n\} = \frac{(b/2)^{2n+\mu}}{\mathcal{I}_\mu(b)\, n!\, \Gamma(n + \mu + 1)}, \quad n = 0, 1, 2, \ldots.$$

Sampling from the transition and bridge distributions of the CEV process relies on the two following results from [3].

Algorithm 1

For all $s_0 > 0$, $t_0 \geq 0$, and $\Delta t > 0$, the value $S(t_0 + \Delta t)$ of a CEV process with drift parameter ν and **without absorption** (i.e., $s = 0$ is a reflecting boundary) conditional on $S(t) = s_0$ is obtained by generating random variables $\eta \sim \mathcal{P}\left(\frac{x_0}{2\Delta\tau}\right)$, $\gamma \sim \mathcal{G}\left(\frac{1}{2\beta} + \eta + 1, 2\Delta\tau\right)$, and then by setting $S(t_0 + \Delta t) := e^{\nu(t_0+\Delta t)}(\gamma\delta^2\beta^2)^{-1/(2\beta)}$, where $x_0 := \left(e^{-\nu t_0} s_0\right)^{-2\beta} / (\delta^2\beta^2)$ and $\Delta\tau := \tau(t_0 + \Delta t) - \tau(t_0)$.

On the other hand, all bridge CEV processes, whether with or without absorption, are simulated as follows.

Algorithm 2

Let $0 \le t_1 < t < t_2$ hold, then for all positive values s_1 and s_2, the value $S(t)$ of the bridge CEV process with drift ν, conditional on $S(t_1) = s_1$ and $S(t_2) = s_2$, is obtained by generating independent random variables $\eta_1(t) \sim \mathcal{P}\left(\frac{1}{2(\tau_2-\tau_1)}\left[\frac{\tau_2-\tau(t)}{\tau(t)-\tau_1}x_1 + \frac{\tau(t)-\tau_1}{\tau_2-\tau(t)}x_2\right]\right)$, $\eta_2 \sim \mathcal{BESS}\left(\frac{1}{2\vartheta}, \frac{\sqrt{x_1 x_2}}{\tau_2-\tau_1}\right)$, and then by setting $S(t) := e^{\nu t}(\gamma(t)\delta^2\beta^2)^{-1/(2\beta)}$, where $\gamma(t) \sim \mathcal{G}\left(\frac{1}{2\vartheta} + \eta_1(t) + 2\eta_2 + 1, \theta(t)\right)$, $\theta(t) := \frac{2(\tau(t)-\tau_1)(\tau_2-\tau(t))}{\tau_2-\tau_1}$, $\tau_i := \tau(t_i)$, and $x_i := \frac{\left(e^{-\nu t_i} s_i\right)^{-2\beta}}{\delta^2\beta^2}$, for $i = 1, 2$.

Another method of sampling CEV paths utilizes the non-central chi-square distribution $\chi^2_k(\lambda)$ with $k > 0$ degrees of freedom and non-centrality parameter $\lambda > 0$.

Algorithm 3

For all $S_0 > 0$, $t_0 \ge 0$, and $\Delta t > 0$, the value $S(t_0 + \Delta t)$ of a CEV process with drift parameter ν and **without absorption** conditional on $S(t_0) = s_0$ is obtained by generating a random variable $\chi \sim \chi^2_k(\lambda)$ with $k = 2 + \frac{1}{\beta}$ and $\lambda = \frac{x_0}{\Delta\tau}$, and then by setting $S(t_0 + \Delta t) := e^{\nu(t_0+\Delta t)}(\chi\delta^2\beta^2)^{-1/(2\beta)}$, where $x_0 := \left(e^{-\nu t_0} s_0\right)^{-2\beta}/(\delta^2\beta^2)$ and $\Delta\tau := \tau(t_0 + \Delta t) - \tau(t_0)$.

4.2 The Variance Gamma Model

The variance gamma (VG) process is a three-parameter generalization of the Brownian motion model for the dynamics of the logarithm of the stock price. It is obtained by evaluating the Brownian motion with drift at a random time given by a Gamma process (see [11]).

Let $B(t; \theta, \sigma) = \theta t + \sigma W(t)$, $t \ge 0$, denote a Brownian motion with drift θ and variance parameter σ. Here $W(t)$ is a standard Brownian motion. The Gamma process $G(t; \mu, \nu)$ with mean rate μ and variance rate ν is a random process with independent Gamma increments over nonoverlapping intervals of time. The increment $G(t + \tau; \mu, \nu) - G(t; \mu, \nu)$ over time interval $(t, t + \tau)$, $t, \tau \ge 0$ has the Gamma distribution with mean $\mu\tau$ and variance $\nu\tau$.

The VG process $X(t; \sigma, \nu, \theta)$ is defined in terms of the Brownian motion with drift $B(t; \theta, \sigma)$ and the Gamma process with unit mean rate, $G(t; 1, \nu)$ as

$$X(t; \sigma, \nu, \theta) := B(G(t; 1, \nu); \theta, \sigma).$$

The PDF of the VG process at time t can be expressed conditional on the realization of the Gamma time change G as a normal density function. The risk neutral process for the asset price is given by

$$S(t) := S_0 \exp((r - q - \omega)t + X(t; \sigma, \nu, \theta)), \tag{15}$$

where r and q has the same meaning as in Sect. 2 and Sect. 4.1, and the constant $\omega = \ln(1-\theta\nu-\sigma^2\nu/2)/\nu$ is chosen so that the discounted asset price process $e^{-(r-q)t}S(t)$ is a true martingale.

Sampling the variance gamma process relies on the normal, gamma, and beta probability distributions. One needs first to sample the gamma process and then to sample the Brownian motion conditional on the obtained values of the stochastic time process. Let $\mathcal{N}(\mu,\sigma^2)$ denote the normal distribution with mean μ and variance σ^2, and $\mathcal{B}(\alpha,\beta)$ denote the beta distribution. The following algorithm is used for sampling paths of the VG process.

Algorithm 4

A gamma process $G(t)$ with parameters (μ,ν) and a Brownian motion with drift $B(t)$ with parameters (θ,σ) can be simulated as follows (see [1] for details).

Sequential sampling of the gamma process: For any $0<t_1<t_2$ the increment $\Delta G = G(t_2)-G(t_1)$ has the $\mathcal{G}((t_2-t_1)\mu^2/\nu,\nu/\mu)$ distribution.

Bridge sampling of the gamma process: For any $0<t_1<t<t_2$ the conditional distribution of $G(t)$ given $G(t_1)=G_1$ and $G(t_2)=G_2$ is the same as $G_1+(G_2-G_1)Y$, where $Y\sim\mathcal{B}((t-t_1)\mu^2/\nu,(t_2-t)\mu^2/\nu)$.

Sequential sampling of Brownian motion: For any $0<g_1<g_2$ the increment $\Delta B = B(g_2)-B(g_1)$ has the $\mathcal{N}(\theta(g_2-g_1),\sigma^2(g_2-g_1))$ distribution.

Bridge sampling of Brownian motion: For any $0<g_1<g<g_2$ the conditional distribution of $B(g)$ given $B(g_1)=B_1$ and $B(g_2)=B_2$ is the normal distribution $\mathcal{N}(aB_1+(1-a)B_2, a(g-g_1)\sigma^2)$, where $a := \frac{g_2-g}{g_2-g_1}$.

5 Numerical Results

In our numerical examples, we deal with discretely-monitored Asian-style options whose payoff functions depend on an arithmetic average A_n of the underlying asset values given by $A_n = \frac{1}{n}\sum_{k=1}^{n} S_k$.

In this paper, all computations presented are done for Asian floating strike options, with the payoff functions $\Lambda_{FS}^{C} = (S_N - A_N)^+$ and $\Lambda_{FS}^{P} = (A_N - S_N)^+$ for the call and put options, respectively, where we define $(x)^+ := \max\{x,0\}$.

Below, we present numerical results for the two asset price models from Sect. 4. For each model we test and compare several algorithms: (1) the crude sequential MC simulation (with pseudorandom numbers used); (2) the partial QMC algorithm with quasirandom (QR) point sets used for numerical integration in the effective subspace $\mathbf{s}^1$; and (3) the adaptive integration methods, where either the Suave method from [7] or the PWCPDF method presented in Sect. 3.2 with or without QR points is used for the variables of $\mathbf{s}^1$. For the partial QMC and adaptive integration methods, we use the bridge sampling (with pseudorandom numbers used) in the subspace $\mathbf{s}^2$.

For the randomized quasi-Monte Carlo methods, we employ the algorithm 823 code from [8], where four different constructions of digital sequences (proposed by

Sobol', Faure, Niederreiter, and Niederreiter&Xing, respectively) and two types of random scrambling (proposed by Owen and Faure&Tezuka, respectively) are implemented. The results reported here only for the Sobol' and Faure point sets randomized by the Owen-scrambling method (see [8] for details).

For the length N of the quasi– or pseudo-random points set we use several values equal to powers of 2, namely, $N = 2^k$, $k = 13, 14, 15, 16$. To estimate the variance, $M = 100$ randomizations of each multidimensional point set were performed. For comparing the hybrid algorithms with a plain Monte Carlo method, a crude estimate was calculated using $N \cdot M$ sample paths with $N = 2^{16}$. Here, we analyse the variance reduction and computational cost reduction. The variance reduction factor of one estimate with respect to a crude MC estimate is given by the inverse ratio of their sample variances. A computation cost reduction factor (a speed-up factor) is defined as the inverse ratio of the sample computational costs that are given by the product of the sample variance and the average running time for one sample. For all hybrid methods, the internal integral from (9) is estimated by one sample value.

The hybrid QMC/MC algorithms are tested for values of the dimension d equal to powers of 2 from $d = 2$ to $d = 32$. The Suave algorithm works well for values of d up to 8. The PWCPDF method can be practically applied for values up to $d = 16$, since the number of elements in the partition of a unit hypercube grows exponentially. When d equals 2, 4, 8, or 16 each side of $(0, 1)^d$ is partitioned into 64, 8, 4, or 2 equal parts, respectively. For the sake of simplicity in the implementation of the PWCPDF method, each side of a unit hypercube is partitioned into $m = 2^l$ equal parts (for some integer l). The total number of subcubes obtained is equal to $m^d = 2^{l \cdot d}$. For every choice of d, we select l trying to make the method feasible (so $2^{l \cdot d}$ should not be too big) and efficient (so $m = 2^l$ should not be too small).

A piecewise constant PDF is constructed by averaging over 10 points sampled uniformly and independently in each subcube C. If a QR point set is applied, the piecewise constant density is constructed once and then it is reused for sampling with all randomized point sets. Note that for a Monte Carlo algorithm with pseudorandom point sets applied, the values used for constructing the density can be utilized for the resulting sample estimate as well.

For the adaptive integration algorithms we study two methods of utilizing QR point sets. One method is when the dimension of QR point sets is the same as the dimension d of the adaptive integration method (e.g., see the left and middle plots in Fig. 3). The other method is when the dimension of QR points is higher than d, so the first d coordinates of a QR point are used in an adaptive integration routine and the rest is used in the bridge sampling of the internal integral from (9) (e.g., see the right plot in Fig. 3).

5.1 Modelling with the VG Process

We use the same VG model parameters as those from [1]: $T = 0.40504$, $\nu = 0.2505$, $\theta = -0.2859$, $\sigma = 0.1927$, $r = 0.0548$, and $q = 0$. The spot price is $S_0 = 100$ and

the strike price (where applicable) is $K = 100$. The number of time partition points is $n = 128$. Since for sampling each value of $S(t)$ two random variables are required, the actual dimensionality of the path-integral problem is not 128 but 256. The bridge sampling method is used for modeling paths. Since we work exclusively with dyadic partitions of the time interval, a special method from [9] of sampling from the symmetric (with respect to $1/2$) beta distribution is employed.

The numerical results are presented in Figures 1–3. The use of bridge sampling and adaptive integration methods has no significant effect on the speed of sampling algorithms, so we report only the variance reduction factors. In the figures we show the variance reduction achieved for various values of d and N. Recall that the VG process is obtained as a superposition of Brownian motion and a gamma process. The PDF of each S_i is a mixture of normal and gamma density functions. Therefore, the actual dimensionality of the problem is $2n$, since every value S_i is a function of two independent stochastic factors. Consequently, the actual dimensionality of the effective subspace $\mathbf{s}^1$ is $2d$.

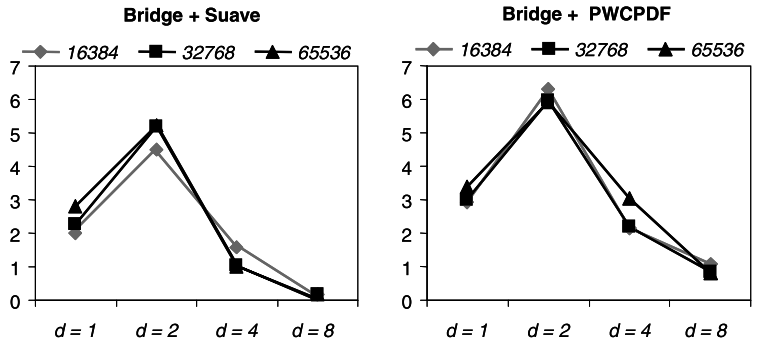


Fig. 1 Variance reduction for the adaptive integration methods with using pseudorandom point sets of lengths $2^{14}, 2^{15}, 2^{16}$ as compared to a plain MC algorithm. The test problem is the pricing of a floating strike put option under the VG model.

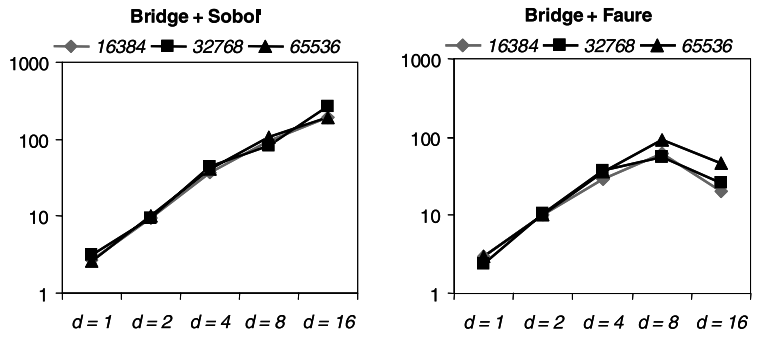


Fig. 2 Variance reduction for the partial QMC methods with using randomized Sobol' (the left plot) and Faure (the right plot) QR point sets of lengths $2^{14}, 2^{15}, 2^{16}$ as compared to a plain MC algorithm. The test problem is the pricing of a floating strike put option under the VG model.

As follows from results presented in Fig. 1, the adaptive integration methods do reduce the variance. Both Suave and PWCPDF methods demonstrate their best results for $d = 2$. The efficiency of the Suave method drops down considerably with rise of d from 2 to 8. For $d = 8$, the Suave method increases the variance, so the variance becomes much larger than that of the crude MC method. In this case, the variance reduction factor is very close to zero (less that 1%). The PWCPDF demonstrates more stable performance while increasing d.

In Figures 2 and 3 we present numerical results obtained with QR point sets used in place of pseudorandom numbers. Since Sobol' point sets demonstrated much better results than Faure point sets for all test conditions, we present the results achieved with the adaptive integration methods only with the Sobol' point sets used. The PWCPDF method with QR point sets works better than Suave because of a simpler yet more efficient use of points. For small values of d equal to 1, 2 or 4, they produce similar or better performance results than the plain QMC method. For example, if $d = 2$ and $N = 2^{16}$, the variance reduction factors of the partial QMC, QMC+Suave, QMC+PWCPDF methods are equal to 10.3, 11.7, and 17.4, respectively. For d equal to 8 or 16, the efficiency of the adaptive integration is lower, and the plain QMC method outperforms both Suave and PWCPDF. The best performance of the PWCPDF method is achieved when $d = 8$ and the dimension of QR point sets is 32.

5.2 Modelling with the CEV Process

To generate a discretized path $\mathbf{S}$ of the CEV process we use a combination of Algorithms 3 and 2. By sampling from the noncentral chi-square distribution (using the inverse method) we obtain the skeleton $\mathbf{S}_1$. The variables of $\mathbf{S}_2$ conditional on $\mathbf{S}_1$ are then sampled from the randomized gamma distribution of the second type.

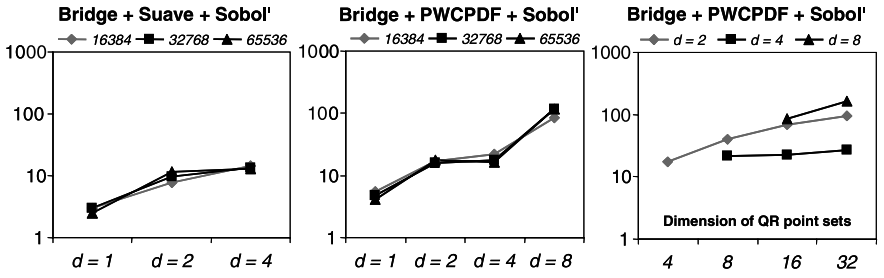


Fig. 3 Variance reduction for the adaptive integration methods with using randomized Sobol' QR point sets of lengths $2^{14}, 2^{15}, 2^{16}$ as compared to a plain MC algorithm. The left and middle plots demonstrate the performance of Suave and PWCPDF methods, respectively. The right plot shows the performance of PWCPDF with varying dimensionality of the QR point sets. The test problem is the pricing of a floating strike put option under the VG model.

For the plain sequential-modeling Monte-Carlo method, Algorithm 1 is the only necessary tool.

Algorithms 1 and 3 can be applied only to the CEV diffusion model without absorption. To obtain such a model from an absorbing diffusion, the modified Bessel function $\mathcal{I}_{\frac{1}{2|\beta|}}$ in the transition PDF (12) has to be replaced by $\mathcal{I}_{\frac{1}{2\beta}}$ where $\beta < -0.5$. This modification of the transition PDF has to be applied to the density function $\mathbf{f}_1$ in the path integral (5) since this function is just a product of consecutive one-dimensional PDFs given by (12) and (14). As a result, $\mathbf{f}_1$ in (5) is replaced by a product of a non-absorbing density function and the weight function W given below:

$$W(\mathbf{S}^1) = \prod_{k=1}^{d} \frac{\mathcal{I}_{\frac{1}{2|\beta|}}(Y_k)}{\mathcal{I}_{\frac{1}{2\beta}}(Y_k)}, \text{ where } Y_k = \frac{e^{\beta v\left(t_{\frac{kn}{d}} - t_{\frac{(k-1)n}{d}}\right)} S_{\frac{kn}{d}}^{-\beta} S_{\frac{(k-1)n}{d}}^{-\beta}}{\delta^2\beta^2\left(\tau(t_{\frac{kn}{d}}) - \tau(t_{\frac{(k-1)n}{d}})\right)}.$$

The weighted estimator is then a product of the sample value of the payoff function $\Lambda(\mathbf{S}^1, \mathbf{S}^2)$ and the weight function $W(\mathbf{S}^1)$. Clearly, the resulting weighted estimator is an unbiased estimator of V and has a finite variance.

We use the following CEV model parameters: $\delta = 2500$, $\beta = -2.0$, $r = 0.05$, and $q = 0$. The spot price is $S_0 = 100$ and the strike price (where applicable) is $K = 100$. The parameter δ is chosen so that the local volatility $\sigma(S)/S$ at the spot S_0 is 25%. The number of time partition points is $n = 128$.

For the CEV model, the bridge sampling and adaptive integration methods slows down the computation considerably (see Fig. 4). Therefore, we have to analyze the computational cost of algorithms rather than the variance reduction only. There are two main reasons for such a slowdown. First, the bridge sampling involves the simulation of the Bessel discrete random variable. The acceptance-rejection

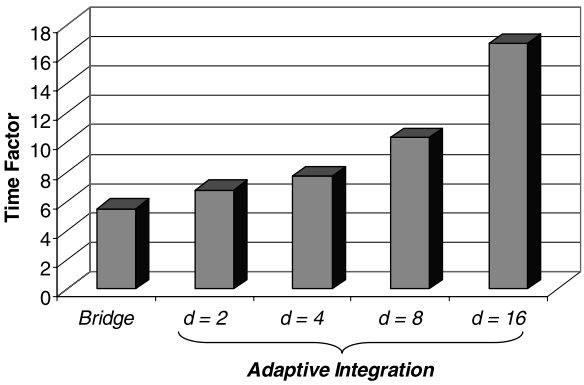


Fig. 4 Slowdown factors for the bridge and hybrid algorithms as compared to a plain sequential Monte Carlo algorithm for a typical problem of option pricing under the CEV process with $n = 128$.

method proposed by Devroye in [6] is not very efficient though faster methods may be developed. Second, the computation of the inverse CDF of the non-central chi-square distribution requires many arithmetic operations. Another possible approach is to sample from the randomized gamma distribution (of the first kind) by successively inverting the Poisson and gamma CDFs. Unfortunately, this method doubles the problem dimensionality (like for the VG model) since every sample value of the asset price process now requires sampling two random variables.

6 Conclusions

In this paper, we present the path-integral decomposition method and demonstrate its applicability to pricing path-dependent options under the two asset pricing models. This method allows us to combine the (R)QMC method and/or (adaptive) variance reduction techniques with the bridge sampling MC method. We refer to the obtained methods as the hybrid methods. Here, we mainly study two variance reduction techniques—the Suave and PWCPDF methods. From the results of our numerical tests, we derive the following conclusions. **(1)** The hybrid methods are more efficient than a crude MCM only if they applied to a problem with a relatively small effective dimension d. The variance reduction reaches its maximum at $d = 2$ or $d = 4$ and rapidly drops down with the increase of d. Interestingly, the PWCPDF method is a bit more efficient than the more complicated adaptive Suave method. **(2)** The partial RQMC method generally provides a larger variance reduction than the adaptive/hybrid methods with the increase of the dimensionality d. The Sobol' QR point sets perform better than those of Faure. **(3)** For models with the very expensive computation of the inverse CDF, hybrid methods may be more efficient than a plain QMC method. The slowdown factor can almost completely remove the variance reduction with the increase of d (so d should be small enough). Therefore, the splitting approach becomes very useful for these models. Notice that one can fur-

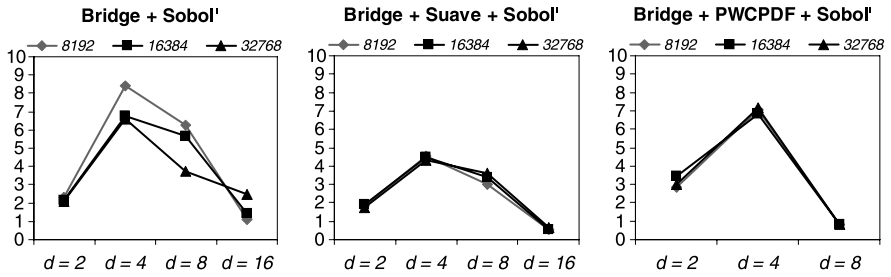


Fig. 5 Computation cost reduction for the partial QMC method (the left plot) and adaptive integration methods (the middle and right plots) with using randomized Sobol' QR point sets of lengths $2^{14}, 2^{15}, 2^{16}$ as compared to a plain MC algorithm. The test problem is the pricing of a floating strike put option under the CEV model.

ther improve the efficiency of hybrid methods by avoiding the computation of the inverse CDF of $\mathbf{S}^1$.

Acknowledgements The author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) for a discovery research grant as well as the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) in providing computational facilities.

References

1. Avramidis, A. N., L'Ecuyer, P.: Efficient Monte Carlo and quasi-Monte Carlo option pricing. Management Science **52**(12), 1930–1944 (2006)
2. Caflisch, R. E., Morokoff, W., Owen, A.: Valuation of mortgage-backed securities using brownian bridges to reduce effective dimension. Journal of Computational Finance **1**, 27–46 (1997)
3. Campolieti, G., Makarov, R.: Pricing path-dependent options on state dependent volatility models with a Bessel bridge. International Journal of Theoretical and Applied Finance **10**(1), 1–38 (2007)
4. Campolieti, G., Makarov, R.: Monte Carlo path integral pricing of Asian options on state dependent volatility models using high performance computing. Quantitative Finance **8**(2), 147–161 (2008)
5. Cox, J.: Notes on option pricing I: Constant elasticity of variance diffusions. Journal of Portfolio Management **22**, 15–17 (1996). Published first as a working paper, Stanford University, 1975
6. Devroye, L.: Simulating Bessel random variables. Statistics and Probability Letters **57**, 249–257 (2002)
7. Hann, T.: Cuba—a library for multidimensional numerical integration. Computer Physics Communications **168**, 78–95 (2005)
8. Hong, H. S., Hickernell, F. J.: Algorithm 823: Implementing scrambled digital sequences. ACM Transactions on Mathematical Software **29**(2), 95–109 (2003)
9. L'Ecuyer, P., Simard, R. J.: Inverting the symmetrical beta distribution. ACM Transactions on Mathematical Software **32**(4), 509–520 (2006)
10. Lepage, G. P.: A new algorithm for adaptive multidimensional integration. Journal of Computational Physics **27**, 192–203 (1978)
11. Madan, D. B., Seneta, E.: The variance gamma (V.G.) model for share market returns. Journal of Business **63**(4), 511–524 (1990)
12. Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P.: Numerical recipes in FORTRAN 77. Cambridge University Press (1992)
13. Yuan, L., Kalbfleisch, J. D.: On the Bessel distribution and related problems. Annals of the Institute of Statistical Mathematics **52**(3), 438–477 (2000)

Monte Carlo Simulation of Stochastic Integrals when the Cost of Function Evaluation Is Dimension Dependent

Ben Niu and Fred J. Hickernell

Abstract In mathematical finance, pricing a path-dependent financial derivative, such as a continuously monitored Asian option, requires the computation of $\mathbb{E}[g(B(\cdot))]$, the expectation of a payoff functional, g, of a Brownian motion, $B(t)$. The expectation problem is an infinite dimensional integration which has been studied in [1], [5], [7], [8], and [10]. A straightforward way to approximate such an expectation is to take the average of the functional over n sample paths, $B_1, \ldots, B_n$. The Brownian paths may be simulated by the Karhunen-Loéve expansion truncated at d terms, $\hat{B}_d$. The cost of functional evaluation for each sampled Brownian path is assumed to be $\mathcal{O}(d)$. The whole computational cost of an approximate expectation is then $\mathcal{O}(N)$, where $N = nd$. The (randomized) worst-case error is investigated as a function of both n and d for payoff functionals that arise from Hilbert spaces defined in terms of a kernel and coordinate weights. The optimal relationship between n and d given fixed N is studied and the corresponding worst-case error as a function of N is derived.

1 Introduction

Infinite-dimensional integration is widely applied, e.g., in mathematical finance and quantum physics, and, moreover, it is used as a computational tool to solve parabolic or elliptic partial differential equations. This paper is motivated by the application in mathematical finance. The price of a financial option can be computed by taking an average over possible movements of the underlying asset prices [12]:

Ben Niu
Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA
e-mail: nben@iit.edu

Fred J. Hickernell
Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA
e-mail: hickernell@iit.edu

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_35,

$$\text{option price} = \mathbb{E}[\text{payoff}].$$

The payoff function depends on the underlying stochastic model for the asset price, $S(t)$, which is governed by a stochastic differential equation, for example, the classical Black-Scholes model takes the form

$$dS(t) = rS(t)\,\mathrm{d}t + \sigma S(t)\,\mathrm{d}B(t), \quad S(t) = S(0)e^{(r-\sigma^2/2)t+\sigma B(t)}.$$

Denoting the option price as μ, an arithmetic mean Asian call option is defined as:

$$\begin{aligned}\text{payoff}(B(\cdot)) &= e^{-rT}\max\left(\frac{1}{T}\int_0^T S(t)\,\mathrm{d}t - K, 0\right),\\ \mu &= \mathbb{E}[\text{payoff}(B(\cdot))], \quad \text{where } B(t) \text{ is a Brownian motion.}\end{aligned}$$

In order to approximate μ, one may simulate n different Brownian sample paths, $B_1, B_2, \ldots$, and then take the sample average, i.e. [2],

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n}\text{payoff}(B_i(\cdot)). \tag{1}$$

One important task is to simulate the Brownian path accurately and efficiently. The most straightforward way is the time discretization methodology based on the independent increment property of the Brownian motion. The time interval $[0, T]$ may be divided into s subintervals, and then $B(t_k)$ at a specific time $t_k = kT/s,\ k = 1, \ldots, s$ is given by [6]:

$$B(t_k; X_1, X_2, \ldots) = \sqrt{\frac{T}{s}}(X_1 + X_2 + \cdots + X_k), \tag{2}$$

where $X_1, X_2, \ldots$ are i.i.d. $N(0, 1)$.

An alternative approach is the Karhunen-Loéve expansion, which is used in [11] for financial option pricing. By solving the eigenvalue problem of the covariance operator of $B(t)$, i.e., $\mathrm{cov}(B(t), B(s)) = \min(t, s)$, the Brownian motion $B(t)$ can be expanded as an infinite series:

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

where $X_1, X_2, \ldots$ are i.i.d. $N(0, 1)$. In practical computation, the Karhunen-Loéve expansion is truncated at a finite dimension, d. The i^{th} approximation of $B(t)$ to be used in (1) is

$$B_i(t) \approx \hat{B}_d(t; x_{i,1}, \ldots, x_{i,d}) = \sqrt{2T}\sum_{j=1}^{d} x_{i,j} \frac{\sin\left(\left(j-\frac{1}{2}\right)\pi t/T\right)}{\left(j-\frac{1}{2}\right)\pi}, \tag{3}$$

where $\{\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}) \in \mathbb{R}^d\}_{i=1}^n$ might a be simple random sequence, grid, centroidal Voronoi tessellation, Latin hypercube, or low discrepancy sequence. The truncated Karhunen-Loéve expansion corresponds formally setting, $x_{i,d+1} = x_{i,d+2} = \cdots = 0$. The aim of this article is to investigate how the choice of n, d, and $\{\mathbf{x}_i\}_{i=1}^n$ together all affect the accuracy of $\hat{\mu}$. Here, possible errors in evaluating the payoff are ignored, e.g., for the continuously monitored Asian option, the possible errors in integrating a stock path over $[0, T]$ are ignored.

In Section 2, a simple example is shown to illustrate the general idea of the algorithm, and the tradeoff between choosing n large or d large. Section 3 provides the worst-case error of the approximation, and also deduces the optimal choice of n and d given a computational cost budget $N = nd$, as done in [1], [7] and [8]. In Section 4, a numerical experiment involving the computation of a continuously monitored geometric mean Asian option price is presented.

2 Illustrative Example

Section 1 describes the problem in the option pricing setting. Here a more general case is addressed. The payoff function is replaced by g, i.e., payoff$(B(\cdot)) = g(B(\cdot, X_1, X_2, \ldots))$. The aim is to evaluate $\mu = \mathbb{E}[g(B(\cdot; X_1, X_2, \ldots))]$, where in the discussion above μ is the option price. The approximation of μ takes the form of an equally weighted sample average:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} g\left(\hat{B}_d(\cdot, ; x_{i,1}, \ldots, x_{i,d})\right). \tag{4}$$

Before developing a general theory for this problem, a simple example is given to illustrate how the error of $\hat{\mu}$ depends both on n and d and how the error analysis is facilitated by splitting the error into two parts.

Example 1. Evaluate $\mathbb{E}\left[\int_0^1 B^2(t)\,\mathrm{d}t\right]$ using (4). In this case

$$g\left(B(\cdot; X_1, X_2, \ldots)\right) = \int_0^1 [B(\cdot; X_1, X_2, \ldots)]^2\,\mathrm{d}t = \sum_{j=1}^{\infty} \frac{X_j^2}{(j - \frac{1}{2})^2 \pi^2},$$

$$\mu = \mathbb{E}[g(B(\cdot; X_1, X_2, \ldots))] = \sum_{j=1}^{\infty} \frac{\mathbb{E}[X_j^2]}{(j - \frac{1}{2})^2 \pi^2} = \sum_{j=1}^{\infty} \frac{1}{(j - \frac{1}{2})^2 \pi^2} = \frac{1}{2}.$$

The first d terms in the series for μ can be identified as

$$\begin{aligned} \mu_d &= \mathbb{E}[g(B(\cdot; X_1, X_2, \ldots))|X_{d+1} = X_{d+2} = \cdots = 0] \\ &= \sum_{j=1}^{d} \frac{\mathbb{E}[X_j^2]}{(j - \frac{1}{2})^2 \pi^2} = \sum_{j=1}^{d} \frac{1}{(j - \frac{1}{2})^2 \pi^2}. \end{aligned}$$

The expression μ_d is introduced here to facilitate splitting the error. From (4), the estimator for μ is

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} g(\hat{B}_d(\cdot; x_{i,1}, \ldots, x_{i,d})) = \sum_{j=1}^{d} \frac{\frac{1}{n}\sum_{i=1}^{n} x_{i,j}^2}{(j-\frac{1}{2})^2\pi^2}.$$

Then, the error of approximating μ can be written as a sum of two parts:

$$\begin{aligned}\overbrace{\mu - \hat{\mu}}^{\text{error}} &= \overbrace{\mu - \mu_d}^{\text{truncated expansion error}} + \overbrace{\mu_d - \hat{\mu}}^{\text{finite sample error}} \\ &= \underbrace{\sum_{j=d+1}^{\infty} \frac{1}{\left(j-\frac{1}{2}\right)^2\pi^2}}_{\text{independent of design}} + \underbrace{\sum_{j=1}^{d} \frac{\mathbb{E}[X_j^2] - \frac{1}{n}\sum_{i=1}^{n} x_{i,j}^2}{\left(j-\frac{1}{2}\right)^2\pi^2}}_{\text{independent of } X_{d+1},\ldots}.\end{aligned}$$

The first term is independent of the design $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, whereas the second term is independent of the coordinates after dimension d.

Simple Monte Carlo Sampling

If $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a set of d-dimensional i.i.d. standard normal random variables, the explicit mean square error (MSE) can be written as a sum of the squared bias and the variance:

$$\begin{aligned}\mathrm{MSE}(\hat{\mu}) &= \mathbb{E}\left(\mu - \hat{\mu}\right)^2 = (\mu - \mu_d)^2 + \mathbb{E}(\mu_d - \hat{\mu})^2 \\ &= \underbrace{\left(\sum_{j=d+1}^{\infty} \frac{1}{\left(j-\frac{1}{2}\right)^2\pi^2}\right)^2}_{\text{bias}^2} + \underbrace{\frac{1}{n}\sum_{j=1}^{d} \frac{2}{\left(j-\frac{1}{2}\right)^4\pi^4}}_{\text{variance}}.\end{aligned}$$

The bias depends primarily on the truncated dimension d, while the variance depends primarily on the sample size n. They can be approximated by

$$\text{bias} = \sum_{j=d+1}^{\infty} \frac{1}{\pi^2(j-1/2)^2} \sim \int_d^{\infty} \frac{1}{\pi^2 x^2}\,\mathrm{d}x = \frac{1}{\pi^2 d}, \quad \text{as } d \to \infty.$$

$$\begin{aligned}\text{variance} &= \frac{2}{n}\sum_{j=1}^{d} \frac{1}{\pi^4(j-1/2)^4} = \frac{2}{n}\frac{1}{\pi^4}\Big[\sum_{j=1}^{\infty} \frac{1}{(j-1/2)^4} - \sum_{j=d+1}^{\infty} \frac{1}{(j-1/2)^4}\Big] \\ &\sim \frac{2}{n}\frac{1}{\pi^4}\left(\frac{\pi^4}{6} - \int_d^{\infty} \frac{1}{x^4}\,\mathrm{d}x\right) = \frac{1}{3n}\left(1 - \frac{2}{\pi^4 d^3}\right), \quad \text{as } d \to \infty.\end{aligned}$$

Hence, the root mean square error (RMSE) is:

$$\mathrm{RMSE}(\hat{\mu}) \sim \sqrt{\frac{1}{\pi^4 d^2} + \frac{1}{3n}\left(1 - \frac{2}{\pi^4 d^3}\right)} \sim \sqrt{\frac{1}{\pi^4 d^2} + \frac{1}{3n}}, \quad \text{as } d \to \infty.$$

Quasi-Monte Carlo Sampling

If $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a set of d-dimensional vectors, whose components are the inverse normal transformations of a low discrepancy sequence, e.g., a scrambled Sobol' sequence, it is technically more difficult to get an explicit expression for the variance, however, the bias is the same. It can be observed from the result of numerical experiments that

$$\mathrm{RMSE}(\hat{\mu}) \sim \sqrt{\frac{1}{\pi^4 d^2} + \frac{C}{n^2}}, \quad \text{as } d \to \infty.$$

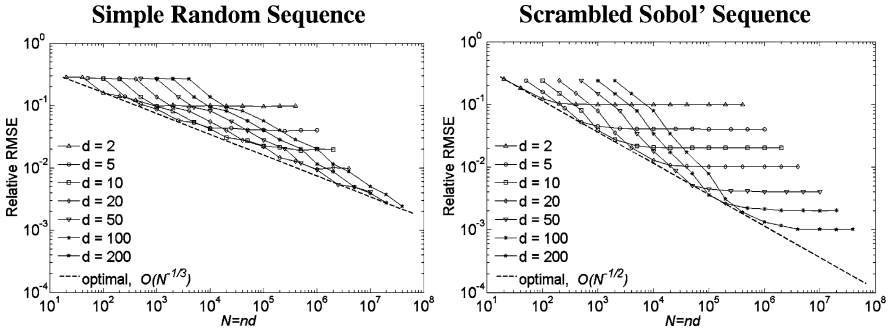


Fig. 1 The relative root mean square error (RMSE) and the empirical optimal convergence rate.

Figure 1 shows that for small, fixed values of d, the relative RMSE converges very quickly to the limiting value, i.e., the bias. For large values of d, the relative RMSE is dominated by the sampling error for moderate values of n. The errors obtained using the optimal n and d for a given $N = nd$ are obtained empirically. The optimal convergence rate in the simple random sequence case is $\mathcal{O}(N^{-1/3})$, whereas the scrambled Sobol' sequence can achieve a superior $\mathcal{O}(N^{-1/2})$ convergence rate. These empirical orders of convergence, $\beta = 1/3$ and $1/2$ are determined by the formula

$$\beta = \underset{b}{\mathrm{argmax}}\, C(b), \quad \text{where } C(b) = \max\{C : C\,(N/N_0)^{-b} \le \mathrm{RMSE}(N)\ \forall N\}$$

The value of N_0 corresponds to the midpoint of the values of N considered in the numerical experiments.

In the derivation above, the approximation error of μ is conveniently split into two parts, i.e., the truncated expansion error, and the finite sample error. This split is exploited in the following section where a worst-case error bound for a general functional is derived.

3 Worst-Case Error Bound

The problem is to measure how well $\mu = \mathbb{E}[g(B(\cdot; X_1, X_2, \ldots))]$ is approximated by $\hat{\mu}$. Here a functional f is defined by $f(X_1, X_2, \ldots) = g(B(\cdot; X_1, X_2, \ldots))$. Hence, $\mu = \mathbb{E}[f(X_1, X_2, \ldots)]$, where $(X_1, X_2, \ldots)$ is an i.i.d. random sequence with common probability density function $\rho_1(\cdot)$. The support of ρ_1 is assumed to be $\overline{I}$, where I is some open, half-open, or closed interval, which may be finite, semi-infinite or infinite. For example, ρ_1 might be the uniform density on $\overline{I} = [0, 1]$ or the standard normal density, which is defined on $\overline{I} = \mathbb{R}$. The domain of f can be considered to be $I^{\mathbb{N}}$, where $\mathbb{N}$ is the set of natural numbers. Here $I^{\mathbb{N}}$ is the set of infinite sequences whose elements lie in I.

The functional f, which depends on a countably infinite number of variables, is constructed as the countable sum of functions of a finite number of variables. The Hilbert space containing f is likewise constructed as the tensor product space of a countable number of reproducing kernel Hilbert spaces. See similar work in [7] and [10].

Let $K_\emptyset = 1$, and so the reproducing kernel Hilbert space $\mathcal{H}(K_\emptyset)$, with the reproducing kernel $K_\emptyset$ is the Hilbert space of constant functionals, i.e., $\langle f_\emptyset, g_\emptyset \rangle_{\mathcal{H}(K_\emptyset)} = f_\emptyset g_\emptyset$ for all $f_\emptyset, g_\emptyset \in \mathcal{H}(K_\emptyset)$.

Next, let K_1 be a finite valued, symmetric, positive semi-definite kernel on $I \times I$, i.e.,

$$K_1(x, y) = K_1(y, x), \quad \forall x, y \in I,$$

$$\sum_{i,j=1}^{n} b_i K_1(x_i, x_j) b_j \geq 0, \quad \forall \mathbf{b} \in \mathbb{R}^n, \ \forall x_1, \ldots, x_n \in I, \ \forall n \in \mathbb{N}.$$

Moreover, let there be some anchor $c \in I$ such that

$$K_1(x, c) = 0, \quad \forall x \in I. \tag{5a}$$

This means that the K_1 is the reproducing kernel for a Hilbert space, $\mathcal{H}(K_1)$, of functions on I that vanish at c, and the only constant function in $\mathcal{H}(K_1)$ is the zero function. An example is $I = \overline{I} = \mathbb{R}$, $c = 0$ and $K_1(x, y) = \frac{1}{2}|x| + \frac{1}{2}|y| - \frac{1}{2}|x - y|$. Further assumptions are made on the finiteness of K_1 and its integrability with respect to the probability density ρ_1, namely,

$$h_1(x) := \int_{\overline{I}} K_1(x, y)\, \rho_1(y)\, \mathrm{d}\, y \quad \forall x \in I, \tag{5b}$$

$$h_1 \in \mathcal{H}(K_1), \tag{5c}$$

$$m := \int_{\overline{I}^2} K_1(x,y)\,\rho_1(x)\rho_1(y)\,\mathrm{d}x\,\mathrm{d}y < \infty, \tag{5d}$$

$$M := \int_{\overline{I}} K_1(x,x)\rho_1(x)\,\mathrm{d}x < \infty. \tag{5e}$$

For the example of K_1 mentioned above and ρ_1, the standard normal density function, these conditions are satisfied. The distinction between I and $\overline{I}$ above means that the evaluation of $f \in \mathcal{H}(K_1)$ at any point $x \in I$ is a bounded functional, whereas the evaluation of f at $x \in \overline{I} \setminus I$ may be an unbounded functional.

The kernel K_1 is the building block used to construct the reproducing kernel for Hilbert spaces of functions of several variables. Let $\mathbf{c} = (c, c, \ldots)$. In the previous examples, $\mathbf{c}$ was chosen to be $\mathbf{0}$. Also, let $1:d$ denote the set $\{1, 2, \ldots, d\}$. For any $u \subset \mathbb{N}$, let $|u|$ denote its cardinality. Define $\mathbb{U}$ as the set of subsets of $\mathbb{N}$ with finite cardinality, i.e.,

$$\mathbb{U} = \{u \subset \mathbb{N} : |u| < \infty\}.$$

Furthermore, let $\mathbf{x}_u$ denote the vector containing the coordinates of $\mathbf{x}$ whose indices are in u. Let $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots)$ be a sequence of non-increasing non-negative weights that satisfy the following summability condition:

$$\sum_{j=d}^{\infty} \gamma_j < \alpha d^{-2q}, \quad \gamma_1 \geq \gamma_2 \geq \cdots \geq 0, \tag{6}$$

for some positive constants α and q. Given any set $u \in \mathbb{U}$, the symmetric, positive semi-definite kernel

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

defines the Hilbert space $\mathcal{H}(K_u)$ of functions of $|u|$ variables, and these functions vanish if any one or more of the coordinates is set to c. The domain of the functions in $\mathcal{H}(K_u)$ can be denoted as I^u. It is also, possible to think of $\mathcal{H}(K_u)$ as a Hilbert space of functionals defined on $I^{\mathbb{N}}$ that are constant with respect to the variables x_j with $j \notin u$.

Following the argument in [5], it can be shown that $\mathcal{H}(K_u) \bigcap \mathcal{H}(K_v) = \{0\}$ for any $u \neq v$ because $\mathcal{H}(1) \bigcap \mathcal{H}(K_1) = \{0\}$. An outline of this argument is as follows. Let $f \in \mathcal{H}(K_u) \bigcap \mathcal{H}(K_v)$, and without loss of generality, assume that there exists a $j \in u \setminus v$. Since $f \in \mathcal{H}(K_u)$, it follows that $f(\mathbf{x}) = 0$ for all $\mathbf{x} \in I^{u \cup v}$ with $x_j = c$. On the other hand, since $f \in \mathcal{H}(K_v)$, it follows that f does not depend on the value of x_j, so $f(\mathbf{x}) = 0$ for all $\mathbf{x} \in I^{u \cup v}$.

The Hilbert space of functionals on $I^{\mathbb{N}}$ is now defined as an infinite sum of functions from the reproducing kernel Hilbert spaces $\mathcal{H}(K_u)$ using the approach introduced by [7]. For any sequence of functions $\{f_u\}_{u \in \mathbb{U}}$ with $f_u \in \mathcal{H}(K_u)$, define $f \in \mathcal{H}(K)$ as a sum of its pieces, sometimes called effects in the statistics literature:

$$f = \sum_{u \in \mathbb{U}} f_u,$$

and define the inner product on this Hilbert space as

$$\langle f, g\rangle_{\mathcal{H}(K)} = \sum_{u\in\mathbb{U}} \langle f_u, g_u\rangle_{\mathcal{H}(K_u)}.$$

Because $\mathcal{H}(K_u)$ and $\mathcal{H}(K_v)$ have only the zero function in common for $u \neq v$, they are orthogonal subspaces of $\mathcal{H}(K)$, i.e.,

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

The kernel, K, referred to here is formally defined as

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

The infinite product defining $K(\mathbf{x}, \mathbf{y})$ is not necessarily finite for all $\mathbf{x}, \mathbf{y} \in I$, particularly for unbounded K_1, and for that reason, the kernel K is not necessarily a reproducing kernel. Likewise, $\mathcal{H}(K)$ is not necessarily a reproducing kernel Hilbert space, since function evaluation may not be a bounded functional at every point in $I^{\mathbb{N}}$.

However, for certain points $\mathbf{y}$, function evaluation is bounded. Specifically, consider point of the form $\mathbf{y} = (\mathbf{y}_u, \mathbf{c}) \in I^{\mathbb{N}}$ where $u \in \mathbb{U}$, i.e., $y_j = c$ for $j \notin u$. By the definition of the Hilbert spaces $\mathcal{H}(K_v)$ it follows that $f_v(\mathbf{y}_u, \mathbf{c})$ vanishes for $v \not\subseteq u$, and so $f(\mathbf{y}_u, \mathbf{c}) = \sum_{v\subseteq u} f_v(\mathbf{y}_v)$. Since this sum is finite, one may write

$$\begin{aligned} f(\mathbf{y}_u, \mathbf{c}) &= \sum_{v\subseteq u} f_v(\mathbf{y}_v) = \sum_{v\subseteq u} \langle K_v(\cdot, \mathbf{y}_v), f_v\rangle_{\mathcal{H}(K_v)} \\ &= \sum_{v\subseteq u} \langle K_v(\cdot, \mathbf{y}_v), f_v\rangle_{\mathcal{H}(K_v)} + \sum_{\substack{v\in\mathbb{U}\\ v\not\subseteq u}} \langle 0, f_v\rangle_{\mathcal{H}(K_v)} \\ &= \left\langle \sum_{v\subseteq u} K_v(\cdot, \mathbf{y}_v), \sum_{v\in\mathbb{U}} f_v \right\rangle_{\mathcal{H}(K)} = \left\langle \sum_{v\subseteq u} \prod_{j\in v} \gamma_j K_1(\cdot, y_j), f \right\rangle_{\mathcal{H}(K)} \\ &= \left\langle \prod_{j\in u} [1 + \gamma_j K_1(\cdot, y_j)], f \right\rangle_{\mathcal{H}(K)} = \langle K(\cdot, (\mathbf{y}_u, \mathbf{c})), f\rangle_{\mathcal{H}(K)}. \end{aligned} \tag{7}$$

One can claim that $|f(\mathbf{y}_u, \mathbf{c})| < \infty$ since

$$|f(\mathbf{y}_u, \mathbf{c})| = \left|\langle K(\cdot, (\mathbf{y}_u, \mathbf{c})), f\rangle_{\mathcal{H}(K)}\right| \leq \|K(\cdot, (\mathbf{y}_u, \mathbf{c}))\|_{\mathcal{H}(K)} \|f\|_{\mathcal{H}(K)}.$$

In addition, (6) implies that

$$\|K(\cdot, (\mathbf{y}_u, \mathbf{c}))\|^2_{\mathcal{H}(K)} = K((\mathbf{y}_u, \mathbf{c}), (\mathbf{y}_u, \mathbf{c})) = \prod_{j\in u} \left[1 + \gamma_j K_1(y_j, y_j)\right]$$

$$\leq \prod_{j\in u}\left[1+\gamma_j L\right] = \prod_{j\in u}\exp\left(\log(1+\gamma_j L)\right) \leq \exp\left(L\sum_{j\in u}\gamma_j\right) < \infty, \tag{8}$$

where $L = \max_{j\in u} K_1(y_j, y_j)$.

Thus, $K(\cdot, (\mathbf{y}_u, \mathbf{c}))$ is the representer for function evaluation at $(\mathbf{y}_u, \mathbf{c})$. From the formula for $f(\mathbf{y}_u, \mathbf{c})$ in (7), one can recursively write the effects or pieces of f in terms of the anchor $\mathbf{c}$ as in [5]:

$$f_\emptyset = f(\mathbf{c}), \quad f_u(\mathbf{y}_u) = f(\mathbf{y}_u, \mathbf{c}) - \sum_{v\subset u} f_v(\mathbf{y}_v), \quad u \in \mathbb{U}.$$

From the argument above, the representer for function evaluation at $(\mathbf{y}_{1:d}, \mathbf{c}_{d+1:\infty})$ is $K^{(d)}(\cdot, \mathbf{y}_{1:d})$, where $K^{(d)}$ is defined by

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

The orthogonality of the subspaces $\mathcal{H}(K_u)$ and $\mathcal{H}(K_v)$ for $u \neq v$ implies that

$$f_1(\cdot, \mathbf{c}_{d+1:\infty}) \perp f_2 - f_2(\cdot, \mathbf{c}_{d+1:\infty}), \quad \forall f_1, f_2 \in \mathcal{H}(K), \tag{9}$$

a fact used later to split the error of approximating the integral into two parts. Note that the Monte Carlo type estimator for the expectation, $\hat{\mu}$, based on $\{\mathbf{x}_{i,1:d}\}_{i=1}^n \subset I^d$ can be represented as an inner product in $\mathcal{H}(K)$:

$$\begin{aligned} \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_{i,1:d}, \mathbf{c}) &= \left\langle \frac{1}{n}\sum_{i=1}^{n} K^{(d)}(\cdot, \mathbf{x}_{i,1:d}), f \right\rangle_{\mathcal{H}(K)} \\ &= \left\langle \frac{1}{n}\sum_{i=1}^{n} K\left(\cdot, (\mathbf{x}_{i,1:d}, \mathbf{c})\right), f \right\rangle_{\mathcal{H}(K)}. \end{aligned} \tag{10}$$

Having defined the Hilbert space of functionals defined in $I^{\mathbb{N}}$, it is now possible to define the expectation of such functionals, f. This expectation is constructed in terms of the expectations of pieces of f. For any $u \in \mathbb{U}$, $\mathbb{E}[f_u(\mathbf{X}_u)]$ is a $|u|$-dimensional integral, which may represented as

$$\begin{aligned} \mathbb{E}[f_u(\mathbf{X}_u)] &= \int_{I^u} f_u(\mathbf{x}_u)\prod_{j\in u}\rho_1(x_j)\,\mathrm{d}\mathbf{x}_u = \langle h_u, f_u\rangle_{\mathcal{H}(K_u)} \\ &= \langle h_u, f\rangle_{\mathcal{H}(K)}, \quad \forall f_u \in \mathcal{H}(K_u),\ f \in \mathcal{H}(K), \\ h_u(\mathbf{x}_u) &= \int_{I^{|u|}} K_u(\mathbf{x}_u, \mathbf{y}_u)\Big(\prod_{j\in u}\rho_1(y_j)\Big)\,\mathrm{d}\mathbf{y}_u \\ &= \int_{I^{|u|}} \Big(\prod_{j\in u}\gamma_j K_1(x_j, y_j)\rho_1(y_j)\Big)\,\mathrm{d}\mathbf{y}_u = \prod_{j\in u}\gamma_j h_1(x_j) \in \mathcal{H}(K_u). \end{aligned}$$

The formula for h_u follows from the reproducing property of the kernel K_u, and the fact that $h_u \in \mathcal{H}(K_u)$ follows from assumption (5c) on h_1. The d-dimensional approximate expectation μ_d can be represented as

$$\begin{aligned}\mu_d &= \mathbb{E}[f(X_1,\dots,X_d,\mathbf{c})] = \mathbb{E}[f(\mathbf{X})|X_{d+1:\infty}=\mathbf{c}]\\ &= \sum_{u\subseteq 1:d}\mathbb{E}[f_u(\mathbf{X}_u)] = \sum_{u\subseteq 1:d}\mu_u = \sum_{u\subseteq 1:d}\langle h_u, f\rangle_{\mathcal{H}(K)}\\ &= \Big\langle \sum_{u\subseteq 1:d} h_u, f\Big\rangle_{\mathcal{H}(K)} = \Big\langle h^{(d)}, f\Big\rangle_{\mathcal{H}(K)},\\ h^{(d)}(\mathbf{x}_{1:d}) &= \sum_{u\subseteq 1:d} h_u(\mathbf{x}_u) = \prod_{j=1}^{d}\big[1+\gamma_j h_1(x_j)\big],\\ \|h^{(d)}\|^2_{\mathcal{H}(K)} &= \sum_{u\subseteq 1:d}\|h_u\|^2_{\mathcal{H}(K)} = \prod_{j=1}^{d}[1+\gamma_j m] < \infty\end{aligned}$$

by (6). The finiteness of $\|h^{(d)}\|^2_{\mathcal{H}(K)}$ is justified by the same argument as in (8).

The expectation of the whole functional is defined as the countable sum of the expectations of the parts, namely,

$$\mu = \mathbb{E}[f(X_1,X_2,\dots)] := \sum_{u\in\mathbb{U}}\langle h_u, f\rangle_{\mathcal{H}(K)} = \langle h, f\rangle_{\mathcal{H}(K)}, \tag{11}$$

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

$$\|h\|^2_{\mathcal{H}(K)} = \sum_{u\in\mathbb{U}}\|h_u\|^2_{\mathcal{H}(K)} = \prod_{j=1}^{\infty}[1+\gamma_j m] \le e^{\alpha m} < \infty. \tag{12}$$

By definition and the summability condition on $\boldsymbol{\gamma}$ in (6), it follows that $h \in \mathcal{H}(K)$, and so it represents a bounded linear functional on $\mathcal{H}(K)$, namely, the expectation. Note that $h^{(d)}(\mathbf{x}_{1:d}) = h(\mathbf{x}_{1:d}, \mathbf{c}_{d+1:\infty})$, so it follows that

$$\mu_d = \Big\langle h^{(d)}, f\Big\rangle_{\mathcal{H}(K)} = \langle h(\cdot, \mathbf{c}_{d+1:\infty}), f\rangle_{\mathcal{H}(K)}. \tag{13}$$

It can be shown by Theorem 1 below that $\lim_{d\to\infty}\mu_d = \mu$, since the worst-case bias in (16) vanishes as $d \to \infty$.

Note that $\hat{\mu}, \mu$ and μ_d defined in (10), (11) and (13) together all depend on f. In the discussion that follows, this f dependence is sometimes written explicitly. In light of the definitions and derivations above, it is now possible to prove the following worst case error bound for the approximation of the infinite dimensional integral (expectation) by a finite sum of a finite dimensional approximation to the functional.

Theorem 1. *Suppose that K_1 is a symmetric, real-valued, positive semi-definite kernel function defined on I^2 that satisfies the assumptions (5). Here I may be a finite semi-infinite, or infinite interval. Consider a Hilbert space $\mathcal{H}(K)$ of functionals*

$f : I^{\mathbb{N}} \to \mathbb{R}$, which is defined above in terms of K_1 and the weights γ_j satisfying assumptions (6). Then the worst-case error for approximating the expectation of these functionals by a Monte Carlo type algorithm is

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

where

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

Proof. The error $\mu - \hat{\mu}$ can be written explicitly as a sum of two inner products, using the expressions derived in (10), (11), and (13):

$$\begin{aligned} \overbrace{\mu - \hat{\mu}}^{\text{error}} &= \overbrace{\mu - \mu_d}^{\text{truncated expansion error}} + \overbrace{\mu_d - \hat{\mu}}^{\text{finite sample error}} \\ &= \langle h - h(\cdot, \mathbf{c}_{d+1:\infty}), f \rangle_{\mathcal{H}(K)} \\ &\quad + \Big\langle h(\cdot, \mathbf{c}_{d+1:\infty}) - \frac{1}{n} \sum_{i=1}^{n} K(\cdot, (\mathbf{x}_{i,1:d}, \mathbf{c})), f \Big\rangle_{\mathcal{H}(K)}. \end{aligned} \tag{14}$$

The two functionals on the left sides of the inner products in (14) are orthogonal by (9). This orthogonality, along with the Pythagorean theorem allows one to compute a tight error bound as follows:

$$\begin{aligned} \sup_{\|f\|_{\mathcal{H}(K)} \le 1} |\mu(f) - \hat{\mu}(f)|^2 &= \|h - h(\cdot, \mathbf{c}_{d+1:\infty})\|_{\mathcal{H}(K)}^2 \\ &\quad + \Big\| h(\cdot, \mathbf{c}_{d+1:\infty}) - \frac{1}{n} \sum_{i=1}^{n} K(\cdot, (\mathbf{x}_{i,1:d}, \mathbf{c})) \Big\|_{\mathcal{H}(K)}^2. \end{aligned} \tag{15}$$

The first of these terms is identified as the squared bias, because it does not vanish even as the sample size tends to infinity:

$$\begin{aligned} \text{worst-bias}^2(d; K) &:= \|h - h(\cdot, \mathbf{c}_{d+1:\infty})\|_{\mathcal{H}(K)}^2 = \|h\|_{\mathcal{H}(K)}^2 - \|h(\cdot, \mathbf{c}_{d+1:\infty})\|_{\mathcal{H}(K)}^2 \\ &= \prod_{j=1}^{d} [1+\gamma_j m] \Big[\prod_{j=d+1}^{\infty} [1+\gamma_j m] - 1 \Big]. \end{aligned} \tag{16}$$

The second term in (15), which is the worst-case error for approximating a d-dimensional integral by a sample average of function values, is the squared discrepancy, which has arisen in many similar error analysis, e.g., [3]. □

How fast d goes to ∞ affects how fast the bias vanishes. From the summability condition on the weights γ_j in (6) it follows that

$$\prod_{j=d+1}^{\infty}(1+\gamma_j m) = \exp\Big(\sum_{j=d+1}^{\infty}\log(1+\gamma_j m)\Big) \le \exp\Big(\sum_{j=d+1}^{\infty}\gamma_j m\Big) \le e^{\alpha m(d+1)^{-2q}}.$$

From the above and (12), the squared worst-case bias satisfies

$$\text{worst-bias}^2(d;K) \le e^{\alpha m}\left(e^{\alpha m(d+1)^{-2q}}-1\right) \le \frac{C_1^2}{d^{2q}}, \tag{17}$$

where C_1 is some constant.

Next, we investigate the mean square discrepancies for the simple i.i.d. random sequence in the randomized worst-case setting and the discrepancy for the low discrepancy sequence in the worst-case setting.

Simple Monte Carlo Sampling:

For simple Monte Carlo sampling, the mean square discrepancy is given by [3],

$$\mathbb{E}\Big[\mathcal{D}^2\left(\{\mathbf{x}_{i,1:d}\}_{i=1}^n; K^{(d)}\right)\Big] = \frac{1}{n}\left\{\prod_{j=1}^{d}[1+\gamma_j M]-\prod_{j=1}^{d}[1+\gamma_j m]\right\} \le \frac{C_2^2}{n},$$

where $C_2^2 = \prod_{j=1}^{\infty}[1+\gamma_j M]-\prod_{j=1}^{\infty}[1+\gamma_j m]$. Although the convergence rate is relatively slow, it is dimension independent. Combining this with (17), the randomized squared worst-case error satisfies

$$\mathbb{E}\Big[\sup_{\|f\|_{\mathcal{H}(K)}\le 1}\big|\hat{\mu}(f)-\mu(f)\big|\Big]^2 \le \frac{C_1^2}{d^{2q}}+\frac{C_2^2}{n}.$$

Quasi-Monte Carlo Sampling:

Results on strong tractability, e.g., [13], show that it is possible to obtain

$$\mathcal{D}\left(\{\mathbf{x}_{i,1:d}\}_{i=1}^n; K^{(d)}\right) \le \frac{C_2}{n^p},$$

for good designs $\{\mathbf{x}_{i,1:d}\}$ with $\gamma_j \to \infty$ fast enough. Here C_2 is a constant independent of d and n but dependent on $\boldsymbol{\gamma}$. This implies that the squared worst-case error for the low discrepancy sequence is

$$\sup_{\|f\|_{\mathcal{H}(K)}\le 1} \left|\hat{\mu}(f)-\mu(f)\right|^2 \le \frac{C_1^2}{d^{2q}}+\frac{C_2^2}{n^{2p}}. \tag{18}$$

The value of p is determined by the specific sequence used. Table 1 gives some values for p depending on q for rank-1 lattices and the Niederreiter (T,s) sequence [9, Sec. 4.5]. The simple random i.i.d sequence is a specific choice with $p=1/2$.

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

Design	p
Simple random sequence	1/2
Rank-1 lattices [4]	$\min\left(1, q+\frac{1}{2}\right)-\varepsilon$
Niederreiter (T,s)-sequence $(q>1/2)$ [14]	$\min\left(1, \frac{q}{2}+\frac{1}{4}\right)-\varepsilon$

The explicit upper error bounds for the simple random sequence and the low discrepancy sequence lead to the minimization of the worst-case error, with which an optimal relationship between the sample size, n, and the truncated dimension, d, can be derived. The cost of the approximation algorithm is $N=nd$, as in the fixed subspace sampling cost model from [1].

By minimizing the upper bound in (18) with respect to d and n, given a budget of $N=nd$, it is found that d should be chosen as

$$\begin{aligned} d &= \left\lceil \sqrt{\frac{q}{p}}\frac{C_1}{C_2} n^p \right\rceil^{\frac{1}{q}} = \mathcal{O}\left(n^{\frac{p}{q}}\right), \\ \min_{\substack{n,d \\ nd=N}} \text{worst-err}(\mathbf{x}_{i,1:d}; K) &\le \left[\left(\frac{q}{p+q}\right)^{-q/2} C_1^p C_2^r N^{-pq} \right]^{\frac{1}{p+q}} \\ &= \mathcal{O}\left(N^{\frac{-pq}{p+q}}\right). \end{aligned}$$

The above equation describes how to choose d as a function of p,q. The value of p is commonly determined by the smoothness of the functional and the quality of the designs as shown in Table 1. The choice of q depends on how fast the weight γ_j decays to 0. Its choice for practical applications is still an open question. Note that for Example 1, $q=1$ and $p=1/2$ (simple Monte Carlo) or 1 (quasi-Monte Carlo). This yields values of $\beta=pq/(p+q)$ corresponding to $1/3$ for simple Monte Carlo sampling and $1/2$ for quasi-Monte Carlo sampling, as observed empirically.

4 Numerical Experiment

Section 1 briefly describes the simulation algorithm for option pricing. Here, a continuously monitored geometric mean Asian option is priced. The payoff of the option is determined by the geometric mean value of the stock price among the life of

the option. The payoff function of options involves the max function, which typically make the integrand non-smooth. Finding a proper kernel K and corresponding Hilbert space to match realistic option payoff functionals is an open problem, even in the case of discretely monitored options (finite d). However, most kernels used in the analysis of quasi-Monte Carlo algorithm assume moderate smoothness.

The numerical experiment given here shows how both d and n affect the approximation error. The explicit analytical formula for the geometric mean Asian option makes the exact computation of the error possible. The option price and payoff function under the Black-Scholes model are:

$$\text{option price} = \mathbb{E}[\text{payoff}(B(\cdot))], \text{ where } B(t) \text{ is a Brownian motion, and}$$

$$\text{payoff} = e^{-rT} \max\left(\exp\left(\frac{1}{T}\int_0^T \log\left(S(0)e^{(r-\sigma^2/2)t+\sigma B(t)}\right) \mathrm{d}t\right) - K, 0\right).$$

The parameters of the model are: $S(0) = 100, K = 100, T = 1, r = 0.03, \sigma = 0.3$. The simulation algorithm for this pricing problem is from (4). To compare the Karhunen-Loéve expansion and the discrete time algorithm for generating Brownian motions, their values at times $kT/s, k = 1, \ldots, s$ are generated by both methods. These values are then used to approximate the payoff function. A mid-point rule is used to approximate the integral with respect to time above. A choice of $s = 52$ corresponds to weekly sampling and makes the error of approximating the integral with respect to time negligible compared to the other errors for n up to at least 10^5 and d up to at least 6. This can be seen from Figure 2 where the relative RMSE continues to decrease as n increases for $d = 6$.

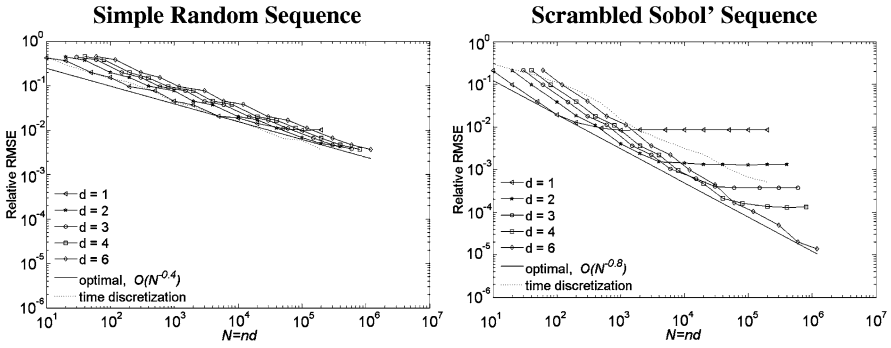


Fig. 2 The relative root mean square error and the empirical optimal convergence rate.

In both plots, the truncated dimension is chosen to be $d = 1, 2, 3, 4, 6$ for the pricing algorithms. The $x-$axis is the computational cost $N = nd$. In the MC case, a pseudo-random sequence is used, while in the QMC case, a scrambled Sobol' sequence is used. At each stage, 30 replications are implemented to compute the root mean square error. In addition, the discrete time algorithm is implemented with $N = n$ to facilitate the comparison with the Karhunen-Loéve expansion method.

For a Karhunen-Loéve expansion with fixed d, the relative RMSE using either i.i.d. or scrambled Sobol' sampling initially decreases as N increases, but eventually reaches the $n = \infty$ limit corresponding to the bias. The Sobol' sampling scheme reaches this limit for smaller n (or N) than is the case for i.i.d. sampling because the former is a superior sampling scheme. It is found empirically by the method described at the end of Section 2 that the convergence rate using the optimal choices of n and d with $N = nd$ in the simple random sampling case is approximately $\mathcal{O}(N^{-0.4})$, while in the scrambled Sobol' sequence case, the optimal convergence rate is close to $\mathcal{O}(N^{-0.8})$.

Sobol' sampling combined with the Karhunen-Loéve expansion yields a smaller error than Sobol' sampling combined with the time discretization algorithm. This is because the time discretization algorithm requires a Sobol' sequence of dimension equalling the number of discrete times ($s = 52$), whereas the Karhunen-Loéve algorithm requires a Sobol' sequence of dimension equalling the number of terms in the expansion, here corresponding to $d \leq 6$. The Karhunen-Loéve expansion concentrates the low frequency behavior of the Brownian motion in the early terms of the expansion, where the Sobol' sequence has especially good equi-distribution properties, something that the time discretization algorithm cannot do.

Acknowledgements This work is supported by grant NSF-DMS-0713848. We are grateful to Igor Cialenco, Thomas Müller-Gronbach, Art Owen, Klaus Ritter, Grzegorz Wasilkowski, and Henryk Woźniakowski for valuable comments and suggestions.

References

1. Creutzig, J., Dereich, S., Müller-Gronbach, T., Ritter, K.: Infinite-dimensional quadrature and approximation of distributions. Found. Comput. Math., online (2008)
2. Glasserman, P.: Monte Carlo methods in financial engineering, *Applications of Mathematics (New York)*, vol. 53. Springer-Verlag, New York (2004). Stochastic Modelling and Applied Probability
3. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Math. Comp. **67**(221), 299–322 (1998)
4. Hickernell, F.J., Sloan, I.H., Wasilkowski, G.W.: The strong tractability of multivariate integration using lattice rules. In: Monte Carlo and quasi-Monte Carlo methods 2002, pp. 259–273. Springer, Berlin (2004)
5. Hickernell, F.J., Wang, X.: The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimension. Math. Comp. **71**(240), 1641–1661 (2002)
6. Higham, D.J.: An algorithmic introduction to numerical simulation of stochastic differential equations. SIAM Review, Education Section **43**, 525–546 (2001)
7. Kuo, F.Y., Sloan, I.H., Wasilkowski, G.W., Woźniakowski, H.: Liberating the dimension. submitted for publication (2009)
8. Müller-Gronbach, T., Ritter, K.: Variable subspace sampling and multi-level algorithms. In this volume (2009)
9. Niederreiter, H.: Random number generation and quasi-Monte Carlo methods, *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1992)
10. Owen, A.B.: Latin supercube sampling for very high dimensional problems. ACM Transactions on Modeling and Computer Simulations **8(1)**, 71–102 (January, 1998)

11. Pagés, G.: Quadratic optimal functional quantization of stochastic processes and numerical applications. In: Monte Carlo and quasi-Monte Carlo methods 2006, pp. 101–142. Springer, Berlin (2008)
12. Shreve, S.E.: Stochastic Calculus for Finance. Volume II - Continuous Time Models. Springer-Verlag, New York (2004)
13. Woźniakowski, H.: Efficiency of quasi-Monte Carlo algorithms for high dimensional integrals. In: Monte Carlo and quasi-Monte Carlo methods 1998 (Claremont, CA), pp. 114–136. Springer, Berlin (2000)
14. Yue, R.X., Hickernell, F.J.: Strong tractability of multivariate integration over Banach spaces. SIAM J. Numer. Anal. **44**, 2559–2583 (2006)

Recent Progress in Improvement of Extreme Discrepancy and Star Discrepancy of One-Dimensional Sequences

Victor Ostromoukhov

Abstract In this communication, we report on recent progress in improvement of extreme discrepancy and star discrepancy of one-dimensional sequences. Namely, we present a permutation of "Babylonian" sequences in base 60, which improves the best known results for star discrepancy obtained by Henri Faure in 1981 [Bull. Soc. Math. France, 109, 143–182 (1981)], and a permutation of sequences in base 84, which improves the best known results for extreme discrepancy obtained by Henri Faure in 1992 [J. Numb. Theory, 42, 47–56 (1992)]. Our best result for star discrepancy in base 60 is $32209/(35400\log 60) \approx 0.222223$ (Faure's best result in base 12 is $1919/(3454\log 12) \approx 0.223585$); our best result for extreme discrepancy in base 84 is $130/(83\log 84) \approx 0.353494$ (Faure's best result in base 36 is $23/(35\log 6) \approx 0.366758$).

1 Introduction

Variance reduction in quasi-Monte Carlo integration is tightly related to uniformity of distributions of the point sets, which sample the integrand. Among different metrics for evaluation of the uniformity of distributions, *star discrepancy* and *extreme discrepancy* play a special role. In fact, it has been shown [9] that the variance of an integral estimation is bounded by an expression which depends on star discrepancy and extreme discrepancies. Schmidt [10] estimated the lower bounds of star and extreme discrepancies for an arbitrary sequence of points. This theoretical estimation has been later improved by Béjian [1]. A thorough description of the problem, the main results and the relevant bibliography can be found in Niederreiter's book [9].

The first low-discrepancy sequences are due to van der Corput [5]. Béjian and Faure [2] estimated the asymptotic behavior of star and extreme discrepancies of

Victor Ostromoukhov
Université de Montréal and LIRIS: CNRS, Université Lyon 1, France
e-mail: ostrom@iro.umontreal.ca

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_36,

the van der Corput sequences. Different constructions for building low-discrepancy sequences have been proposed and evaluated by Borel [3], Braaten and Weller [4], Lapeyre and Pagès [8] and Thomas [11]. In 1981, Faure [6] proposed different generalized (permuted) van der Corput sequences in base 12, having the smallest asymptotic star and extreme discrepancies. In 1989, Thomas [11] improved Faure's result for extreme discrepancy by a small amount. In 1992, Faure [7] further improved extreme discrepancy, using generalized van der Corput sequences in base 36. Faure's constructions for star discrepancy (1981, base 12) and for extreme discrepancy (1992, base 36) remain the best known to date results for one-dimensional sequences.

In this paper, we improve Faure's results for extreme discrepancy and star discrepancy of one-dimensional sequences. Our best result for star discrepancy in base 60 is $32209/(35400\log 60) \approx 0.222223$ (Faure's best result in base 12 is $1919/(3454\log 12) \approx 0.223585$); our best result for extreme discrepancy in base 84 is $130/(83\log 84) \approx 0.353494$ (Faure's best result in base 36 is $23/(35\log 6) \approx 0.366758$).

First, let us recall some definitions commonly used in the specialized literature [6, 7, 9].

Let $X = (x_n)_{n\geq 1}$ be a sequence defined on one-dimensional interval $[0, 1]$, and $A(\alpha, N, X)$ the number of $n \leq N$ such that $0 \leq x_n < \alpha$. The *remainder* E is defined as $E(\alpha, N, X) = A(\alpha, N, X) - \alpha N$; $E([\alpha, \beta[; N, X) = E(\beta, N, X) - E(\alpha, N, X)$, where $0 \leq x_n < \alpha < \beta \leq 1$.

The *extreme discrepancy* is defined as $D(N, X) = \sup_{\alpha,\beta} |E([\alpha, \beta[; N, X)|$, and the *star discrepancy* is defined as $D^*(N, X) = \sup_{\alpha,\beta} |E(\alpha, N, X)|$.

The superior limits of extreme and star discrepancy are defined as

$$s(X) = \overline{\lim_N}(D(N)/\log(N)$$

and

$$s^*(X) = \overline{\lim_N}(D^*(N)/\log(N).$$

Given an integer $n \geq 1$ in b-adic representation $\sum_{j=0}^{\infty} a_j(n)n^j$ and the sequences of permutations $(\sigma_j)_{j\geq 0}$ of the set $\{0, 1, \ldots, b-1\}$, the generalized van der Corput sequence $S_{b,\sigma}$ in fixed base b is defined by

$$S_{b,\sigma} = \sum_{j=0}^{\infty} \sigma_j(a_j(n))n^{-j-1}. \tag{1}$$

In this article, we consider only position-independent permutations, that is permutations $(\sigma_j)_{j\geq 0}$ which are identical for any position j in the generalized van der Corput sequence in Equation (1); j can be omitted.

Let $Z_b^\sigma = (\sigma(0)/b, \ldots, \sigma(b-1)/b)$. For any integer h such that $0 \leq h < b-1$, the functions $\Psi_{b,\sigma}^-$, $\Psi_{b,\sigma}^+$ and $\Psi_{b,\sigma}$ are defined as follows:

$$\Psi^{+}_{b,\sigma}(x) = \begin{cases} \max_h(A([0,h/b[;k;Z^{\sigma}_b) - hx) & if \quad 0 \le h \le \sigma(h-1), \\ \max_h((b-h)x - A([h/b,1[;k;Z^{\sigma}_b)) & if \quad \sigma(h-1) < h < b, \end{cases} \tag{2}$$

$$\Psi^{-}_{b,\sigma}(x) = \begin{cases} \max_h(hx - A([0,h/b[;k;Z^{\sigma}_b)) & if \quad 0 \le h \le \sigma(h-1), \\ \max_h(A([h/b,1[;k;Z^{\sigma}_b) - (b-h)x) & if \quad \sigma(h-1) < h < b, \end{cases} \tag{3}$$

and

$$\Psi_{b,\sigma}(x) = \Psi^{+}_{b,\sigma}(x) + \Psi^{-}_{b,\sigma}(x). \tag{4}$$

The terms $\alpha^{+}_{b,\sigma}, \alpha^{-}_{b,\sigma}$ and $\alpha_{b,\sigma}$ are defined as follows:

$$\begin{aligned} \alpha^{+}_{b,\sigma} &= \inf_{n\ge 1}\sup_{x\in R}\left(\frac{1}{n}\sum_{j=1}^{n}\Psi^{+}_{b,\sigma}\left(\frac{x}{b^j}\right)\right), \\ \alpha^{-}_{b,\sigma} &= \inf_{n\ge 1}\sup_{x\in R}\left(\frac{1}{n}\sum_{j=1}^{n}\Psi^{-}_{b,\sigma}\left(\frac{x}{b^j}\right)\right), \quad \text{and} \\ \alpha_{b,\sigma} &= \inf_{n\ge 1}\sup_{x\in R}\left(\frac{1}{n}\sum_{j=1}^{n}\Psi_{b,\sigma}\left(\frac{x}{b^j}\right)\right). \end{aligned} \tag{5}$$

Three theorems by Faure [6] relate the terms of extreme discrepancy $D(S_{b,\sigma}, N)$ and star discrepancy $D^*(S_{b,\sigma}, N)$, as well as the partial terms $D^+(S_{b,\sigma}, N)$ and $D^-(S_{b,\sigma}, N)$, with functions $\Psi^{+}_{b,\sigma}, \Psi^{-}_{b,\sigma}$ and $\Psi_{b,\sigma}$ defined in Equations (2) to (4). Also, they allow to express the superior limits of extreme discrepancy $s(S_b, \sigma)$ and star discrepancy $s^*(S_b, \sigma)$ in terms of $\alpha^{+}_{b,\sigma}, \alpha^{-}_{b,\sigma}$ and $\alpha_{b,\sigma}$:

Theorem 1 (Faure 1981) *The terms of extreme and star discrepancy of $S_{b,\sigma}$ can be expressed, for any $N \ge 1$, as follows*

$$\begin{aligned} D^+(S_{b,\sigma}, N) &= \sum_{j=1}^{\infty}\Psi^{+}_{b,\sigma}\left(\frac{N}{b^j}\right), \\ D^-(S_{b,\sigma}, N) &= \sum_{j=1}^{\infty}\Psi^{-}_{b,\sigma}\left(\frac{N}{b^j}\right), \\ D(S_{b,\sigma}, N) &= \sum_{j=1}^{\infty}\Psi_{b,\sigma}\left(\frac{N}{b^j}\right), \quad \textit{and} \\ D^*(S_{b,\sigma}, N) &= \max(D^+(S_{b,\sigma}, N), D^-(S_{b,\sigma}, N)). \end{aligned}$$

Theorem 2 (Faure 1981) *The asymptotic term of the extreme discrepancy of $S_{b,\sigma}$ can be expressed in terms of the constant $\alpha_{b,\sigma}$, defined in Equation (5):*

$$s(S_{b,\sigma}) = \overline{\lim}_{N\to\infty}\frac{D(S_{b,\sigma}, N)}{\log N} = \frac{\alpha_{b,\sigma}}{\log b}.$$

Theorem 3 (Faure 1981) *Let $A \subset N$ defined as $A = \bigcup_{H=1}^{\infty} A_H$ and $A_H = \{H(H-1)+1, \ldots, H^2\}$. Let σ be any permutation of $\{0, \ldots, b-1\}$, and τ be a permutation defined as $\tau(k) = b-1-k$, where $0 \le k \le b-1$. Then, the permutation $\Sigma_A = (\sigma_j)_{j \ge 1}$ is defined as $\sigma_j = \sigma$ if $j \in A$ and $\sigma_j = \tau \circ \sigma$ if $j \notin A$. The asymptotic behavior of the star discrepancy of S_{b,Σ_A} can be expressed in terms of $\alpha^+_{b,\sigma}$ and $\alpha^-_{b,\sigma}$ as follows:*

$$s^*(S_{b,\Sigma_A}) = \overline{\lim}_{N \to \infty} \frac{D^*(S_{b,\sigma}, N)}{\log N} = \frac{\alpha^+_{b,\sigma} + \alpha^-_{b,\sigma}}{2 \log b}.$$

2 Main Results

Let σ_{84} be a permutation in base 84:

$$\begin{aligned}\sigma_{84} = (&0, 22, 64, 32, 50, 76, 10, 38, 56, 18, 72, 45, 6, 28, 59, 79, 41, 13, 67, 25, 54,\\ &2, 36, 70, 16, 48, 81, 30, 61, 8, 43, 74, 20, 52, 4, 34, 66, 15, 46, 77, 26, 11, 62,\\ &39, 82, 57, 23, 69, 33, 3, 51, 19, 73, 42, 7, 60, 29, 80, 47, 14, 65, 35, 1, 53, 24,\\ &68, 12, 40, 78, 58, 27, 5, 44, 71, 17, 55, 37, 83, 21, 49, 75, 9, 31, 63).\end{aligned} \tag{6}$$

Theorem 4 *Let base $b = 84$ and the permutation σ_{84}, defined in Equation (6). The superior limit of the extreme discrepancy of the sequence $S_{84,\sigma_{84}}$ is*

$$s(S_{84,\sigma_{84}}) = 130/(83 \log 84) \approx 0.353494.$$

Let σ_{60} be a permutation in base 60:

$$\begin{aligned}\sigma_{60} = (&0, 15, 30, 40, 2, 48, 20, 35, 8, 52, 23, 43, 12, 26, 55, 4, 32, 45, 17, 37,\\ &6, 50, 28, 10, 57, 21, 41, 13, 33, 54, 1, 25, 46, 18, 38, 5, 49, 29, 9, 58,\\ &22, 42, 14, 34, 53, 3, 27, 47, 16, 36, 7, 51, 19, 44, 31, 11, 56, 24, 39, 59).\end{aligned} \tag{7}$$

Theorem 5 *Let base $b = 60$ and the permutation σ_{60}, defined in Equation (7). The superior limit of the star discrepancy of the sequence S_{60,Σ_A} is*

$$s^*(S_{60,\Sigma_A}) = 32209/(35400 \log 60) \approx 0.222223.$$

3 Upper and Lower Bounds of $s(S_{84,\sigma_{84}})$ and $s^*(S_{60,\Sigma_A})$

It may be interesting to evaluate numerically the upper and lower bounds of the extreme discrepancy $s(S_{84,\sigma_{84}})$. Here, we follow Faure's method presented in [6], Section 5.2.1.

To obtain a lower bound of $s(S_{84,\sigma_{84}})$, we compute $(1/\nu)F_\nu(a/(b^\nu-1))$ for given integers a and ν so that $1 \le a \le b^\nu$. For $\nu = 1$ and $a = 16$, we get $s(S_{84,\sigma_{84}}) \ge 0.353494\cdots = 130/(83\log 84)$. Note that we get the same value of $s(S_{84,\sigma_{84}})$ by exact calculation, presented in Section 4.2. To obtain an upper bound, we need to compute $F_n(x)$ up to a sufficiently big n, then evaluate the expression α. Namely, for $n = 6$, $F_6(x)$ reaches its maximum at $x = 120475271600/84^6$. For this x, $s(S_{84,\sigma_{84}})$ can be calculated: $s(S_{84,\sigma_{84}}) = 207668158967/(131736761856\log 84) \approx 0.355778$. Therefore, our numerical evaluation of lower and upper bounds of $s(S_{84,\sigma_{84}})$ can be formulated as follows:

$$0.353494 \le s(S_{84,\sigma_{84}}) \le 0.355778.$$

Note that this numerical estimation already surpasses the best Faure's result of $s(S_{36,\sigma_{36}})$ in base 36.

Similarly, to obtain a lower bound of $s^*(S_{60,\Sigma_A})$, we compute $(1/\nu)F_\nu(a/(b^\nu - 1))$ for given integers a and ν so that $1 \le a \le b^\nu$. For $\nu = 2$ and $a = 1239$, we get $s^*(S_{60,\Sigma_A}) \ge 0.222218\cdots = 111/(122\log 60)$. To obtain an upper bound, we need to compute $F_n(x)$ up to a sufficiently big n, then evaluate the expression α. Namely, for $n = 8$, $F_8(x)$ reaches its maximum at $x = 57822845901639/60^8$. For this x, $s^*(S_{60,\Sigma_A})$ can be calculated: $s^*(S_{60,\Sigma_A}) \approx 0.223424$. Therefore, our numerical evaluation of lower and upper bounds of $s^*(S_{60,\Sigma_A})$ can be formulated as follows:

$$0.222218 \le s^*(S_{60,\Sigma_A}) \le 0.223424.$$

Note that this numerical estimation already improves the best Faure's result of $s^*(S_{12,\Sigma_A})$ in base 12.

4 Proofs

The proofs of Theorems 4 and 5 follow the main line of the proofs provided by Henri Faure in [6, 7].

First, we build the functions $\Psi^+_{b,\sigma}(x)$, $\Psi^-_{b,\sigma}(x)$ and $\Psi_{b,\sigma}(x)$. Then, based on Theorems 1 and 2, we express $s(S_{b,\sigma})$ in terms of $\Psi_{b,\sigma}$. We perform numerical investigation of this function, make an induction hypothesis and prove it.

Similarly, we express $s^*(S_b, \sigma)$ in terms of $\Psi^-_{b,\sigma}$ and $\Psi^+_{b,\sigma}$, based on theorems Theorems 1 and 3. We make an induction hypothesis and prove it.

As in [6, 7], we introduce the function

$$F_n(x) = \sum_{k=0}^{n-1} \Psi(xb^k), \tag{8}$$

where $\Psi(xb^k)$ is the piecewise affine function defined in Equation (4), and express $\alpha = \inf_{n\ge 1}(\max_{x\in[0,1]} F_n(x)/n)$.

4.1 Function $\Psi_{84,\sigma_{84}}(x)$

Finding $\Psi^+_{b,\sigma}(x)$, $\Psi^-_{b,\sigma}(x)$ and $\Psi_{b,\sigma}(x)$ is a tedious work. These functions should be presented as piecewise affine functions on well-defined intervals. As, for definition of $s(S_{84,\sigma_{84}})$, we need the function $\Psi_{84,\sigma_{84}}(x)$ only, we omit here, for the reasons of compactness, the intermediate expressions for $\Psi^+_{84,\sigma_{84}}(x)$ and $\Psi^-_{84,\sigma_{84}}(x)$.

The exact definition of the function $\Psi_{84,\sigma_{84}}(x)$ defined on intervals $I^1_h = [h/84, (h+1)/84]$ is presented in Table 1. Each interval I^1_h can also be expressed as a set of affine subintervals. Thus, the interval $[0, 1]$ is expressed as a set of 216 affine subintervals. Figure 1 (left) shows the function $\Psi_{84,\sigma_{84}}(x)$ visually.

4.2 Proof of Theorem 4

Following [6, 7], we define $\Psi^-_{b,\sigma_{84}}$, $\Psi^+_{b,\sigma_{84}}$ and $\Psi_{b,\sigma_{84}}$ on intervals $I^n_h = [h/b^n, (h+1)/b^n]$. The interval I^n_h is called *dominated* if there exists a set J of integers with $h \notin J$ such that $F_n(x) \le \max_{j\in J} F_n(x + (j-h)/b^n)$ for all $x \in I^n_h$. Otherwise, the interval is *dominant*.

Numerical investigations shows that there are three dominant intervals when $n = 1$: J^1_{28}, J^1_{52} and J^1_{55}. But, for higher n, there are exactly two dominant intervals. For example, when $n = 2$, the dominant intervals are J^2_{2420} and J^2_{4636}. Further numerical investigations allow us to make the following induction hypothesis: for any $n > 1$, the index h_n of dominant intervals $J^n_{h_n}$ is either

$$h_n = -\frac{16}{83} + \frac{509}{83} 3^{n+1} 28^{n-1}$$

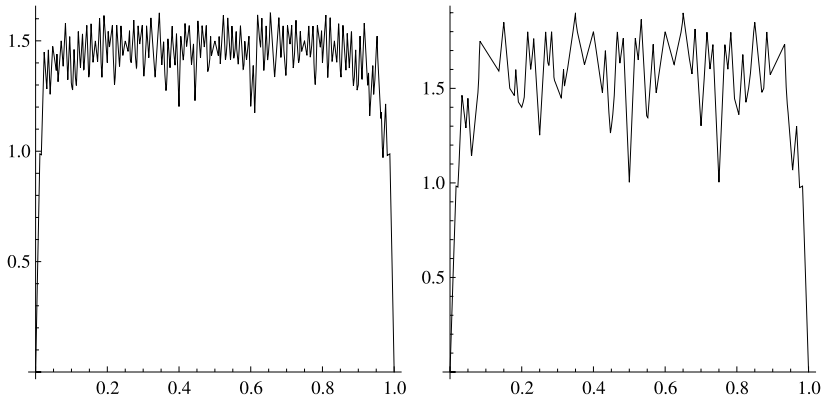


Fig. 1 Graphical representation of the functions $\Psi_{84,\sigma_{84}}(x)$ and $\Psi^+_{60,\sigma_{60}}(x)$, as defined in Equations (2) to (4), for two particular cases explored in this paper. Left: the function $\Psi_{84,\sigma_{84}}(x)$. Right: the function $\Psi^+_{60,\sigma_{60}}(x)$. Both are defined on the interval $[0, 1]$. Note that $\Psi^-_{60,\sigma_{60}}(x) = 0$ on $[0, 1]$.

Table 1 $\Psi_{84,\sigma_{84}}(x)$ defined on intervals $I_h^1 = [h/84, (h+1)/84]$.

h	$\Psi_{84,\sigma_{84}}(x)$, maximum of linear functions	h	$\Psi_{84,\sigma_{84}}(x)$, maximum of linear functions
0	$\{83x\}$	42	$\{5-7x, 44x-21\}$
1	$\{1-x, 61x\}$	43	$\{22-40x, 26x-12\}$
2	$\{2-23x, 41x\}$	44	$\{21-37x, 14x-6, 59x-30\}$
3	$\{3-43x, 31x\}$	45	$\{15-25x, 21x-10, 54x-28\}$
4	$\{2-11x, 41x-1\}$	46	$\{18-30x, 8x-3, 26x-13\}$
5	$\{4-43x, 21x\}$	47	$\{34-58x, 11-17x, 29x-15\}$
6	$\{3-21x, 31x-1\}$	48	$\{17-27x, 11x-5, 54x-30\}$
7	$\{5-41x, 37x-2\}$	49	$\{12-18x, 16x-8\}$
8	$\{6-47x, 2-7x, 51x-4\}$	50	$\{42-68x, 27x-15\}$
9	$\{5-33x, 3-15x, 29x-2, 55x-5\}$	51	$\{36-57x, 51x-30, 64x-38\}$
10	$\{5-29x, 27x-2\}$	52	$\{14-20x, 9-12x, 39x-23\}$
11	$\{9-57x, 5-27x, 25x-2\}$	53	$\{30-45x, 21x-12, 46x-28\}$
12	$\{8-45x, 9x, 49x-6\}$	54	$\{26-38x, 13x-7, 33x-20, 59x-37\}$
13	$\{7-35x, 15x-1\}$	55	$\{18-25x\}$
14	$\{4-15x, 37x-5\}$	56	$\{23x-14\}$
15	$\{10-47x, 3-9x, 45x-7, 61x-10\}$	57	$\{24-33x, 24x-15\}$
16	$\{6-23x, 62x-11\}$	58	$\{43-60x, 16-21x, 32x-21, 72x-49\}$
17	$\{6-22x, 12x-1\}$	59	$\{10-12x, 19x-12\}$
18	$\{14-58x, 9-35x, 15x-2, 60x-12\}$	60	$\{38-51x, 10-12x, 20x-13, 38x-26\}$
19	$\{7-24x, 23x-4, 57x-12\}$	61	$\{35-46x, 19-24x, 6x-3, 59x-42\}$
20	$\{8-27x, 10x-1\}$	62	$\{20-25x, 6x-3\}$
21	$\{3-6x, 25x-5\}$	63	$\{9-10x, 27x-19\}$
22	$\{17-59x, 3-6x, 24x-5, 46x-11\}$	64	$\{29-36x, 11x-7, 24x-17\}$
23	$\{12-38x, 7-20x, 12x-2, 51x-13\}$	65	$\{48-60x, 13-15x, 35x-26, 58x-44\}$
24	$\{7-19x, 12x-2\}$	66	$\{11-12x, 22x-16\}$
25	$\{23-72x, 11-32x, 21x-5, 60x-17\}$	67	$\{51-62x, 23x-17\}$
26	$\{9-24x, 33x-9\}$	68	$\{51-61x, 38-45x, 9x-6, 47x-37\}$
27	$\{9-23x\}$	69	$\{32-37x, 15x-11\}$
28	$\{25x-7\}$	70	$\{14-15x, 35x-28\}$
29	$\{22-59x, 13-33x, 21x-6\}$	71	$\{43-49x, 9-9x, 45x-37\}$
30	$\{19-49x, 10-24x, 9x-2, 61x-21\}$	72	$\{23-25x, 27x-22, 57x-48\}$
31	$\{10-23x, 33x-11\}$	73	$\{25-27x, 29x-24\}$
32	$\{21-51x, 8-17x, 37x-13\}$	74	$\{50-55x, 27-29x, 15x-12, 33x-28\}$
33	$\{20-47x, 68x-26\}$	75	$\{47-51x, 7x-5, 47x-41\}$
34	$\{8-16x, 18x-6, 47x-18\}$	76	$\{35-37x, 41x-36\}$
35	$\{12-25x, 13x-4\}$	77	$\{30-31x, 23x-20\}$
36	$\{14-29x, 17x-6\}$	78	$\{58-61x, 27x-24\}$
37	$\{31-67x, 41x-17, 61x-26\}$	79	$\{55-57x, 13x-11, 31x-28\}$
38	$\{12-23x, 27x-11, 40x-17\}$	80	$\{52-53x, 31-31x, 23x-21\}$
39	$\{9-16x, 18x-7\}$	81	$\{60-61x, 31-31x, 33x-31\}$
40	$\{33-66x, 22-43x, 7x-2, 40x-18\}$	82	$\{51-51x, x\}$
41	$\{23-44x, 7x-2\}$	83	$\{83-83x\}$

Table 2 $\Psi^{+}_{60,\sigma_{60}}(x)$ defined on intervals $I^1_h = [h/60, (h+1)/60]$. Note that $\Psi^{-}_{60,\sigma_{60}}(x) = 0$ on $[0, 1]$.

h	$\Psi^{+}_{60,\sigma_{60}}(x)$, maximum of linear functions	h	$\Psi^{+}_{60,\sigma_{60}}(x)$, maximum of linear functions
0	$\{59x\}$	30	$\{46x-22\}$
1	$\{1-x, 44x\}$	31	$\{9-14x, 26x-12\}$
2	$\{2-16x, 29x\}$	32	$\{20-34x, 3-3x\}$
3	$\{3-31x, 19x\}$	33	$\{3-3x, 26x-13\}$
4	$\{19x, 57x-3\}$	34	$\{21-34x, 17-27x, 13x-6\}$
5	$\{2-3x\}$	35	$\{13x-6\}$
6	$\{2-3x\}$	36	$\{6-7x\}$
7	$\{2-3x\}$	37	$\{6-7x, 9x-4\}$
8	$\{2-3x, 19x-1\}$	38	$\{9x-4, 26x-15\}$
9	$\{5-21x\}$	39	$\{11-14x\}$
10	$\{2-3x, 36x-5\}$	40	$\{9-11x, 29x-18\}$
11	$\{6-24x, 2-3x\}$	41	$\{23-31x\}$
12	$\{7x, 36x-6\}$	42	$\{29x-19, 36x-24\}$
13	$\{7-24x, 16x-2, 29x-5\}$	43	$\{19-24x, 16x-10\}$
14	$\{9-31x\}$	44	$\{34-44x\}$
15	$\{33x-7\}$	45	$\{44x-32\}$
16	$\{9-27x, 3-5x, 24x-5\}$	46	$\{14-16x, 24x-17\}$
17	$\{12-36x, 3-5x\}$	47	$\{30-36x, 7-7x\}$
18	$\{3-5x, 24x-6\}$	48	$\{7-7x, 29x-22\}$
19	$\{13-36x, 11x-2\}$	49	$\{27-31x, 9x-6\}$
20	$\{14x-3\}$	50	$\{21x-16\}$
21	$\{18-46x, 5-9x\}$	51	$\{18-19x\}$
22	$\{5-9x, 7x-1\}$	52	$\{18-19x, 4x-2, 36x-30\}$
23	$\{7x-1\}$	53	$\{23-24x, 4x-2\}$
24	$\{7-13x\}$	54	$\{4x-2\}$
25	$\{7-13x, 27x-10\}$	55	$\{4x-2\}$
26	$\{16-33x, 12-24x, 14x-5\}$	56	$\{54-56x, 24-24x\}$
27	$\{14x-5, 36x-15\}$	57	$\{24-24x, 21x-19\}$
28	$\{13-24x, 14x-5\}$	58	$\{39-39x, x\}$
29	$\{24-46x\}$	59	$\{59-59x\}$

or

$$h_n = \frac{16}{83} + \frac{797}{83} 3^n 28^{n-1}.$$

In these intervals, F_n is the affine function in form $p_n(x - h_n/84^n) + q_n$, where the coefficients p_n and q_n are either

$$p_n = -\frac{61}{83} + \frac{703384^n}{83}; \quad q_n = \frac{130n}{83} + \frac{1434^{2-n}21^{-n}}{6889} + \frac{11715}{192892}$$

or

$$p_n = -\frac{23}{83} + \frac{699584^n}{83}; \quad q_n = \frac{130n}{83} + \frac{1434^{2-n}21^{-n}}{6889} + \frac{11715}{192892}.$$

In both cases, $\max\{F_n(x) \mid x \in J^n_{h_n}\} = q_n$.

Our induction hypothesis can be easily checked for $n = 1$. Let us suppose that it holds for an arbitrary $n \geq 1$. To check that it holds for $n+1$, we need to add $\Psi(xb^n)$ to $F_n(x)$ on $J^n_{h_n}$ and check that $F_{n+1}(x)$ is still dominant on $J^{n+1}_{h_{n+1}}$. We performed

this checking for each affine subinterval of definition of the function $\Psi_{84,\sigma_{84}}(x)$, and verified that our induction hypothesis holds: the intervals $J_{h_{n+1}}^{n+1}$ are dominant.

There, we have proved that

$$d_n = \max_{x\in[0,1]} F_n(x)/n = q_n$$

and

$$\alpha_{84,\sigma_{84}} = \inf_{n\geq 1} d_n/n = \lim_{n\to\infty} d_n/n = 130/83.$$

Consequently,

$$s(S_{84,\sigma_{84}}) = 130/(83\log 84) \approx 0.353494.$$

4.3 Functions $\Psi^{+}_{60,\sigma_{60}}(x)$ and $\Psi^{-}_{60,\sigma_{60}}(x)$

For the definition of $s^*(S_{60,\Sigma_A})$, we need the function $\Psi^{+}_{60,\sigma_{60}}(x)$ and $\Psi^{-}_{60,\sigma_{60}}(x)$.

The exact definition of the function $\Psi^{+}_{60,\sigma_{60}}(x)$ defined on intervals $I_h^1 = [h/60, (h+1)/60]$ is presented in Table 2. Each interval I_h^1 is also expressed as a set of affine subintervals. Thus, the interval [0, 1] is expressed as a set of 102 affine subintervals. $\Psi^{-}_{60,\sigma_{60}}(x) = 0$ on [0, 1]. Figure 1 (right) shows the function $\Psi^{+}_{60,\sigma_{60}}(x)$ visually.

4.4 Proof of Theorem 5

In this case, $\Psi^{-}_{60,\sigma_{60}}(x) = 0$ on [0, 1]. Consequently, $\Psi_{60,\sigma_{60}}(x) = \Psi^{+}_{60,\sigma_{60}}(x)$.

Numerical investigations shows that there are are exactly two dominant intervals, for any $n \geq 1$. When $n = 1$, the dominant intervals are J_{21}^1 and J_{39}^1. When $n = 2$, the dominant intervals are J_{1239}^2 and J_{2361}^2. Further numerical investigations allow us to make the following induction hypothesis: for any $n \geq 1$, the index h_n of dominant intervals $J_{h_n}^n$ is either

$$h_n = \frac{21}{61}\left((-1)^n - 60^n\right)$$

or

$$h_n = \frac{1}{61}\left(21(-1)^n + 2^{2n+3}3^n5^{n+1}\right).$$

In these intervals, F_n is the affine function in form $p_n(x - h_n/60^n) + q_n$, where the coefficients p_n and q_n are either

$$p_n = \frac{1793}{59}2^{2n+1}15^n - \frac{46}{59}; \quad q_n = \frac{32209n}{17700} + \frac{492^{1-2n}3^{1-n}5^{-n}}{3481} + \frac{82369}{104430}$$

or

$$p_n = \frac{2}{59}\left(118760^n - 7\right); \quad q_n = \frac{32209n}{17700} + \frac{492^{1-2n}3^{1-n}5^{-n}}{3481} + \frac{82369}{104430},$$

and $\max\{F_n(x) \mid x \in J^n_{h_n}\} = q_n$.

Our induction hypothesis can be easily checked for $n = 1$. Let us suppose that it holds for an arbitrary $n \geq 1$. To check that it holds for $n+1$, we need to add $\Psi^+(xb^n)$ to $F_n(x)$ on $J^n_{h_n}$ and check that $F_{n+1}(x)$ is still dominant on $J^{n+1}_{h_{n+1}}$. We performed this checking for each affine subinterval of definition of the function $\Psi^+_{60,\sigma_{60}}(x)$, and verified that our induction hypothesis holds: the intervals $J^{n+1}_{h_{n+1}}$ are dominant.

There, we have proved that

$$d_n = \max_{x\in[0,1]} F_n(x)/n = q_n$$

and

$$\alpha^+_{60,\sigma_{60}} = \inf_{n\geq 1} d_n/n = \lim_{n\to\infty} d_n/n = 32209/17700; \quad \alpha^-_{60,\sigma_{60}} = 0.$$

Consequently,

$$s^*(S_{60,\Sigma_A}) = (\alpha^+_{60,\sigma_{60}} + \alpha^-_{60,\sigma_{60}})/(2\log 60) = 32209/(35400\log 60) \approx 0.222223.$$

5 Search Method

Looking for good permutations for large bases b is a difficult task. In fact, an exhaustive search would require explicit evaluation of $b!$ cases. For example, in base 84, this would require studying $84! > 10^{127}$ sequences, which is obviously not tractable with modern computers. In this section, we shortly sketch our search method, which makes this difficult task manageable. We illustrate our method using base $b = 84$.

Our search method consists in three separate steps. First, we build a (pruned) tree of all possible permutations. The root node is, by convention, 0. The first level of the tree contains 83 branches: it can be any number in the range [1, 83]. Every branch of the first level contains 82 branches of the second level, etc. We build the tree, starting from the root. When a new branch is added, this corresponds to building a partial permutation. For example, adding the branch '3' to the root '0', is equivalent to building the beginning of the permutation sequence $\sigma = (0, 3, \ldots)$. At this point, the whole permutation sequence σ is unknown; therefore, we can not build the functions $\Psi^-_{b,\sigma}, \Psi^+_{b,\sigma}$ and $\Psi_{b,\sigma}$, as defined in Equations (2) to (4). Nevertheless, we can evaluate the discrepancy for this partial subset of k elements $(\sigma(0)/b, \ldots, \sigma(k-1)/b), k < b$. If the discrepancy value of this particular sequence is bigger than a certain pruning threshold value T, the branch is pruned away. The choice for the pruning threshold is a delicate task: if it is too large, the tree after all pruning operations may contain a huge number of branches. If the pruning threshold T is too small, the final tree may contain no branches at all. Choosing the right threshold value T requires many trial-and-errors and some intuition. At the end of

the first step, the entire tree of all possible permutations is built. Thanks to pruning, it contains a reasonably small number of branches. For each possible permutation sequence $\sigma^{(i)}$, which corresponds to one leaf of the tree, the discrepancy of the first 84 elements of the sequence is below the threshold T.

At the beginning of the second step, we have a list of permutation sequences $\sigma^{(i)}$, one sequence per leaf of the tree. For each sequence $\sigma^{(i)}$, the functions $\Psi^{-}_{b,\sigma}$, $\Psi^{+}_{b,\sigma}$ and $\Psi_{b,\sigma}$ are built according to Equations (2) to (4). Consequently, the terms $F_n(x)$ can be evaluated according to Equation (8). We sort the sequences $\sigma^{(i)}$ according to the value of $F_2(x)$, calculated for the first 84^2 of each permutation sequence $\sigma^{(i)}$.

During the third step, we study more carefully the permutation sequences $\sigma^{(i)}$ with the smallest values of $F_2(x)$. The behavior of $F_n(x)$ is studied for $n > 2$; for each n, the maxima of $F_n(x)$ are determined. Finally, an induction hypothesis is emitted, and the value of $p_n, q_n, \alpha_{84}, \sigma_{84}$, etc. are calculated. A special program checks the induction hypothesis (validation of the dominant intervals), as described in Section 4.2.

6 Conclusions

In this contribution, we have shown two permutations in bases 60 and 84, which improve the best known values of asymptotic star and extreme discrepancies of one-dimensional sequences. Our numerical exploration, based on the methodology described in Section 5, has shown that, in general, asymptotic terms of star and extreme discrepancies decrease as the values of the base b become bigger. The decrease is not linear, and some particular bases, namely $b = 60$ and $b = 84$, allow particularly low asymptotic terms of star and extreme discrepancies. Our current methodology allows the exploration of integer bases $b < 100$. A challenging future step would be developing a more powerful method of search for "good permutations" in larger bases, which could approach the theoretical lower bounds of star and extreme discrepancies, predicted by Schmidt and Béjian.

Acknowledgements We would like to thank Henri Faure and Art Owen for insightful discussions and help during the preparation of the article. Computational resources were provided by the Réseau Québécois de Calcul de Haute Performance (RQCHP).

References

1. Béjian, R.: Minoration de la discrépance d'une suite quelconque sur T. Acta Arith. **41**, 185–202 (1982)
2. Béjian, R., Faure, H.: Discrépance de la suite de van der Corput. C. R. Acad. Sci. Paris, Série A **285**, 313–316 (1977)
3. Borel, J.P.: Self-similar measures and sequences. J. of Numb. Theory **31**(2), 208–241 (1989)

4. Braaten, E., Weller, W.: An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration. J. Comput. Phys. **33**, 249–258 (1979)
5. van der Corput, J.G.: Verteilungsfunktionen I. Akademie van Wetenschappen **38**, 813–821 (1935)
6. Faure, H.: Discrépance des suites associées à un système de numération (en dimension un). Bull. Soc. Math. France **109**, 143–182 (1981)
7. Faure, H.: Good permutations for extreme discrepancy. J. Numb. Theory **42**, 47–56 (1992)
8. Lapeyre, B., Pagès, G.: Familles de suites discrépance faible obtenues par itération de transformations de [0,1]. Note aux C.R.A.S., Série I **17**, 507–509 (1989)
9. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia, PA (1992)
10. Schmidt, W.M.: Irregularities of distribution. VII. Acta Arith. **21**, 45–50 (1972)
11. Thomas, A.: Discrépance en dimension un. Ann. Fac. Sci. Toulouse, Série Math. **10**(3), 369–399 (1989)

Discrepancy of Hyperplane Nets and Cyclic Nets

Friedrich Pillichshammer and Gottlieb Pirsic

Abstract Digital nets are very important representatives in the family of low-discrepancy point sets which are often used as underlying nodes for quasi-Monte Carlo integration rules. Here we consider a special sub-class of digital nets known as cyclic nets and, more general, hyperplane nets. We show the existence of such digital nets of good quality with respect to star discrepancy in the classical as well as weighted case and we present effective search algorithms based on a component-by-component construction.

1 Introduction

For a finite point set $\mathcal{P}$ consisting of N (not necessarily distinct) points $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}$ in the s-dimensional unit-cube $[0,1)^s$ the *star discrepancy* is defined by

$$D_N^*(\mathcal{P}) = \sup_B \left| \frac{|\{0 \le n < N : \boldsymbol{x}_n \in B\}|}{N} - \lambda_s(B) \right| \tag{1}$$

where the supremum is extended over all subintervals B of $[0,1)^s$ of the form $B = \prod_{i=1}^s [0, b_i)$, $0 < b_i \le 1$ for all $1 \le i \le s$. This is a quantitative measure for the deviation of the empirical distribution of $\mathcal{P}$ from uniform distribution modulo one. The star discrepancy is also intimately connected with the error of a quasi-Monte Carlo (QMC) rule via the well-known Koksma-Hlawka inequality

Friedrich Pillichshammer
Institut für Finanzmathematik, Universität Linz, Altenbergerstraße 69, A-4040 Linz, Austria
e-mail: friedrich.pillichshammer(AT)jku.at

Gottlieb Pirsic
Institut für Finanzmathematik, Universität Linz, Altenbergerstraße 69, A-4040 Linz, Austria
e-mail: gpirsic(AT)gmail.com

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_37,

$$\left| \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{N} \sum_{\boldsymbol{x}\in\mathcal{P}} f(\boldsymbol{x}) \right| \le D_N^*(\mathcal{P}) V(f), \tag{2}$$

where $V(f)$ denotes the variation of f in the sense of Hardy and Krause and $\mathcal{P}$ consists of N points in $[0,1)^s$. See [4, 8, 11] for further informations.

Apart from the above (classical) concept one often studies a "weighted version" of the star discrepancy. This concept has been introduced by Sloan and Woźniakowski [19] with the idea that different coordinates of integrands may have different influence on the quality of approximation of an integral by a QMC rule.

Let $\mathcal{D} = \{1, \ldots, s\}$ be the set of coordinate indices and let $\boldsymbol{\gamma} = (\gamma_i)_{i\ge 1}$ denote a sequence of non-negative real numbers, the so-called "weights" associated to the coordinate directions $i = 1, 2, \ldots$. To avoid a trivial case, we will always assume that not all weights are 0. For $\emptyset \ne \mathfrak{u} \subseteq \mathcal{D}$ let $\gamma_{\mathfrak{u}} = \prod_{i\in\mathfrak{u}} \gamma_i$ be the weight associated to the coordinate directions given by $\mathfrak{u}$, let $|\mathfrak{u}|$ the cardinality of $\mathfrak{u}$, and for a vector $\boldsymbol{z} \in [0,1]^s$ or a subset $B \subseteq [0,1]^s$ let $\boldsymbol{z}(\mathfrak{u})$ or $B(\mathfrak{u})$ denote the projection of the vector or the subset to the components given by $\mathfrak{u}$. Hence $\boldsymbol{z}(\mathfrak{u}) \in [0,1]^{|\mathfrak{u}|}$ and $B(\mathfrak{u}) \subseteq [0,1]^{|\mathfrak{u}|}$.

For a point set $\mathcal{P}$ of N points $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}$ in $[0,1)^s$ and given weights $\boldsymbol{\gamma}$, the *weighted star discrepancy* is defined by

$$D_{N,\boldsymbol{\gamma}}^*(\mathcal{P}) = \sup_B \max_{\emptyset\ne\mathfrak{u}\subseteq\mathcal{D}} \gamma_{\mathfrak{u}} \left| \frac{|\{0 \le n < N : \boldsymbol{x}_n(\mathfrak{u}) \in B(\mathfrak{u})\}|}{N} - \lambda_{|\mathfrak{u}|}(B(\mathfrak{u})) \right|,$$

where the supremum is extended over all subintervals B of $[0,1)^s$ of the form $B = \prod_{i=1}^s [0, b_i)$, $0 < b_i \le 1$ for all $1 \le i \le s$.

This is a generalization of the classical star discrepancy (1) which is recovered if we choose $\gamma_i = 1$ for all $i \ge 1$. Furthermore, the error bound (2) can also be generalized by replacing the star discrepancy with the weighted star discrepancy and the variation by a weighted version of the variation (see [19] for more details).

Good constructions of finite point sets with low star discrepancy are based on the concept of (t, m, s)-nets in base q. A detailed theory of (t, m, s)-nets was developed by Niederreiter [10] (see also [11, Chapter 4] and [14] for surveys of this theory). We refer to [11] and [14] for the definition of (t, m, s)-nets. The crucial fact is that (t, m, s)-nets in a base q provide sets of q^m points in the s-dimensional unit cube $[0,1)^s$ which are extremely well distributed if the quality parameter t is "small". Explicit constructions of (t, m, s)-nets are based on the digital construction scheme which we recall in the following.

From now on let p be a prime and let $q = p^r$, where $r \in \mathbb{N}$, denote a prime-power. Let $\mathbb{F}_q$ be the finite field of q elements and let $\mathbb{F}_q^* := \mathbb{F}_q \setminus \{0\}$, where 0 is the neutral element with respect to addition. Let $\mathbb{Z}_q = \{0, 1, \ldots, q-1\} \subseteq \mathbb{Z}$ with ring operations modulo q and let $\varphi_1 : \mathbb{Z}_q \to \mathbb{F}_q$ be a fixed bijection with $\varphi_1(0) = 0$. We extend φ_1 to a mapping $\varphi : \mathbb{Z}_{q^m} \to \mathbb{F}_q^m$ by setting

$$\varphi(k) := (\varphi_1(\kappa_0), \ldots, \varphi_1(\kappa_{m-1}))^\top \tag{3}$$

for $k = \kappa_0 + \kappa_1 q + \cdots + \kappa_{m-1} q^{m-1}$ with $\kappa_0, \ldots, \kappa_{m-1} \in \mathbb{Z}_q$. Here $\boldsymbol{x}^\top$ means the transpose of the vector $\boldsymbol{x}$.

Definition 1 (digital (t, m, s)-nets). Let $s \geq 1$ and $m \geq 1$ be integers. Let $C_1, \ldots, C_s$ be $m \times m$ matrices over $\mathbb{F}_q$. Now we construct q^m points in $[0, 1)^s$: For $1 \leq i \leq s$ and for $k \in \mathbb{Z}_{q^m}$ multiply the matrix C_i by the vector $\varphi(k)$, i.e.,

$$C_i \varphi(k) =: (y_{i,1}(k), \ldots, y_{i,m}(k))^\top \in \mathbb{F}_q^m,$$

and set

$$x_{k,i} := \frac{\varphi_1^{-1}(y_{i,1}(k))}{q} + \cdots + \frac{\varphi_1^{-1}(y_{i,m}(k))}{q^m}.$$

If for some integer t with $0 \leq t \leq m$ the point set consisting of the points

$$\boldsymbol{x}_k = (x_{k,1}, \ldots, x_{k,s}) \quad \text{for} \quad k \in \mathbb{Z}_{q^m},$$

is a (t, m, s)-net in base q, then it is called a *digital (t, m, s)-net over $\mathbb{F}_q$*, or, in brief, a *digital net (over $\mathbb{F}_q$)*. The C_i are called its *generating matrices*.

Many constructions of digital nets are inspired by a close connection between coding theory and the theory of digital nets (see, for example, Niederreiter [13] or [15]). The construction considered here has been introduced by Niederreiter [13] and it is an analogue to a special type of codes, namely to cyclic codes which are well known in coding theory. Later this construction has been generalized by Pirsic, Dick and Pillichshammer [18] to so-called hyperplane nets.

Definition 2 (hyperplane nets). Let integers $m \geq 1, s \geq 2$ and a prime-power q be given. Let $\mathbb{F}_{q^m}$ be a finite field with q^m elements and fix an element $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_s) \in \mathbb{F}_{q^m}^s$. Let $\mathcal{F}$ be the space of linear forms

$$\mathcal{F} := \{f(x_1, \ldots, x_s) = x_1 \gamma_1 + \cdots + x_s \gamma_s : \gamma_1, \ldots, \gamma_s \in \mathbb{F}_{q^m}\} \subseteq \mathbb{F}_{q^m}[x_1, \ldots, x_s]$$

and consider the subset

$$\mathcal{F}_{\boldsymbol{\alpha}} := \{f \in \mathcal{F} : f(\alpha_1, \ldots, \alpha_s) = 0\}.$$

For each $1 \leq i \leq s$ choose an ordered basis $\mathcal{B}_i$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and define the mapping $\phi : \mathcal{F} \to \mathbb{F}_q^{ms}$ by

$$f = \sum_{i=1}^{s} \gamma_i x_i \in \mathcal{F} \mapsto (\gamma_{1,1}, \ldots, \gamma_{1,m}, \ldots, \gamma_{s,1}, \ldots, \gamma_{s,m}) \in \mathbb{F}_q^{ms},$$

where $(\gamma_{i,1}, \ldots, \gamma_{i,m})$ is the coordinate vector of γ_i with respect to the chosen basis $\mathcal{B}_i$.

We denote by $\mathcal{C}_{\boldsymbol{\alpha}}$ the orthogonal subspace in $\mathbb{F}_q^{ms}$ of the image $\mathcal{N}_{\boldsymbol{\alpha}} := \phi(\mathcal{F}_{\boldsymbol{\alpha}})$. Let

$$C_{\boldsymbol{\alpha}} = (C_1^\top \cdots C_s^\top) \in \mathbb{F}_q^{m \times sm}$$

be a matrix whose row space is $\mathcal{C}_{\boldsymbol{\alpha}}$. Then $C_1, \ldots, C_s \in \mathbb{F}_q^{m \times m}$ are the generating matrices of a *hyperplane net over* $\mathbb{F}_q$ *with respect to* $\mathcal{B}_1, \ldots, \mathcal{B}_s$ and $C_{\boldsymbol{\alpha}}$ is its overall generating matrix. This hyperplane net will be denoted by $\mathcal{P}_{\boldsymbol{\alpha}}$ and we say $\mathcal{P}_{\boldsymbol{\alpha}}$ is the hyperplane net associated to $\boldsymbol{\alpha}$. We shall from now on assume a fixed choice of bases $\mathcal{B}_1, \ldots, \mathcal{B}_s$ and will therefore not explicitly mention them anymore.

Remark 1. In Definition 2 above, if $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^s$ is of the special form $\boldsymbol{\alpha} = \alpha^{(s)} := (1, \alpha, \alpha^2, \ldots, \alpha^{s-1})$ with some $\alpha \in \mathbb{F}_{q^m}$, then we obtain a *cyclic digital net* as introduced initially by Niederreiter [13]. This cyclic net will be denoted by $\mathcal{P}_{\alpha^{(s)}}$ and we say $\mathcal{P}_{\alpha^{(s)}}$ is the cyclic net associated to α.

For a concise representation of the generator matrices $C_1, \ldots, C_s$ of a hyperplane net in terms of $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_s)$ we refer to [18]. We remark here that polynomial lattice point sets can be considered as a (proper) sub-class of hyperplane nets. This has been shown in [17].

In [16] it has been shown that for m large enough there always exists a vector $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^s$ such that the quality parameter t of the hyperplane net $\mathcal{P}_{\boldsymbol{\alpha}}$ satisfies a certain upper bound. From this result it follows that for the star discrepancy of $\mathcal{P}_{\boldsymbol{\alpha}}$ we have $D_{q^m}^*(\mathcal{P}_{\boldsymbol{\alpha}}) = O(m^{2s-2} q^{-m})$.

In this paper we investigate the (weighted) star discrepancy of hyperplane nets. We show by an averaging argument that there exist hyperplane nets and even cyclic nets with "low" (weighted) star discrepancy. Thereby we improve the above mentioned bound from [16]. Furthermore, such point sets can be constructed with a component-by-component algorithm which allows to investigate the asymptotic behavior as s goes to infinity. For the weighted star discrepancy, under certain conditions on the weights, it turns out that our discrepancy bounds do not depend on the dimension s. Such a behavior is known as *strong tractability* (see [19]). We remark here that similar results are already known for polynomial lattice point sets but only in *prime* bases q (see [1, 2]). However, we point out that our results are valid for the much more general class of hyperplane nets. Beside this, here we consider arbitrary prime-power bases q instead of prime bases only as done so far ([1, 2, 3]). For cyclic nets we further show that they can be extended in the dimension s.

2 Prerequisites

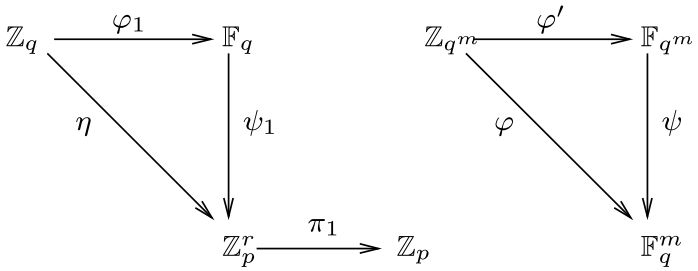
We use the definitions of $q = p^r$, $\mathbb{F}_q$, $\mathbb{Z}_q$, φ_1 and φ from Section 1. Moreover denote by ψ_1 the isomorphism of additive groups $\psi_1 : \mathbb{F}_q \to \mathbb{Z}_p^r$ and define $\eta := \psi_1 \circ \varphi_1$. For $1 \le i \le r$ denote by π_i the projection $\pi_i : \mathbb{Z}_p^r \to \mathbb{Z}_p$, $\pi_i(x_1, \ldots, x_r) = x_i$.

Let $\mathbb{F}_{q^m} = \mathbb{F}_q[\omega]$, such that $\{1, \omega, \ldots, \omega^{m-1}\}$ forms a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. If we have the representation of $\alpha \in \mathbb{F}_{q^m}$ as $\alpha = \sum_{l=0}^{m-1} a_l \omega^l$, where $a_0, \ldots, a_{m-1} \in \mathbb{F}_q$, define

$$\psi(\alpha) := (a_0, \ldots, a_{m-1}) \in \mathbb{F}_q^m.$$

Furthermore, for $k = \sum_{l=0}^{m-1} \kappa_l q^l \in \mathbb{Z}_{q^m}$ let $\varphi'(k) := \sum_{l=0}^{m-1} \varphi_1(\kappa_l)\omega^l$. Observe that $\varphi(k) = \psi(\varphi'(k))$.

We have the following commutative diagrams:



For $1 \le i \le s$ we define the permutations $\tau_i : \mathbb{Z}_{q^m} \to \mathbb{Z}_{q^m}$ by $\tau_i(k) = \varphi^{-1}(B_i\varphi(k))$, where $B_i = (\psi(b_{i,1}), \dots, \psi(b_{i,m})) \in \mathbb{F}_q^{m\times m}$, and where the $b_{i,l}$ constitute the chosen basis $\mathcal{B}_i$.

From now on for $s \in \mathbb{N}$ and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_s) \in \mathbb{F}_{q^m}^s$ we define the set

$$\mathcal{K}_{\boldsymbol{\alpha}} := \left\{ \boldsymbol{k} = (k_1, \dots, k_s) \in \mathbb{Z}_{q^m}^s \setminus \{\boldsymbol{0}\} : \sum_{j=1}^{s} \alpha_j \varphi'(\tau_j(k_j)) = 0 \right\},$$

which is often referred to as the *dual net* of the digital net $\mathcal{P}_{\boldsymbol{\alpha}}$.

Proposition 1. *Let $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^s$. For the star discrepancy of the hyperplane net $\mathcal{P}_{\boldsymbol{\alpha}}$ we have*

$$D_{q^m}^*(\mathcal{P}_{\boldsymbol{\alpha}}) \le 1 - \left(1 - \frac{1}{q^m}\right)^s + 2R_q(\boldsymbol{\alpha}) \le \frac{s}{q^m} + 2R_q(\boldsymbol{\alpha}), \tag{4}$$

where

$$R_q(\boldsymbol{\alpha}) = \sum_{\boldsymbol{k}\in\mathcal{K}_{\boldsymbol{\alpha}}} r_q(\boldsymbol{k}),$$

where for $\boldsymbol{k} = (k_1, \dots, k_s) \in \mathbb{Z}_{q^m}^s$ we write $r_q(\boldsymbol{k}) = r_q(k_1)\cdots r_q(k_s)$ and for $k \in \mathbb{Z}_{q^m}$,

$$r_q(k) = \begin{cases} 1 & \text{if } k = 0, \\ \frac{C}{q^{r+1}} & \text{if } k = \kappa_0 + \kappa_1 q + \cdots + \kappa_r q^r,\ \kappa_r \ne 0, \end{cases}$$

with $C := 1 + \max\limits_{1\le x<q} \max\limits_{1\le y<q} \left| \sum_{a=0}^{y-1} \prod_{i=1}^{r} \exp\left(2\pi\sqrt{-1}\,\frac{(\pi_i\circ\eta)(x)(\pi_i\circ\eta)(a)}{p}\right)\right|$. (Note that $C = C(q) \le q$.)

This result follows from [5, Theorem 1] in combination with [18, Corollary 2.12].

For the weighted star discrepancy $D_{N,\boldsymbol{\gamma}}^*$ of a point set $\mathcal{P}$ of N points in $[0,1)^s$ we find from the definition (or see [3]) that

$$D_{N,\boldsymbol{\gamma}}^*(\mathcal{P}) \le \sum_{\emptyset\ne\mathfrak{u}\subseteq\mathcal{D}} \gamma_{\mathfrak{u}} D_N^*(\mathcal{P}(\mathfrak{u})),$$

where $\mathcal{P}(\mathfrak{u})$ denotes the projection of the point set $\mathcal{P}$ to the coordinates given by $\mathfrak{u}$. If we consider the hyperplane net $\mathcal{P}_{\boldsymbol{\alpha}}$, $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^s$, then (4) yields

$$D_{q^m}^*(\mathcal{P}_{\boldsymbol{\alpha}}(\mathfrak{u})) \le 1 - \left(1 - \frac{1}{q^m}\right)^{|\mathfrak{u}|} + 2R_q(\boldsymbol{\alpha}_{\mathfrak{u}})$$

for $\mathfrak{u} \ne \emptyset$, where $\boldsymbol{\alpha}_{\mathfrak{u}} = (\alpha_j)_{j \in \mathfrak{u}} \in \mathbb{F}_{q^m}^{|\mathfrak{u}|}$. Hence for the weighted star discrepancy of the hyperplane net $\mathcal{P}_{\boldsymbol{\alpha}}$, $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^s$, we get

$$D_{q^m,\boldsymbol{\gamma}}^*(\mathcal{P}_{\boldsymbol{\alpha}}) \le \Gamma_{s,q^m,\boldsymbol{\gamma}} + 2\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha}), \tag{5}$$

where

$$\Gamma_{s,q^m,\boldsymbol{\gamma}} := \sum_{\emptyset \ne \mathfrak{u} \subseteq \mathcal{D}} \gamma_{\mathfrak{u}} \left(1 - \left(1 - \frac{1}{q^m}\right)^{|\mathfrak{u}|}\right) \quad \text{and} \quad \widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha}) := \sum_{\emptyset \ne \mathfrak{u} \subseteq \mathcal{D}} \gamma_{\mathfrak{u}} R_q(\boldsymbol{\alpha}_u).$$

Remark 2. It was proven by Joe [7] that if the sequence of weights $(\gamma_i)_{i \ge 1}$ satisfies $\sum_{i=1}^{\infty} \gamma_i < \infty$, then, with $\Lambda := \sum_{i=1}^{\infty} \frac{\gamma_i}{1+\gamma_i} < \infty$, we have $\Gamma_{s,q^m,\boldsymbol{\gamma}} \le \max(1, \Lambda) \exp(\sum_{i=1}^{\infty} \gamma_i) q^{-m}$ for all $m, s \ge 1$.

In the following proposition we obtain a formula for $\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha})$. The proof of this result is nearly the same as that of [2, Proposition 3.2].

Proposition 2. *We have*

$$\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha}) = \sum_{\boldsymbol{k} \in \mathcal{K}_{\boldsymbol{\alpha}}} \widetilde{r}_q(\boldsymbol{k}, \boldsymbol{\gamma}),$$

where for $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{Z}_{q^m}^s$ *we write* $\widetilde{r}_q(\boldsymbol{k}, \boldsymbol{\gamma}) = \widetilde{r}_q(k_1, \gamma_1) \cdots \widetilde{r}_q(k_s, \gamma_s)$ *and for* $k \in \mathbb{Z}_{q^m}$,

$$\widetilde{r}_q(k, \gamma) = \begin{cases} 1 + \gamma & \text{if } k = 0, \\ \gamma r_q(k) & \text{if } k \ne 0. \end{cases}$$

Proposition 2 shows that R_q and $\widetilde{R}_{q,\boldsymbol{\gamma}}$ only differ by the definitions of r_q and $\widetilde{r}_q$. For this reason we will provide the proofs of the forthcoming results only for the unweighted case. The proofs for the weighted case apply accordingly. In the Appendix (Proposition 3) it is shown how for $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^s$ one can compute $R_q(\boldsymbol{\alpha})$ and $\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha})$ at a cost of $O(sq^m)$ operations.

3 The Results

First we determine the average value of $R_q(\boldsymbol{\alpha})$ respectively $\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha})$ over all possible $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m}^*)^s$. We denote $c_q := C\frac{q-1}{q} \le q - 1$ where C is as in Proposition 1.

Theorem 1. *We have*

$$\frac{1}{|\mathbb{F}_{q^m}^*|^s}\sum_{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s} R_q(\boldsymbol{\alpha}) = \frac{1}{q^m-1}\left(\left(1+mc_q\right)^s - 1 - smc_q\right)$$

and

$$\frac{1}{|\mathbb{F}_{q^m}^*|^s}\sum_{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s} \widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha}) = \frac{1}{q^m-1}\sum_{\substack{\mathfrak{u}\subseteq\mathcal{D}\\ |\mathfrak{u}|\geq 2}}\prod_{i\in\mathfrak{u}}\left(\gamma_i mc_q\right)\prod_{i\notin\mathfrak{u}}(1+\gamma_i)\,.$$

Proof. First observe that $|\mathbb{F}_{q^m}^*| = q^m - 1$. We have

$$\begin{aligned}\frac{1}{|\mathbb{F}_{q^m}^*|^s}\sum_{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s} R_q(\boldsymbol{\alpha}) &= \frac{1}{(q^m-1)^s}\sum_{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s}\sum_{\boldsymbol{k}\in\mathcal{K}_{\boldsymbol{\alpha}}} r_q(\boldsymbol{k})\\ &= \frac{1}{(q^m-1)^s}\sum_{\boldsymbol{k}\in\mathbb{Z}_{q^m}^s\setminus\{\mathbf{0}\}} r_q(\boldsymbol{k})\sum_{\substack{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s\\ \boldsymbol{k}\in\mathcal{K}_{\boldsymbol{\alpha}}}} 1,\end{aligned}$$

where we substituted for $R_q(\boldsymbol{\alpha})$ and changed the order of summation. Note that the τ_j's are permutations and that $\tau_j(k) = 0$ if and only if $k = 0$.

If $\boldsymbol{k}\in\mathbb{Z}_{q^m}^s\setminus\{\mathbf{0}\}$ is of the form $\boldsymbol{k} = (0,\ldots,0,k_i,0,\ldots,0)$ with $k_i\neq 0$, then there is no $\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s$ such that $\alpha_1\varphi'(\tau_1(k_1))+\cdots+\alpha_s\varphi'(\tau_s(k_s)) = \alpha_i\varphi'(\tau_i(k_i)) = 0$, since $\mathbb{F}_{q^m}$ is an integral domain. Otherwise, the number of $\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s$ which satisfy $\alpha_1\varphi'(\tau_1(k_1))+\cdots+\alpha_s\varphi'(\tau_s(k_s)) = 0$ is exactly $(q^m-1)^{s-1}$. Therefore we have (note that $r_q(0) = 1$)

$$\frac{1}{|\mathbb{F}_{q^m}^*|^s}\sum_{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s} R_q(\boldsymbol{\alpha}) = \frac{1}{q^m-1}\left(\sum_{\boldsymbol{k}\in\mathbb{Z}_{q^m}^s\setminus\{\mathbf{0}\}} r_q(\boldsymbol{k}) - \sum_{i=1}^{s}\sum_{k_i\in\mathbb{Z}_{q^m}^*} r_q(k_i)\right).$$

Now the result follows from

$$\sum_{k=0}^{q^m-1} r_q(k) = 1+mc_q \tag{6}$$

which is easily verified. □

The following consequence of Theorem 1 gives an improvement of [16, Corollary 2].

Corollary 1. *Let $0\leq\varepsilon<1$. Then there are more than $\varepsilon|\mathbb{F}_{q^m}^*|^s$ vectors $\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s$ such that*

$$D_{q^m}^*(\mathcal{P}_{\boldsymbol{\alpha}}) \leq \frac{s}{q^m} + \frac{2}{(1-\varepsilon)(q^m-1)}\left(1+mc_q\right)^s$$

respectively

$$D_{q^m,\boldsymbol{\gamma}}^*(\mathcal{P}_{\boldsymbol{\alpha}}) \leq \Gamma_{s,q^m,\boldsymbol{\gamma}} + \frac{2}{(1-\varepsilon)(q^m-1)}\prod_{i=1}^{s}\left(1+\gamma_i\left(1+mc_q\right)\right).$$

Proof. Let $\delta > 0$, then we obtain from Theorem 1,

$$\frac{1}{q^m-1}\left(1+mc_q\right)^s \geq \frac{1}{|\mathbb{F}_{q^m}^*|^s}\sum_{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s} R_q(\boldsymbol{\alpha})$$

$$> \frac{\delta}{q^m-1}\left(1+mc_q\right)^s \frac{1}{|\mathbb{F}_{q^m}^*|^s}\left|\left\{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s : R_q(\boldsymbol{\alpha}) > \frac{\delta}{q^m-1}\left(1+mc_q\right)^s\right\}\right|.$$

Hence

$$\left|\left\{\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s \,:\, R_q(\boldsymbol{\alpha}) \leq \frac{\delta}{q^m-1}\left(1+mc_q\right)^s\right\}\right| > |\mathbb{F}_{q^m}^*|^s\left(1-\frac{1}{\delta}\right),$$

and the result follows from Proposition 1 by substituting $\delta = (1-\varepsilon)^{-1}$. □

From the previous results it follows that there exists a sufficiently large number of vectors $\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s$ which yield hyperplane nets of good quality with respect to (weighted) star discrepancy. As for polynomial lattices (see [1, 2]), such vectors can be found by computer search using a component-by-component construction. We state the algorithm for the star- and the weighted star discrepancy.

Algorithm 1

Given a prime-power q, a sequence of ordered bases $(\mathcal{B}_i)_{i\geq 1}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ (and a sequence $\boldsymbol{\gamma} = (\gamma_i)_{i\geq 1}$ of weights).

1. Choose $\alpha_1 = 1$, the multiplicative unity element in $\mathbb{F}_{q^m}$.
2. For $d > 1$, assume we have already constructed $\alpha_1, \ldots, \alpha_{d-1}$. Then find $\alpha_d \in \mathbb{F}_{q^m}^*$ which minimizes $R_q(\alpha_1, \ldots, \alpha_{d-1}, \alpha_d)$ (or alternatively $\widetilde{R}_{q,\boldsymbol{\gamma}}(\alpha_1, \ldots, \alpha_{d-1}, \alpha_d)$ in the weighted case) as a function of α_d.

In the Appendix we show that $R_q(\boldsymbol{\alpha})$ and $\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha})$ can be computed at a cost of $O(sq^m)$ operations. Hence the cost of the algorithm is of $O(s^2q^{2m})$ operations.

In the following theorem we show that Algorithm 1 is guaranteed to find a good vector $\boldsymbol{\alpha}\in(\mathbb{F}_{q^m}^*)^s$.

Theorem 2. *Let q be prime-power, $m \geq 1$ and $\boldsymbol{\gamma} = (\gamma_i)_{i\geq 1}$ be a sequence of weights. Suppose $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_s) \in (\mathbb{F}_{q^m}^*)^s$ is constructed according to Algorithm 1 using R_q (respectively $\widetilde{R}_{q,\boldsymbol{\gamma}}$). Then for all $d = 1, 2, \ldots, s$ we have*

$$D_{q^m}^*(\mathcal{P}_{(\alpha_1,\ldots,\alpha_d)}) \leq \frac{d}{q^m} + \frac{2}{q^m-1}\left(1+mc_q\right)^d,$$

respectively

$$D_{q^m,\boldsymbol{\gamma}}^*(\mathcal{P}_{(\alpha_1,\ldots,\alpha_d)}) \leq \Gamma_{d,q^m,\boldsymbol{\gamma}} + \frac{2}{q^m-1}\prod_{i=1}^{d}\left(1+\gamma_i\left(1+mc_q\right)\right).$$

Proof. By Proposition 1 it is enough to show that

$$R_q((\alpha_1,\dots,\alpha_d)) \le \frac{1}{q^m-1}\left(1+mc_q\right)^d \quad \text{for all} \quad d=1,\dots,s. \tag{7}$$

Since $\varphi'(\tau_1(k))=0$ if and only if $k=0$ it follows that $R_q(1)=0$ and (7) is true for $d=1$. Suppose now that for some $1\le d<s$ we have already constructed $\boldsymbol{\alpha}=(\alpha_1,\dots,\alpha_d)\in(\mathbb{F}_{q^m}^*)^d$ such that $R_q(\boldsymbol{\alpha})\le\frac{1}{q^m-1}\left(1+mc_q\right)^d$. Then we have

$$\begin{aligned} R_q((\boldsymbol{\alpha},\alpha_{d+1})) &= \sum_{(\boldsymbol{k},k_{d+1})\in\mathcal{K}_{(\boldsymbol{\alpha},\alpha_{d+1})}} \prod_{i=1}^{d+1} r_q(k_i) \\ &= \sum_{\boldsymbol{k}\in\mathcal{K}_{\boldsymbol{\alpha}}} \prod_{i=1}^{d} r_q(k_i) + \theta(\alpha_{d+1}) = R_q(\boldsymbol{\alpha})+\theta(\alpha_{d+1}), \end{aligned}$$

where

$$\theta(\alpha_{d+1}) = \sum_{k_{d+1}\in\mathbb{Z}_{q^m}^*} r_q(k_{d+1}) \sum_{\substack{\boldsymbol{k}\in\mathbb{Z}_{q^m}^d \\ (\boldsymbol{k},k_{d+1})\in\mathcal{K}_{(\boldsymbol{\alpha},\alpha_{d+1})}}} \prod_{i=1}^{d} r_q(k_i).$$

Since α_{d+1} is a minimizer of $R_q((\boldsymbol{\alpha},\cdot))$ it follows that α_{d+1} is also a minimizer of $\theta(\cdot)$ and hence we obtain

$$\begin{aligned} \theta(\alpha_{d+1}) &\le \frac{1}{q^m-1}\sum_{z\in\mathbb{F}_{q^m}^*}\theta(z) \\ &= \frac{1}{q^m-1}\sum_{z\in\mathbb{F}_{q^m}^*}\sum_{k_{d+1}\in\mathbb{Z}_{q^m}^*} r_q(k_{d+1}) \sum_{\substack{\boldsymbol{k}\in\mathbb{Z}_{q^m}^d \\ (\boldsymbol{k},k_{d+1})\in\mathcal{K}_{(\boldsymbol{\alpha},z)}}} \prod_{i=1}^{d} r_q(k_i) \\ &= \frac{1}{q^m-1}\sum_{k_{d+1}\in\mathbb{Z}_{q^m}^*} r_q(k_{d+1}) \sum_{\boldsymbol{k}\in\mathbb{Z}_{q^m}^d} \prod_{i=1}^{d} r_q(k_i) \sum_{\substack{z\in\mathbb{F}_{q^m}^* \\ (\boldsymbol{k},k_{d+1})\in\mathcal{K}_{(\boldsymbol{\alpha},z)}}} 1. \end{aligned}$$

The condition $(\boldsymbol{k},k_{d+1})\in\mathcal{K}_{(\boldsymbol{\alpha},z)}$ is equivalent to the equation

$$z\varphi'(\tau_{d+1}(k_{d+1})) = -(\alpha_1\varphi'(\tau_1(k_1))+\cdots+\alpha_d\varphi'(\tau_d(k_d)))$$

which has exactly one solution $z\in\mathbb{F}_{q^m}^*$ if $\alpha_1\varphi'(\tau_1(k_1))+\cdots+\alpha_d\varphi'(\tau_d(k_d))\ne 0$ and no solution if $\alpha_1\varphi'(\tau_1(k_1))+\cdots+\alpha_d\varphi'(\tau_d(k_d))=0$. Therefore we obtain

$$\theta(\alpha_{d+1}) \le \frac{1}{q^m-1}\sum_{k_{d+1}\in\mathbb{Z}_{q^m}^*} r_q(k_{d+1}) \sum_{\boldsymbol{k}\in\mathbb{Z}_{q^m}^d} \prod_{i=1}^{d} r_q(k_i)$$

$$= \frac{1}{q^m - 1} \left(1 + mc_q\right)^d \sum_{k_{d+1} \in \mathbb{Z}_{q^m}^*} r_q(k_{d+1}).$$

Now we obtain

$$\begin{aligned} R_q((\boldsymbol{\alpha}, \alpha_{d+1})) &= R_q(\boldsymbol{\alpha}) + \theta(\alpha_{d+1}) \\ &\le R_q(\boldsymbol{\alpha}) + \frac{1}{q^m - 1} \left(1 + mc_q\right)^d \sum_{k_{d+1} \in \mathbb{Z}_{q^m}^*} r_q(k_{d+1}) \\ &\le \frac{1}{q^m - 1} \left(1 + mc_q\right)^d \sum_{k_{d+1} \in \mathbb{Z}_{q^m}} r_q(k_{d+1}) = \frac{1}{q^m - 1} \left(1 + mc_q\right)^{d+1} \end{aligned}$$

where we have used Eq. (6). Now (7) follows by induction on d. □

Now we are interested in the behavior of the weighted star discrepancy of hyperplane nets as the dimension s goes to infinity (note that Algorithm 1 is extensible in the dimension s). The following result follows from Theorem 2 and can be proved in the same way as [3, Corollary 8].

Corollary 2. *Let q be a prime-power, $s \ge 2$, $m \ge 1$ and $\boldsymbol{\gamma} = (\gamma_i)_{i \ge 1}$ be a sequence of weights. If $\sum_{i=1}^{\infty} \gamma_i < \infty$, then for any $\delta > 0$ there exists a $\widetilde{c}_{q,\boldsymbol{\gamma},\delta} > 0$, independent of s and m, such that the weighted star discrepancy of the hyperplane net $\mathcal{P}_{\boldsymbol{\alpha}}$ where $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m}^*)^s$ is constructed according to Algorithm 1 using $\widetilde{R}_{q,\boldsymbol{\gamma}}$ satisfies*

$$D_{q^m,\boldsymbol{\gamma}}^*(\mathcal{P}_{\boldsymbol{\alpha}}) \le \frac{\widetilde{c}_{q,\boldsymbol{\gamma},\delta}}{q^{m(1-\delta)}}. \tag{8}$$

Let $N \in \mathbb{N}$ have q-adic expansion $N = \nu_1 q^{m_1} + \cdots + \nu_r q^{m_r}$ with digits $1 \le \nu_i < q$ for $1 \le i \le r$. For each $1 \le i \le r$ construct a vector $\boldsymbol{\alpha}_i \in \mathbb{F}_{q^{m_i}}^s$ according to Algorithm 1 and let $\mathcal{P}_{N,s}$ be the superposition of ν_i copies of the hyperplane net $\mathcal{P}_{\boldsymbol{\alpha}_i}$ for all $1 \le i \le r$. Hence $\mathcal{P}_{N,s}$ contains N elements in $[0,1)^s$. We point out that for any N, s the point set $\mathcal{P}_{N,s}$ can be constructed explicitly. Using Corollary 2 and the same arguments as used in the proof of [6, Theorem 3] we obtain the following result.

Corollary 3. *Let $N, s \in \mathbb{N}$ and assume that $\sum_{i=1}^{\infty} \gamma_i < \infty$. Then for the weighted star discrepancy of the point set $\mathcal{P}_{N,s} \subseteq [0,1)^s$ of cardinality N constructed above, for any $\delta > 0$ we have*

$$D_{N,\boldsymbol{\gamma}}^*(\mathcal{P}_{N,s}) \le \frac{C_{q,\delta,\boldsymbol{\gamma}}}{N^{1-\delta}},$$

where $C_{q,\delta,\boldsymbol{\gamma}} > 0$ is independent of s and N. Hence the weighted star discrepancy of $\mathcal{P}_{N,s}$ achieves a strong tractability bound (with ε-exponent equal to 1).

Obviously we can restrict the search space for $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m}^*)^s$ when we search for cyclic nets only. The subsequent theorem, which improves the second part of [16, Corollary 2], shows that there is a sufficiently large number of good α's in $\mathbb{F}_{q^m}^*$. The cost of a full search for the best $\alpha \in \mathbb{F}_{q^m}^*$ is of $O(sq^{2m})$ operations.

Theorem 3. *Let q be a prime-power, $s \geq 2$, $m \geq 1$ and $\boldsymbol{\gamma} = (\gamma_i)_{i\geq 1}$ be a sequence of weights. For $0 \leq \varepsilon < 1$ there are more than $\varepsilon|\mathbb{F}_{q^m}^*|$ elements $\alpha \in \mathbb{F}_{q^m}^*$ such that*

$$D_{q^m}^*(\mathcal{P}_{\alpha^{(s)}}) \leq \frac{s}{q^m} + \frac{2(s-1)}{(1-\varepsilon)(q^m-1)}\left(1+mc_q\right)^s,$$

respectively

$$D_{q^m,\boldsymbol{\gamma}}^*(\mathcal{P}_{\alpha^{(s)}}) \leq \Gamma_{s,q^m,\boldsymbol{\gamma}} + \frac{2(s-1)}{(1-\varepsilon)(q^m-1)}\prod_{i=1}^{s}\left(1+\gamma_i\left(1+mc_q\right)\right).$$

Proof. We have

$$\begin{aligned}\frac{1}{q^m-1}\sum_{\alpha\in\mathbb{F}_{q^m}^*} R_q(\alpha^{(s)}) &= \frac{1}{q^m-1}\sum_{\alpha\in\mathbb{F}_{q^m}^*}\sum_{\boldsymbol{k}\in\mathcal{K}_{\alpha^{(s)}}}\prod_{i=1}^{s} r_q(k_i)\\ &= \frac{1}{q^m-1}\sum_{\boldsymbol{k}\in\mathbb{Z}_{q^m}^s\setminus\{\mathbf{0}\}}\prod_{i=1}^{s} r_q(k_i)\sum_{\substack{\alpha\in\mathbb{F}_{q^m}^*\\ \boldsymbol{k}\in\mathcal{K}_{\alpha^{(s)}}}} 1.\end{aligned}$$

We have $\boldsymbol{k} \in \mathcal{K}_{\alpha^{(s)}}$ if and only if $\sum_{j=1}^{s}\alpha^{j-1}\varphi'(\tau_j(k_j)) = 0$. As the polynomial $\sum_{j=1}^{s}\alpha^{j-1}\varphi'(\tau_j(k_j))$ over the finite field $\mathbb{F}_{q^m}$ of degree at most $s-1$ has at most $s-1$ zeros $\alpha \in \mathbb{F}_{q^m}^*$ we obtain

$$\frac{1}{q^m-1}\sum_{\alpha\in\mathbb{F}_{q^m}^*} R_q(\alpha^{(s)}) \leq \frac{s-1}{q^m-1}\sum_{\boldsymbol{k}\in\mathbb{Z}_{q^m}^s}\prod_{i=1}^{s} r_q(k_i) = \frac{s-1}{q^m-1}(1+mc_q)^s. \tag{9}$$

For the rest of the proof one just has to follow the proof of Corollary 1. □

There even exists an α such that $\mathcal{P}_{\alpha^{(s)}}$ is of low star discrepancy for arbitrary dimensions $s \geq 1$. One says that $\mathcal{P}_{\alpha^{(s)}}$ is *extensible in the dimension s*. In the special case of polynomial lattices this was shown by Niederreiter [12, Theorem 9].

Corollary 4. *Let q be a prime-power, $m \geq 1$, $(\mathcal{B}_i)_{i\geq 1}$ a sequence of ordered bases of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and $\boldsymbol{\gamma} = (\gamma_i)_{i\geq 1}$ be a sequence of weights. Then for $c > \sum_{s=1}^{\infty}(s(\log(s+1))^2)^{-1}$ there exists an element $\alpha \in \mathbb{F}_{q^m}^*$ such that for all $s \geq 1$ we have*

$$D_{q^m}^*(\mathcal{P}_{\alpha^{(s)}}) \leq \frac{s}{q^m} + \frac{2cs(s-1)(\log(s+1))^2}{q^m-1}(1+mc_q)^s$$

respectively

$$D_{q^m,\boldsymbol{\gamma}}^*(\mathcal{P}_{\alpha^{(s)}}) \leq \Gamma_{s,q^m,\boldsymbol{\gamma}} + \frac{2cs(s-1)(\log(s+1))^2}{q^m-1}\prod_{i=1}^{s}\left(1+\gamma_i\left(1+mc_q\right)\right).$$

In fact, in both cases for arbitrary small $\varepsilon > 0$ we can get at least $(1-\varepsilon)(q^m-1)$ such elements α by choosing $c > 0$ large enough.

Proof. The proof uses arguments from [12]. Let

$$E_s := \left\{ \alpha \in \mathbb{F}_{q^m}^* \,:\, R_q(\alpha^{(s)}) > \frac{cs(s-1)(\log(s+1))^2}{q^m-1}(1+mc_q)^s \right\}$$

where the constant $c > 0$ is chosen such that $c > \sum_{s=1}^{\infty}(s(\log(s+1))^2)^{-1}$. Using (9) we obtain for any $s \geq 1$,

$$(s-1)(1+mc_q)^s \geq \sum_{\alpha \in \mathbb{F}_{q^m}^*} R_q(\alpha^{(s)}) \geq |E_s| \frac{cs(s-1)(\log(s+1))^2}{q^m-1}(1+mc_q)^s$$

and hence $|E_s| \leq \frac{q^m-1}{cs(\log(s+1))^2}$. For $E := \bigcup_{s\geq 1} E_s$ we hence obtain

$$|E| \leq \sum_{s=1}^{\infty} |E_s| \leq \frac{q^m-1}{c} \sum_{s=1}^{\infty} \frac{1}{s(\log(s+1))^2} < q^m - 1 = |\mathbb{F}_{q^m}^*|.$$

Especially, there exists an element $\alpha \in \mathbb{F}_{q^m}^* \setminus E$ and for this element we have

$$R_q(\alpha^{(s)}) \leq \frac{cs(s-1)(\log(s+1))^2}{q^m-1}(1+mc_q)^s \quad \text{for all} \quad s \geq 1.$$

Now the result follows from Proposition 1. □

Appendix: Calculation of R_q and $\widetilde{R}_{q,\boldsymbol{\gamma}}$

We will give an explicit form of the quantities R_q and $\widetilde{R}_{q,\boldsymbol{\gamma}}$ which can be computed efficiently. For this computation we will employ Walsh functions which we briefly recall in the following (notations are defined as in Section 2).

Definition 3 (Walsh functions). Let $q = p^r$ with a prime p and a positive integer r, let $k \in \mathbb{N}_0$ with base q representation $k = \kappa_0 + \kappa_1 q + \cdots + \kappa_{m-1} q^{m-1}$ where $\kappa_l \in \mathbb{Z}_q$ and let $x \in [0,1)$ with base q representation $x = \xi_1/q + \xi_2/q^2 + \cdots$. Then the k-th *Walsh function over the finite field* $\mathbb{F}_q$ with respect to the bijection φ_1 is defined by

$${}_{\mathbb{F}_q,\varphi_1}\mathrm{wal}_k(x) := \prod_{l=0}^{m-1} \prod_{i=1}^{r} \exp\left(2\pi\sqrt{-1}\,\frac{(\pi_i \circ \eta)(\kappa_l)(\pi_i \circ \eta)(\xi_l)}{p}\right).$$

For convenience we will in the rest of the paper omit the subscript and simply write wal_k if there is no ambiguity.

Multivariate Walsh functions are defined by multiplication of the univariate components, i.e., for $\boldsymbol{x} = (x_1, \ldots, x_s) \in [0,1)^s$, $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s$ where $s > 1$, we set

$$\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{s} \mathrm{wal}_{k_j}(x_j).$$

Specifically we will need the following lemma that gives an important indicator function.

Lemma 1. *Let $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^s$ and let $\mathcal{P}_{\boldsymbol{\alpha}}$ be the hyperplane net associated to $\boldsymbol{\alpha}$. Then for any $\boldsymbol{k} \in \mathbb{Z}_{q^m}^s$ we have*

$$\frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{\alpha}}} \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{k} \in \mathcal{K}_{\boldsymbol{\alpha}} \cup \{\boldsymbol{0}\}, \\ 0 & \text{else.} \end{cases}$$

For a proof of this result see [18, Corollary 2.12].

Proposition 3. *Let $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m}^*)^s$ and let $\mathcal{P}_{\boldsymbol{\alpha}}$ be the associated hyperplane net. Then for the quantities $R_q(\boldsymbol{\alpha})$ and $\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha})$ we get the formulas*

$$R_q(\boldsymbol{\alpha}) = -1 + \frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{\alpha}}} \prod_{i=1}^{s} \left(1 + C\left(\frac{q-1}{q} m_0(x_i) - 1\right)\right),$$

$$\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha}) = -\prod_{i=1}^{s}(1+\gamma_i) + \frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{\alpha}}} \prod_{i=1}^{s} \left(1 + \gamma_i + \gamma_i C\left(\frac{q-1}{q} m_0(x_i) - 1\right)\right),$$

with C as in the definition of r_q in Proposition 1 and for $x \in q^{-m}\mathbb{Z}_{q^m} \setminus \{0\}$, $m_0(x) := \max\{l \le m : x < q^{-(l-1)}\} = \lceil -\log_q x \rceil$ and $m_0(0) := m + q/(q-1)$. Hence $R_q(\boldsymbol{\alpha})$ and $\widetilde{R}_{q,\boldsymbol{\gamma}}(\boldsymbol{\alpha})$ can be computed at a cost of $O(sq^m)$ operations.

Proof. By definition we have

$$R_q(\boldsymbol{\alpha}) = \sum_{\boldsymbol{k} \in \mathcal{K}_{\boldsymbol{\alpha}}} \prod_{i=1}^{s} r_q(k_i).$$

Using Lemma 1 we can let the sum range over all $\boldsymbol{k} \in \mathbb{Z}_{q^m}^s$. We get

$$\begin{aligned} 1 + R_q(\boldsymbol{\alpha}) &= \sum_{\boldsymbol{k} \in \mathbb{Z}_{q^m}^s} \frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{\alpha}}} \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) \prod_{i=1}^{s} r_q(k_i) \\ &= \frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{\alpha}}} \sum_{\boldsymbol{k} \in \mathbb{Z}_{q^m}^s} \prod_{i=1}^{s} r_q(k_i) \mathrm{wal}_{k_i}(x_i) \\ &= \frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{\alpha}}} \prod_{i=1}^{s} \left(1 + \sum_{k \in \mathbb{Z}_{q^m} \setminus \{0\}} r_q(k) \mathrm{wal}_k(x_i)\right). \end{aligned} \tag{10}$$

Since $r_q(k)$ depends only on the "digit length" of k we get for $x \in [0,1)$, by [9, Lemma 4] (note that it is enough to consider $x \in q^{-m}\mathbb{Z}_{q^m}$ only)

$$\begin{aligned}\sum_{k\in\mathbb{Z}_{q^m}\setminus\{0\}} r_q(k)\mathrm{wal}_k(x) &= \sum_{l=1}^{m}\frac{C}{q^l}\sum_{k=q^{l-1}}^{q^l-1}\mathrm{wal}_k(x)\\ &= \sum_{l=1}^{m}\frac{C}{q^l}q^{l-1}\times\begin{cases} q-1 & \text{if } x < q^{-l},\\ -1 & \text{if } q^{-l}\le x < q^{-(l-1)},\\ 0 & \text{else,}\end{cases}\\ &= \frac{C}{q}\Big((q-1)\big(m_0(x)-1\big)-1\Big),\end{aligned}$$

where for $x \in q^{-m}\mathbb{Z}_{q^m}$ the quantity $m_0(x)$ is defined in Proposition 3. Inserting the formula into (10) gives the claimed result for $R_q(\boldsymbol{\alpha})$.

The derivation of the weighted case from the unweighted one can be carried out as in [2]. □

Acknowledgements The authors are supported by the Austrian Science Foundation (FWF), Project S9609, that is part of the Austrian National Research Network "Analytic Combinatorics and Probabilistic Number Theory".

References

1. Dick, J., Kritzer, P., Leobacher, G. and Pillichshammer, F.: Constructions of general polynomial lattice rules based on the weighted star discrepancy. Finite Fields Appl. **13**: 1045–1070, 2007.
2. Dick, J., Leobacher, G. and Pillichshammer, F.: Construction algorithms for digital nets with low weighted star discrepancy. SIAM J. Numer. Anal. **43**: 76–95, 2005.
3. Dick, J., Niederreiter, H. and Pillichshammer, F.: Weighted star discrepancy of digital nets in prime bases. In *Monte Carlo and quasi-Monte Carlo methods 2004*, pages 77–96. Springer, Berlin, 2006.
4. Drmota, M. and Tichy, R. F.: *Sequences, Discrepancies and Applications*. Lecture Notes in Mathematics 1651, Springer, Berlin, 1997.
5. Grozdanov, V. S. and Stoilova, S. S.: The inequality of Erdős-Turan-Koksma: Walsh and Haar functions over finite groups. Math. Balkanica (N.S.) **19**: 349–366, 2005.
6. Hinrichs, A., Pillichshammer, F. and Schmid, W. Ch.: Tractability properties of the weighted star discrepancy. J. Complexity **24**: 134–143, 2008.
7. Joe, S.: Construction of good rank-1 lattice rules based on the weighted star discrepancy. In *Monte Carlo and quasi-Monte Carlo methods 2004*, pages 181–196. Springer, Berlin, 2006.
8. Kuipers, L. and Niederreiter, H.: *Uniform Distribution of Sequences.* John Wiley, New York, 1974; reprint, Dover Publications, Mineola, NY, 2006.
9. Larcher, G. and Pirsic, G.: Base change problems for generalized Walsh series and multivariate numerical integration. Pacific J. Math. **189**: 75–105, 1999.
10. Niederreiter, H.: Point sets and sequences with small discrepancy. Monatsh. Math. **104**: 273–337 1987.
11. Niederreiter, H.: *Random Number Generation and Quasi-Monte Carlo Methods.* No. 63 in CBMS-NSF Series in Applied Mathematics. SIAM, Philadelphia, 1992.
12. Niederreiter, H.: The existence of good extensible polynomial lattice rules. Monatsh. Math. **139**: 295–307, 2003.
13. Niederreiter, H.: Digital nets and coding theory. In *Coding, Cryptography and Combinatorics*, pages 247–257. Birkhäuser, Basel, 2004.

14. Niederreiter, H.: Constructions of (t, m, s)-nets and (t, s)-sequences. Finite Fields Appl. **11**: 578–600, 2005.
15. Niederreiter, H.: Nets, (t, s)-sequences and codes. In *Monte Carlo and quasi-Monte Carlo methods 2006*, pages 83–100. Springer, Berlin, 2008.
16. Pillichshammer, F. and Pirsic, G.: The quality parameter of cyclic nets and hyperplane nets. Uniform Distribution Theory **4**: 69–79, 2009.
17. Pirsic, G.: A small taxonomy of integration node sets. Österreich. Akad. Wiss. Math.-Natur. Kl. Sitzungsber. II **214** (2005): 133–140, 2006.
18. Pirsic, G., Dick, J. and Pillichshammer, F.: Cyclic digital nets, hyperplane nets, and multivariate integration in Sobolev spaces. SIAM J. Numer. Anal. **44**: 385–411, 2006.
19. Sloan, I. H. and Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? J. Complexity **14**: 1–33, 1998.

A PRNG Specialized in Double Precision Floating Point Numbers Using an Affine Transition

Mutsuo Saito and Makoto Matsumoto

Abstract We propose a pseudorandom number generator specialized to generate double precision floating point numbers. It generates 52-bit pseudorandom patterns supplemented by a constant most significant 12 bits (sign and exponent), so that the concatenated 64 bits represents a floating point number obeying the IEEE 754 format. To keep the constant part, we adopt an affine transition function instead of the usual $\mathbb{F}_2$-linear transition, and extend algorithms computing the period and the dimensions of equidistribution to the affine case. The resulted generator generates double precision floating point numbers faster than the Mersenne Twister, whoes output numbers only have 32-bit precision.

1 Introduction

In [19], we proposed a fast version of the Mersenne twister (MT) of [14], that exploits the single instruction multiple data (SIMD) feature of some recent CPUs, which processes 128 bits at a time [20]. This new pseudorandom number generator (PRNG), named SFMT (which stands for SIMD-oriented fast Mersenne twister), is faster than the original MT and also has better equidistribution. The proposal of [19] also features a block generation procedure, which returns a large array of pseudorandom numbers at each call.

Mutsuo Saito
Department of Mathematics, Graduate School of Science, Hiroshima University, Hiroshima, Japan
e-mail: saito@math.sci.hiroshima-u.ac.jp

Makoto Matsumoto
Department of Mathematics, Graduate School of Science, Hiroshima University, Hiroshima, Japan
e-mail: m-mat@math.sci.hiroshima-u.ac.jp

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_38,

In this article, we propose PRNGs specialized in generating floating point numbers, which we call dSFMT (double precision floating point SFMT). It generates a sequence of 64-bit patterns with constant 12 most significant bits (MSBs), so that each of 64-bit patterns represents a double precision floating point numbers in a fixed interval in the standard IEEE 754 format. Instead of the usual $\mathbb{F}_2$-linear transition function, we adopt an $\mathbb{F}_2$-affine transition function to keep the fixed constant in the 64 bits (§4). We extended to the affine case some of the existing algorithms to compute the period and distribution. As a result, we implemented this type of generators whose periods are multiples of 6 Mersenne primes from $2^{521}-1$ to $2^{19937}-1$, respectively. These generators are shown to be faster than MT, SFMT and WELL generators, and have satisfactorily high dimensions of equidistribution (much higher than MT, but lower than WELL, which attains the theoretical bounds).

2 Generating Floating Point Numbers

Usually, floating point pseudorandom numbers are obtained by converting integer pseudorandom numbers. One may consider recursion in floating point numbers for PRNG, but it may accumulate approximation errors. Since the rounding-off is not standardized, the generated sequence often depends on CPUs. Consequently, usual PRNGs generate integer random numbers by integer recursion, and converts them to floating point numbers by multiplying by a constant. However, this method requires a conversion from an integer to a floating point number, which consumes about 50% of the CPU time in the generation, according to our experiments using the 64-bit MT [15].

A faster conversion is given by bit operations fitting a standard floating point format. We recall the most widely-used standard, IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEEE Std 754-2008) [6], which we shall refer as IEEE 754. The standard was defined in 1985 and revised in 2008, and here we treat the 64-bit binary format valid for both. The 64 bits are separated in to the sign bit (the most significant bit, MSB), the exponent (the next 11 most significant bits, representing an integer between 0 to 2047, denoted by e) and the remaining 52 bits (representing a real number in the interval $[1,2)$. This 52-bit pattern $\mathtt{xxx}\ldots$ is interpreted as a binary floating number $1.\mathtt{xxx}\ldots$, denoted by f). When $0 < e < 2047$, the 64 bits represents a floating point number $\pm f \times 2^{e-1023}$ with the sign determined by the sign bit. Thus, if the sign bit is 0 and $e = 1023$ (or equivalently the 12 MSBs are `0x3ff` in hexadecimal form), then the represented number is in $[1,2)$. If the 52-bit fraction part is uniformly randomly chosen, then the represented number is uniformly randomly distributed over $[1,2)$ with 52-bit precision. In the C language, this conversion of a 64-bit integer `x` is described as follows:

```
x = (x >> 12) | 0x3FF0000000000000ULL;
y = *((double *)&x);
```

where the first line shifts `x` to the right by 12 bits and set the 12 MSBs to the constant `0x3ff`, and the second line regards the 64-bit pattern as an IEEE 754 format. This method is less portable than the conversion by multiplication, because it depends on a particular format, but consumes only 5% to 10% of the CPU time for the conversion, according to our experiments with the 64-bit MT. This method goes back to at least 1997: Agner Fog used this method in his open source library [4], and others seemed to have invented it independently, too.

A pseudorandom number r in $[1,2)$ can be converted into $[0,1)$ (respectively $(0,1]$) by taking $r-1$ (respectively $2-r$). In practice, it is often the case that random numbers r in the range $[1,2)$ can be used without converting into $[0,1)$: for example, the Box-Muller transformation converts two uniform random numbers s_1, s_2 in $[0,1)$ into two normally distributed numbers

$$\sqrt{-2\log(1-s_1)}\sin(2\pi s_2), \quad \sqrt{-2\log(1-s_1)}\cos(2\pi s_2).$$

If r_1, r_2 are two uniform random numbers in $[1,2)$, then the conversion can be done by

$$\sqrt{-2\log(2-r_1)}\sin(2\pi r_2), \quad \sqrt{-2\log(2-r_1)}\cos(2\pi r_2).$$

3 LFSR with Lung

Our proposal is to use a linear recursion over $\mathbb{F}_2$ to generate a sequence of 64-bit patterns with the 12 MSBs being `0x3ff` as above, by a Linear Feedback Shift Register (LFSR) with additional memory called the 'lung.' We identify the set of bits $\{0, 1\}$ with the two element field $\mathbb{F}_2$. This means that every arithmetic operation is done modulo 2. A b-bit register or memory is identified with a horizontal vector in $\mathbb{F}_2^b$, and $+$ denotes the sum as vectors (i.e., bit-wise exclusive or). We consider an array of b-bit integers of size N in computer memory as the vector space $(\mathbb{F}_2^b)^N$.

An LFSR generates a sequence $\mathbf{w_0}, \mathbf{w_1}, \mathbf{w_2}, \ldots$ of elements $\mathbb{F}_2^b$ by a recursion

$$\mathbf{w}_{i+N} = g(\mathbf{w}_i, \ldots, \mathbf{w}_{i+N-1}), \quad (i = 0, 1, 2, \ldots)$$

where g is an $\mathbb{F}_2$-linear map $(\mathbb{F}_2^b)^N \to \mathbb{F}_2^b$. In a naive implementation, this recursion is computed by using an array `W[0..N-1]` of N words of b-bit size, by the simultaneous substitutions

```
W[0] ← W[1], W[1] ← W[2],...,W[N-2] ← W[N-1],
                                W[N-1] ← g(W[0],...,W[N-1]).
```

The first $N-1$ substitutions shift the content of the array, hence the name of LFSR. Note that in the implementation we may use an indexing technique to avoid computing these substitutions, see [7, P.28 Algorithm A]. Before starting the gener-

ation, we need to set the (array) state to some initial values; this is the initialization. Mersenne Twister [14] (MT) is an example of such an LFSR.

An LFSR with lung generates a sequence $\mathbf{w_0}$,$\mathbf{w_1}$, $\mathbf{w_2}$, ... of elements $\mathbb{F}_2^b$ by a recursion

$$\mathbf{w}_i = g(\mathbf{w}_{i-N+1}, ..., \mathbf{w}_{i-1}, \mathbf{u}_{i-1}), \tag{1}$$

$$\mathbf{u}_i = h(\mathbf{w}_{i-N+1}, ..., \mathbf{w}_{i-1}, \mathbf{u}_{i-1}). \tag{2}$$

where g and h are $\mathbb{F}_2$-linear maps $(\mathbb{F}_2^b)^N \to \mathbb{F}_2^b$ and $\mathbf{w}_i, \mathbf{u}_i \in \mathbb{F}_2^b$. In the implementation, the $\mathbf{w}_i$'s are kept in an array `W[0..N-1]`, and $\mathbf{u}_i$ is (expected to be) kept in a register of the CPU, which is called the *lung*. We denote the register by `U`. The first line (1) renews the array `W[0..N-1]`, and the second line (2) renews the register (lung) `U`. The idea of LFSR with lung appeared in the talk of Hiroshi Haramoto the MCM 2005 conference, and is also used in the WELL PRNG [17]. The lung realizes a short feedback loop, which improves some measures of randomness such as higher dimensional equidistributions and the density of nonzero coefficients in the characteristic polynomial.

4 Affinity Introduced by the Constant Part

Our idea is to design the functions g and h in the recursion (1) (2) for the LFSR with lung, so that if the initial values $\mathbf{w}_0, \ldots, \mathbf{w}_{N-1}$ are set to have `0x3ff` at their 12 MSBs, then the following $\mathbf{w}_i$ have the same property, regardlessly of the value of $\mathbf{u}_0$. According to our experiments, this method is 5% to 10% faster than the bit-masking conversion explained in §2.

A new difficulty in this approach is that the state transition is far from being maximal periodic. A linear state transition function is said to be maximal periodic, if every non-zero state lies on the same orbit. The existence of the constant implies that, if the initial state is chosen as above, then the 12 MSBs of each member of the array of `W[0..N-1]` are constant in the orbit, and the transition can not be maximal periodic. This makes it difficult to apply standard techniques to compute the period and high-dimensional equidistribution property.

A natural solution to this problem is to redefine the state space by excluding the constant part, and consider the transition function as an affine function. More concretely, let $\mathbf{w}'_i$ denote the lower 52-bit of $\mathbf{w}_i$. Since the upper 12 bits is a constant, the recursion formula (1), (2) can be described by

$$\mathbf{w}'_i = g'(\mathbf{w}'_{i-N+1}, ..., \mathbf{w}'_{i-1}, \mathbf{u}_{i-1}), \tag{3}$$

$$\mathbf{u}_i = h'(\mathbf{w}'_{i-N+1}, ..., \mathbf{w}'_{i-1}, \mathbf{u}_{i-1}). \tag{4}$$

Here, it is easy to see that the linearity of g (resp. h) implies the affinity of g' (resp. h'). (Here affine means linear plus a constant.)

Let b_w denote the number of variable bits in each `W[i]` (52 in the above case), and b_u denote the number of bits in the lung `U`. This LFSR with lung (not linear but affine) is considered as an automaton, with the state space $S = \mathbb{F}_2^{b_u+b_w\times(N-1)}$. The state transition function $F : S \to S$ is given by

$$(\mathbf{w}'_0, \ldots, \mathbf{w}'_{N-2}, \mathbf{u}_0) \mapsto (\mathbf{w}'_1, \ldots, \mathbf{w}'_{N-2}, g'(\mathbf{w}'_0, \ldots, \mathbf{w}'_{N-2}, \mathbf{u}_0), h'(\mathbf{w}'_0, \ldots, \mathbf{w}'_{N-2}, \mathbf{u}_0)).$$

As a b_w-bit vector generator (i.e., removing the constant bits), the output function is

$$o : S \to {\mathbb{F}_2}^{b_w}; \quad (\mathbf{w}'_0, \ldots, \mathbf{w}'_{N-2}, \mathbf{u}_0) \mapsto \mathbf{w}'_0.$$

Now, both F and o are not linear but affine. Namely, they have the form $x \mapsto Ax + c$ where x is a vector, A is an $\mathbb{F}_2$ matrix, and c is a constant vector. (If $c = 0$, it is linear.)

5 Reduction from Affine to Linear: Fixed Points

Let f denote the linear part of F, namely, put $c := F(0)$ and

$$F(x) = f(x) + c \tag{5}$$

with linear $f : S \to S$. If F has a fixed point $F(z) = z$, then $F(x+z) = f(x+z) + c = f(x) + z$, and consequently $F^n(x+z) = f^n(x) + z$. Thus, for the state transition $x_0, x_1, x_2, \ldots$ by F, its translation $x_0 + z, x_1 + z, \ldots$ by the constant z is obtained by the linear state transition f, hence can be analyzed by the existing methods. Since the period and the distribution property of the sequence is unchanged by a parallel translation, computation of those for the affine F is reduced to those for the linear f. If f has the maximal period, then the equidistribution property can be computed as usual.

The equation $F(z) = z$ is equivalent to $(f - \mathrm{Id})(z) = c$, where Id denotes the identity transformation on S. Thus, a fixed point exists if the characteristic polynomial χ_f of f does not have 1 as a root, in particular if it is irreducible with degree ≥ 2.

6 Reducible Transition Function in Affine Case

Usually, to make sure that the period is maximal, we need to check the primitivity of χ_f. This is often computationally difficult, since we need the integer factorization of $2^{\deg(\chi(t))} - 1$, which is hard if the degree is high (say, > 10000). There are two methods to avoid this: (1) to tune the size of the state space to be a Mersenne exponent (i.e. a prime number p such that $2^p - 1$ is also prime) where $2^{\deg(\chi(t))} - 1$ is a prime, and (2) to use f such that χ_f has an irreducible factor of a Mersenne

prime degree denoted by p. We here adopt the latter method, named reducible transition method (RTM) in [19]. This is advantageous over the former in the generation speed, because of no need for discarding a part of the state array (as was required in MT [14] and WELL [17]). Note that this idea appeared in somewhat different purposes previously in [5, 1, 2].

We here recall RTM very briefly. Let $f : S \to S$ be an $\mathbb{F}_2$-linear transition function, $o : S \to O$ be an $\mathbb{F}_2$-linear output function. Assume that a linear transition function $f : S \to S$ has a decomposition $S = V_p \oplus V_r$, $f = f_p \oplus f_r$ with $f_p : V_p \to V_p$, $f_r : V_r \to V_r$. In other words, f is the combined generator obtained from the two generators (f_p, V_p, o_p) and (f_r, V_r, o_r), in the sense of §2.3 of [10]. A linear output function $o : S \to O$ is then the sum of the restrictions $o_p : V_p \to O$ and $o_r : V_r \to O$. The output of the combined generator is obtained by taking the xor of the outputs of each generator. The period of the combined generator (f, S, o) is the least common multiple of the periods of the two generators. Thus, once we know that (f_p, V_p, o_p) has a large period, then the combined generator has at least that period.

Our strategy is to fix a Mersenne prime p, to determine the size N of the state array so that $p \le \dim S$, and then search for parameters with a factorization $\chi_f = \phi_p \phi_r$, where ϕ_p is irreducible of degree p and ϕ_r has degree r with $r < p$. Then, it is automatic to have a decomposition $S = V_p \oplus V_r$ into p-dimensional and r-dimensional subspaces, so that the restriction f_p (respectively f_r) of f to V_p (respectively to V_r) has the characteristic polynomial ϕ_p (respectively ϕ_r). Once we have such decomposition, then the component $f_p : V_p \to V_p$ has the Mersenne exponent dimension p, and hence an existing method searches for the parameters that assure the period of $2^p - 1$. Then we can assure $2^p - 1$ as the lower bound on the period of the combined generator, provided that the initial state $s = s_p \oplus s_r \in S = V_p \oplus V_r$ has the non zero component $s_p \neq 0$.

In the case of affine transition $F(x) = f(x) + c$, we assume that its linear part f satisfies the above factorizing condition $\chi_f = \phi_p \phi_r$. Let us decompose $c = c_p \oplus c_r$ and $x = x_p \oplus x_r$ along $V_p \oplus V_r$, then

$$F(x) = f(x) + c = (f_p(x_p) + c_p) \oplus (f_r(x_r) + c_r) =: F_p(x_p) \oplus F_r(x_r). \quad (6)$$

This implies that the affine generator (F, S, o) is obtained by combining two affine generators (F_p, V_p, o_p) and (F_r, V_r, o_r). Now f_p is irreducible, and the fixed point argument in §5 reduce the computation of the periods and the high-dimensional equidistribution property for F_p to those for f_p.

7 Period Certification

We explain how to choose parameters realizing the period $2^p - 1$, for a given Mersenne exponent p. For the linear transition function, the method is described in [19], which we briefly recall. Let N be the smallest length of the array such that

the dimension of the state space $S = \mathbb{F}_2^{b_u+b_w\times(N-1)}$ is greater than or equal to p. Thus, $r := \dim S - p < b_w$ holds.

We randomly choose parameters for the recursion (3) and (4). Let $F : S \to S$ be the corresponding affine transition function, and $f : S \to S$ be its linear part. We compute the characteristic polynomial $\chi_f(t)$ by using Berlekamp-Massey algorithm, and check whether it decomposes to

$$\chi_f = \phi_p \phi_r$$

where ϕ_p is a primitive polynomial of degree p and ϕ_r is a polynomial of degree $r := \dim S - p < b_w$. We assume $r < p$, which is natural in our context where p is large, and also $b_w \le b_u$, since b_w is the number of the non-constant part in a b_u-bit word. We continue the random search of parameters, until we obtain a primitive ϕ_p.

Once we found such a set of parameter, then we have $S = V_p \oplus V_r$ and the projector $P_p : S \to V_p$. To assure the period of a multiple of $2^p - 1$ for the initial state $s \in S$, it suffices to assure $s_p := P_p(s) \neq 0$. In the implementation, to compute $P_p(s)$ is a time-consuming procedure in the initialization. Instead, we propose the following method, named period certification vector (PCV) method, by which the period is certified by looking at one word in the state.

Let V_U denote the b_u-dimensional vector space corresponding to the lung `U` in (4). To certify the period for the initial state $s \in S$, it suffices to show that $s \notin V_r$. Let $\pi : S \to V_U$ be the projection obtained by extracting the lung from the state space S. Since we assumed $b_u = \dim(V_U) > r$, the image $\pi(V_r)$ is a proper subspace of V_U. Hence, there is a nonzero vector q in V_U which is orthogonal to every vector in $\pi(V_r)$. We call such a vector PCV. For a given initial state s, if the inner product $\pi(s) \cdot q$ is nonzero, then $\pi(s) \notin \pi(V_r)$ and hence $s \notin V_r$, and the period is certified. If the inner product is zero, then we can make the inner product nonzero by reversing one bit in $\pi(s)$.

The period certification for affine case easily reduces to the linear case. Let $z_p \in V_p$ be a fixed point of F_p. For the initial state $s \in S$, it suffices to show that $s - z_p \notin V_r$ to assure the period. This can be done by precomputing $\pi(z_p)$, and check that $(\pi(s) - \pi(z_p)) \cdot q \neq 0$. In this method, only two constant b_u-bit words $\pi(z_p)$ and q need to be precomputed and stored, and at the initialization stage, only the last inner product need to be computed.

8 Computation of the Dimension of Equidistribution

We briefly recall the definition of dimension of equidistribution (cf. [3, 8, 19]).

Definition 1. Let $F : S \to S$ be an affine transition function over $\mathbb{F}_2$. Let v be an integer, and $o : S \to \mathbb{F}_2^v$ be a v-bit affine output function. The generator (S, F, o) is said to be k-dimensionally equidistributed, if the map

$$S \to (\mathbb{F}_2^v)^k, \quad s \mapsto (o(s), o(F(s)), o(F^2(s)), \ldots, o(F^{k-1}(s)))$$

is surjective. The largest value of such k is called the dimension of equidistribution (DE).

For a b-bit integer generator, its *dimension of equidistribution at v-bit accuracy* $k(v)$ is defined as the DE of the v-bit sequence, obtained by extracting the v MSBs from each of the b-bit integers.

Let $P = 2^p - 1$ be the period of the generated sequence. Then, there is an upper bound $k(v) \le \lfloor p/v \rfloor$, and their gap $d(v)$ is called the dimension defect at v of the sequence, and their sum Δ over $v = 1, \ldots, b$ is called the total dimension defect, namely:

$$d(v) := \lfloor p/v \rfloor - k(v) \text{ and } \Delta := \sum_{v=1}^{b} d(v). \tag{7}$$

We adopt RTM as in §6, and the dimensions of the equidistribution of the larger component (F_p, V_p, o_p) gives the lower bound of these dimensions [9] [19]. Accordingly, we define $k(v)$ and $d(v)$ of RTM to be those for this larger component. Let f_p be the linear part of F_p. Since χ_{f_p} is irreducible, there is a fixed point of F_p as explained in §5. Thus, computation of $k(v)$ for F_p is reduced to that for the linear part f_p, which was done in [19].

9 Implementation of dSFMT

As a result of the preceding discussion, we propose a generator using SIMD features, an affine transition function to keep the MSBs constant, and reducible characteristic polynomial. The generator is named dSFMT (double precision floating point SIMD-oriented Fast Mersenne Twister).

Remark 1. In the homepage [18], we released "dSFMT" in 2007, but no corresponding article exists. The generator proposed here is its improved version, by adopting the lung and a more efficient recursion, and is referred to as dSFMT version 2 in the homepage. In this manuscript, we call the former dSFMT-old, and the latter simply dSFMT.

The dSFMT generator is an LFSR with lung, whose recursion formulas are (1) and (2) with

$$h(\mathbf{w}_0, \ldots, \mathbf{w}_{N-2}, \mathbf{u}_0) = \mathbf{w}_0 A + \mathbf{w}_M + \mathbf{u}_0 B, \tag{8}$$

$$g(\mathbf{w}_0, \ldots, \mathbf{w}_{N-2}, \mathbf{u}_0) = \mathbf{w}_0 + h(\mathbf{w}_0, \ldots, \mathbf{w}_{N-2}, \mathbf{u}_0) C, \tag{9}$$

where $\mathbf{w}_i$'s and $\mathbf{u}$ are 128-bit integers regarded as horizontal vectors in $\mathbb{F}_2^{128}$, and A, B, C are linear transformations described below, computable by a few SIMD operations. The number b_w of variable bits is $128 - 12 \times 2 = 104$, while $b_u = 128$. It generates two 52-bit precision floating point numbers at each step.

- $\mathbf{w}A := \mathbf{w} \overset{64}{<<} \mathrm{SL1}$

This notation means that **w** is regarded as two 64-bit memories, and **w**A is the result of the left-shift of each 64 bits by SL1 bits. There is such a SIMD operation in the Pentium SSE2, and can be emulated in the PowerPC AltiVec SIMD. SL1 is a parameter with $12 \leq \mathrm{SL1} < 64$.

- $\mathbf{u}B := \mathbf{u}\,\mathrm{perm}(4, 3, 2, 1)$
  This notation means that **u** is regarded as four 32-bit memories and $\mathbf{u}\,\mathrm{perm}(4, 3, 2, 1)$ is the result of reversing the order of the 32-bit blocks in the 128 bits. The permutation can be done by one SIMD operation.
- $\mathbf{u}C := (\mathbf{u} \overset{64}{>>} 12) + (\mathbf{u} \,\&\, \mathrm{MASK})$
  The notation $\mathbf{u} \overset{64}{>>} 12$ means that **u** is regarded as two 64-bit memories and each right-shifted by 12 bit. The notation & means a 128-bit bitwise logical '**AND**' with a 128-bit constant vector **MASK**, defined as the concatenation of two 64-bit vectors with 0s in the 12 MSBs for both.

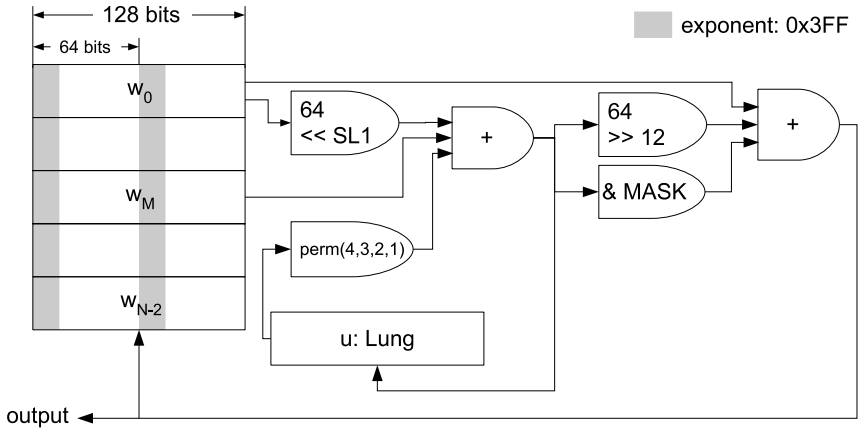


Fig. 1 Diagram of dSFMT.

Fig. 1 shows the recursion in a circuit-like diagram. Note that the recursion (8) and (9) is linear, and the constant `0x3ff` (in hexadecimal) of the IEEE 754 exponent part does not appear. The recursion is carefully selected so that once the initial values $w_0, \ldots, w_{N-2}$ have `0x3ff` in their 12 MSBs, then these constant parts are preserved through the recursion. This trick contributes to the generation speed, by avoiding constant-setting.

Table 1 lists the parameters for dSFMTs with various sizes. Table 2 lists the corresponding fixed points and PCVs, as discussed in §7.

Table 1 Parameter sets. MEXP denotes the Mersenne exponents. The column MASK(HIGH) shows the higher 64 bits of the constant mask, and the column MASK(LOW) shows the lower 64 bits in hexadecimal.

MEXP	N	M	SL1	MASK(LOW)	MASK(HIGH)
521	5	3	25	0x000fbfefff77efff	0x000ffeebfbdfbfdf
1279	13	9	19	0x000efff7ffddffee	0x000fbfffffff77fff
2203	21	7	19	0x000fdfffff5edbfff	0x000f77ffffffffbfe
4253	41	19	19	0x0007b7fffef5feff	0x000ffdffeffefbfc
11213	108	37	19	0x000fffffffdf7fffd	0x000dfffffff6bfff
19937	192	117	19	0x000ffaffffffffb3f	0x000ffdfffc90fffd

Table 2 Fixed points and PCVs. Two 64-bit integers (in hexadecimal) piled in one place represent one 128-bit integer with higher (respectively lower) 64-bit being the upper (respectively lower) piled integer. For example, the PCV in the first row is 0xccaa588000000000000000000000001.

MEXP	Fixed Point	PCV
521	0xcfb393d661638469 0xc166867883ae2adb	0xccaa588000000000 0x0000000000000001
1279	0xb66627623d1a31be 0x04b6c51147b6109b	0x7049f2da382a6aeb 0xde4ca84a40000001
2203	0xb14e907a39338485 0xf98f0735c637ef90	0x8000000000000000 0x0000000000000001
4253	0x80901b5fd7a11c65 0x5a63ff0e7cb0ba74	0x1ad277be12000000 0x0000000000000001
11213	0xd0ef7b7c75b06793 0x9c50ff4caae0a641	0x8234c51207c80000 0x0000000000000001
19937	0x90014964b32f4329 0x3b8d12ac548a7c7a	0x3d84e1ac0dc82880 0x0000000000000001

10 Comparison of Speed

We compared generators MT19937, 64-bit MT19937, SFMT19937, dSFMT-old19937 and dSFMT19937, with and without SIMD instructions. For MT and SFMT, 'mask' means the conversion by bit operation described in §2 from 64-bit integers, and '$\times$ const' means the conversion by multiplying 2^{-64}. Note that the original MT and SFMT do not use 'mask' conversion.

We measured the speeds for five different CPUs: Pentium M 1.4GHz, Pentium IV 3GHz, core 2 duo 1.83GHz (32-bit mode, using one core), AMD Athlon 64 3800+ (64-bit mode), and PowerPC G4 1.33GHz. In returning the random values, we used two different methods. One is sequential generation, where one double floating point random number is returned per call. The other is block generation, where an array of random double floating point numbers is generated per call. We used Intel C Compiler for intel CPUs (Pentium M, Pentium IV, core 2 duo) and GNU C Compiler for others (AMD Athlon, Power PC G4).

We measured the consumed CPU time in second, for generating 10^8 floating point numbers in the range $[0, 1)$ to compare with other generators. In case of the block generation, we generate 10^5 floating point numbers per call, and this is iterated 10^3 times. For sequential generation, the same 10^8 floating point numbers are generated, one per call. We used the inline declaration `inline` to avoid the function call. Implementations without SIMD are written in ISO/IEC 9899 : 1999(E) C Programming Language, Second Edition (which we shall refer to as C99 in the rest of this article), whereas those with SIMD use some standard SIMD extension of C99 supported by the Intel C compiler and GNU C Compiler.

Table 3 summarizes the speed comparison using SIMD and Table 4 shows the speed comparison without SIMD. The 64-bit MT is not listed in Table 3, because we do not have the SIMD version. The first two lines list the CPU time (in seconds) needed to generate 10^8 floating point numbers, for a Pentium-M CPU. The first line lists the timings for the block-generation scheme, and the second line lists those for the sequential generation scheme. The result is that dSFMT is the fastest for all CPUs, all returning methods, using SIMD and without using SIMD. Table 5 shows the speed of other generators. Although dSFMT has 52-bit precision while the others have only 32-bit precision, dSFMT's sequential generation using standard C (i.e. the slowest case) is faster than the other generators, except xorshift128 [13], whose quality is reported to be questionable in [16].

Table 3 The CPU time (sec.) for 10^8 generations using SIMD.

		dSFMT (new)	dSFMT-old (old)	MT mask	SFMT mask	SFMT × const
Pentium M 1.4 Ghz	blk	0.626	0.867	1.526	0.928	2.636
	seq	1.422	1.761	3.181	2.342	3.671
Pentium 4 3 Ghz	blk	0.254	0.640	0.987	0.615	3.537
	seq	0.692	1.148	3.339	3.040	3.746
core 2 duo 1.83GHz	blk	0.199	0.381	0.705	0.336	0.532
	seq	0.380	0.457	1.817	1.317	2.161
Athlon 64 2.4GHz	blk	0.362	0.637	1.117	0.623	1.278
	seq	0.680	0.816	1.637	0.763	1.623
PowerPC G4 1.33GHz	blk	0.887	1.151	2.175	1.657	8.897
	seq	1.212	1.401	5.624	2.994	7.712

11 Dimension of Equidistribution

We calculated $d(v)$s for our generators, by using the method described in §8. Table 6 lists the dimension defects $d(v)$ of dSFMT, for Mersenne exponent (mexp) = $521, 1279, 2203, 4253, 11213, 19937$ and $v = 1, 2, \ldots, 52$. The $d(v)$ for $1 \le v \le 22$ are very small. The larger mexp seems to lead to the larger $d(v)$ for $v > 22$. Still,

Table 4 The CPU time (sec.) for 10^8 generations (without SIMD).

		dSFMT (new)	dSFMTold (old)	MT 64 mask	MT mask	SFMT mask	SFMT × const
Pentium M	blk	1.345	2.023	2.031	3.002	2.026	3.355
1.4 Ghz	seq	2.004	2.386	2.579	3.308	2.835	3.910
Pentium 4	blk	1.079	1.128	1.432	2.515	1.929	3.762
3 Ghz	seq	1.431	1.673	3.137	3.534	3.485	4.331
core 2 duo	blk	0.899	1.382	1.359	2.404	1.883	1.418
1.83GHz	seq	0.777	1.368	1.794	1.997	1.925	2.716
Athlon 64	blk	0.334	0.765	0.820	1.896	1.157	1.677
2.4GHz	seq	0.567	0.970	1.046	2.134	1.129	2.023
PowerPC G4	blk	1.834	3.567	2.297	4.326	4.521	12.685
1.33GHz	seq	1.960	2.865	4.090	5.489	5.464	9.110

Table 5 The CPU time (sec.) for 10^8 generations for other generators, where conversion to floating point numbers uses constant multiplication.

	WELL1024	WELL19937	MT19937	XORSHIFT128
Pentium M	2.076	2.876	2.028	1.233
Pentium 4	1.626	2.031	1.232	1.023
core 2 duo	1.165	1.913	1.032	0.653
Athlon 64	0.804	1.191	0.971	0.975
Power PC G4	2.947	7.524	3.082	2.267

the case mexp=19937 has total dimension defect $\Delta = 2608$, which is smaller than the defect of the 32-bit SFMT19937' and the 32-bit MT19937, which are $\Delta = 4188$ and $\Delta = 6750$, respectively. Note that it is natural to guess that Δ increases at least proportionally to the word size b, by its definition (7).

Remark 2. The number of non-zero terms in $\chi_f(t)$ is an index measuring the amount of bit-mixing. The column "weight" in Table 7 shows these numbers: dSFMT19937 has the ratio $9756/19992 = 0.488$ which is higher than those of MT (135/19937=0.00677), WELL19937a ($8585/19937 = 0.431$) and WELL19937b ($9679/19937 = 0.485$).

The dSFMT generators passed the DIEHARD statistical tests [12]. They also passed TestU01 [11] consisting of 144 different tests, except for `LinearComp` (fail unconditionally) and MatrixRANK tests (fail if the size of dSFMT is smaller than the matrix size). These tests measure the $\mathbb{F}_2$-linear dependency of the outputs, and reject $\mathbb{F}_2$-linear generators, such as MT, SFMT and WELL.

We shall keep the latest version of the codes in the web page [18].

Acknowledgements This study is partially supported by JSPS/MEXT Grant-in-Aid for Scientific Research No.19204002, No.18654021, and JSPS Core-to-Core Program No.18005. The second author is partially supported as a visiting professor of The Institute of Statistical Mathematics. The authors are thankful to the anonymous referees and the editor for valuable comments.

Table 6 $d(v)$ $(1 \le v \le 52)$ of 52-bit fraction part of dSFMT.

	521	1279	2203	4253	11213	19937		521	1279	2203	4253	11213	19937
d(1)	0	1	0	0	4	0	d(27)	0	0	1	1	33	4
d(2)	0	1	1	0	0	1	d(28)	0	6	7	28	33	10
d(3)	0	2	1	0	0	1	d(29)	1	5	7	23	28	67
d(4)	0	0	0	0	1	1	d(30)	3	3	15	18	80	126
d(5)	0	0	0	0	0	0	d(31)	2	6	13	15	68	107
d(6)	0	1	1	0	1	0	d(32)	4	4	10	10	58	88
d(7)	0	0	0	0	0	1	d(33)	6	12	25	43	120	220
d(8)	0	0	0	0	0	1	d(34)	6	12	23	44	114	202
d(9)	0	1	0	0	0	0	d(35)	5	11	21	40	105	185
d(10)	1	0	0	0	0	0	d(36)	5	10	20	37	96	169
d(11)	0	0	0	0	0	0	d(37)	5	9	18	33	88	155
d(12)	0	0	0	0	0	0	d(38)	4	8	16	30	80	141
d(13)	0	0	0	0	0	0	d(39)	4	7	15	28	72	128
d(14)	0	0	0	0	0	1	d(40)	4	6	14	25	65	115
d(15)	0	0	0	0	0	1	d(41)	3	6	12	22	58	103
d(16)	0	0	0	0	0	1	d(42)	3	5	11	20	51	91
d(17)	0	0	0	0	0	0	d(43)	3	4	10	17	45	80
d(18)	0	0	0	0	0	0	d(44)	2	4	9	15	39	70
d(19)	0	0	0	0	0	0	d(45)	2	3	7	13	34	60
d(20)	1	0	0	0	0	0	d(46)	2	2	6	11	28	50
d(21)	0	0	0	0	7	0	d(47)	2	2	5	9	23	41
d(22)	0	0	0	0	0	134	d(48)	1	1	4	7	18	32
d(23)	0	0	7	16	22	94	d(49)	1	1	3	5	13	23
d(24)	0	1	3	9	19	58	d(50)	1	0	3	4	9	15
d(25)	0	1	0	6	7	25	d(51)	1	0	2	2	4	7
d(26)	0	0	0	0	0	0	d(52)	1	0	1	0	0	0
total dimension defect Δ								73	135	291	531	1423	2608

Table 7 The number of non-zero terms in $\chi_f(t)$.

mexp	521	1279	2203	4253	11213	19937
degree of $\chi_f(t)$	544	1376	2208	4288	11256	19992
weight	273	673	1076	2233	5684	9756
ratio	0.50	0.49	0.49	0.52	0.50	0.49

References

1. R.P. Brent and P. Zimmermann. Random number generators with period divisible by a Mersenne prime. In *Computational Science and its Applications - ICCSA 2003*, volume 2667, pages 1–10, 2003.
2. R.P. Brent and P. Zimmermann. Algorithms for finding almost irreducible and almost primitive trinomials. *Fields Inst. Commun.*, 41:91–102, 2004.
3. R. Couture, P. L'Ecuyer, and S. Tezuka. On the distribution of k-dimensional vectors for simple and combined Tausworthe sequences. *Math. Comp.*, 60(202):749–761, 1993.
4. Agner Fog. Pseudo random number generators. http://www.agner.org/random/.
5. M. Fushimi. Random number generation with the recursion $x_t = x_{t-3p} \oplus x_{t-3q}$. *Journal of Computational and Applied Mathematics*, 31:105–118, 1990.

6. IEEE standard for binary floating-point arithmetic 754, 2008. http://ieeexplore.ieee.org/servlet/opac?punumber=4610933.
7. D. E. Knuth. *The Art of Computer Programming. Vol.2. Seminumerical Algorithms*. Addison-Wesley, Reading, Mass., 3rd edition, 1997.
8. P. L'Ecuyer. Maximally equidistributed combined tausworthe generators. *Math. Comp.*, 65(213):203–213, 1996.
9. P. L'Ecuyer and J. Granger-Piché. Combined generators with components from different families. *Mathematics and Computers in Simulation*, 62:395–404, 2003.
10. P. L'Ecuyer and F. Panneton. $\mathbb{F}_2$-linear random number generators. In *Advancing the Frontiers of Simulation: A Festschrift in Honor of George Samuel Fishman*, pages 175–200, 2009. C. Alexopoulos, D. Goldsman, and J. R. Wilson Eds.
11. P. L'Ecuyer and R. Simard. TestU01: A C library for empirical testing of random number generators. *ACM Transactions on Mathematical Software*, 15(4):346–361, 2006.
12. G. Marsaglia. The Marsaglia Random Number CDROM, with the DIEHARD Battery of Tests of Randomness. Department of Statistics, Florida State University, (1996) http://www.stat.fsu.edu/pub/diehard/.
13. G. Marsaglia. Xorshift RNGs. *Journal of Statistical Software*, 8(14):1–6, 2003.
14. M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1):3–30, January 1998. http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html.
15. T. Nishimura. Tables of 64-bit mersenne twisters. *ACM Trans. on Modeling and Computer Simulation*, 10(4):348–357, October 2000.
16. F. Panneton and P. L'Ecuyer. On the Xorshift random number generators. *ACM Transactions on Modeling and Computer Simulation*, 15(4):346–361, 2005.
17. F. Panneton, P. L'Ecuyer, and M. Matsumoto. Improved long-period generators based on linear reccurences modulo 2. *ACM Transactions on Mathematical Software*, 32(1):1–16, 2006.
18. M. Saito and M. Matsumoto. SFMT Homepage. http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/index.html.
19. M. Saito and M. Matsumoto. SIMD-oriented fast Mersenne twister : a 128-bit pseudorandom number generator. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*, LNCS, pages 607–622. Springer, 2008.
20. SIMD From Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/SIMD.

On the Behavior of the Weighted Star Discrepancy Bounds for Shifted Lattice Rules

Vasile Sinescu and Pierre L'Ecuyer

Abstract We examine the question of constructing shifted lattice rules of rank one with an arbitrary number of points n, an arbitrary shift, and small weighted star discrepancy. An upper bound on the weighted star discrepancy, that depends on the lattice parameters and is easily computable, serves as a figure of merit. It is known that there are lattice rules for which this upper bound converges as $O(n^{-1+\delta})$ for any $\delta > 0$, uniformly over the shift, and lattice rules that achieve this convergence rate can be found by a component-by-component (CBC) construction. In this paper, we examine practical aspects of these bounds and results, such as: What is the shape of the probability distribution of the figure of merit for a random lattice with a given n? Is the CBC construction doing much better than just picking the best out of a few random lattices, or much better than using a randomized CBC construction that tries only a small number of random values at each step? How does the figure of merit really behave as a function of n for the best lattice, and on average for a random lattice, say for n under a million? Do we observe a convergence rate near $O(n^{-1})$ in that range of values of n? Finally, is the figure of merit a tight bound on the true discrepancy, or is there a large gap between the two?

1 Introduction and Background Results

We are concerned with the approximation of an integral over the d-dimensional unit cube,

$$I_d(f) = \int_{[0,1]^d} f(\mathbf{u})\,\mathrm{d}\mathbf{u}$$

Vasile Sinescu and Pierre L'Ecuyer
Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, C.P.6128 Succ. Centre-Ville, Montréal QC,H3C 3J7, Canada
e-mail: {sinescuv, lecuyer}@iro.umontreal.ca

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_39,

where $f : [0,1)^d \to \mathbb{R}$, by a shifted lattice rule of rank one with generating vector $\mathbf{z} \in \mathbb{Z}^d$ and arbitrary shift $\boldsymbol{\Delta} \in [0,1)^d$, i.e., by the average

$$Q_{n,d}(f) = \frac{1}{n} \sum_{k=0}^{n-1} f\left(\left\{ \frac{k\mathbf{z}}{n} + \boldsymbol{\Delta} \right\} \right),$$

where n is the number of points in the rule [14, 22]. That is, the approximation is the average of the values of f over the set of quadrature points

$$P_n = \{\{k\mathbf{z}/n + \boldsymbol{\Delta}\},\ 0 \le k \le n-1\}. \tag{1}$$

We assume that each coordinate of $\mathbf{z}$ is relatively prime to n. Thus, the set of admissible generating vectors $\mathbf{z}$ is $(\mathbb{Z}'_n)^d$ where $\mathbb{Z}'_n$ denotes the set of integers in $\{1, \ldots, n-1\}$ that are relatively prime to n. This set has cardinality $|\mathbb{Z}'_n| = \varphi(n)$, where φ is Euler's totient function. When n is prime, we have $\varphi(n) = n-1$. (In this paper, $|\cdot|$ denotes the cardinality if the argument is a set and the absolute value if it is a real number.)

It is well known that the integration error $|Q_{n,d}(f) - I_d(f)|$ can be bounded in different ways by the product of a discrepancy measure of the point set used in the rule and the corresponding measure of variation $V(f)$ of the function f [5, 12, 14]. The discrepancy measure considered in this paper is the weighted star discrepancy, an L_∞-type discrepancy defined below. This measure (with weights) was also used in [20], for example. The L_∞-type discrepancies are of interest in particular because their corresponding $V(f)$ is finite under weaker smoothness assumptions than the other types of L_p discrepancies found in the literature.

For any $\mathbf{x} = (x_1, \ldots, x_d) \in [0,1)^d$ and an arbitrary point set P_n, we define the local star discrepancy at $\mathbf{x}$ by

$$\mathrm{disc}(\mathbf{x}, P_n) := \frac{|[\mathbf{0}, \mathbf{x}) \cap P_n|}{n} - \prod_{j=1}^{d} x_j.$$

For any set of indices $\mathfrak{u} \subseteq \mathcal{D} := \{1, \ldots, d\}$, let $\mathbf{x}_\mathfrak{u}$ denote the vector in $[0,1]^{|\mathfrak{u}|}$ that contains the components of $\mathbf{x}$ whose indices belong to $\mathfrak{u}$, and let $(\mathbf{x}_\mathfrak{u}, \mathbf{1}) \in [0,1]^d$ be the vector whose j-th component is x_j if $j \in \mathfrak{u}$ and 1 if $j \notin \mathfrak{u}$. The *weighted star discrepancy* of P_n is defined by

$$D^*_{\boldsymbol{\gamma}}(P_n) := \max_{\mathfrak{u} \subseteq \mathcal{D}} \boldsymbol{\gamma}_\mathfrak{u} \sup_{\mathbf{x}_\mathfrak{u} \in [0,1]^{|\mathfrak{u}|}} |\mathrm{disc}((\mathbf{x}_\mathfrak{u}, \mathbf{1}), P_n)|, \tag{2}$$

where $\boldsymbol{\gamma}_\mathfrak{u} > 0$ is the weight given to $\mathfrak{u}$, for each $\mathfrak{u} \subseteq \{1, \ldots, d\}$. The weight $\boldsymbol{\gamma}_\mathfrak{u}$ should reflect the importance of the component that corresponds to the subset $\mathfrak{u}$ of coordinates, in the ANOVA decomposition of f [12, 13, 17]. Then a weighted variant of the Koksma-Hlawka inequality [14, 20] gives

$$\left| Q_{n,d}(f) - I_d(f) \right| \le D^*_{\boldsymbol{\gamma}}(P_n) \times V(f) \tag{3}$$

if $V(f)$ exists, where

$$V(f) = \sum_{\mathfrak{u} \subseteq \mathcal{D}} \boldsymbol{\gamma}_{\mathfrak{u}}^{-1} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}}{\partial \mathbf{x}_{\mathfrak{u}}} f((\mathbf{x}_{\mathfrak{u}}, \mathbf{1})) \right| \mathrm{d}\mathbf{x}_{\mathfrak{u}}$$

measures the variation of f. Note that this variation, and the worst-case error bound (3), can be finite only for bounded integrands f.

Later in this paper we shall assume that the weights $\boldsymbol{\gamma}_{\mathfrak{u}}$ have the following product form, as was done in [4, 25] and several other places:

$$\boldsymbol{\gamma}_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j, \tag{4}$$

where $\gamma_j > 0$ is the weight associated with coordinate j. We also assume that $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_d$. We say that a family of lattice rules indexed by n and with point sets P_n has *low-discrepancy* if $D^*_{\boldsymbol{\gamma}}(P_n) = O(n^{-1+\delta})$ for any $\delta > 0$.

The weight γ_j reflects the importance of coordinate j in the discrepancy measure. We should take it larger [smaller] if we believe that f depends more [less] on the jth coordinate of $\mathbf{u}$. For example, in situations where we have a low effective dimension in the truncation sense [2, 12], the first few random numbers have much more importance than the other ones for the realization of $f(\mathbf{u})$ and the importance decreases quickly with j. The weights should then decrease accordingly. In other applications where we have low effective dimension in the superposition sense [2, 12], all coordinates of $\mathbf{u}$ have similar importance, but the importance of a subset $\mathfrak{u}$ in the ANOVA decomposition decreases quickly with the cardinality of $\mathfrak{u}$. By taking equal weights $\gamma_j = \gamma < 1$, we assume implicitly that this decrease is geometric in $|\mathfrak{u}|$.

Given that no efficient algorithm is available for computing $D^*_{\boldsymbol{\gamma}}(P_n)$, we will follow the common practice of using an easily computable upper bound on $D^*_{\boldsymbol{\gamma}}(P_n)$ as a figure of merit. This upper bound $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ will be written in terms of the generating vector $\mathbf{z}$ and will be independent of the shift $\boldsymbol{\Delta}$.

Shifted lattice rules (often randomly shifted) for the approximation or estimation of integrals over the unit cube have been used for a long time [3, 13, 22]. Shifted lattice rules with low discrepancy have been constructed in [9, 10, 23, 24, 27], for example, under the assumption that n was prime, but with a different definition of discrepancy that required stronger smoothness assumptions on the integrands. Moreover, in [10, 23, 27] the authors considered the average discrepancy over all shifts, whereas in [24], the shift was optimized to minimize the discrepancy. In our case, the bounds are valid for an arbitrary (worst-case) shift. Under the additional condition that $\sum_{j=1}^{\infty} \gamma_j < \infty$ (the weights are summable), the $O(n^{-1+\delta})$ bound is also independent of the dimension d (we have strong tractability).

Rank-1 lattice rules that achieve this convergence rate can be found by a greedy-type component-by-component (CBC) construction. The CBC construction algorithm has been used by several authors recently, including [7, 9, 20, 24]. It defines the generating vector $\mathbf{z}$ coordinate by coordinate. At the sth step, for $s = 2, \ldots, d$, it selects the sth coordinate of $\mathbf{z}$ as (one of) the integer(s) in $\mathbb{Z}'_n$ for which the discrep-

ancy bound $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ is minimized for the corresponding s-dimensional point set P_n. Once a coordinate is selected, it is never modified again. With this algorithm, one computes the discrepancy (2) for at most $|\mathbb{Z}'_n|d = \varphi(n)d$ generating vectors rather than for all $\varphi(n)^d$ possibilities, which would take an excessive amount of time when d is large.

When n is large, one could also think of sampling only a limited number of integers from $\mathbb{Z}'_n$ and then picking the best one, at each step of the CBC algorithm, instead of trying all $\varphi(n)$ possibilities. This randomized CBC construction was already proposed in [26], where the authors also suggested to check if the retained rule had a figure of merit at least as small as the (known) average over all $\varphi^d(n)$ possibilities. This method is much simpler and can be faster than standard CBC when n is large. If it also provides a $\mathbf{z}$ whose figure of merit is practically as good with high probability, then one might prefer it for its simplicity. Our empirical investigations indicate that this is indeed the case. They also indicate that we can do almost as well with a very naive method that just generates, say, r generating vectors $\mathbf{z}$ randomly and uniformly in $(\mathbb{Z}'_n)^d$, and picking the best one. To get proper insight on those issues, we approximate (empirically) the distribution function of the figure of merit $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ for a random $\mathbf{z}$, and for a $\mathbf{z}$ constructed from the randomized CBC construction, for given choices of r, n, d, and the weights.

We also examine the behavior of the figure of merit as a function of n, for the best lattice, and on average for a random lattice, for "reasonable" values of n (under a million). We see that unless d is very small or the weights γ_{u} converge very quickly as a function of $|\mathrm{u}|$ (which is almost equivalent), the observed rate of decrease in that range of values of n is much slower than n^{-1}. This type of reality check for the behavior of the figure of merit is important from the practical viewpoint. Similar illustrations of the behavior as a function of n have been given earlier in [18, 19] for a bound on the classical (unweighted) star discrepancy for other types of low-discrepancy point sets, namely those produced by the Halton, Sobol', and Niederreiter-Xing sequences.

Another important reality check, given the slow decrease of the bound in the practical range of values of n, is to see how close is the bound from the true discrepancy. We provide a partial answer by computing the true discrepancy for the cases where we can (for $d = 2$ for all n, and for $d = 3$ with small n) and comparing it with the bound. The gap turns out to be significant, and seems to increase with the dimension. Our conclusion discusses the practical meaning of this fact.

2 Bounds on the Weighted Star Discrepancy

Proposition 1 below provides a conveniently computable bound on the discrepancy (2) for any given $\mathbf{z}$. This proposition puts together some known results to provide a bound for a discrepancy with general weights and arbitrary shift of the lattice.

Proposition 1. *For any $n > 1$, any $\mathbf{z} \in (\mathbb{Z}'_n)^d$, arbitrary weights γ_{u}, and point set P_n defined in (1), we have*

$$D^*_{\boldsymbol{\gamma}}(P_n) \le \bar{D}^*_{\boldsymbol{\gamma}}(P_n) := \sum_{\mathfrak{u}\subseteq\mathcal{D}} \boldsymbol{\gamma}_{\mathfrak{u}} \left(1-(1-1/n)^{|\mathfrak{u}|}\right) + \frac{1}{2} e^2_{n,d}(\mathbf{z}), \tag{5}$$

where the sum on the right does not depend on $\mathbf{z}$*,*

$$e^2_{n,d}(\mathbf{z}) = \sum_{\mathfrak{u}\subseteq\mathcal{D}} \boldsymbol{\gamma}_{\mathfrak{u}} \left(\frac{1}{n} \sum_{k=0}^{n-1} \prod_{j\in\mathfrak{u}} \left(1 + \sum_{-n/2<h\le n/2}{}' \frac{e^{2\pi \mathrm{i} h k z_j/n}}{|h|} \right) - 1 \right),$$

and $\sum'$ *denotes the sum over the nonzero integers* h*.*

For the case of product weights of the form (4), we can also write

$$\sum_{\mathfrak{u}\subseteq\mathcal{D}} \boldsymbol{\gamma}_{\mathfrak{u}} \left(1-(1-1/n)^{|\mathfrak{u}|}\right) = \prod_{j=1}^{d} \beta_j - \prod_{j=1}^{d} (\beta_j - \gamma_j/n) = O(n^{-1}) \tag{6}$$

and

$$e^2_{n,d}(\mathbf{z}) = \frac{1}{n} \sum_{k=0}^{n-1} \prod_{j=1}^{d} \left(\beta_j + \gamma_j \sum_{-n/2<h\le n/2}{}' \frac{e^{2\pi \mathrm{i} h k z/n}}{|h|} \right) - \prod_{j=1}^{d} \beta_j, \tag{7}$$

where $\beta_j = 1+\gamma_j$*.*

Proof. The inequality (5) is obtained by applying Lemma 6 of [4] to bound the maximum in (2) for each $\mathfrak{u}$, and then bounding the maximum over $\mathfrak{u}$ by the sum over $\mathfrak{u}$. Then it suffices to note that $e^2_{n,d}(\mathbf{z})$ is the same as $\sum_{\mathfrak{u}\subseteq\mathcal{D}} \boldsymbol{\gamma}_{\mathfrak{u}} R_1(\mathbf{h}, \mathbf{1}, n, \mathfrak{u})$, where R_1 is defined in Lemma 6 of [4]. We recognize that bounding the max by the sum is likely to give a loose bound, but this is the standard approach used by other authors [4, 7, 21]. The second part follows by identical arguments as in [7].

For the remainder of the paper, we assume that the weights have the product form (4). Observe that the term (6) decays linearly with n for any choice of weights. It does not depend on $\mathbf{z}$, so it is just a constant part in the figure of merit $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$. In fact, the bound in (5) depends essentially on the quantity $e^2_{n,d}(\mathbf{z})$. We will now focus on this quantity. We compute it as explained in [21, page 657], via asymptotic expansions from [8] and by storing the products during the construction.

The average of this quantity over all admissible generating vectors is

$$M_{n,d,\boldsymbol{\gamma}} = \frac{1}{\varphi^d(n)} \sum_{\mathbf{z}\in(\mathbb{Z}'_n)^d} e^2_{n,d}(\mathbf{z}).$$

The corresponding average value of $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ is

$$\Gamma_{n,d,\boldsymbol{\gamma}} = \prod_{j=1}^{d} \beta_j - \prod_{j=1}^{d} (\beta_j - \gamma_j/n) + \frac{M_{n,d,\boldsymbol{\gamma}}}{2}.$$

If n is prime, we have the following explicit formula [7]:

$$M_{n,d,\boldsymbol{\gamma}} = \frac{1}{n}\prod_{j=1}^{d}(\beta_j + \gamma_j S_n) + \frac{n-1}{n}\prod_{j=1}^{d}\left(\beta_j - \gamma_j \frac{S_n}{n-1}\right) - \prod_{j=1}^{d}\beta_j, \tag{8}$$

where $S_n = \sum'_{-n/2<h\le n/2} 1/|h|$. For the general case where n can be composite, we do not have an explicit formula for $M_{n,d,\boldsymbol{\gamma}}$, but an upper bound is given in [21]. In any case, regardless of n (unless it is very small), $M_{n,d,\boldsymbol{\gamma}}$ is well approximated by its dominant term

$$M_{n,d,\boldsymbol{\gamma}} \approx T_{n,d,\boldsymbol{\gamma}} := \frac{1}{n}\prod_{j=1}^{d}(\beta_j + \gamma_j S_n) = O(n^{-1+\delta}),$$

with an approximation error of $O((\log\log(n+1))/n)$. In numerical experiments with small values of n, in which the average was computed explicitly, it has been observed that $T_{n,d,\boldsymbol{\gamma}}$ was actually always larger than $M_{n,d,\boldsymbol{\gamma}}$. For the case where n is prime, it is easy to prove that $T_{n,d,\boldsymbol{\gamma}}$ is always larger [7].

It is also known (see Lemma 3 in [4]) that $T_{n,d,\boldsymbol{\gamma}} = O(n^{-1+\delta})$ for any $\delta > 0$ when d is fixed and $n \to \infty$, and uniformly over d when the weights γ_j are summable. Then a simple argument that the best is at least as good as the average leads to the following result (see also [4, Theorem 7]):

Proposition 2. *For any n there is a generating vector $\mathbf{z}$ such that the weighted star discrepancy of the corresponding shifted lattice rule satisfies*

$$D^*_{\boldsymbol{\gamma}}(P_n) = O(n^{-1+\delta})$$

for any $\delta > 0$, where the implied constant depends on δ and the weights, but does not depend on n and on the shift. If the weights are summable, then the implied constant can also be taken independent of the dimension d.

3 The CBC Construction and Random Search Methods

The CBC algorithm constructs the generating vector $\mathbf{z} = (z_1, z_2, \ldots, z_d)$ as follows. We suppose that $n \ge 2$, and that d and the weights are fixed.

CBC construction algorithm:
Let $z_1 := 1$;
For $s = 2, 3, \ldots, d$, find $z_s \in \mathbb{Z}'_n$ that minimizes $e^2_{n,s}(z_1, z_2, \ldots, z_s)$, defined in (7), while $z_1, \ldots, z_{s-1}$ remain unchanged.

The following result, proved in [21, Theorem 2], combined with Proposition 1, implies that the algorithm produces a generating vector $\mathbf{z}$ whose corresponding weighted star discrepancy (2) has the same order of magnitude as the bound provided by Proposition 2.

Proposition 3. *This CBC algorithm returns a vector $\mathbf{z}$ for which*

$$e_{n,d}^2(\mathbf{z}) \le \frac{1}{n}\prod_{j=1}^{d}\left(\beta_j + \alpha\gamma_j \ln n\right) = O(n^{-1+\delta}),$$

where $\alpha > 0$ is an absolute constant.

This bound implies that for a fixed d, we have $e_{n,d}^2(\mathbf{z}) = O(n^{-1+\delta})$. And if the weights are summable, then this holds uniformly in d. The costs for the CBC construction algorithm using the fast implementation of [15, 16], is $O(nd\log n)$ computing time and $O(n)$ space for storage (see also [21] for further details). A more straightforward implementation requires $O(dn^2)$ time. The fast CBC construction is actually based on the fast Fourier transform, allowing to reduce the typical $O(n^2)$ operations required by a matrix-vector multiplication to $O(n\log n)$. However, the $O(n\log n)$ term has a larger hidden constant, which depends on the respective implementations. We did not use the fast Fourier transform in our implementation; it would have taken much more time to implement it than what we were ready to spend, and our goal was not really to compare speeds.

The following randomized CBC construction algorithm is simpler to implement and can reduce the computing cost by examining only a small number of integers $z_s \in \mathbb{Z}_n'$ (chosen at random) at each step.

Randomized CBC construction algorithm (R-CBC):
Let $z_1 := 1$;
For $s = 2, 3, \ldots, d$,
  choose r integers z_s at random in $\mathbb{Z}_n'$, and select the one that minimizes $e_{n,s}^2(z_1, z_2, \ldots, z_s)$, while $z_1, \ldots, z_{s-1}$ remain unchanged.

A similar algorithm was proposed in [26], with the additional feature that for any given s, new integers z_s are examined until $e_{n,s}^2(z_1, \ldots, z_s)$ is less than the average $M_{n,d,\boldsymbol{\gamma}}$. An even simpler (and more naive) algorithm is a uniform random search in $(\mathbb{Z}_n')^d$, as follows (we can also stop only when $e_{n,d}^2(\mathbf{z}) \le M_{n,d,\boldsymbol{\gamma}}$):

Uniform random search algorithm:
Choose r vectors $\mathbf{z}$ at random in $(\mathbb{Z}_n')^d$, and select the one that minimizes $e_{n,d}^2(\mathbf{z})$.

In the next section, we compare empirically the performance of these three algorithms, in terms of the figures of merit of the returned vectors $\mathbf{z}$. We know a priori that the CBC construction should provide better figures of merit, but is the difference really significant?

4 Empirical Assessments

Our aim in this section is to explore the behavior of the discrepancy bound (or figure of merit) $\bar{D}_{\boldsymbol{\gamma}}^*(P_n)$ defined in (5), empirically, from various angles, for n not exceeding one million. We first examine its distribution function when $\mathbf{z}$ is drawn

uniformly from $(\mathbb{Z}'_n)^d$ (so the figure of merit is a random variable). This distribution function F is defined by $F(x) = \mathbb{P}[\bar{D}^*_{\boldsymbol{\gamma}}(P_n) \le x]$. We find that typically, this distribution is positively skewed, and the median is smaller than the mean $\Gamma_{n,d,\boldsymbol{\gamma}}$, so the probability $q_{n,d,\boldsymbol{\gamma}}$ of a value smaller than the mean is more than 1/2 (often more than 0.75). This implies that a vector $\mathbf{z}$ whose figure of merit is smaller than the mean (and thus satisfies the bound in Proposition 2) is easy to find by uniform random search. By applying this algorithm with r trials, the probability of finding such a vector is $1-(1-q_{n,d,\boldsymbol{\gamma}})^r$. With $q_{n,d,\boldsymbol{\gamma}} = 0.75$, this probability is approximately $1-9.5\times10^{-7}$ for $r=10$, and $1-6.2\times10^{-61}\approx 1$ for $r=100$, for example. That is, finding a $\mathbf{z}$ smaller than the mean is very easy, even with the most naive method.

To approximate the distribution function F and the density f of $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ for a random $\mathbf{z}$, we generated a sample of $r=10^5$ generating vectors $\mathbf{z}$ and computed the empirical distribution function $\hat{F}$ of the r realizations. We also computed a kernel estimator $\hat{f}$ of the corresponding density, using a Gaussian kernel, with the bandwidth selected as suggested in [6, pages 308–309]. Note that these density estimates inflate the tails (compare with the empirical distribution). This tail inflation can be reduced by reducing the bandwidth, but then the curve becomes less smooth. Thus, the empirical distribution seems to give a better idea of the distribution. The computations and plots were made with SSJ [11]. We did this with various choices of n, d, and the weights, and the shape of the empirical distribution (with proper scaling) was very much the same in all cases.

Figure 1 (upper panel) gives an illustration with $n=32749$ (a prime number), $d=10$, and weights $\gamma_j = 1/j^2$ for all j. Here the figure of merit obtained via the CBC construction was 0.14996, while the best one among the 10^5 random $\mathbf{z}$ was 0.15048, the median was 0.15247 (indicated by the leftmost vertical line), the empirical mean was 0.15367, and theoretical mean $\Gamma_{n,d,\boldsymbol{\gamma}}$ is 0.15386 (the rightmost vertical line). Here the probability $q_{n,d,\boldsymbol{\gamma}}$ is slightly more than 0.75.

The lower panel of the same figure provides another illustration with $n=1048573$, $d=5$, and weights $\gamma_j = 1/4$ for all j. Here, we know a priori that the weighted discrepancy cannot exceed 1/4. The CBC construction gave a figure of merit of 0.01646, the best random vector had 0.01649, the median was 0.01668, and both the empirical and theoretical means were 0.01683. The probability $q_{n,d,\boldsymbol{\gamma}}$ is again very close to 0.75.

Table 1 summarizes the figures of merit obtained for other values of n, d, and the weights. The CBC algorithm usually returned a value slightly smaller than the best values from the two randomized methods. However, in absolute terms, the values returned by all three algorithms are typically very close to each other. The difference between the corresponding error bounds can be deemed negligible. Moreover, those best values are not much smaller than the median and the mean. We also observe that unless the weights decrease very quickly with j or are all small, the discrepancy bounds become larger than the trivial bound of $\gamma_{\max} = \max_j \gamma_j$ already in 5 dimensions, even for $n = 1048573 \approx 2^{20}$. For $\gamma_j = 1$, $d=10$ and $n=131071$, the best bound is approximately 4.15×10^8. Any discrepancy bound (or figure of merit) larger than $\gamma_{\max}$ is in fact totally useless, because the discrepancy itself is never larger than $\sup_j \gamma_j \le 1$.

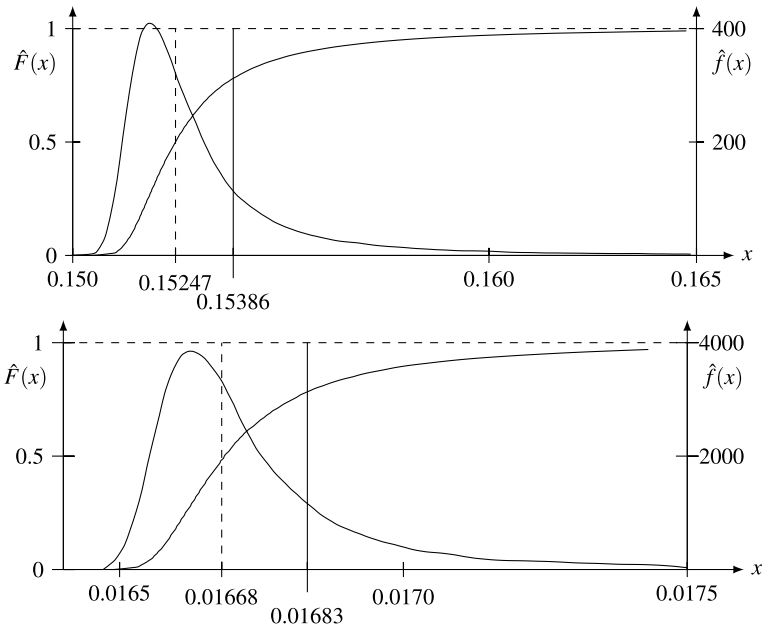


Fig. 1 Estimated distribution function $\hat{F}$ (increasing curve) and density $\hat{f}$ (other curve) of the figure of merit $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ defined in (5). Above: $n = 32749$, $d = 10$, and $\gamma_j = 1/j^2$. Below: $n = 1048573$, $d = 5$, and $\gamma_j = 1/4$. The solid and dashed vertical lines indicate the mean and the median, respectively.

We made some experiments to estimate the distribution function of the (random) figure of merit returned by the randomized CBC algorithm. The minimal value was usually slightly larger than that returned by the CBC algorithm, but on rare occasions the R-CBC algorithm did a bit better. The latter can happen in situations where the CBC path is not optimal and the randomized method finds a better one by chance. As an illustration, for $n = 32749$, $d = 10$ and weights $\gamma_j = 1/j^2$ (as in the upper panel of Figure 1 and in row 6 of the table), the CBC construction gave a figure of merit of 0.1499629, and the randomized CBC algorithm gave figures of merit between 0.1500882 and 0.1504918 (from 1000 independent runs of the algorithm), with a median of 0.1502565 and a mean of 0.1502607. Figure 2 shows the estimated distribution function and density of the value returned by the R-CBC algorithm, from the 1000 runs. The density is slightly asymmetric and is concentrated in a narrow interval.

Figure 3 shows the best figure of merit obtained by the CBC construction, as a function of n, in a log-log scale, for various choices of the weights and dimension. These plots provide good insight on how the best bound behaves as a function of n in general. We see that unless the dimension is very small (e.g., 2 or 3, as in the upper panel) or the weights converge extremely fast (as in the bottom panel), the observed convergence rate for n up to one million is much slower than $O(1/n)$. That is, we

Table 1 Values of the figure of merit obtained by the CBC algorithm (CBC), the randomized CBC construction with $r = 5$ (R-CBC), and uniform random search with $r = 10^5$ (Best-R), for various choices of weights, d, and n. The last three columns also provide the median and the mean of the empirical distribution (Median and Mean), and the exact mean $\Gamma_{n,d,\boldsymbol{\gamma}}$, for comparison.

γ_j	d	n	CBC	R-CBC	Best-R	Median	Mean	$\Gamma_{n,d,\boldsymbol{\gamma}}$
$1/j^2$	5	8009	0.0719	0.0729	0.0723	0.0759	0.0787	0.0792
		32749	0.0292	0.0294	0.0293	0.0306	0.0317	0.0318
		131071	0.0114	0.0115	0.0115	0.0119	0.0123	0.0123
		1048573	0.0026	0.0026	0.0026	0.0027	0.0028	0.0028
"	10	8009	0.3125	0.3135	0.3137	0.3192	0.3223	0.3229
		32749	0.1499	0.1500	0.1504	0.1524	0.1536	0.1538
		131071	0.0689	0.0691	0.0691	0.0698	0.0702	0.0703
		1048573	0.0198	0.0198	0.0199	0.0200	0.0201	0.0201
"	20	8009	0.7315	0.7347	0.7337	0.7400	0.7432	0.7439
		32749	0.3934	0.3943	0.3943	0.3967	0.3980	0.3981
		131071	0.2021	0.2025	0.2025	0.2033	0.2039	0.2039
		1048573	0.0686	0.0686	0.0687	0.0688	0.0689	0.0689
$1/j$	3	8009	0.0733	0.0762	0.0735	0.0802	0.0874	0.0881
		32749	0.0266	0.0270	0.0266	0.0290	0.0317	0.0319
		131071	0.0093	0.0095	0.0093	0.0101	0.0111	0.0112
		1048573	0.0018	0.0018	0.0018	0.0020	0.0022	0.0022
"	5	32749	1.1037	1.1078	1.1044	1.1179	1.1249	1.1252
		131071	0.4728	0.4759	0.4732	0.4782	0.4811	0.4811
		1048573	0.1206	0.1211	0.1207	0.1217	0.1222	0.1222
1	3	8009	0.4036	0.4206	0.4033	0.4349	0.4659	0.4682
		32749	0.1480	0.1498	0.1476	0.1590	0.1712	0.1718
		131071	0.0526	0.0536	0.0522	0.0562	0.0608	0.0610
		1048573	0.0096	0.0107	0.0103	0.0111	0.0120	0.0120
1/4	3	8009	0.0081	0.0083	0.0081	0.0090	0.0100	0.0102
		32749	0.0029	0.0029	0.0029	0.0032	0.0036	0.0036
		131071	0.0010	0.0010	0.0010	0.0011	0.0012	0.0012
"	5	32749	0.1568	0.1583	0.1572	0.1601	0.1617	0.1618
		131071	0.0660	0.0663	0.0662	0.0671	0.0678	0.0678
		1048573	0.0165	0.0166	0.0165	0.0167	0.0168	0.0168

know that the slope of all the curves in the figure converges to -1, asymptotically, when $n \to \infty$, but for the observed values of n, the curves have a concave shape and the slope is often much less than -1. This behavior is typical and was observed in our plots for several other parameters as well.

In the case of the fast-decaying weights $\gamma_j = 1/2^j$ (bottom panel), increasing the dimension has eventually almost no visible effect: the upper curve appears a bit thicker because it contains the curves for $d = 10$, 20, and 40, which almost overlap. The reason is simple: the weights decrease so fast that the high-dimensional coordi-

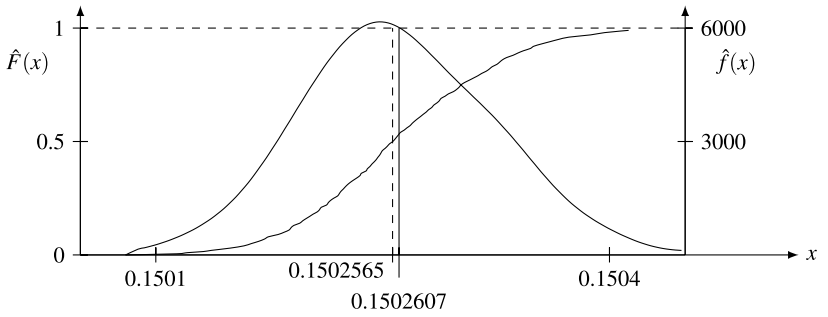


Fig. 2 Empirical distribution $\hat{F}$ and density estimate $\hat{f}$ of $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ for the point set P_n returned by the R-CBC algorithm with $r = 5$, when $n = 32749$, $d = 10$, and $\gamma_j = 1/j^2$ (based on 1000 replicates). The vertical lines indicate the median (dashed) and the mean.

nates have a negligible contribution to the discrepancy. This effect also appears to a lesser extent for $\gamma_j = 1/j^2$ (middle panel).

In the upper panel, we also show the true value of the discrepancy $D^*_{\boldsymbol{\gamma}}(P_n)$ for the same point sets P_n, for $\gamma_j = 1$, for the cases where we have been able to compute it, namely for $d = 2$ and for $d = 3$ with small n. For this, we implemented the algorithm given in [1] (the required work increases as $O(n^d)$). One can see that the upper bound $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ (the figure of merit) is much larger than the true discrepancy $D^*_{\boldsymbol{\gamma}}(P_n)$, and the gap seems to increase rapidly with the dimension d. As an illustration, when $d = 2$ and $n = 32749$, the true discrepancy is 0.00014 and the bound is 0.00382. For $d = 3$ and $n = 8009$, the true discrepancy is 0.0017, whereas the bound is 0.4036, which is about 240 times larger! This gap is orders of magnitude larger than the gain of the CBC algorithm over the two randomized methods (see Table 1), even with small r.

Conclusion

We have provided a reality check on the practical meaning of known upper bounds on the weighted star discrepancy, the convergence rate of these bounds for the best rank-1 lattices, the distribution of the value of the bound for a random rank-1 lattice, and the behavior of the CBC construction algorithm as well as randomized algorithms based on these bounds. We saw that the best achievable value of the upper bound is typically not much smaller than the average value over all admissible generating vectors, and that a value close to the minimum can easily be found by a simple randomized algorithm and even by naive random search. Our experiments confirm the popular belief that these discrepancy bounds are not always tight and that they may converge rather slowly. Even though the best achievable value of the bound decreases as $O(n^{-1+\delta})$ for any $\delta > 0$ asymptotically, its rate of decrease is typically slower than this asymptotic rate for reasonable values of n. For n less than

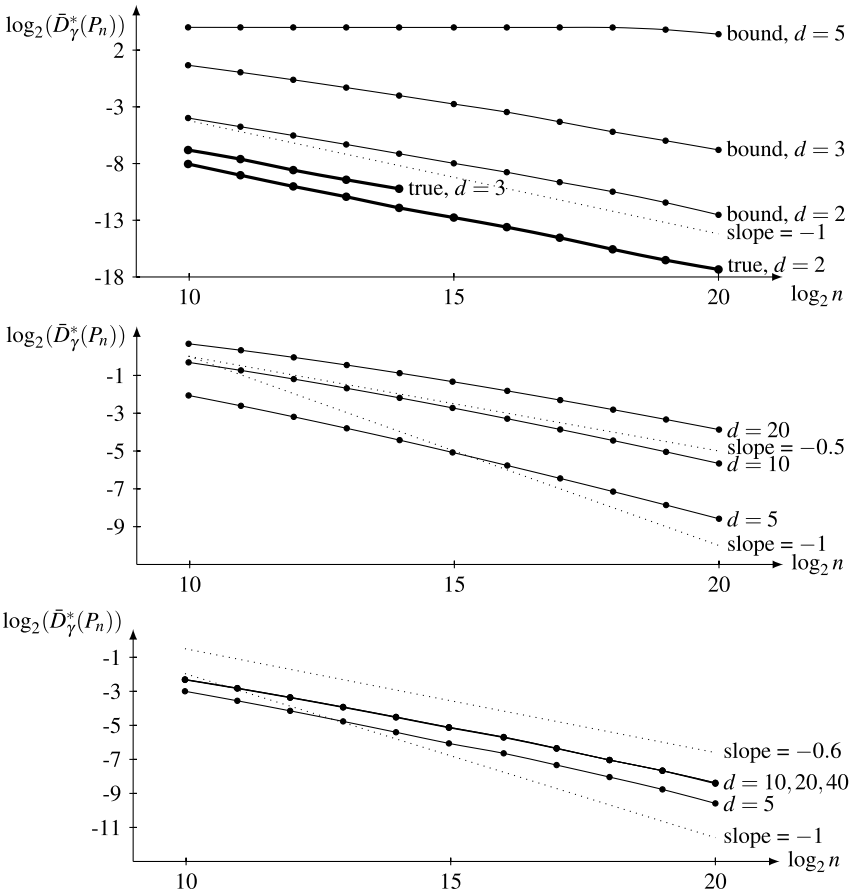


Fig. 3 The best bound $\bar{D}^*_{\boldsymbol{\gamma}}(P_n)$ obtained with the CBC construction as a function of n, in a log-log scale, in d dimensions, when $\gamma_j = 1$ for all j (upper panel), $\gamma_j = 1/j^2$ (middle panel), and $\gamma_j = 1/2^j$ (lower panel). In the upper panel, we also have the true weighted star discrepancy $D^*_{\boldsymbol{\gamma}}(P_n)$ (the thick lines) for the cases where we were able to compute it (for $d = 2$ and for $d = 3$ with small n). Selected slopes are shown for reference. All these functions have a slope of -1 asymptotically when $n \to \infty$.

a million (say), the bound turns out to be practically useless unless the dimension d is very small or the weights converge very quickly. One implication might be that other types of discrepancies, which can be computed exactly and converge faster than the bounds considered here [5, 12], provide more appropriate figures of merit from the practical viewpoint.

Acknowledgements This research has been supported by NSERC-Canada grant No. ODGP0110050 and a Canada Research Chair to P. L'Ecuyer. The authors are grateful to Richard Simard who helped with the computations.

References

1. Bundschuh, P., Zhu, Y.: A method for exact calculation of the discrepancy of low-dimensional finite point sets I. Abh. Math. Sem. Univ. Hamburg **63**, 115–133 (1993)
2. Caflisch, R.E., Morokoff, W., Owen, A.: Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. The Journal of Computational Finance **1**(1), 27–46 (1997)
3. Cranley, R., Patterson, T.N.L.: Randomization of number theoretic methods for multiple integration. SIAM Journal on Numerical Analysis **13**(6), 904–914 (1976)
4. Hickernell, F., Niederreiter, H.: The existence of good extensible rank-1 lattices. J. Complexity **19**, 286–300 (2003)
5. Hickernell, F.J.: What affects the accuracy of quasi-Monte Carlo quadrature? In: H. Niederreiter, J. Spanier (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1998, pp. 16–55. Springer-Verlag, Berlin (2000)
6. Hörmann, W., Leydold, J., Derflinger, G.: Automatic Nonuniform Random Variate Generation. Springer-Verlag, Berlin (2004)
7. Joe, S.: Construction of good rank-1 lattice rules based on the weighted star discrepancy. In: H. Niederreiter, D. Talay (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 181–196. Springer (2006)
8. Joe, S., Sloan, I.: On computing the lattice rule criterion R. Math. Comp **59**, 557–568 (1992)
9. Kuo, F.Y.: Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces. Journal of Complexity **19**(3), 301–320 (2003)
10. Kuo, F.Y., Wasilkowski, G.W., Waterhouse, B.J.: Randomly shifted lattice rules for unbounded integrands. Journal of Complexity **22**(5), 630–651 (2006)
11. L'Ecuyer, P.: SSJ: A Java Library for Stochastic Simulation (2008). Software user's guide, available at http://www.iro.umontreal.ca/~lecuyer
12. L'Ecuyer, P.: Quasi-Monte Carlo methods with applications in finance. Finance and Stochastics, **13**(3), 307–349 (2009)
13. L'Ecuyer, P., Lemieux, C.: Variance reduction via lattice rules. Management Science **46**(9), 1214–1235 (2000)
14. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
15. Nuyens, D., Cools, R.: Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces. Math. Comp **75**, 903–920 (2006)
16. Nuyens, D., Cools, R.: Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points. J. Complexity **22**, 4–28 (2006)
17. Owen, A.B.: Latin supercube sampling for very high-dimensional simulations. ACM Transactions on Modeling and Computer Simulation **8**(1), 71–102 (1998)
18. Schlier, C.: Discrepancy behaviour in the non-asymptotic regime. Appl. Numer. Math. **50**, 227–238 (2004)
19. Schlier, C.: Error trends in Quasi-Monte Carlo integration. Comp. Phys. Comm. **159**, 93–105 (2004)
20. Sinescu, V., Joe, S.: Good lattice rules based on the general weighted star discrepancy. Mathematics of Computation **76**(258), 989–1004 (2007)
21. Sinescu, V., Joe, S.: Good lattice rules with a composite number of points based on the product weighted star discrepancy. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 645–658. Springer (2008)
22. Sloan, I.H., Joe, S.: Lattice Methods for Multiple Integration. Clarendon Press, Oxford (1994)
23. Sloan, I.H., Kuo, F.Y., Joe, S.: Constructing randomly-shifted lattice rules in weighted Sobolev spaces. SIAM Journal on Numerical Analysis **40**, 1650–1665 (2002)
24. Sloan, I.H., Kuo, F.Y., Joe, S.: On the step-by-step construction of quasi-Monte Carlo rules that achieve strong tractability error bounds in weighted Sobolev spaces. Mathematics of Computation **71**, 1609–1640 (2002)

25. Sloan, I.H., Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals. Journal of Complexity **14**, 1–33 (1998)
26. Wang, X., Sloan, I.H.: Efficient weighted lattice rules with applications to finance. SIAM Journal on Scientific Computing **28**(2), 728–750 (2006)
27. Waterhouse, B.J., Kuo, F.Y., Sloan, I.H.: Randomly shifted lattice rules on the unit cube for unbounded integrands in high dimensions. Journal of Complexity **22**(1), 71–101 (2006)

Ergodic Estimations of Upscaled Coefficients for Diffusion in Random Velocity Fields

Nicolae Suciu and Călin Vamoş

Abstract Upscaled coefficients for diffusion in ergodic velocity fields are derived by summing up correlations of increments of the position process, or equivalently of the Lagrangian velocity. Ergodic estimations of the correlations are obtained from time averages over finite paths sampled on a single trajectory of the process and a space average with respect to the initial positions of the paths. The first term in this path decomposition of the diffusion coefficients corresponds to Markovian diffusive behavior and is the only contribution for processes with independent increments. The next terms describe memory effects on diffusion coefficients until they level off to the value of the upscaled coefficients. Since the convergence with respect to the path length is rather fast and no repeated Monte Carlo simulations are required, this method speeds up the computation of the upscaled coefficients over methods based on long-time limit and ensemble averages by four orders of magnitude.

1 Introduction

Direct Monte Carlo estimations of diffusion coefficients by averaging over ensembles of realizations of the process and taking the large time limit often constitute a numerical challenge in simulation studies. Owing to the ergodicity of the process, the numerical burden can be reduced to a great extent.

For ergodic transport processes, the ensemble average of the observables can be estimated by the arithmetic mean of the observables resulting from repeated simulations of diffusion, done for the same realization of velocity field and for point-like

Nicolae Suciu
Mathematics Department, Friedrich-Alexander University of Erlangen-Nuremberg, Erlangen, Germany
url: http://www1.am.uni-erlangen.de/~suciu/

Călin Vamoş
Tiberiu Popoviciu Institute of Numerical Analysis, Romanian Academy, Cluj-Napoca, Romania
url: http://www.ictp.acad.ro/vamos/

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_40,

sources with different locations uniformly distributed over large enough spatial domains [8]. Even though promising results can be obtained in this way, this "ergodic simulations method" depends on the quality of the numerically generated processes: worse are the ergodic properties of the latter, greater is the number of simulations for different initial positions required to achieve the desired accuracy. For instance, results of two-dimensional ergodic simulations of diffusion in random velocity fields presented in [8], obtained by averaging over a moderate number of 121 initial positions, indicated the approach of ergodic estimates to the corresponding ensemble averages but the accuracy of the upscaled diffusion coefficients was not yet satisfactory. To increase the accuracy, more initial positions should be considered, which would increase the computational costs and render the ergodic simulations less competitive with respect to the direct Monte Carlo approach.

The "path decomposition method" proposed in Sect. 2 provides ergodic estimates of diffusion coefficients by sums of correlations of increments on paths of increasing but finite lengths on a single trajectory of the diffusion process. To increase the accuracy, the path correlations are further averaged over a large number of paths. In the case of diffusion in random velocity fields, the upscaled diffusion coefficients can be explicitly written in terms of correlations of the Lagrangian velocity, sampled on the same trajectory (Sect. 3). Summing up autoregressive processes of order 1, diffusion processes with memory and exactly computable diffusion coefficients are constructed and are used to test the path decomposition method (Sect. 4). Finally, in Sect. 5, the new approach is applied to a problem of diffusion in random velocity fields which occurs in modeling groundwater contamination. Some conclusions are drawn in Sect. 6.

2 Path Decomposition of Diffusion Coefficients

Let $X = \{X_t, t \geq 0\}$ be a stochastic process of mean zero starting from $X_0 = 0$. If after a transient time X behaves as a normal diffusion, the diffusion coefficient is related to the expectation $E\{X_t^2\}$ by the Einstein formula [1, 15]

$$D = \lim_{t \longrightarrow \infty} \frac{1}{2t} E\{X_t^2\}. \tag{1}$$

Dividing a finite time interval $[0, t]$ in S subintervals of equal length τ, $t = S\tau$, the position X_t and its square X_t^2 can be expressed in terms of random variables $\delta X_s = X_{s\tau} - X_{(s-1)\tau}$,

$$X_t = \sum_{s=1}^{S} \delta X_s,\ X_t^2 = \sum_{s=1}^{S} (\delta X_s)^2 + 2 \sum_{r=1}^{S-1} \sum_{s=1}^{S-r} \delta X_s \delta X_{s+r}.$$

Defining the (time average) correlations

$$\rho(r) = \frac{1}{S-r}\sum_{s=1}^{S-r} \delta X_s \delta X_{s+r}$$

one obtains

$$X_t^2 = S\rho(0) + 2\sum_{r=1}^{S-1}(S-r)\rho(r). \tag{2}$$

The expectation of (2) is a discrete form of Taylor's formula (see e.g. [3]). If X is an ergodic process, the diffusion coefficient can be estimated as $D = \frac{1}{2S\tau}X_t^2$, without taking the expectation of X_t^2. For some lattice gas systems and molecular dynamics simulations the hydrodynamic long-time limit in (1) can be indeed replaced by $\tau \longrightarrow 0$ (fixed t) and expectations can be approximated by time averages [15]. The space average of (2) over N different paths of length S sampled on a single trajectory of the process X further improves the estimates. Replacing the correlations in (2) by their space averages $\varrho = \frac{1}{N}\sum_{n=1}^{N}\varrho_n$ and using (1) one obtains the following estimation of the diffusion coefficient for a given length S of the paths:

$$D = \frac{1}{2\tau}\rho(0) + \frac{1}{\tau}\sum_{r=1}^{S-1}\left(1-\frac{r}{S}\right)\rho(r). \tag{3}$$

For a Markovian diffusive behavior, as for instance random walk or Wiener process, the diffusion coefficients is solely defined by the first term of (3). The correlations $\rho(r)$, with $r > 0$, describe the transient regime of diffusion processes with memory. One expects that if the process takes place in a random environment, (3) also estimates the up-scaled diffusion coefficient provided the environment has suitable ergodic properties.

3 Upscaled Coefficients for Diffusion in Random Fields

The advection-dispersion model for transport in heterogeneous media such as turbulent atmosphere, plasmas, or aquifers [7] is a vector valued process described by the Itô equation $d\mathbf{Y}(t) = \mathbf{V}(\mathbf{Y}(t))dt + d\mathbf{W}(t)$, where $\mathbf{V}$ is a realization of a random velocity field and $\mathbf{W}$ a Wiener process of mean zero and variance $2D_0t$. The diagonal ll components of the upscaled diffusion coefficient are given by the long time limit of $D_{ll}^* = \frac{1}{2t}\Sigma_{ll}$, where $\Sigma_{ll} = \langle E\{X_l^2\}\rangle$ is the average over the ensemble of velocity realizations of the expected squared displacement from the mean plume center of mass, $X_l = Y_l - \langle E\{Y_l\}\rangle$ [5]. If the Eulerian velocity field $\mathbf{V}$ is statistically homogeneous and has suitable smoothness properties which ensure the existence of pathwise unique solutions of the Itô equation, then the Lagrangian velocity field $\mathbf{V}(\mathbf{Y})$ is also statistically homogeneous [6]. Under these conditions, from the Itô-Euler scheme $\delta X_{l,s} = u_l(\mathbf{Y}_{s-1})\tau + \delta W_{l,s}$, where $u_l = V_l - \langle E\{V_l\}\rangle$ is the velocity fluctuation, one obtains

$$\langle E\{\delta X_{l,s}\delta X_{l,s+r}\}\rangle = \delta_{r,0}2D_0\tau + \tau^2\rho_{u,ll}(r),$$

where

$$\rho_{u,ll}(r) = \langle E\{u_l(Y_{l,s-1})_l u_l(Y_{l,s-1+r})\}\rangle.$$

If the Eulerian velocity field $\mathbf{V}$ has a finite correlation range then it is ergodic [14]. Assuming that the Lagrangian velocity field inherits the ergodicity property, one expects that the velocity correlation above can be approximated by an average over N paths of length S on a single trajectory of the process $\mathbf{X}(t)$,

$$\rho_{u,ll}(r) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{S-r}\sum_{s=1}^{S-r} u_l(Y_{l,n+s-1})u_l(Y_{l,n+s-1+r})$$

and from (3) one obtains

$$D_{ll}^{*} = D_0 + \frac{\tau}{2}\rho_{u,ll}(0) + \tau\sum_{r=1}^{S-1}\left(1-\frac{r}{S}\right)\rho_{u,ll}(r). \tag{4}$$

In Sect. 5 below it is shown that the heuristic formula (4) yields quite good estimates of upscaled coefficients, which, in turn, supplies a numerical indication for the ergodicity of the Lagrangian velocity field.

The path correlations decomposition (4) of the diffusion coefficients resembles a discrete Green-Kubo formula [1] and has been used in [9] to calibrate parameters for massive parallel simulations of transport in saturated aquifers [4, 5].

4 Exactly Computable Diffusion Coefficients

The trajectory of a diffusion process can be simulated numerically by summing up realizations of a Gaussian white noise, $Z = \{Z_n, n = 0, 1, 2, ...\}$, of mean zero and constant variance σ_0^2. This is a particular case of the more general algorithm for autoregressive processes of order 1 (AR(1))

$$X_n = \phi X_{n-1} + Z_n. \tag{5}$$

AR(1) processes are often used to generate synthetic time series with given statistical properties [10, 13]. For $\phi = 1$, (5) defines a discrete time diffusion process, for $\phi = 0$ it reduces to the white noise, for $0 < \phi < 1$ one obtains a correlated AR(1) process, and for $-1 < \phi < 0$ one obtains an anticorrelated AR(1) [12]. For $|\phi| < 1$ the infinite AR(1) processes, $\{X_n, n = 0, \pm 1, \pm 2, ...\}$, are stationary and, as follows from (5), their expectation is zero, $E\{X_n\} = 0$, and the variance and the autocovariance are given by the relations

$$\sigma^2 = \frac{\sigma_0^2}{1-\phi},\ \gamma(r) = E\{X_n X_{n+r}\} = \sigma^2\phi^r. \tag{6}$$

Since in numerical simulations AR(1) processes always have a finite length, there is a transient regime until the process reaches a stationary state. If the first term X_1 in (5) is a realization of a random variable whose variance equals the variance σ^2 of the theoretical AR(1) of infinite length, then the transient regime is completely eliminated and one obtains an AR(1) process of finite length with the variance and autocovariance (6) of the infinite AR(1) process [12].

In the following we show that AR(1) processes can be used to generate diffusion processes with memory. This can be achieved by replacing in the algorithm for the generation of the discrete time diffusion processes the Gaussian noise by the AR(1) noise $\{X_n\}$ of mean zero and constant variance, defined by the relation (5) with $|\phi| < 1$. The process $\{Y_n\}$ starting from $Y_1 = 0$ is then generated by

$$Y_n = Y_{n-1} + X_n = \sum_{s=1}^{n} X_s. \tag{7}$$

From (7) it follows that $E\{Y_n\} = 0$, and using (2) and the stationarity of the AR(1) noise $\{X_n\}$ one finds that the variance is given by a Taylor formula,

$$\sigma_{Y_n}^2 = E\{Y_n^2\} = n\sigma^2 + 2\sum_{r=1}^{n-1}\sum_{s=1}^{n-r} E\{X_s X_{s+r}\} = n\sigma^2 + 2\sum_{r=1}^{n-1}\sum_{s=1}^{n-r}\gamma(r)$$
$$= n\sigma^2 + 2\sum_{r=1}^{n-1}(n-r)\gamma(r).$$

According to Einstein's relation (1), the diffusion coefficient is

$$D = \lim_{n\longrightarrow\infty}\frac{\sigma_{Y_n}^2}{2n\tau} = \frac{\sigma^2}{2\tau} + \frac{1}{\tau}\lim_{n\longrightarrow\infty}\sum_{r=1}^{n-1}(1-\frac{r}{n})\gamma(r) = \frac{\sigma^2}{2\tau} + \frac{1}{\tau}\lim_{n\longrightarrow\infty}\sum_{r=1}^{n-1}\gamma(r),$$

and can be effectively computed, for $|\phi| < 1$, by using the explicit form of the autocovariance (6),

$$D = \frac{\sigma^2}{2\tau} + \frac{\sigma^2}{\tau}\lim_{n\longrightarrow\infty}\phi\frac{1-\phi^{n-1}}{1-\phi} = \frac{\sigma^2}{2\tau} + \frac{\sigma^2}{\tau}\frac{\phi}{1-\phi}.$$

Denoting by $D_0 = \sigma^2/(2\tau)$ the diffusion coefficient of the process generated by a white noise of variance σ^2, the diffusion coefficient of the process generated by a general AR(1) noise with $|\phi| < 1$ takes the explicit form

$$D = D_0\left(1 + 2\frac{\phi}{1-\phi}\right). \tag{8}$$

Since the finite limit (8) exists, the process (7) generated by a AR(1) noise is a diffusion process.

To test the path decomposition method against exact results given by (8), we consider a variance $\sigma^2 = 1$ m^2 and a time step $\tau = 0.5$ s, so that the diffusion

coefficient for the case $\phi = 0$ takes on the value $D_0 = \sigma^2/(2\tau) = 1$ m^2/s. Using $N = 50{,}000$ paths with fixed length $S = 100$ we computed the correlations $\rho(r)$ entering the path decomposition (3) for diffusion processes generated by AR(1) noises with different values of the parameter ϕ.

Figure 1 shows the progress of partial sums in (3), $D(r)$, for increasing r. One can see that the path decomposition (3) converges, after a transient regime, to the exact coefficients (8), represented by straight lines in left panel of Figure 1. The right panel Figure 1 shows that the super-diffusive transient regime is expanded as ϕ tends to 1 when, according to (8), D tends to infinity.

The computation time, of about 4 seconds in all cases presented in Figure 1, is still of the order of the CPU time required by direct estimations of diffusion coefficients with the Einstein's relation (2). However, the path decomposition method demonstrates its advantages when applied to diffusion processes in random velocity fields, as shown by the example presented in the next section.

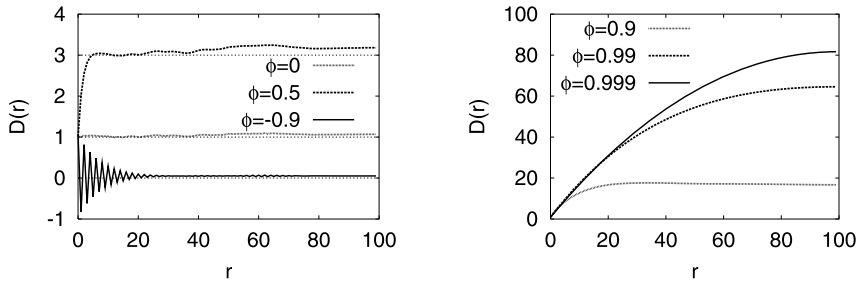


Fig. 1 Diffusion coefficients for processes generated by AR(1) noises with ϕ = 0, 0.5, -0.9 (left), and with ϕ = 0.9, 0.99, and 0.999 (right).

5 Estimated Upscaled Coefficients for Diffusion in Random Velocity Fields

We consider a two dimensional process of diffusion in random velocity fields, used to model a typical situation of contaminant transport in groundwater systems, consisting of a superposition of a Wiener process with diffusion coefficient $D_0 = 0.01$ m^2/day and a random velocity field of mean $U = 1$ m/day, exponentially correlated, with finite correlation range $\lambda = 1$ m and variance of the order of 0.01 m/day [4, 5, 6]. The diffusion processes were simulated by simultaneously tracking 10^{10} computational particles in every velocity realization with the "global random walk"

algorithm [11]. The simulations and the Monte Carlo estimations of the upscaled diffusion coefficients are described in detail in [4].

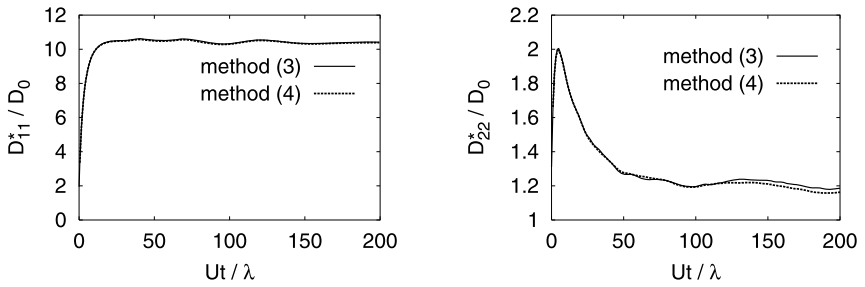


Fig. 2 Longitudinal (left) and transverse (right) upscaled diffusion coefficients estimated by the methods (3) and (4).

To estimate upscaled diffusion coefficients we considered a time step $\tau = 0.5$ day and $N = 2{,}000{,}000$ paths of fixed length $S = 400$, on a single trajectory of the process, which required a CPU time of about 65 minutes. Figure 2 shows the partial sums in (3) and (4) as functions of dimensionless time Ut/λ. The longitudinal direction is that of the mean velocity. The upscaled coefficients computed by the method (4) are practically identical with those given by (3). This result supplies a numerical indication that the Lagrangian velocity inherits the ergodic properties of the Eulerian velocity field.

For comparison with the Monte Carlo estimations presented in [4, 6], we used a reduced number of paths, $N = 200{,}000$, and we estimated mean values and standard deviations of the upscaled coefficients by averaging over 100 independent estimations (3) performed for different realizations of the velocity field. Figure 3 shows that the path decomposition method yields estimations of the upscaled coefficients close to the Monte Carlo results in the limits of one standard deviation.

The Monte Carlo convergence with the number of velocity realizations R of a statistical estimate f can be assessed with an estimation error defined, for given increment ΔR of the number of realizations, by

$$E(f(R)) = |f(R + \Delta R) - f(R)|. \tag{9}$$

Figure 4 shows results on the convergence of the mean coefficients $\langle D^*_{ll} \rangle$ estimated by the Monte Carlo simulations presented in Figure 3. For longitudinal quantities ($l = 1$) the errors (9) were computed at dimensionless times of 1 and 20 (where the ensemble averages were most sensitive to R) while for transverse ones ($l = 2$) the dimensionless times were 1 and 10. The number of realizations was gradually increased by $\Delta R = 32$ in both cases. For $R \geq 1000$ realizations E falls under the desired accuracy threshold of $D_0/2$. The errors (9) for the path decomposition method

(3), computed at the dimensionless time of 200 with an increment of the number of realizations of $\Delta R = 5$, are presented in Figure 5. The results show that the oscillations of both mean values $\langle D^*_{ll} \rangle$ and standard deviations $SD(D^*_{ll})$, $l = 1, 2$ are already smaller than D_0 for about 10 realizations.

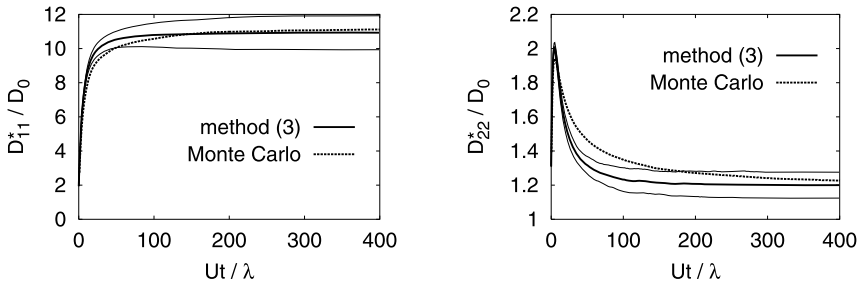


Fig. 3 Longitudinal (left) and transverse (right) upscaled diffusion coefficients estimated by method (3) and by Monte Carlo simulations. The thick full lines represent the arithmetic average $\langle D^*_{ll} \rangle$ of 100 estimations (3) and the thin full lines represent $\langle D^*_{ll} \rangle \pm SD(D^*_{ll})$.

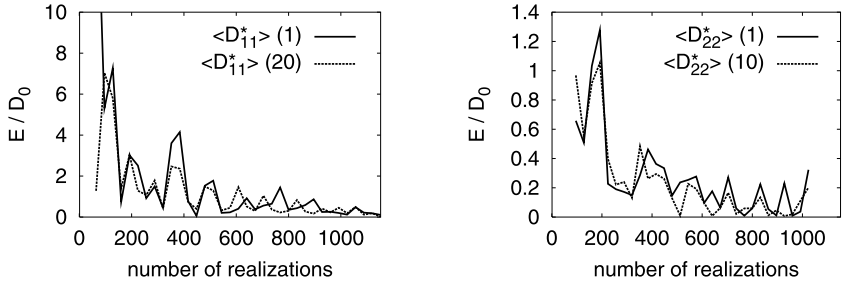


Fig. 4 Convergence of the mean longitudinal (left) and transverse (right) upscaled diffusion coefficients for the Monte Carlo simulations.

A single estimation (3) of the upscaled diffusion coefficients lasted about 6 minutes, so that the total computation time for the statistical estimations presented in Figure 3 was of about 10 CPU hours. Thus the estimates with the decomposition (3) were thousand times faster than those obtained with the Monte Carlo approach (1024 realizations and about 12 CPU hours / realization, on the same computing platform). Moreover, since quite good estimates of the upscaled coefficients can

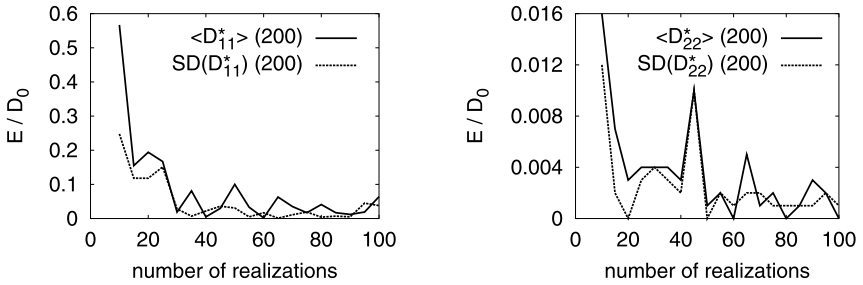


Fig. 5 Convergence of the mean and standard deviation of the longitudinal (left) and transverse (right) upscaled diffusion coefficients estimated by the path decomposition method (3).

be obtained by using a single trajectory of the process (see Figure 3), the effective speed up of the computations is of about four orders of magnitude.

6 Conclusions

The path decomposition method proposed in this paper avoids the cumbersome tasks of taking the long time hydrodynamic limit and of averaging over large statistical ensembles, like in usual Monte Carlo approaches. Instead, the diffusion coefficients are computed as contributions of correlations of the process increments on paths of finite lengths, sampled on a single trajectory.

The advantage of this method is obvious for processes with memory, consisting of diffusion in random velocity fields. As compared with the Monte Carlo approach, the speed up of the computations can be as large as four orders of magnitude. The comparison presented in Figure 3 shows that the path decomposition method yields fairly good estimations of the upscaled coefficients.

Since the method allows fast estimation of the diffusion coefficients, it can be used to calibrate more complex simulations for large scale transport problems. A promising field of applications is the Lagrangian approach for turbulent dispersion, where the Lagrangian velocity samples are readily available as solutions of Itô-type equations [2].

The decomposition of the upscaled diffusion coefficients in terms of velocity correlations is another feature, which, as illustrated by the results presented in Figure 2, is useful in investigations on relationships between ergodic properties of the Eulerian and Lagrangian velocity fields.

Acknowledgements The work presented in this paper was supported by the Deutsche Forschungsgemeinschaft grant SU 415/1-2 and the Romanian Ministry of Education and Research grant 2-CEx06-11-96.

References

1. Harayama, T., Klages, R., Gaspard, P.: Deterministic diffusion in flower shape billiards. Phys. Rev. E., **66**, 02611 (2002)
2. Kurbanmuradov, O., Sabelfeld, K.: Lagrangian stochastic models for turbulent dispersion in the atmospheric boundary layer. Boundary-Layer Metheorology, **97**, 191-218, (2000)
3. Mandelbrot, B. B., Wallis, J. R.: Noah, Joseph, and operational hydrology. Water Resour. Res., **4**(5), 909-918 (1968)
4. Suciu, N., Vamoş, C., Vanderborght, J., Hardelauf, H., Vereecken, H.: Numerical investigations on ergodicity of solute transport in heterogeneous aquifers. Water Resour. Res., **42**, W04409, doi:10.1029/2005WR004546 (2006)
5. Suciu, N., Vamoş, C., Eberhard, J.: Evaluation of the first-order approximations for transport in heterogeneous media. Water Resour. Res., **42**, W11504, doi:10.1029/2005WR004714 (2006)
6. Suciu, N., Vamoş, C., Vereecken, H., Sabelfeld, K., Knabner, P.: Memory effects induced by dependence on initial conditions and ergodicity of transport in heterogeneous media. Water Resour. Res., **44**, W08501, doi:10.1029/2007WR006740 (2008)
7. Suciu, N., Vamoş, C., Vereecken, H., Sabelfeld, K., Knabner, P.: Itô equation model for dispersion of solutes in heterogeneous media. Revue D'Analyse Numérique et de Théorie D'Approximation, **37**(2), 221-238, (2008)
8. Suciu, N., Vamoş, C., Sabelfeld, K.: Ergodic simulations of diffusion in random velocity fields. In: Keller, A., Heinrich, S., and Niederreiter, H. (eds) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 659-668. Springer, Heidelberg (2008)
9. Vamoş, C.: C++ codes 'descin.cpp' and 'descre.cpp'. Unpublished manuscript (2001)
10. Vamoş, C.: Automatic algorithm for monotone trend removal. Phys. Rev. E **75**, 036705, doi: 10.1103/PhysRevE.75.036705 (2007)
11. Vamoş, C., Suciu, N., Vereecken, H.: Generalized random walk algorithm for the numerical modeling of complex diffusion processes. J. Comput. Phys., **186**(2), 527–544 (2003)
12. Vamoş, C., Şoltuz, Ş. M., Crăciun, M.: Order 1 autoregressive processes of finite length. E-print arXiv:079.2963v1 [physics.data-an], 19 Sep 2007 (2007)
13. Vamoş, C., Crăciun, M.: Serial correlation of detrended time series. Phys. Rev. E **78**, 036707 (2008)
14. Yaglom, A. M.: Correlation Theory of Stationary and Related Random Functions, Volume I: Basic Results, Springer, New York (1987)
15. Ying, S. C., Vattulainen, J., Merikoski, J., Hjelt, T., Als-Nissila, T.: Memory expansion for diffusion coefficients. Phys. Rev. B, **58**(4), 2170-2178 (1998)

Green's Functions by Monte Carlo

David White and Andrew Stuart

Abstract We describe a new numerical technique to estimate Green's functions of elliptic differential operators on bounded open sets. The algorithm utilizes SPDE based function space sampling techniques in conjunction with Metropolis-Hastings MCMC. The key idea is that neither the proposal nor the acceptance probability require the evaluation of a Dirac measure. The method allows Green's functions to be estimated via ergodic averaging. Numerical examples in both 1D and 2D, with second and fourth order elliptic PDE's, are presented to validate this methodology.

1 Introduction

Green's functions play a central role in many areas of mathematics and statistics: they provide fundamental solutions used as the basic building block to construct solutions of inhomogeneous PDEs; they act as the representers for reproducing kernel Hilbert spaces; and the covariance function of a Gaussian random field may be viewed as the Green's function for the precision operator.

This article describes a new numerical technique to estimate Green's functions of elliptic differential operators on bounded open sets. The algorithm utilizes SPDE based function space sampling techniques [3] in conjunction with Metropolis-Hastings MCMC [7]. The key idea is that neither the proposal nor the acceptance probability require the evaluation of a Dirac measure. The method estimates Green's functions via an ergodic average of sampled functions. The basic framework is that probability measures defined on a Hilbert space [4] are sampled using techniques designed specifically for this infinite dimensional setting.

Mathematics Institute
Zeeman Building
University of Warwick
Warwickshire, UK
url: http://www.maths.warwick.ac.uk

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_41,

In Section 2 it is shown that a Gaussian measure on function space can be constructed with mean corresponding to the desired Green's function. The algorithm samples functions from this measure and the sample mean provides a good estimate of the Green's function of interest.

This idea is validated numerically by examples with known analytic solutions in Section 3. Green's functions of second and fourth order elliptic operators in both 1D and 2D are presented in this section.

The concept presented here is independent of the particular methodology used for sampling function space. Section 4 considers an alternative proposal for a function space MCMC sampling method and demonstrates the algorithm in this context via one of the numerical examples shown in Section 3.

2 Function Space Sampling and Algorithm Description

Sampling from a measure on function space is central to this algorithm. It is shown below that if the measure and function space sampling algorithm are constructed appropriately, then the ergodic average of suitably sampled functions converges to a Green's function of choice.

We begin by defining a probability measure π on a Hilbert space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$. The measure π is constructed so that its mean is the desired Green's function on a bounded open set $D \subset \mathbb{R}^n$. Throughout this article the measure, π, has a Radon-Nikodym derivative with respect to a Gaussian measure π_0:

$$\frac{d\pi}{d\pi_0} \propto \exp\left(-\Theta\left(x\right)\right). \tag{1}$$

We chose a mean zero Gaussian reference measure $\pi_0 = \mathcal{N}\left(0, \mathcal{C}\right)$ where $\mathcal{C}$ is a trace class, self adjoint, positive definite operator on $\mathcal{H}$ so that $\pi_0\left(\mathcal{H}\right) = 1$. For equation (1) we require that $\Theta : \mathcal{H} \to \mathbb{R}$ is π_0-measurable and integrable. The definition of π_0 may be combined with equation (1) to write the following informal expression for the target density as:

$$\pi\left(dx\right) \propto \exp\left(-\Theta\left(x\right) - \frac{1}{2}\left\langle x, \mathcal{C}^{-1}x\right\rangle\right) dx. \tag{2}$$

This expression has no rigorous status because there is no infinite dimensional equivalent of Lebesgue measure. However it conveys intuition about the measure which may be useful to the reader.

The algorithmic ideas presented in this paper apply to sampling general measures π of the form given by equation (1), in the case where π_0 is Gaussian [3]. However we now look at a particular choice of Θ arising in the application to the construction of Green's functions.

In order to obtain the Green's function of some elliptic differential operator $\mathcal{L}$ incorporating the boundary conditions through its domain, the covariance operator of the reference Gaussian measure is selected to be $\mathcal{C} = -\mathcal{L}^{-1}$. The function Hilbert

space $\mathcal{H} = L^2(D)$ and Θ is chosen to be $\Theta(x) = \langle x, \delta_s \rangle$. Here δ_s is the Dirac delta function centered at $s \in D$.

By completing the square in equation (2) we deduce that $\pi \sim \mathcal{N}(\hat{x}, \mathcal{C})$ where $\hat{x} = -\mathcal{C}\delta_s$ or

$$\mathcal{L}\hat{x} = \delta_s. \tag{3}$$

Then π is absolutely continuous with respect to π_0 whenever $\hat{x} \in \mathrm{Im}\left(\mathcal{C}^{\frac{1}{2}}\right)$, by the Feldman-Hajek Theorem [4].

The measure π is invariant for the SPDE:

$$\frac{dx}{dt} = \mathcal{L}x - \delta_s + \sqrt{2}\frac{dw}{dt}. \tag{4}$$

Lemma 2.2 in [5] shows that this equation is well defined and ergodic.

Since equation (4) is invariant with respect to the target measure, π, the Green's function of interest may be obtained by time marching the SPDE and averaging the sampled functions to estimate $\hat{x}$. In practice, this requires direct evaluation of Dirac delta functions, which introduces further complications.

This difficulty may be circumvented as follows. Instead of equation (4) consider:

$$\frac{dx}{dt} = \mathcal{L}x + \sqrt{2}\frac{dw}{dt}. \tag{5}$$

Lemma 2.2 in [5] shows that equation (5) is π_0 invariant rather than π invariant. However equation (5) does not involve a Dirac delta function. If we use proposals based on discretising equation (5) then the Metropolis-Hastings accept/reject mechanism may be used to create a π invariant Markov chain.

Discretising the SPDE (5) using Crank-Nicolson gives equation:

$$\frac{y-x}{\Delta t} = \frac{\mathcal{L}x + \mathcal{L}y}{2} + \sqrt{\frac{2}{\Delta t}}\xi \tag{6}$$

where ξ represents a spatial white noise which is independent of the current state x. Re-arranging we obtain the proposal y given a current function x:

$$(2 - \Delta t\mathcal{L})\, y = (2 + \Delta t\mathcal{L})\, x + \sqrt{8\Delta t}\xi. \tag{7}$$

The acceptance probability for the proposal y given x is $\alpha(x, y)$ where:

$$\alpha(x, y) = \exp(0 \wedge R(x, y)), \tag{8a}$$

$$R(x, y) = \Theta(x) - \Theta(y) = x(s) - y(s). \tag{8b}$$

Notice that the acceptance probability only requires computation of the difference between the current and proposed functions at a single point. This is computationally inexpensive to evaluate and at no point in the algorithm do we need to evaluate a delta function.

This completes our explanation concerning the construction of the measure and the sampling algorithm. A more detailed explanation of function space sampling algorithms can be found in [3], for non-Gaussian measures, using SPDE which are invariant for π given by (1) see [5] and [6].

3 Examples and Numerics

This section numerically validates the above algorithm via three examples. The examples have known analytic solutions derived by techniques described in [1], [2] and [8].

Throughout this section we use the standard notation for Sobolev spaces H^s of functions with s square integrable derivatives, possibly incorporating periodic $\left(H^s_{\text{per}}\right)$ or Dirichlet $\left(H^s_0\right)$ boundary conditions.

Example 1. As a first example, we consider the elliptic differential operator $\mathcal{L} = \frac{d^2}{du^2}$ with Dirichlet boundary conditions:

$$\mathcal{L} = \frac{d^2}{du^2} \text{ on } (0,1) \tag{9a}$$

$$\text{with } D(\mathcal{L}) = \left\{x \in H^1_0(0,1) \cap H^2(0,1)\right\}. \tag{9b}$$

It may be shown theoretically that the Green's function for $\mathcal{L}$ is:

$$G(u,s) = \begin{cases} s(u-1) & \forall s \le u \\ u(s-1) & \forall s > u. \end{cases} \tag{10}$$

Figure 1 shows a numerical estimate of the Green's function (with $s = 0.3$) using 10^4 burn in steps and 10^5 actual steps of the MCMC method described in this article. A spatial discretisation of $\Delta u = 10^{-3}$ and time step of $\Delta t = 1$ were used here to generate these estimates. The initial function and the last sampled function are also displayed to demonstrate the stochastic origins of the estimate. The estimate appears to be approximately piecewise linear with a minimum at $u = 0.3$. These observed features are in agreement with the theory.

Example 2. The first example is generalised by introducing a second term into the differential operator. Equations (11a) and (11b) show the operator, interval and boundary conditions.

$$\mathcal{L} = \frac{d^2}{du^2} - k^2 \text{ on } (0,1) \tag{11a}$$

$$D(\mathcal{L}) = \left\{x \in H^1_0(0,1) \cap H^2(0,1)\right\} \tag{11b}$$

It may be shown theoretically that the Green's function for $\mathcal{L}$ is:

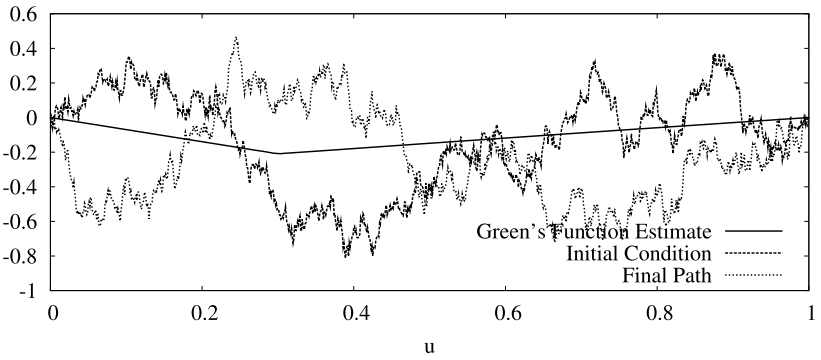


Fig. 1 Green's Function of $\mathcal{L} = \frac{d^2}{du^2}$ with $s = 0.3$.

$$G(u,s) = \begin{cases} \frac{e^{-k}}{2k} \frac{e^{ks} - e^{k(2-s)}}{e^k - e^{-k}} \left(e^{ku} - e^{-ku}\right) & \forall u \leq s \\ \frac{e^{-k}}{2k} \frac{e^{ks} - e^{-ks}}{e^k - e^{-k}} \left(e^{ku} - e^{k(2-u)}\right) & \forall u > s. \end{cases} \tag{12}$$

The algorithm was tested using this problem with $\Delta u = 10^{-3}$, $\Delta t = 10^{-2}$ with a 10% burn in period. The initial condition function was chosen to be identically zero across $[0, 1]$.

Figure 2 shows both (a) the numerical estimates of the Green's functions and (b) the L^2 normed error of these estimates for $k = 10$. (a) shows the Green's function estimate for 10^5 iterations and 10^8 iterations. The former has visible deviations from the correct solution and the latter is visually identical to the true solution. (b) shows the normed error of the algorithm's estimates for 10^5, 10^6, 10^7 and 10^8 samples. It is evident from these plots that the algorithm's output does converge to the correct solution.

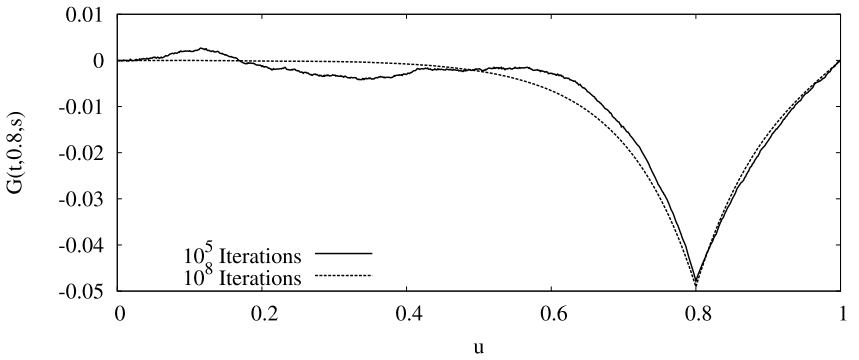
Example 3. We now consider a two-dimensional Green's function, arising from the biharmonic operator. The objective here is to test the algorithm on a higher dimensional problem. Define $\mathcal{L}$ by:

$$\mathcal{L} = -\triangle^2 = -\left(\frac{\partial^2}{\partial u_1^2} + \frac{\partial^2}{\partial u_2^2}\right)^2 \text{ on } E = (0, \ell_1) \times (0, \ell_2), \tag{13a}$$

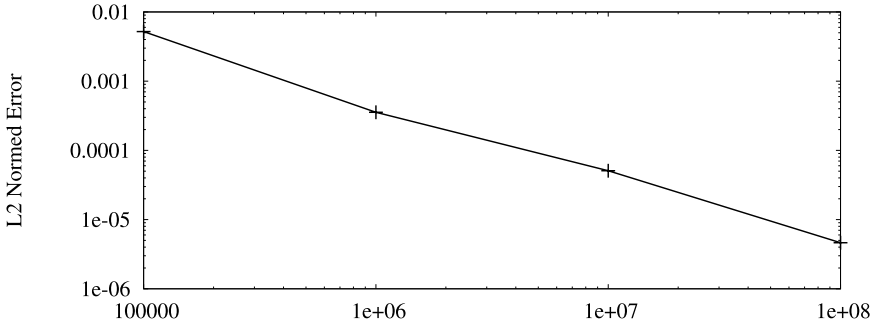
$$D(\mathcal{L}) = \left\{ x \in H^4_{\text{per}}(E) \,\middle|\, \int_E x du = 0 \right\}. \tag{13b}$$

The constraint shown in (13b) is required to uniquely define the Green's function. Without this, any constant may be added to a Green's function of $\mathcal{L}$ to obtain another valid Green's function.

Equation (14) shows the Green's function of this problem, calculated using Fourier series expansions:



(a) Numerical Green's Functions Estimates.



(b) Error in MCMC Estimates.

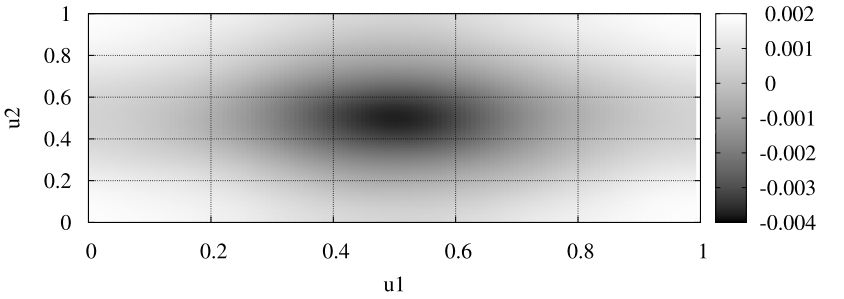
Fig. 2 Green's Function of $\mathcal{L} = \frac{d^2}{du^2} - 100$ with $s = 0.8$.

$$G\left(\underline{u}, \underline{s}\right) = -\frac{1}{16\pi^4 \ell_1 \ell_2} \sum_{(p,q)\in\mathbb{K}} \frac{\exp\left(\frac{2\pi i p(u_1 - s_1)}{\ell_1}\right) \exp\left(\frac{2\pi i q(u_2 - s_2)}{\ell_2}\right)}{\left(\frac{p^2}{\ell_1^2} + \frac{q^2}{\ell_2^2}\right)^2}. \tag{14}$$

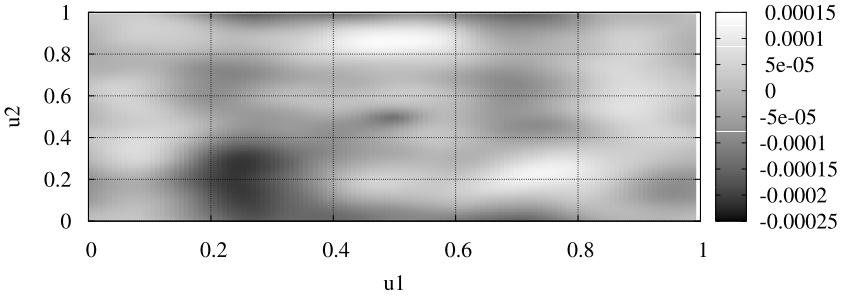
Here $\mathbb{K} = \mathbb{Z}^2 \setminus \{(0,0)\}$.

An FFT based approach was used to calculate proposals on $[0,1]^2$ using (7) and the accept/reject step was based on the function value at $s = \left(\frac{1}{2}, \frac{1}{2}\right)$. The discretisations and time steps used were $\Delta u_1 = \Delta u_2 = \frac{1}{128}$, $\Delta t = 1$ with 3.2×10^6 MCMC steps preceded by 10^5 burn in steps. The initial function was chosen to be identically zero on D.

Figure 3 shows (a) the resulting Green's function estimate and (b) the error estimate. It is clear from these plots that the algorithm functions correctly for this problem.



(a) Numerical Green's Functions Estimate.



(b) Error in MCMC Estimate.

Fig. 3 Green's Function of $\mathcal{L} = -\triangle^2$ with $s = (0.5, 0.5)$.

4 Other Related Proposals

A requirement of the algorithm presented in this paper is the invariance of the measure π to the SPDE stated in equation (4). However, this SPDE is not the only SPDE with this property. An alternative SPDE is (see [6], equation (2.14) and Theorem 3.6):

$$\frac{dx}{dt} = -x + \mathcal{C}\delta_s + \sqrt{2\mathcal{C}}\frac{dw}{dt}. \tag{15}$$

As above, $\mathcal{C}$ is the covariance operator and δ_s is the Dirac delta measure with centre at s.

The general definition of the square root of the self-adjoint operator $\mathcal{C}$ is, of course, through diagonalization in an orthonormal basis, as for matrices. Note however that if ξ is spatial white noise then $\sqrt{\mathcal{C}}\xi$ is simply a draw from the measure π_0; this may sometimes be achieved without constructing $\sqrt{\mathcal{C}}$ explicitly, for example if $\mathcal{C}$ is the covariance operator of Brownian bridge.

Similarly the following SPDE is π_0 invariant:

$$\frac{dx}{dt} = -x + \sqrt{2\mathcal{C}}\frac{dw}{dt}. \tag{16}$$

Similarly to the development in Section 2, this equation may be discretised and used to generate proposals for a Metropolis-Hastings Markov chain. The Crank-Nicolson discretisation gives:

$$\frac{y-x}{\Delta t} = -\frac{x+y}{2} + \sqrt{\frac{2\mathcal{C}}{\Delta t}}\xi \tag{17}$$

which re-arranges into:

$$y = \frac{2-\Delta t}{2+\Delta t}x + \frac{\sqrt{8\Delta t\mathcal{C}}}{2+\Delta t}\xi. \tag{18}$$

The re-arrangement has a special form, the proposal, y, is a linear combination of the current solution x and $\sqrt{\mathcal{C}}\xi$ where ξ is spatial white noise independent of x. In particular, $\sqrt{\mathcal{C}}\xi$ may be drawn directly from π_0. Also notice that:

$$\left(\frac{2-\Delta t}{2+\Delta t}\right)^2 + \frac{8\Delta t}{(2+\Delta t)^2} = 1. \tag{19}$$

This ensures that y is drawn from a measure which is absolutely continuous with respect to the Gaussian reference measure π_0. The acceptance probability for this proposal is again that shown in equations (8a) and (8b), (Theorem 4.1 in [3]).

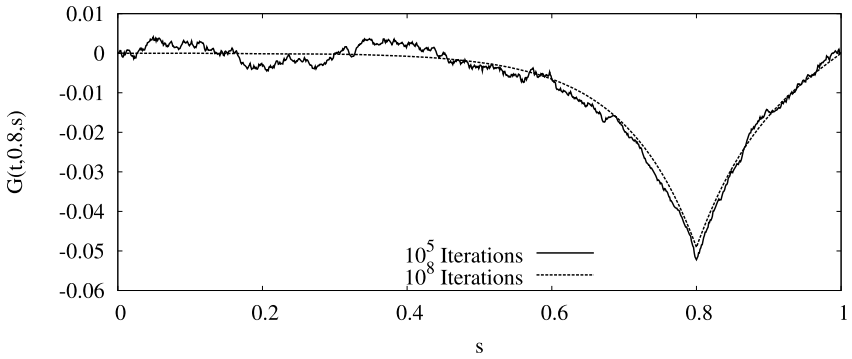
In one dimension, for the operator $\mathcal{L}$ given in Examples 1 and 2, $\sqrt{\mathcal{C}}\xi$ is Brownian bridge measure and draws from it can be made from linear combinations of Brownian motion.

The algorithm was tested using this problem with $\Delta u = 10^{-3}$, $\Delta t = 0.5$ with a 10% burn in period. The initial condition function was chosen to be identically zero across $[0, 1]$.

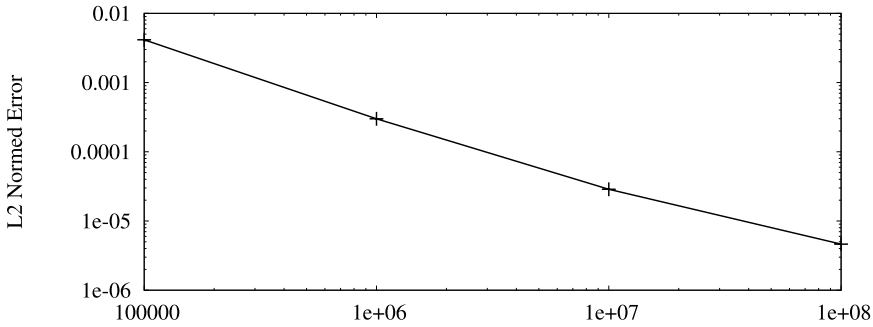
Figure 4 (a) shows estimates of the Green's function problem described in Example 2 and (b) shows the corresponding error norms produced using this alternate proposal. It is clear from these plots that the algorithm converges for this new proposal. This result is particularly interesting in view of the fact that the basic building block is simulation of Brownian motions (and hence Brownian bridges) and no inversion of $\mathcal{L}$ or $\mathcal{L}^{\frac{1}{2}}$ was required to generate this estimate.

5 Conclusions and Further Work

In this article we have introduced a new Metropolis-Hastings based approach to calculating Green's functions of elliptic operators on bounded open sets. It was shown that if the target measure is constructed on a function space in a particular way, the



(a) Numerical Green's Function Estimates.



(b) Error in MCMC Estimates.

Fig. 4 Green's Function of $\mathcal{L} = \frac{d^2}{du^2} - 100$ with $s = 0.8$ using the alternate SPDE proposal of Section 4.

ergodic average of the sampled functions converges to the Green's function of an elliptic differential operator.

The method was validated via three numerical examples, for which the Green's function was known analytically.

In addition to the work presented in this article, it has been observed that this algorithm is trivially parallelisable. Existing direct PDE based methods for calculating Green's functions are serial by necessity. So this potential for parallelism places a considerable advantage over existing methods. This work is on going and more details will appear in [9].

Acknowledgements The work described in this article was developed during MCQMC 08 after listening to many interesting talks about using MCMC to estimate solutions to PDEs. We would like to thank both the organisers and the speakers for providing the catalyst for this work. We would also like to thank the referees for helpful suggestions regarding the presentation of this work.

David White and Andrew Stuart are grateful to EPSRC for financial support. They are also very grateful to both the University of Warwick's High Performance Systems Group and the Centre for Scientific Computing for use of the Condor and IBM clusters.

References

1. Barton, G.: Elements of Greens functions and propagation. Oxford Science Publications. The Clarendon Press Oxford University Press, New York (1989). Potentials, diffusion, and waves
2. Beck, J.V., Cole, K.D., Haji-Sheikh, A., Litkouhi, B.: Heat conduction using Greens functions. Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Hemisphere Publishing Corp., London (1992)
3. Beskos, A., Roberts, G., Stuart, A., Voss, J.: MCMC methods for diffusion bridges. Stochastics and Dynamics 8(3), 319–350 (2008)
4. Da Prato, G., Zabczyk, J.: Stochastic equations in infinite dimensions, Encyclopedia of Mathematics and its Applications, vol. 44. Cambridge University Press, Cambridge (1992)
5. Hairer, M., Stuart, A.M., Voss, J.: Analysis of SPDEs arising in path sampling. II. The nonlinear case. Ann. Appl. Probab. 17(5-6), 1657–1706 (2007)
6. Hairer, M., Stuart, A.M., Voss, J., Wiberg, P.: Analysis of SPDEs arising in path sampling. I. The Gaussian case. Commun. Math. Sci. 3(4), 587–603 (2005)
7. Hastings, W.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57(1), 97–109 (1970)
8. Roach, G.F.: Greens functions, second edn. Cambridge University Press, Cambridge (1982)
9. White, D.: PhD Thesis - Infinite Dimensional Sampling (2009)

Tractability of Multivariate Integration for Weighted Korobov Spaces: My 15 Year Partnership with Ian Sloan

Henryk Woźniakowski

Abstract This paper is intended as a birthday present for Ian Sloan who celebrated his 70th birthday during MCQMC'08 in Montreal. In the first paper with Ian we studied multivariate integration for the unweighted Korobov spaces of smooth and periodic functions equipped with L_∞-type norms expressed in terms of Fourier coefficients. We proved that this problem is intractable and suffers from the curse of dimensionality. To break intractability, weighted Korobov spaces are studied in this paper. Product weights are mainly considered, and finite-order weights are only briefly mentioned. Necessary and sufficient conditions for strong polynomial tractability, polynomial tractability and weak tractability are presented. The necessary and sufficient conditions coincide only for weak tractability, whereas there is a gap between them for strong polynomial and polynomial tractability. In terms of the exponent of strong polynomial tractability, the lower and upper bounds differ at most by a factor of two. Nevertheless, these bounds prove that the exponent of strong polynomial tractability depends on the decay of weights.

1 Introduction

There was a special session during MCQMC'08 in Montreal honoring Ian Sloan on his 70th birthday. I was pleased to mention during this session that our partnership with Ian started about 15 years ago. I met Ian for the first time during the Oberwolfach conference on multivariate integration in 1992. It was obvious from the very beginning that multivariate integration is our favorite research subject, especially the integration of d-variate functions with large d. I visited Ian in Sydney for the first time in 1994, and we now often meet in various places all over the world. Ian has visited me in Poland at least 5 times, and I have visited him in Australia even more often.

Department of Computer Science, Columbia University, and Institute of Applied Mathematics, University of Warsaw
e-mail: henryk@cs.columbia.edu

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5_42,

Our first paper [13] was published in 1997 and I will return to this paper later. In 1998 we published the paper "When are quasi Monte Carlo algorithms efficient for high dimensional integrals?", see [14], where weighted spaces were introduced. Weights monitor the importance of successive variables and groups of variables. We wanted to explain why QMC algorithms are so efficient for many finance applications where functions of 360 or more variables are integrated. We proved that if the weights decay sufficiently fast, then the worst case error of some QMC algorithms does not depend on d or depends only polynomially on d. The idea of weighted spaces has turned out to be quite rich. We are pleased that today many people are studying multivariate integration as well as other multivariate problems defined over weighted spaces.

So far, I have written with Ian 15 papers, some of them jointly with Josef Dick, Frances Kuo, Erich Novak, Xiaoqun Wang and Grzegorz Wasilkowski. On behalf of all collaborators, I wish to say that it has been a great honor (and a lot of fun) to work with Ian. We all wish Ian many fruitful years to come, and we wish ourselves the pleasure of writing many more papers with Ian.

The reader may notice that Ian and I have a nice average of published papers; namely one paper a year. Since we are working on a few more papers, I hope to maintain this average for many years to come. I would be pleased to celebrate Ian's 80th, 90th and 100th birthday, repeating that our average is still one paper a year at each celebration.

Let me now return to our first paper [13], where we studied multivariate approximation for smooth periodic functions. This was done for an unweighted Korobov space with an L_∞-type norm expressed in terms of Fourier coefficients, where all variables and groups of variables play the same role. We proved that this problem is intractable. More precisely, we proved that it is necessary to compute at least 2^d function values if we want to have an error of ε with $\varepsilon < 1$. This means that the problem suffers from the curse of dimensionality. To break intractability and the curse of dimensionality, we must shrink the original unweighted space by introducing decaying weights. The main challenge is to find necessary and sufficient conditions on weights such that the curse of dimensionality is broken. As I mentioned before, many weighted spaces have been analyzed for multivariate integration. Usually this has been done for Hilbert spaces, but the weighted analogue of the Korobov space from [13] has not yet been analyzed.

This is the subject of the present paper. The so-called *product* weights are mainly considered, and *finite-order* weights are mentioned only at the end of the paper. The main result of this paper is to provide necessary and sufficient conditions on weights to guarantee polynomial or weak tractability. More precisely, let $n(\varepsilon, d)$ denote the minimal number of function values needed to find an algorithm using these values with (worst case) error at most ε for the d-variate case. Polynomial tractability means that $n(\varepsilon, d)$ is bounded by a polynomial in ε^{-1} and d, whereas strong polynomial tractability means that this polynomial is independent of d. The exponent of strong polynomial tractability is the minimal (or the infimum of) b for which $n(\varepsilon, d)$ is bounded by a multiple of ε^{-b}. Weak tractability means that $n(\varepsilon, d)$ does not depend exponentially on ε^{-1} and d.

I now explain the proof technique of the paper. Multivariate integration for the Korobov space with the L_∞-type norm is not easier than its analogue for the L_2-type norm. The latter problem has been thoroughly studied and the lower bound results for the L_2 case are also lower bound results for the L_∞ case. Since the weighted Korobov space studied here is not a Hilbert space, the results on decomposable kernels from [9] do not apply. We present another lower bound entirely in terms of the product weights in Theorem 1. This theorem generalizes the result of [13] to the weighted case. The upper error bounds are obtained by showing that lattice rules may also be used in the L_∞ case and their error is related to the error for the L_2 case with appropriately changed parameters. This allows us to present necessary and sufficient conditions for strong polynomial tractability, polynomial tractability and weak tractability. The necessary and sufficient conditions only match for weak tractability. There is a gap between the necessary and sufficient conditions for strong polynomial and polynomial tractability; this is probably because Theorem 1 is not sharp.

One of the open problems for tractability of multivariate integration is the question of whether the exponent of strong polynomial tractability depends on how quickly the weights decay. It is known that if the weights decay sufficiently fast then we can achieve nearly the same exponent as for the univariate case, and it is conjectured that such a decay of the weights is also necessary. We partially prove that this is the case by showing a lower bound on the exponent in terms of the decay of the weights. This bound is unfortunately not sharp and may differ from the upper bound by a factor of 2.

2 Weighted Korobov Spaces

Weighted Korobov spaces consist of periodic complex valued functions f defined on the d-dimensional unit cube $[0,1]^d$ with a controlled decay of their Fourier coefficients. This decay depends on the weights and on the smoothness parameter.

More precisely, for $h = [h_1, h_2, \ldots, h_d] \in \mathbb{Z}^d$ with integers

$$h_j \in \mathbb{Z} := \{\ldots, -1, 0, 1, \ldots\},$$

and for $f \in L_2([0,1]^d)$, let $\hat{f}(h)$ denote the Fourier coefficient of f,

$$\hat{f}(h) = \int_{[0,1]^d} f(x) \exp(-2\pi \,\mathrm{i}\, h \cdot x)\, \mathrm{d}x,$$

with $\mathrm{i} = \sqrt{-1}$, $x = [x_1, x_2, \ldots, x_d]$ for $x_j \in [0,1]$, and the inner product $h \cdot x = h_1 x_1 + h_2 x_2 + \cdots + h_d x_d$. For $d \in \mathbb{N} := \{1, 2, \ldots\}$, let

$$1 = \gamma_{d,0} \geq \gamma_{d,1} \geq \cdots \geq \gamma_{d,d} \geq 0$$

be a given sequence of weights. For any $\mathfrak{u} \subseteq [d] := \{1, 2, \ldots, d\}$, denote $\gamma_{d,\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_{d,j}$. For $h \in \mathbb{Z}^d$, define

$$\mathfrak{u}(h) = \{j \in [d] \mid h_j \neq 0\}.$$

Finally, by $\gamma = \{\gamma_{d,j}\}_{d\in\mathbb{N}, j\in[d]}$ we denote a sequence of all weights. Such weights are called *product weights*, see [10] for more details.

For a given smoothness parameter α, we consider two classes of Korobov spaces equipped with the L_2 and L_∞-type norms. The Korobov space $F_2 = F_{d,\alpha,\gamma,2}$ is defined for $\alpha > 1/2$ by

$$F_2 = \left\{ f \mid \|f\|_2 := \left(\sum_{h\in\mathbb{Z}^d} |\hat{f}(h)|^2 \frac{\prod_{j\in\mathfrak{u}(h)} |h_j|^{2\alpha}}{\gamma_{d,\mathfrak{u}(h)}} \right)^{1/2} < \infty \right\},$$

and the Korobov space $F_\infty = F_{d,\alpha,\gamma,\infty}$ is defined for $\alpha > 1$ by

$$F_\infty = \left\{ f \mid \|f\|_\infty := \sup_{h\in\mathbb{Z}^d} |\hat{f}(h)| \frac{\prod_{j\in\mathfrak{u}(h)} |h_j|^{\alpha}}{\sqrt{\gamma_{d,\mathfrak{u}(h)}}} < \infty \right\}.$$

If $\gamma_{d,\mathfrak{u}(h)} = 0$ then we assume that $\hat{f}(h) = 0$ and interpret $0/0$ as 0. Functions from F_2 and F_∞ are continuous. For large α, they are also sufficiently smooth.

More precisely, assume that $f \in F_2$. For $d = 1$ and $\alpha = r \geq 1$ being an integer, f is periodic, with absolutely continuous derivatives of order up to $r-1$ and with the rth derivative belonging to $L_2([0,1])$. Furthermore, we have

$$\|f\|_2 = \left(\left| \int_0^1 f(x)\,\mathrm{d}x \right|^2 + \frac{1}{(2\pi)^{2r}\,\gamma_{1,1}} \int_0^1 |f^{(r)}(x)|^2\,\mathrm{d}x \right)^{1/2}.$$

For $d \geq 1$, the space $F_{d,r,\gamma,2}$ is a tensor product reproducing kernel Hilbert space

$$F_{d,r,\gamma,2} = F_{1,r,\gamma_{d,1},2} \otimes F_{1,r,\gamma_{d,2},2} \otimes \cdots \otimes F_{1,r,\gamma_{d,d},2}.$$

Its reproducing kernel is

$$K(x,y) = \prod_{j=1}^{d} \left(1 + 2\gamma_{d,j} \sum_{h=1}^{\infty} \frac{\cos(2\pi(x_j - y_j))}{h^{2\alpha}} \right) \quad \text{for all } x, y \in [0,1]^d.$$

For $h \in \mathbb{Z}^d$, define

$$e_h(x) = \exp(2\pi\,\mathrm{i}\,h\cdot x) \quad \text{for all } x \in [0,1]^d. \tag{1}$$

Then for $f \in F_2$ we have

$$f(x) = \sum_{h\in\mathbb{Z}^d} \hat{f}(h)\, e_h(x). \tag{2}$$

For $\beta = [\beta_1, \beta_2, \ldots, \beta_d]$ with $\beta_j \in \mathbb{N}_0 := \{0, 1, \ldots\}$ and $|\beta| := \sum_{j=1}^d \beta_j$, we have

$$(D^\beta f)(x) = \frac{\partial^{|\beta|}}{\partial^{\beta_1} x_1 \cdots \partial^{\beta_d} x_d} f(x) = (2\pi \mathrm{i})^{|\beta|} \sum_{h \in \mathbb{Z}^d} \hat{f}(h)\, e_h(x) \prod_{j=1}^{d} h_j^{\beta_j},$$

with the convention that $0^0 = 1$. The last series is absolutely convergent if $\alpha - \beta_j > 1/2$ for all $j \in [d]$. Indeed, we have

$$\begin{aligned}
\frac{|(D^\beta f)(x)|}{(2\pi)^{|\beta|}} &\le \sum_{h \in \mathbb{Z}^d} \frac{|\hat{f}(h)| \prod_{j \in \mathrm{u}(h)} |h_j|^\alpha}{\sqrt{\gamma_{d,\mathrm{u}(h)}}} \sqrt{\gamma_{d,\mathrm{u}(h)}} \prod_{j \in \mathrm{u}(h)} |h_j|^{-(\alpha-\beta_j)} \\
&\le \|f\|_2 \Bigg(\sum_{h \in \mathbb{Z}^d} \gamma_{d,\mathrm{u}(h)} \prod_{j \in \mathrm{u}(h)} |h_j|^{-2(\alpha-\beta_j)} \Bigg)^{1/2} \\
&= \|f\|_2 \prod_{j=1}^{d} \big[1 + 2\gamma_{d,j}\, \zeta(2(\alpha - \beta_j))\big]^{1/2} < \infty.
\end{aligned}$$

Here $\zeta(x) = \sum_{j=1}^{\infty} j^{-x}$ is the Riemann zeta function defined for $x > 1$. More about the space $F_{d,\alpha,\gamma,2}$ can be found in, e.g., Appendix A of [10].

Assume now that $f \in F_\infty$. Then the derivative $D^\beta f$ exists if $\alpha - \beta_j > 1$ for all $j \in [d]$. Indeed, we now have

$$\begin{aligned}
\frac{|(D^\beta f)(x)|}{(2\pi)^{|\beta|}} &\le \sum_{h \in \mathbb{Z}^d} \frac{|\hat{f}(h)| \prod_{j \in \mathrm{u}(h)} |h_j|^\alpha}{\sqrt{\gamma_{d,\mathrm{u}(h)}}} \sqrt{\gamma_{d,\mathrm{u}(h)}} \prod_{j \in \mathrm{u}(h)} |h_j|^{-(\alpha-\beta_j)} \\
&\le \|f\|_\infty \sum_{h \in \mathbb{Z}^d} \sqrt{\gamma_{d,\mathrm{u}(h)}} \prod_{j \in \mathrm{u}(h)} |h_j|^{-(\alpha-\beta_j)} \\
&= \|f\|_\infty \prod_{j=1}^{d} \big[1 + 2\sqrt{\gamma_{d,j}}\, \zeta(\alpha - \beta_j)\big] < \infty.
\end{aligned}$$

3 Multivariate Integration and Tractability

We consider multivariate integration defined for F_p with $p \in \{2, \infty\}$. That is, we want to approximate

$$I_d(f) = \int_{[0,1]^d} f(x)\, \mathrm{d}x \quad \text{for all} \quad f \in F_p = F_{d,\alpha,\gamma,p}.$$

We assume that we can compute function values at any point from $[0,1]^d$, and approximate $I_d(f)$ by algorithms $A_{n,d}$ that use at most n function values, so that

$$A_{n,d}(f) = \varphi_n(f(t_1), f(t_2), \ldots, f(t_n))$$

for some points t_j and a scalar mapping φ_n. The points t_j can be chosen adaptively, that is, t_j can depend in an arbitrary way on the previously computed points

$t_1, t_2, \ldots, t_{j-1}$ and values $f(t_1), f(t_2), \ldots, f(t_{j-1})$, and also the mapping φ_n can be arbitrary.

The error of the algorithm $A_{n,d}$ is defined in the worst case setting as

$$e(A_{n,d}, p) = \sup_{f \in F_p,\, \|f\|_p \le 1} |I_d(f) - A_{n,d}(f)|.$$

Let

$$e(n, d, p) = \inf_{A_{n,d}} e(A_{n,d}, p)$$

denote the minimal worst case error that can be achieved by using n function values, It turns out that the last infimum is attained for linear algorithms that use nonadaptive choice of points t_j. That is, it is enough to consider algorithms $A_{n,d}$ of the form

$$A_{n,d}(f) = \sum_{j=1}^{n} a_j f(t_j)$$

for some complex numbers a_j and points t_j chosen independently of f. Then

$$e(n, d, p) = \inf_{a_j, t_j} \sup_{f \in F_p,\, \|f\|_p \le 1} \left| I_d(f) - \sum_{j=1}^{n} a_j f(t_j) \right|.$$

Optimality of linear algorithms was proved by Smolyak in his PhD thesis in 1965, whereas optimality of non-adaption by Bakhvalov in 1971. Both results may be found in [1], see also [10, 16].

Note that for $n = 0$, the only algorithms that are allowed are constant. In fact, it is easy to see that the best constant is zero, and

$$e(0, d, p) = \sup_{\|f\|_p \le 1} |\hat{f}(0)| = 1$$

for both $p = 2$ and $p = \infty$. This means that multivariate integration for both F_2 and F_∞ is properly normalized for all d.

We will be using relations between the spaces F_2 and F_∞. For $p \in \{2, \infty\}$, let

$$B_p = B_{d,\alpha,\gamma,p} := \{ f \in F_p \mid \|f\|_{F_p} \le 1 \}$$

be the unit ball of F_p. Since $\|f\|_2 \ge \|f\|_\infty$ for all $f \in F_2$, we have

$$F_2 \subset F_\infty \quad \text{and} \quad B_2 \subset B_\infty.$$

This implies that multivariate integration over F_∞ is not easier than over F_2, i.e.,

$$e(n, d, 2) \le e(n, d, \infty) \quad \text{for all } n \in \mathbb{N}_0 \text{ and } d \in \mathbb{N}.$$

In particular, all lower error estimates for F_2 are true for F_∞, and all upper bounds for F_∞ are also true for F_2.

Let F_p^{unw} denote the unweighted space for which $\gamma_{d,\mathfrak{u}} = 1$ for all $\mathfrak{u} \subseteq [d]$. The unit ball of F_p^{unw} will be denoted by B_p^{unw}. Then for all weights γ we have

$$B_p \subset B_p^{\mathrm{unw}}.$$

This means that multivariate integration over the weighted Korobov spaces is no harder than over the unweighted Korobov spaces; hence, weights can only help. For the unweighted spaces, it can be derived from the results in [8] and [12] that

$$e(n, d, \infty) = \mathcal{O}(n^{-r}) \quad \text{for all} \quad r < \alpha$$

with the factor in the $\mathcal{O}$ notation independent of n but dependent on d. Hence, for large α the rate of convergence is excellent also for all weighted spaces $F_p = F_{d,\alpha,\gamma,p}$ with $p \in \{2, \infty\}$.

It is natural to ask how long we have to wait to enjoy this rate of convergence. This leads us to tractability. For $\varepsilon \in (0, 1)$, let

$$n(\varepsilon, d, p) = \min\{n \mid \ e(n, d, p) \le \varepsilon\}$$

be the minimal number of function values needed to obtain an error at most ε. Clearly,

$$n(\varepsilon, d, 2) \le n(\varepsilon, d, \infty) \quad \text{for all} \quad \varepsilon \in (0, 1) \quad \text{and} \quad d \in \mathbb{N}. \tag{3}$$

We now recall the notions of tractability, see [10] for general discussions, motivation and history. We say that $I = \{I_d\}$ is *weakly tractable* iff

$$\lim_{\varepsilon^{-1}+d \to \infty} \frac{\ln n(\varepsilon, d, p)}{\varepsilon^{-1} + d} = 0,$$

and I is *polynomially tractable* iff there exist non-negative numbers C, a and b such that

$$n(\varepsilon, d, p) \le C\, d^{\,a}\, \varepsilon^{-b} \quad \text{for all} \quad \varepsilon \in (0, 1) \quad \text{and} \quad d \in \mathbb{N}.$$

If $a = 0$ in the inequality above, then I is *strongly polynomially tractable*, and the infimum of b satisfying the inequality above with $a = 0$ is called the *exponent of strong polynomial tractability* and denoted by b^{str}.

Tractability of multivariate integration for weighted spaces F_2 has been studied in [5, 6, 15] for product weights independent of d, i.e., $\gamma_{d,j} = \gamma_j$ but this restriction is not essential for the results of these papers. Tractability for the unweighted space F_∞^{unw} was studied in [13]. Tractability for weighted spaces F_∞ has not yet been studied and, as already mentioned, this is the subject of this paper.

We briefly recall the results for weighted spaces F_2. Weak tractability for F_2 holds iff

$$\lim_{d \to \infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}}{d} = 0. \tag{4}$$

This follows from the following argument. Weak tractability was studied in [3] for reproducing kernel Hilbert spaces whose kernel has a decomposable component,

see [9]. It was proved in [3] that (4) is a necessary and sufficient condition for weak tractability. In [6] it was shown that multivariate integration over F_2 is not easier than integration for a specific subspace of a Sobolev space; said subspace is a reproducing kernel Hilbert space whose kernel has a decomposable component. Therefore (4) is also necessary for weak tractability over F_2. Sufficiency of (4) follows from the known estimate,

$$e(n,d,2) \le n^{-1/2} \prod_{j=1}^{d} (1+2\gamma_{d,j}\zeta(2\alpha))^{1/2},$$

see [4]. This yields

$$n(\varepsilon,d,2) \le \left\lceil \frac{\prod_{j=1}^{d}(1+2\gamma_{d,j}\zeta(2\alpha))}{\varepsilon^2} \right\rceil.$$

Using the bounds $\ln(1+x) \le x$ for $x \ge 0$ and $\lceil x \rceil \le 2x$ for $x \ge 1$, we conclude that

$$\ln n(\varepsilon,d,2) \le 2\zeta(2\alpha) \sum_{j=1}^{d} \gamma_{d,j} \; + \; 2\ln \varepsilon^{-1} + \ln 2.$$

Now it is clear why (4) implies weak tractability for F_2.

It is proved in [6] that polynomial tractability for F_2 holds iff

$$\limsup_{d\to\infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}}{\ln d} < \infty, \tag{5}$$

and strong polynomial tractability for F_2 holds iff

$$\limsup_{d\to\infty} \sum_{j=1}^{d} \gamma_{d,j} < \infty. \tag{6}$$

The paper [6] formally studied product weights that are independent of d, i.e., $\gamma_{d,j} = \gamma_j$. However, the same proof techniques applies for product weights dependent on d with the obvious changes.

Assume that (6) holds, so that we have strong polynomial tractability. We now discuss the exponent b^{str} of strong polynomial tractability. It was shown in [15] that if for some $\beta \in [1/(2\alpha), 1]$ we have

$$\limsup_{d\to\infty} \sum_{j=1}^{d} \gamma_{d,j}^{\beta} < \infty \tag{7}$$

then

$$b^{\mathrm{str}} \in [1/\alpha, 2\beta]. \tag{8}$$

The lower bound on b^{str} is easy since for $d = 1$ we have $e(n, 1, 2) = \Theta(n^{-\alpha})$ which implies that $b^{\mathrm{str}} \geq 1/\alpha$. It is open whether (7) is also necessary, i.e, whether (8) implies (7).

We now turn to the unweighted space F_∞^{unw}. It was proved in [13] that $e(n, d, \infty) = 1$ for all $n = 0, 1, \ldots, 2^d - 1$. This means that

$$n(\varepsilon, d, \infty) \geq 2^d \quad \text{for all } \varepsilon \in (0, 1) \text{ and } d \in \mathbb{N},$$

and multivariate integration over F_∞^{unw} suffers from the curse of dimensionality. Therefore, it is interesting to study conditions on weights for which this curse is vanquished and conditions on weights for which we have polynomial or strong polynomial tractability.

From (3), we know that multivariate integration over F_∞ is not easier than multivariate integration over F_2. Therefore all necessary tractability conditions for F_2 are also necessary tractability conditions for F_∞.

4 Lower Bounds for F_∞

We show a lower bound on $e(n, d, \infty)$. Consider the 2^d weights $\gamma_{d,\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_{d,j}$ for all subsets $\mathfrak{u} \subseteq [d]$, and recall that $1 = \gamma_{d,\emptyset} = 1 \geq \gamma_{d,1} \geq \gamma_{d,2} \geq \cdots \geq 0$. For $j = 0, 1, \ldots, 2^d - 1$, let $\eta_{d,j}$ denote the $(j+1)$st largest weight in the sequence $\{\gamma_{d,\mathfrak{u}}\}$. If two weights are equal then their ordering may be assigned arbitrarily. Clearly, $\eta_{d,0} = 1$, $\eta_{d,1} = \gamma_{d,1}$ and $\eta_{d,2} = \gamma_{d,2}$. If $\gamma_{d,3} \geq \gamma_{d,1}\gamma_{d,2}$ then $\eta_{d,3} = \gamma_{d,3}$; otherwise $\eta_{d,3} = \gamma_{d,1}\gamma_{d,2}$. Finally, we have $\eta_{d,2^d-1} = \prod_{j=1}^{d} \gamma_{d,j}$. Observe that if $\gamma_{d,j} \equiv 1$ then $\eta_{d,j} = 1$ for all $j = 0, 1, \ldots, 2^d - 1$.

Theorem 1.

$$e(n, d, \infty) \in [\eta_{d,n}, 1] \text{ for all } n = 0, 1, \ldots, 2^d - 1.$$

Proof: We modify the proof from [13] to cover product weights. The idea of the proof is to construct a function from the unit ball of F_∞ that vanishes at all n points used by an algorithm with the possibly maximal value of the integral.

Since $e(n, d, \infty) \leq e(0, d, \infty) = 1$, we need to show that $e(n, d, \infty) \geq \eta_{d,n}$ for $n < 2^d$. Take an arbitrary algorithm

$$A_{n,d}(f) = \varphi_n(f(t_1), f(t_2), \ldots, f(t_n))$$

for some (perhaps non-linear) mapping φ_n and some (perhaps adaptively) chosen points

$$t_k = t_k(t_1, t_2, \ldots, t_{k-1}, f(t_1), f(t_2), \ldots, f(t_{k-1})).$$

Take first $f = 0$ and obtain points $t_k = t_k(t_1, t_2, \ldots, t_{k-1}, 0, 0, \ldots, 0)$ with 0 occurring $k - 1$ times.

Each $\eta_{d,j} = \gamma_{d,\mathfrak{u}_j}$ for some $\mathfrak{u}_j$, where $\{\mathfrak{u}_0, \mathfrak{u}_1, \ldots, \mathfrak{u}_{2^d-1}\}$ is an enumeration of the 2^d subsets of $[d]$. For $j \in [0, 2^d - 1]$, define the vector $h_j = [h_{j,1}, h_{j,2}, \ldots, h_{j,d}]$ by letting $h_{j,k} = 1$ if $k \in \mathfrak{u}_j$, and $h_{j,k} = 0$ if $k \notin \mathfrak{u}_j$ for $k = 1, 2, \ldots, d$. Then $h_j \in \{0, 1\}^d$ and $\mathfrak{u}(h_j) = \mathfrak{u}_j$. Therefore $\eta_{d,j} = \gamma_{d,\mathfrak{u}(h_j)}$ and

$$\eta_{d,n} \le \eta_{d,j} = \gamma_{d,\mathfrak{u}(h_j)} \quad \text{for all} \quad j = 0, 1, \ldots, n. \tag{9}$$

We then choose a trigonometric polynomial of the form

$$\theta(x) \sum_{j=0}^{n} a_j \, e^{2\pi \mathrm{i} h_j \cdot x}$$

with a function θ to be specified later, and complex coefficients a_j that are a non-trivial solution of the homogeneous linear system

$$\sum_{j=0}^{n} a_j \, e^{2\pi \mathrm{i} h_j \cdot t_k} = 0 \quad \text{for} \quad k = 1, 2, \ldots, n.$$

Here, we need the assumption that $n < 2^d$. Indeed, we have $n + 1 \le 2^d$ unknowns a_j and n homogeneous linear equations; hence for $n < 2^d$ a non-zero solution exists. The non-zero solution a_j can be normalized and we choose the normalization such that

$$\max_{j \in [0,n]} |a_j| = a_{j^*} = 1,$$

for some $j^* \in [0, n]$. We now define $\theta(x) = e^{-2\pi \mathrm{i} h_{j^*} \cdot x}$. Our function f is given as

$$f(x) = c \sum_{j=0}^{n} a_j \, e^{2\pi \mathrm{i} (h_j - h_{j^*}) \cdot x},$$

where $c = \eta_{d,n}$ if the real part of $\varphi_n(0, 0, \ldots, 0)$ is non-positive, and $c = -\eta_{d,n}$ if the real part of $\varphi_n(0, 0, \ldots, 0)$ is positive.

We now show that f belongs to the unit ball of $F_\infty = F_{d,\alpha,\gamma,\infty}$. Indeed, observe that f is a trigonometric polynomial with

$$h_j - h_{j^*} \in \{-1, 0, 1\}^d \quad \text{for all} \quad j \in [0, n].$$

This implies that

$$\prod_{k \in \mathfrak{u}(h_j - h_{j^*})} |h_{j,k} - h_{j^*,k}|^\alpha = 1.$$

Since the h_j are distinct vectors, we have $|\hat{f}(h_j - h_{j^*})| = |c a_j| \le \eta_{d,n}$ for all $j \in [0, n]$, and $\hat{f}(h) = 0$ for all $h \notin \{h_1 - h_{j^*}, h_2 - h_{j^*}, \ldots, h_n - h_{j^*}\}$. Hence,

$$\|f\|_\infty = \max_{j \in [0,n]} \frac{|\hat{f}(h_j - h_{j^*})|}{\sqrt{\gamma_{d,\mathfrak{u}(h_j - h_{j^*})}}} \le \max_{j \in [0,n]} \frac{\eta_{d,n}}{\sqrt{\gamma_{d,\mathfrak{u}(h_j - h_{j^*})}}}.$$

Observe that

$$u(h_j - h_{j^*}) = \{k \in [d] \mid \ h_{j,k} \neq h_{j^*,k}\} =$$
$$\{k \in [d] \mid \ (h_{j,k} = 1 \wedge h_{j^*,k} = 0) \vee (h_{j,k} = 0 \wedge h_{j^*,k} = 1)\}$$

and

$$\mathfrak{u}(h_j - h_{j^*}) \subseteq \{k \in [d] \mid h_{j,k} = 1\} \cup \{k \in [d] \mid h_{j^*,k} = 1\} = \mathfrak{u}(h_j) \cup \mathfrak{u}(h_{j^*}).$$

Therefore

$$\gamma_{d,\mathfrak{u}(h_j - h_{j^*})} = \prod_{k \in \mathfrak{u}(h_j - h_{j^*})} \gamma_{d,j} \geq \prod_{k \in \mathfrak{u}(h_j)} \gamma_{d,j} \prod_{k \in \mathfrak{u}(h_{j^*})} \gamma_{d,j}$$
$$= \gamma_{d,\mathfrak{u}(h_j)} \gamma_{d,\mathfrak{u}(h_{j^*})}.$$

Using (9), we conclude that

$$\sqrt{\gamma_{d,\mathfrak{u}(h_j - h_{j^*})}} \geq \eta_{d,n} \quad \text{for all} \quad j \in [0, n],$$

and therefore $\|f\|_\infty \leq 1$, as claimed.

Clearly, $f(t_k) = 0$ for all $k = 1, 2, \ldots, n$ and therefore

$$A_{n,d}(f) = \varphi_n(0, 0, \ldots, 0).$$

Furthermore, $I_d(f) = \hat{f}(0) = c\, a_{j^*} = c$, and

$$|I_d(f) - A_{n,d}(f)| = |c - \varphi_n(0, 0, \ldots, 0)| \geq |c - \Re\varphi_n(0, 0, \ldots, 0)| \geq |c|.$$

Hence, $e(A_{n,d}, \infty) \geq \eta_{d,n}$, which completes the proof.

When $\gamma_{d,j} = 1$ for $j = 1, 2, \ldots, d$, Theorem 1 states that $e(n, d, \infty) = 1$ for all $n < 2^d$, and coincides with the same result in [13]. Assume now that $\gamma_{d,j} = 1$ for $j = 1, 2, \ldots, k < d$. Then $e(n, d, \infty) = 1$ for $n < 2^k - 1$, and $n(\varepsilon, d, \infty) \geq 2^k$. Hence, if $k \geq c\, d$ for some positive c independent of d, we have the curse of dimensionality and intractability of multivariate integration for F_∞.

Remark 1. We believe that Theorem 1 is not sharp. First of all, note that we can replace $\eta_{d,n}$ in Theorem 1 by $(\eta_{d,n}\eta_{d,n-1})^{1/2}$ for $n > 0$. However, this is not important, which is why this is not included in Theorem 1. More importantly, for $n \in \{1, 2^d - 1\}$, it is easy to see that $\eta_{d,n}$ can be replaced by $\eta_{d,n}^{1/2}$. We do not know whether $\eta_{d,n}$ can be replaced by $c\,\eta_{d,n}^{1/2}$ for all $n < 2^d$ for some positive constant c independent of d. If this is indeed the case, then the gap between necessary and sufficient conditions on tractability would disappear.

The dependence on $\eta_{d,n}^{1/2}$ can be proved for special weights that are *not* product weights. Namely, this can be done for "almost" constant weights, $\gamma_{d,\emptyset} = 1$ and $\gamma_{d,\mathfrak{u}} = c_d < 1$ for all $\mathfrak{u} \neq \emptyset$. The space F_∞ is defined as before without assuming that

$\gamma_{d,\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_{d,j}$. Then it is easy to apply the same proof as above with $c = c_d^{1/2}$, and prove that Theorem 1 indeed holds with $e(n,d,\infty) \in [\eta_{d,n}^{1/2}, 1]$ for all $n < 2^d$. So this is another indication that Theorem 1 may be not sharp.

To get tractability we must have decaying weights. For $\beta > 0$, note that

$$\prod_{j=1}^{d} \left(1+\gamma_{d,j}^{\beta}\right) = \sum_{\mathfrak{u}\subseteq[d]} \gamma_{d,\mathfrak{u}}^{\beta} = \sum_{n=0}^{2^d-1} \gamma_{d,\beta_n}^{\beta} \le \sum_{n=0}^{2^d-1} e^{\beta}(n,d,\infty).$$

Assume that we have polynomial tractability, so that

$$n(\varepsilon,d,\infty) \le C\, d^{\,a}\, \varepsilon^{-b}$$

for some $C \ge 1$ and $a \ge 0$ and $b > 0$. In fact, it is easy to see that $b > 1/\alpha$ since even for $d = 1$ we must have $b \ge 1/\alpha$, as explained before. For $d = 2$, and ε tending to zero, it is known that $n(\varepsilon,2,2)$ is lower bounded by $\varepsilon^{-1/\alpha}(\ln \varepsilon^{-1})^c$ for some positive c. Since $n(\varepsilon,d,2) \le n(\varepsilon,d,\infty)$, this means that b cannot be equal to $1/\alpha$.

The estimate on $n(\varepsilon,d,\infty)$ yields that $e(\lceil C d^a \varepsilon^{-b}\rceil, d, \infty) \le \varepsilon$ for all $\varepsilon \in (0,1)$. Substituting $n = \lceil C d^a \varepsilon^{-b}\rceil \le 2C d^a \varepsilon^{-b}$ we have

$$e(n,d,\infty) \le (2\,C)^{1/b}\, d^{\,a/b}\, n^{-1/b}.$$

Taking $\beta > b$, we conclude that

$$\sum_{n=0}^{2^d-1} e^{\beta}(n,d,\infty) \le (2C)^{\beta/b}\, d^{\beta a/b}\, \zeta(\beta/b) < \infty.$$

This implies that

$$\limsup_{d\to\infty} \frac{\prod_{j=1}^{d}(1+\gamma_{d,j}^{\beta})}{d^{\,\beta a/b}} < \infty.$$

If $a > 0$, this can happen iff

$$\limsup_{d\to\infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}^{\beta}}{\ln d} < \infty,$$

and if $a = 0$ (strong polynomial tractability) this can happen iff

$$\limsup_{d\to\infty} \sum_{j=1}^{d} \gamma_{d,j}^{\beta} < \infty.$$

Define the exponent of the weight sequence $\gamma = \{\gamma_{d,j}\}$ as

$$\beta(\gamma) = \inf\left\{ \beta :\ \limsup_{d\to\infty} \sum_{j=1}^{d} \gamma_{d,j}^{\beta} < \infty \right\}. \tag{10}$$

Then we have proved the following corollary.

Corollary 1.

- *If multivariate integration is polynomially tractable for F_∞, that is, $n(\varepsilon, d, \infty) \le C\, d^a\, \varepsilon^{-b}$ with $a > 0$ and $b > 1/\alpha$, then for all $\beta > b$ we have*

$$\limsup_{d\to\infty} \frac{\sum_{j=1}^d \gamma_{d,j}^\beta}{\ln d} < \infty.$$

- *If multivariate integration is strongly polynomially tractable for F_∞ with the exponent of strong polynomial tractability b^{str} then*

$$\max\left(\alpha^{-1}, \beta(\gamma)\right) \le b^{\mathrm{str}}.$$

5 Upper Bounds for F_∞

We show an upper bound on $n(\varepsilon, d, \infty)$ by analyzing the worst case errors of lattice rules,

$$A_{n,d}(f) = \frac{1}{n} \sum_{j=0}^{n-1} f\left(\left\{\frac{j}{n} z\right\}\right).$$

Here, n is assumed to be prime and $z \in \{1, 2, \ldots, n-1\}^d$ is the generating vector with integer components. The symbol $\{x\}$ denotes the fractional part of each component of the vector x.

Observe that for $f = e_h$ given by (1) we have

$$A_{n,d}(e_h) = \frac{1}{n} \sum_{j=0}^{n-1} [\exp(2\pi \mathrm{i} h \cdot z/n)]^j = \begin{cases} 0 & \text{if } h \cdot z \ne 0 \bmod n, \\ 1 & \text{if } h \cdot z = 0 \bmod n. \end{cases}$$

Using (2) we conclude that

$$I_d(f) - A_{n,d}(f) = \sum_{h\in\mathbb{Z}^d} \hat{f}(h) \left[I_d(e_h) - A_{n,d}(e_h)\right] = \sum_{h\in\mathbb{Z}^d\setminus\{0\}:\ h\cdot z = 0 \bmod n} \hat{f}(h).$$

From this we obtain the worst case error for F_2 and F_∞,

$$e(A_{n,d}, 2) = \left(\sum_{h\in\mathbb{Z}^d\setminus\{0\}:\ h\cdot z = 0 \bmod n} \gamma_{d,\mathfrak{u}(h)} \prod_{j\in\mathfrak{u}(h)} |h_j|^{-2\alpha} \right)^{1/2},$$

$$e(A_{n,d}, \infty) = \sum_{h\in\mathbb{Z}^d\setminus\{0\}:\ h\cdot z = 0 \bmod n} \sqrt{\gamma_{d,\mathfrak{u}}} \prod_{j\in\mathfrak{u}(h)} |h_j|^{-\alpha}.$$

Using the more precise notation $e(A_{n,d}, p) = e(A_{n,d}, F_{d,\alpha,\gamma,p})$ for $p \in \{2, \infty\}$ we obtain for $\alpha > 1$,

$$e\left(A_{n,d}, F_{d,\alpha,\gamma,\infty}\right) = e^2\left(A_{n,d}, F_{d,\alpha/2,\sqrt{\gamma},2}\right). \tag{11}$$

Here, by $\sqrt{\gamma}$ we mean the product weights generated by $\sqrt{\gamma_{d,j}}$.

This allows us to use upper bounds on the worst case of lattice rules for F_2 from [7] and obtain corresponding upper bounds for F_∞. More precisely, from (11) and Corollary 2 of [7] we know that we can find a generating vector z by the CBC (component-by-component) algorithm such that

$$e(A_{n,d}, F_{d,\alpha,\gamma,\infty}) \le 2^{1/\lambda} n^{-1/\lambda} \prod_{j=1}^{d} \left(1 + 2\gamma_{d,j}^{\lambda/2} \zeta(\alpha\lambda)\right)^{1/\lambda}$$

for all $\lambda \in (1/\alpha, 1]$. Furthermore, the cost of computing such a vector z is proportional to $n\,d\,\ln n$, see [11]. Let

$$n^*(\varepsilon, d) = \left\lceil 2\,\varepsilon^{-\lambda} \prod_{j=1}^{d} \left(1 + 2\gamma_{d,j}^{\lambda/2} \zeta(\alpha\lambda)\right) \right\rceil.$$

Then if we take a minimal prime $n \ge n^*(\varepsilon, d)$ then $e(A_{n,d}, F_{d,\alpha,\gamma,\infty}) \le \varepsilon$. It is known that $n \le 2n^*(\varepsilon, d)$ and this proves the following theorem.

Theorem 2. *For all $\varepsilon \in (0,1)$ and $d \in \mathbb{N}$ we have*

$$n(\varepsilon, d, \infty) \le 2 + 4\,\varepsilon^{-\lambda} \prod_{j=1}^{d} \left(1 + 2\gamma_{d,j}^{\lambda/2} \zeta(\alpha\lambda)\right) \quad \textit{for all} \quad \lambda \in (1/\alpha, 1].$$

6 Tractability

We now combine the lower and upper bounds of the previous sections to present necessary and sufficient conditions on tractability for F_∞.

Theorem 3.
Consider multivariate integration $I = \{I_d\}$ for the space F_∞.

- *If I is strongly polynomially tractable then*

$$\limsup_{d\to\infty} \sum_{j=1}^{d} \gamma_{d,j} < \infty.$$

If

$$\limsup_{d\to\infty} \sum_{j=1}^{d} \gamma_{d,j}^{1/2} < \infty$$

then I is strongly polynomially tractable and its exponent b^{str} satisfies

$$\max\left(\alpha^{-1}, \beta(\gamma)\right) \le b^{\mathrm{str}} \le \max\left(\alpha^{-1}, 2\beta(\gamma)\right).$$

Here $\beta(\gamma)$ is the exponent of the weight sequence $\gamma = \{\gamma_{d,j}\}$ defined by (10).

- *If I is polynomially tractable then*

$$\limsup_{d\to\infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}}{\ln d} < \infty.$$

If

$$A := 2\zeta(\alpha) \limsup_{d\to\infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}^{1/2}}{\ln d} < \infty$$

then I is polynomially tractable and

$$n(\varepsilon, d, \infty) \le 2 + 4\varepsilon^{-1} d^{\,2\zeta(\alpha)\sum_{j=1}^{d} \gamma_{d,j}^{1/2}/\ln d} \le C_\delta\, \varepsilon^{-1} d^{\,A+\delta} \quad \text{for all} \quad \delta > 0.$$

- *I is weakly tractable iff*

$$\lim_{d\to\infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}}{d} = 0.$$

Proof: Necessity of strong polynomial tractability follows from (6), and sufficiency from Theorem 3 with $\lambda = 1$. The lower bound on b^{str} follows from Corollary 2, and the upper bound from Theorem 3 with $\lambda/2$ tending to $\beta(\gamma)$.

Necessity of polynomial tractability follows from (5), and sufficiency from Theorem 3 with $\lambda = 1$. The estimate on $n(\varepsilon, d, \infty)$ easily follows from the bound in Theorem 3 again with $\lambda = 1$.

We turn to weak tractability. Necessity follows from (4). We now show that

$$\lim_{d\to\infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}}{d} = 0 \quad \Longrightarrow \quad \lim_{d\to\infty} \frac{\sum_{j=1}^{d} \gamma_{d,j}^{\lambda}}{d} = 0 \quad \text{for all} \quad \lambda > 0.$$

Indeed for any $\delta \in (0, 1)$, there exists $d(\delta)$ such that $\sum_{j=1}^{d} \gamma_{d,j} \le \delta\, d$ for all $d \ge d(\delta)$. Since the $\gamma_{d,j}$ are non-increasing with j, it follows $j\gamma_{d,j} \le \sum_{j=1}^{d} \gamma_{d,j}$, and therefore $\gamma_{d,j} \le \delta\, d/j$ for all $j \in [d]$. For $j \ge \sqrt{\delta}\, d$ we have $\delta\, d/j \le \sqrt{\delta}$ and

$$\gamma_{d,j}^{\lambda} \le \delta^{\lambda/2} \quad \text{for all} \quad j \ge \sqrt{\delta}\, d \quad \text{and} \quad d \ge d(\delta).$$

Since all $\gamma_{d,j} \le 1$, we have

$$\sum_{j=1}^{d} \gamma_{d,j}^{\lambda} = \sum_{j=1}^{\lceil \sqrt{\delta} d \rceil - 1} \gamma_{d,j}^{\lambda} + \sum_{j=\lceil \sqrt{\delta} d \rceil}^{d} \gamma_{d,j}^{\lambda} \le \delta^{1/2} d + \delta^{\lambda/2}(1 - \delta^{1/2})\, d$$

and

$$\frac{\sum_{j=1}^{d} \gamma_{d,j}^{\lambda}}{d} \le \delta^{1/2} + \delta^{\lambda/2} \quad \text{for all} \quad d \ge d(\delta).$$

Since δ can be arbitrarily small, the limit is zero, as claimed.

Sufficiency of weak tractability now easily follows from Theorem 3 with $\lambda = 1$. Indeed, $\lim_{d\to\infty} \sum_{j=1}^{d} \gamma_{d,j}^{1/2}/d = 0$ implies that

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon, d)}{d} = 0,$$

as needed.

Remark 2. So far we dealt with product weights. We now briefly discuss finite-order weights. They are defined as follows. Let $\gamma = \{\gamma_{d,\mathfrak{u}}\}_{d\in\mathbb{N},\mathfrak{u}\subseteq[d]}$ be a given sequence of non-negative weights. Then γ is called *finite-order weights* iff there is an integer ω such that

$$\gamma_{d,\mathfrak{u}} = 0 \quad \text{for all } d \text{ and } \mathfrak{u} \text{ with } |\mathfrak{u}| > \omega.$$

The smallest ω satisfying the property above is called the *order* of finite-order weights, see [2]. For simplicity we also assume that $\gamma_{d,\mathfrak{u}} \le 1$.

The spaces F_2 and F_∞ are defined as before without assuming that $\gamma_{d,\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_{d,j}$.

Lattice rules for finite-order weights were studied in [2] for the space F_2. For prime n and the CBC algorithm, the result in [2] states that for any $\tau \in [1, \alpha]$ there exists a positive C_τ such that

$$e(A_{n,d}, F_{d,\alpha/2,\sqrt{\gamma}}) \le C_\tau\, d^{\omega\tau/2}\, (n-1)^{\tau/2} \quad \text{for all } \ d \in \mathbb{N}.$$

This and (11) yields

$$n(\varepsilon, d, \infty) = \mathcal{O}\left(\varepsilon^{-1/\tau}\, d^{\omega}\right) \quad \text{for} \quad \tau \in [1, \alpha), \tag{12}$$

where the factor in the $\mathcal{O}$ notation depends only on τ and goes to infinity with τ tending to α. This proves polynomial tractability for all finite-order weights of order ω. For $\gamma_{d,\mathfrak{u}} = 1$ for all $|\mathfrak{u}| \le \omega$, it is easy to modify the proof of Theorem 1 and show that

$$e(n, d, \infty) = 1 \quad \text{for all} \quad n < \binom{d}{\lfloor \omega/2 \rfloor} = \Theta\left(d^{\lfloor \omega/2 \rfloor}\right).$$

Indeed, it is enough to take $h_j \in \{0,1\}^d$ with at most $\lfloor \omega/2 \rfloor$ components equal to 1. Then $h_j - h_j^* \in \{-1, 0, 1\}^d$, as before, and $h_j - h_j^*$ has at most $2\lfloor \omega/2 \rfloor \le \omega$ components equal to 1. Therefore $\gamma_{d,\mathfrak{u}(h_j - h_j^*)} = 1$, as needed. Hence, the upper bound (12) is sharp with respect to d modulo roughly a factor 2 in the exponent of d, and sharp with respect to ε^{-1} since τ can be arbitrarily close to α.

Acknowledgements I thank Frances Kuo, Erich Novak, Art Werschulz and the referees for valuable comments.

References

1. N. S. Bakhvalov, On the optimality of linear methods for operator approximation in convex classes of functions, *USSR Comput. Maths. Math. Phys.* **11**, 244–249 (1971)
2. J. Dick, I. H. Sloan, X. Wang and H. Woźniakowski, Good lattice rules in weighted Korobov spaces with general weights, *Numer. Math.* **103**, 63–97 (2006)
3. M. Gnewuch and H. Woźniakowski, Generalized tractability for linear functionals, in: *Monte Carlo and Quasi-Monte Carlo Methods 2006*, A. Keller, S. Heinrich and H. Niederreiter (eds.), 359–381, Springer, Berlin (2008)
4. F. J. Hickernell, Lattice rules: how well do they measure up?, in: *Random and Quasi-Random Point Sets*, P. Hallekalek and G. Larcher (eds.), 109–166, Springer Verlag, Berlin (1998)
5. F. J. Hickernell and H. Woźniakowski, Integration and approximation in arbitrary dimension, *Adv. Comput. Math.* **12**, 25–58 (2000)
6. F. J. Hickernell and H. Woźniakowski, Tractability of multivariate integration for weighted Korobov classes, *J. Complexity* **17**, 660–682 (2001)
7. F. Y. Kuo, Component-by component construction achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces, *J. Complexity* **19**, 301–320 (2003)
8. H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, vol. 63, SIAM, Philadelphia (1991)
9. E. Novak and H. Woźniakowski, Intractability results for integration and discrepancy, *J. Complexity* **17**, 388–441 (2001)
10. E. Novak and H. Woźniakowski, *Tractability of Multivariate Problems*, Volume I, *European Mathematical Society*, Zürich (2008)
11. D. Nuyens and R. Cools, Fast algorithms for component-by-component construction of rank-1 lattice rules in shift invariant reproducing kernel Hilbert spaces, *Math. Comp.* **75**, 903–920 (2006)
12. I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration*, Clarendon Press, Oxford (1994)
13. I. H. Sloan and H. Woźniakowski, An intractability result for multiple integration, *Math. Comp.* **66**, 1119–1124 (1997)
14. I. H. Sloan and H. Woźniakowski, When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *J. Complexity* **14**, 1–33 (1998)
15. I. H. Sloan and H. Woźniakowski, Tractability of integration for weighted Korobov spaces, *J. Complexity* **17**, 697–721 (2001)
16. J. F. Traub, G. W. Wasilkowski and H. Woźniakowski, *Information-Based Complexity*, Academic Press, New York (1988)

Conference Participants

Christos Alexopoulos
Georgia Tech, School of ISyE, Atlanta, GA, 30332-0205, USA,
e-mail: christos@isye.gatech.edu

Diego Amaya
HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 2A7, Canada, e-mail: diego.amaya@hec.ca

Tatiana Averina
Novosibirsk State University and Institute of Computational Mathematics and Mathematical Geophysics, prospect Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia, e-mail: ata@osmf.sscc.ru

Yan Bai
100 St. George St., Toronto, ON, M5S 3G3, Canada,
e-mail: yanbai@utstat.toronto.edu

Jan Baldeaux
University of New South Wales, Department of Mathematics and Statistics, International House, Sydney, NSW, 2052, Australia,
e-mail: z3177364@science.unsw.edu.au

Serge Barbeau
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: serge_barbeau@hotmail.com

Fabian Bastin
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: bastin@iro.umontreal.ca

Heiko Bauke
Max Planck Institute for Nuclear Physics, Saupfercheckweg 1, Heidelberg, 69117, Germany, e-mail: heiko.bauke@mpi-hd.mpg.de

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5,

Mylène Bédard
Université de Montréal, Département de mathématiques et de statistique, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: bedard@dms.umontreal.ca

Alain Benoit
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: benoit.alain@umontreal.ca

Claude Bernier
HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 2A7, Canada, e-mail: claude.bernier@hec.ca

Alexandros Beskos
University College of London, Department of Statistical Science Torrington Place 1-19, London, WC1E 6BT, United Kingdom e-mail: alex@stats.ucl.ac.uk

Camille Besse
Université Laval, Département d'informatique, Québec, QC, G1K 7P4, Canada,
e-mail: besse@damas.ift.ulaval.ca

Katherine Bhan
University of California, Irvine, Beckman Laser Institute, 1002 Health Sciences Rd., Irvine, CA, 92612, USA, e-mail: kbhan@uci.edu

Jose H. Blanchet
Columbia University, Mudd Building 3rd Floor, 500 West 120th St., New York, NY, 10027-6699, USA, e-mail: jose.blanchet@gmail.com

Denis Bolduc
Université Laval, Département d'économique, 1025, avenue des Sciences-Humaines, Québec, QC, G1V 0A6, Canada,
e-mail: denis.bolduc@ecn.ulaval.ca

Itshak Borosh
Texas A&M University, Milner Hall, College Station, TX, 77843, USA,
e-mail: borosh@math.tamu.edu

Mathieu Boudreault
HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 2A7, Canada, e-mail: mathieu.boudreault@hec.ca

Abdeslam Boularias
Université Laval, Pavillon Parent, chambre 8643, Québec, QC, G1K 7P4, Canada,
e-mail: boularias@gmail.com

Taoufik Bounhar
Société Générale, 17 cours Valmy, Tours Société Générale, Puteaux - La Défense, 92800, France, e-mail: taoufik.bounhar@sgcib.com

Eric Buist
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada, e-mail: buisteri@iro.umontreal.ca

Tim T. Cheam
Université de Montréal, Département de mathématiques et de statistique, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada, e-mail: cheam@dms.umontreal.ca

Zhixiong Chen
Putian University, No.1133, Xueyuan Road, Putian, Fujian, 351100, P.R.China, e-mail: ptczx@126.com

Nan Chen
Chinese University of Hong Kong, 709A, William Mong Engineering Building, CUHK, Sha Tin, Hong Kong, PRC, e-mail: nchen@se.cuhk.edu.hk

Cinzia Cirillo
University of Maryland, Department of civil and environmental engineering, 1179 Glenn M Hall, College Patk, MD, 21114, USA, e-mail: ccirillo@umd.edu

Serge Cohen
Université Paul Sabatier, Institut de Mathématique, Laboratoire de Statistique et Probabilités, Bat 1R1 110, Route de Narbonne, Toulouse, 31062, France, e-mail: Serge.Cohen@math.ups-tlse.fr

Rama Cont
Columbia University, 500 W120th St, New York, NY, 10025, USA, e-mail: Rama.Cont@columbia.edu

Ronald Cools
K.U.Leuven, Department of Computer Science, Celestijnenlaan 200A, Heverlee, B-3001, Belgium, e-mail: Ronald.Cools@cs.kuleuven.be

Daniel Dahan
Société Générale, 17 cours Valmy, Paris La Défense Cedex, 92987, France, e-mail: daniel.dahan@sgcib.com

Patrick Dallaire
Université Laval, DAMAS, 8660, rue de la Pruchière, Québec, QC, G2K1T4, Canada, e-mail: dallaire@damas.ift.ulaval.ca

Sabrina Dammertz
Ulm University, Institute of Media Informatics, Albert-Einstein-Allee 11, Ulm, 89069, Germany, e-mail: sabrina.dammertz@uni-ulm.de

Fred Daum
Raytheon, 318 Acton Street, Carlisle, MA, 01741, USA, e-mail: frederick_e_daum@raytheon.com

Lucia Del Chicca
University of Linz, Institute for Financial Mathematics, Altenbergerstrasse, 69, Linz, Upper Austria, 4040, Austria, e-mail: lucia.delchicca@jku.at

Lih-Yuan Deng
University of Memphis, Department of Mathematical Sciences, Memphis, TX, 38152, USA, e-mail: lihdeng@memphis.edu

Pierre Derian
INSA Toulouse, 3 rue Joutx Aigues, Toulouse, 31000, France,
e-mail: pierre.derian@gmail.com

Josef Dick
UNSW, Kensington, Sydney, NSW, 2052, Australia,
e-mail: josef.dick@unsw.edu.au

Maxime Dion
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: dion.maxime@gmail.com

Benjamin Doerr
Max-Planck Institute for Computer Science, Saarbrücken, Campus E1 4, Saarbrücken, 66123, Germany, e-mail: doerr@mpi-sb.mpg.de

Louis Doray
Université de Montréal, Département de mathématiques et de statistique, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: doray@dms.umontreal.ca

Randal Douc
CITI, Telecom sudParis, 9 rue Charles Fourier, Evry, 91000, France,
e-mail: randal.douc@it-sudparis.eu

Arnaud Doucet
University of British Columbia, Department of Computer Science, 2366 Main Mall, Vancouver, BC, V6T 1Z4, Canada, e-mail: arnaud@cs.ubc.ca

Alain Dubus
Université Libre de Bruxelles, 50, av. F.D. Roosevelt, Bruxelles, B-1050, Belgique,
e-mail: adubus@ulb.ac.be

R. Gabriel Esteves
University of Waterloo, 200 University Ave W, Waterloo, ON, N2L 3G1, Canada,
e-mail: rgesteve@uwaterloo.ca

Pierre Etoré
CMAP, Ecole Polytechnique, Route de Saclay, Palaiseau, 91128, France,
e-mail: etore@cmap.polytechnique.fr

Henri Faure
IML CNRS UMR 6206, 163 Av. de Luminy, case 907, Marseille cedex 9, 13288, France, e-mail: faure@iml.univ-mrs.fr

Daan Frenkel
University of Cambridge, Lensfield Road, Cambridge, CB2 1EW, UK, e-mail: frenkel@amolf.nl

Noufel Frikha
Université Pierre et Marie Curie, 175 rue du Chevaleret, Paris, 75013, France, e-mail: frikha_noufel@hotmail.com

Masanori Fushimi
Nanzan University, 27 Seirei, Seto, Aichi, 489-0863, Japan, e-mail: fushimi@ms.nanzan-u.ac.jp

David Gains
General Dynamics Canada, 350 Legget Drive, 6th Floor, Ottawa, ON, K2K 2W7, Canada, e-mail: dave.gains@gdcanada.com

Geneviève Gauthier
HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 2A7, Canada, e-mail: genevieve.gauthier@hec.ca

Stefan Geiss
Department of Mathematics and Statistics, University of Jyvaeskylae, P.O. Box 35 (MaD), Jyvaeskylae, FIN-40014, Finland, e-mail: geiss@maths.jyu.fi

Alan Genz
Washington State University, Department of Mathematics, Neill Hall, Pullman, WA, 99164-3113, USA, e-mail: alangenz@wsu.edu

Mike Giles
Oxford University Mathematical Institute, 24-29 St Giles, Oxford, OX1 3LB, England, e-mail: mike.giles@maths.ox.ac.uk

Gaston Giroux
410 Vimy, apt. 1, Sherbrooke, QC, J1J 3M9, Canada, e-mail: gastongiroux2000@yahoo.ca

Paul Glasserman
Columbia University, 403 Uris Hall, New York, NY, 10027, USA, e-mail: pg20@columbia.edu

Michael Gnewuch
Christian-Albrechts University Kiel, Department of Computer Science, Christian-Albrechts-Platz 4, Kiel, Schleswig-Holstein, 24098, Germany, e-mail: mig@informatik.uni-kiel.de

Domingo Gomez
University of Cantabria, Facultad de Ciencias, Departamento de Matematicas, Estadistica y Computacion, Avenida de los Castros s/n, 39005, Santander, Cantabria, Spain, e-mail: domingo.gomez@unican.es

Anatoly Gormin
Saint-Petersburg State University, Faculty of Mathematics and Mechanics, Department of Statistical Simulation, Universitetskiy prospect 28, 198504, Saint-Petersburg, Russia, e-mail: Anatoliy.Gormin@pobox.spbu.ru

Yongtao Guan
University of Chicago, 920 E. 58th St, CLSC 412, Chicago, IL, 60637, USA, e-mail: ytguan@uchicago.edu

Hiroshi Haramoto
Kure College of Technology, Department of General Education, Hiroshima, Japan, e-mail: haramoto@math.sci.hiroshima-u.ac.jp

Shin Harase
Hiroshima University, Department of Mathematics, Graduate School of Science, 1-3-1, Kagamiyama, Higashi-Hiroshima, 739-8526, Japan, e-mail: sharase@orange.ocn.ne.jp

Carole Hayakawa
University of California, Irvine, 916 Engineering Tower, Irvine, CA, 92697, USA, e-mail: hayakawa@uci.edu

Stefan Heinrich
University of Kaiserslautern, Department of Computer Science, Erwin-Schrodinger-Strasse, Kaiserslautern, D-67653, Germany, e-mail: heinrich@informatik.uni-kl.de

Fred J. Hickernell
Illinois Institute of Technology, Rm E1-208, 10 W. 32nd Street, Chicago, IL, 60616, USA, e-mail: hickernell@iit.edu

Roswitha Hofer
University of Linz, Institute for Financial Mathematics, Altenbergerstr. 69, Linz, Upperaustria, 4040, Austria, e-mail: roswitha.hofer@jku.at

Wolfgang Hörmann
Bogazici University Istanbul, IE Department, Bebek, Istanbul, 34342, Turkey, e-mail: hormannw@boun.edu.tr

Jim Huang
Raytheon, 116 Walpole St., Dover, MA, 2030, USA, e-mail: jim_huang@raytheon.com

Ronald Jean Paul
9157, 25e Avenue, Montréal, QC, H1Z 4C7, Canada, e-mail: jeanpaul@dms.umontreal.ca

Chan Jiang Tao
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada

Stephen Joe
University of Waikato, Department of Mathematics, Private Bag 3105, Hamilton, 3240, New Zealand, e-mail: stephenj@math.waikato.ac.nz

Galin Jones
University of Minnesota, 1866 Merrill Street, Roseville, MN, 55113, USA, e-mail: galin@stat.umn.edu

Tarek Jouini
University of Windsor, 1177 CHN, 401 Sunset Avenue, Windsor, ON, N9B 3P4, Canada, e-mail: tjouini@uwindsor.ca

Alex Kaganov
University of Toronto, Department of Electrical and Computer Engineering, 40 Elise Terrace, Toronto, ON, M2R 2X1, Canada, e-mail: alex.kaganov@utoronto.ca

Kin Hung Kan
University of Western Ontario, Department of Applied Mathematics, London, ON, N6A 5B7, Canada, e-mail: felix525@gmail.com

Aneta Karaivanova
IPP-BAS, Acad. G. Bonchev St., bl. 25A, Sofia, 1113, Bulgaria, e-mail: anet@parallel.bas.bg

William Kath
Northwestern University, 2145 Sheridan Road, Evanston, IL, 60208-3125, USA, e-mail: kath@northwestern.edu

Reiichiro Kawai
Osaka University, Center for the Study of Finance and Insurance, 5-10-16-201 Kasuga, Toyonaka, Osaka, 560-0052, Japan, e-mail: reiichiro_kawai@ybb.ne.jp

Alexander Keller
mental images GmbH, Fasanenstraße 81, Berlin, 10623, Germany, e-mail: alex@mental.com

Andrew Klapper
University of Kentucky, 325 McDowell Road, Lexington, KY, 40502, USA, e-mail: klapper@cs.uky.edu

Samuel Kou
Harvard University, Department of Statistics, 1 Oxford Street, Cambridge, MA, 2138, USA, e-mail: kou@stat.harvard.edu

Kokoasse Kpomblekou-A
Tuskegee University, 210 Campbell Hall, Tuskegee, AL, 36088, USA,
e-mail: kka@tuskegee.edu

Peter Kritzer
University of New South Wales, School of Mathematics and Statistics, Sydney, New South Wales, 2052, Australia, e-mail: peter.kritzer@gmail.com

Frances Kuo
University of New South Wales, School of Mathematics and Statistics, Sydney, NSW, 2052, Australia, e-mail: f.kuo@unsw.edu.au

Céline Labart
INRIA Rocquencourt, Domaine de Voluceau BP 105, Le Chesnay, 78153, France,
e-mail: celine.labart@inria.fr

Yongyut Laosiritaworn
Chiang Mai University, Department of Physics, Faculty of Science, 239 Huay Kaew Road, Suthep, Muang, Chiang Mai, 502000, Thailand,
e-mail: yongyut_laosiritaworn@yahoo.com

Bernard Lapeyre
Ecole des Ponts, 6 et 8 avenue Blaise Pascal, Marne La Vallée, 77455, France,
e-mail: bernard.lapeyre@enpc.fr

Gerhard Larcher
University Linz, Institut fuer Finanzmathematik, Altenberger Strasse 69, Linz, A-4040, Austria, e-mail: Gerhard.Larcher@jku.at

Adam L'Archevêque-Gaudet
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: larcheva@iro.umontreal.ca

Christian Lécot
Université de Savoie, Laboratoire de Mathématiques, Campus scientifique, Le Bourget-du-Lac, 73376, France,
e-mail: Christian.Lecot@univ-savoie.fr

Pierre L'Ecuyer
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: lecuyer@iro.umontreal.ca

Kevin Leder
Zuckerman Research Center, 415-417 East 68th Street, Suite 1131, New York, NY, 10065, USA, e-mail: kevinleder@gmail.com

Jérôme Lelong
LJK - Tour IRMA, 51, rue des Mathématiques, 38041 Grenoble Cedex 9, France,
e-mail: jerome.lelong@imag.fr

Vincent Lemaire
Université Pierre et Marie Curie, Laboratoire de Probabilités et Modèles Aléatoires, 175 rue du Chevaleret, Paris, 75013, France,
e-mail: vincent.lemaire@upmc.fr

Christiane Lemieux
University of Waterloo, Statistics and Actuarial Science Department, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada,
e-mail: clemieux@math.uwaterloo.ca

Gunther Leobacher
University of Linz, Altenberger Strasse 69, Linz, 4040, Austria,
e-mail: gunther.leobacher@jku.at

Paul Leopardi
The Australian National University, Mathematical Sciences Institute, Building 27, Canberra ACT 0200, Australia, e-mail: paul.leopardi@anu.edu.au

Josef Leydold
WU Wien, Department of Statistics and Mathematics, Augasse 2-6, Vienna, A-1090, Austria, e-mail: josef.leydold@wu-wien.ac.at

Shuo Li
Intel Corporation, 17204 NW Holcomb Drive, Portland, OR, 97229, USA,
e-mail: shuo.li@intel.com

Hsin-Ying Lin
Intel Corporation, 2111 NE 25th Avenue, JF1-13, Hillsboro, OR, 97229, USA,
e-mail: hsin-ying.lin@intel.com

Jun Liu
Harvard University, Département of statistics, 1 Oxford Street, Cambridge, MA, 2138, USA, e-mail: jliu@stat.harvard.edu

Frank Lohse
LBBW, Am Hauptbahnhof 2, Stuttgart, Baden-Wuerttemberg, 70173, Germany,
e-mail: frank.lohse@lbbw.de

Vidaliy Lukinov
Institute of Computational Mathematics, Novosibirsk, Siberia, 630090, Russia,
e-mail: Vidaliy.Lukinov@ngs.ru

Ka Chun Ma
Columbia University, 500 West 120th Street, New York, NY, 10027, USA,
e-mail: km2207@columbia.edu

Sylvain Maire
Université de Toulon (ISITV), Avenue G. Pompidou, La valette du Var, 83262, France, e-mail: maire@univ-tln.fr

Roman Makarov
Wilfrid Laurier University, 75 University Avenue West, Waterloo, ON, N2L 3C5, Canada, e-mail: rmakarov@wlu.ca

Jean-Samuel Marier
Université Laval, DAMAS, 1492, rue Joseph-Napoléon, Québec, QC, G2G 2C1, Canada, e-mail: marier@damas.ift.ulaval.ca

Nathan Martin
INSA, 16 rue Saint-Rémésy, Toulouse, 31000, France, e-mail: nathan.mar7in@gmail.com

Peter Mathé
Weierstrass Institute, Mohrenstrasse 39, Berlin, Germany, e-mail: mathe@wias-berlin.de

Makoto Matsumoto
Hiroshima University, 1-3-1 Kagamiyama, Higashi-Hiroshima, 739-8526, Japan, e-mail: m-mat@math.sci.hiroshima-u.ac.jp

Ilya Medvedev
ICMMG of SB RAS, prospect Lavrentieva 6, Novosibirsk, 630090, Russia, e-mail: medvedev79@ngs.ru

Stephan Mertens
Otto-von-Guericke Universität Magdeburg, Postfach 4120, Magdeburg, 39016, Germany, e-mail: mertens@ovgu.de

Hervé Misere
08, rue Samba Félix. Météo, Brazzaville, Congo, e-mail: hervemisere@yahoo.fr

Walid Mnif
HEC Montréal, GERAD, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 2A7, Canada, e-mail: Walid.Mnif@hec.ca

Hozumi Morohosi
GRIPS, 7-22-1 Roppongi, Minato-ku, Tokyo, 1068677, Japan, e-mail: morohosi@grips.ac.jp

William Morokoff
Standard & Poor's, 55 Water Street, New York City, NY, 10041, USA, e-mail: william_morokoff@sandp.com

Thomas Müller-Gronbach
Lehrstuhl für Mathematische Stochastik, Fakultät für Informatik und Mathematik, Universität Passau, D-94030 Passau, Germany, e-mail: thomas.mueller-gronbach@ovgu.de

Kenji Nagasaka
Hosei University, 39106 Magdeburg, Koganei-Shi, Tokyo, 184-8584, Japan, e-mail: nagasaka@hosei.ac.jp

Christian Nambeu
Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada, e-mail: christian.nambeu@umontreal.ca

Peter Neal
University of Manchester, Oxford Road, Manchester, M13 9PL, UK, e-mail: Peter.Neal@manchester.ac.uk

Harald Niederreiter
National University of Singapore, 2 Science Drive 2, Singapore, 117543, Republic of Singapore, e-mail: ghnied@gmail.com

Wojciech Niemiro
Nicolaus Copernicus University, Chopina 12/18, Torun, 87-100, Poland, e-mail: wniem@mat.uni.torun.pl

Takuji Nishimura
Yamagata University, Department of Mathematical Sciences, 1-4-12, Kojirakawa, Yamagata, 990-8560, Japan, e-mail: nisimura@sci.kj.yamagata-u.ac.jp

Ben Niu
Illinois Institute of Technology, Department of Applied Mathematics, E1 Building, Room 208, 10 West 32nd Street, Chicago, IL, 60616, USA, e-mail: nben@iit.edu

Dirk Nuyens
K.U. Leuven, Department of Computer Science, Celestijnenlaan 200A, Heverlee, B-3001, Belgium, e-mail: dirk.nuyens@cs.kuleuven.be

Damian Oana
HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 2A7, Canada, e-mail: oanada@yahoo.com

Victor Ostromoukhov
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, Succ. Centre-ville, Montréal, QC, H3C 3J7, Canada, e-mail: ostrom@iro.umontreal.ca

Art B. Owen
Stanford University, Department of Statistics, 390 Serra Mall, Stanford, CA, 94305, USA, e-mail: owen@stat.stanford.edu

Ferruh Ozbudak
Middle East Technical University, Inonu Bulvari, Ankara, 6531, Turkey, e-mail: ozbudak@metu.edu.tr

Gilles Pagès
Université Paris 6, Case 188, 4, Place Jussieu, Paris, 75252, France, e-mail: gilles.pages@upmc.fr

Fabien Panloup
INSA Toulouse and Institut de Mathématiques de Toulouse, 135, Avenue de Rangueil, Toulouse, 31077, France, e-mail: fpanloup@insa-toulouse.fr

Christian-Marc Panneton
Industrielle Alliance, 1080, boul. Grande Allée Ouest, Québec, QC, G1K 7M3, Canada, e-mail: Christian-Marc.Panneton@inalco.com

Jean-Sébastien Parent-Chartier
1280, rue St-Jean, Saint-Hyacinthe, QC, J2S 8M3, Canada,
e-mail: js.parent.chartier@gmail.com

François Perron
Université de Montréal, Département de mathématiques et de statistique, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: perronf@dms.umontreal.ca

Friedrich Pillichshammer
University of Linz, Altenbergerstrasse 69, Linz, 4040, Austria,
e-mail: friedrich.pillichshammer@jku.at

Leszek Plaskota
University of Warsaw, Institute of Applied Mathematics and Mechanics, Banacha 2, 02-097, Warsaw, Poland, e-mail: L.Plaskota@mimuw.edu.pl

Marco Pollanen
Trent University, 1600 West Bank Drive, Peterborough, ON, K9J 7B8, Canada,
e-mail: marcopollanen@trentu.ca

Laurent Prigneaux
Société Générale, 17 cours Valmy, Paris La Défense Cedex, 92987, France,
e-mail: laurent.prigneaux@sgcib.com

James Propp
University of Massachusetts Lowell, Department of Mathematical Sciences, 74 Gilbert Road, Belmont, MA, 2478, USA, e-mail: jpropp@cs.uml.edu

Rajiv Ramchurn
Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada, e-mail: rajiv.ramchurn@gmail.com

Mark Reesor
University of Western Ontario, Department of Applied Mathematics, 1151 Richmond Street North, London, ON, N6A 5B7, Canada,
e-mail: mreesor@uwo.ca

Klaus Ritter
Technische Universität Darmstadt, Schlossgartenstrasse 7, Darmstadt, 64289, Germany, e-mail: ritter@mathematik.tu-darmstadt.de

Jeffrey S. Rosenthal
University of Toronto, Department of Statistics, 100 St. George Street, Room 6018, Toronto, ON, M5S 3G3, Canada, e-mail: jeff@math.toronto.edu

Gerardo Rubino
INRIA, Campus de Beaulieu, Rennes, 35042, France,
e-mail: rubino@irisa.fr

Daniel Rudolf
Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, Jena, Thüringen, 7743, Germany, e-mail: danrudolf@web.de

Piergiacomo Sabino
Università degli Studi di Bari, Via E. Orabona 4, Bari, 70125, Italy,
e-mail: sabino@dm.uniba.it

Mutsuo Saito
Hiroshima University, Department of Mathematics, Graduate School of Science, 1-3-1, Kagamiyama, Higashi-Hiroshima, 739-8526, Japan,
e-mail: saito@math.sci.hiroshima-u.ac.jp

Mohamed Sbai
CERMICS-ENPC, 6 et 8 avenue Blaise Pascal, Champs sur Marne, 77420, France,
e-mail: sbai@cermics.enpc.fr

Wolfgang Ch. Schmid
University of Salzburg, Department of Mathematics, Hellbrunnerstr. 34, Salzburg, A-5020, Austria, e-mail: wolfgang.schmid@sbg.ac.at

John Sepikas
Pasadena City College, 1570 E. Colorado, Pasadena, CA, 91106, USA,
e-mail: jsepikas@hotmail.com

Ali Devin Sezer
Middle East Technical University, Eskisehir Yolu, Ankara, 6531, Turkey,
e-mail: devin@metu.edu.tr

Jie Shen
Purdue University, Department of Mathematics, 150 N. Univ. St., West Lafayette, 47907, USA, e-mail: shen@math.purdue.edu

Richard Simard
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: simardr@iro.umontreal.ca

Vasile Sinescu
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, Succ. Centre-ville, Montréal, QC, H3C 3J7, Canada,
e-mail: sinescuv@iro.umontreal.ca

Ian H. Sloan
University of New South Wales, Sydney, NSW, 2086, Australia,
e-mail: i.sloan@unsw.edu.au

Jerome Spanier
University of California, 1002 Health Sciences Rd. E., Irvine, CA, 92612, USA,
e-mail: jspanier@uci.edu

Elaine Spiller
SAMSI / Duke University, 19 TW Alexander PO Box 14009, Research Triangle Park, NC, 27709, USA, e-mail: spiller@math.duke.edu

Jeremy Staum
Northwestern University, 2145 Sheridan Road, Evanston, IL, 60208-3119, USA,
e-mail: j-staum@northwestern.edu

Andrew M. Stuart
University of Warwick, Gibbet Hill Road, Coventry, CV4 7AL, UK,
e-mail: A.M.Stuart@warwick.ac.uk

Nicolae Suciu
Friedrich-Alexander University of Erlangen-Nuremberg, Institute of Applied Mathematics, Martensstrasse 3, Erlangen, 91058, Germany,
e-mail: suciu@am.uni-erlangen.de

Roberto Szechtman
Naval Postgraduate School, Operations Research Department, Montery, CA, 93943, USA, e-mail: rszechtm@nps.edu

Abderrahim Taamouti
Universidad Carlos III de Madrid, Departamento de Economía, Calle Madrid, 126, Getafe (Madrid) 28903, Spain, e-mail: ataamout@eco.uc3m.es

Keizo Takashima
Okayama University of Science, Department of Applied Mathematics, 1-1, Ridai-cho, Okayama, 700-0005, Japan, e-mail: takashim@xmath.ous.ac.jp

Hani Tamim
King Saud bin Abdulaziz University for Health Sciences, College of Medicine/Research Center, Riyadh, Saudi Arabia,
e-mail: hani_t@hotmail.com

Ken Seng Tan
University of Waterloo, Universeity Ave. West, Waterloo, Ontario, N2L 3G1, Canada, e-mail: kstan@uwaterloo.ca

Huei-Wen Teng
Pennsylvania State University, Thomas Building 330B, University Park, PA, 16801, USA, e-mail: hut118@psu.edu

Kurt Thomsen
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada, e-mail: kurt.thomsen@hotmail.com

Thomáš Tichý
Technical University of Ostrava, Department of Finance, Sokolska 33, Ostrava, 701 21, Czech Republic, e-mail: tomas.tichy@vsb.cz

Pierre-Alexandre Tremblay
Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, Canada, e-mail: tremblap@iro.umontreal.ca

Bruno Tuffin
INRIA, Campus Universitaire de Beaulieu, Rennes, 35042, France, e-mail: bruno.tuffin@irisa.fr

Pirooz Vakili
Boston University, 15 Saint Mary's Street, Brookline, MA, 2446, USA, e-mail: vakili@bu.edu

Pascale Valery
HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 2A7, Canada, e-mail: pascale.valery@hec.ca

Suzanne Varet
ONERA, Chemin de la Hunière, Palaiseau, 91761, France, e-mail: Suzanne.Varet@onera.fr

Yi Qi Wang
Wilfrid Laurier University, 9 Lodge Street, Waterloo, ON, N2J 4S8, Canada, e-mail: litflowngo@hotmail.com

Ben J. Waterhouse
University of New South Wales, School of Mathematics and Statistics, Sydney, NSW, 2052, Australia, e-mail: benjw@maths.unsw.edu.au

David White
Warwick University, Gibbet Hill Road, Coventry, Warwickshire, CV4 7AL, UK, e-mail: david.a.white@warwick.ac.uk

James R. Wilson
NC State University, 111 Lampe Dr, Campus Box 7906, Raleigh, NC, 27695-7906, USA, e-mail: jwilson@ncsu.edu

Carola Winzen
Höhenkircherstr.16, Muenchen, Baviera, 81247, Germany, e-mail: Carola.Winzen@gmx.de

Henryk Wozniakowski
Columbia University, Amsterdam Ave., New York, NY, 10027, USA, e-mail: henryk@cs.columbia.edu

Chao Yang
University of Toronto, Department of Mathematics, 40 St. George St., Toronto, ON, M5S 2E4, Canada, e-mail: chaoyang@math.toronto.edu

Hao Yu
University of Western Ontario, Department of Statistics and Actuarial Science, London, ON, N6A 5B7, Canada, e-mail: hyu@uwo.ca

Marta Zalewska
Medical University of Warsaw, Department of Prevention of Environmental Hazards and Allergology, Zwirki i Wigory 61, Warsaw, 02-091, Poland, e-mail: zalewska.marta@gmail.com

Xiaoyan Zeng
Illinois Institute of Technology, Department of Applied Mathematics, 3242 Foxridge Ct., Woodridge, IL, 60517, USA, e-mail: zengxia@iit.edu

Enlu Zhou
University of Maryland, 4305 Rowalt Dr., Apt. 301, College Park, MD, 20740, USA, e-mail: enlu.zhou@gmail.com

Author Index

P. L'Ecuyer, A.B. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008*,
DOI 10.1007/978-3-642-04107-5, © Springer-Verlag Berlin Heidelberg 2009